Title

Randomized Decision Making in Stochastic Control and Revenue Management

Author

Guan, Xinyi

Publication Date

2024

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

Randomized Decision Making in Stochastic Control and Revenue Management

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Management

by

Xinyi Guan

2024

ABSTRACT OF THE DISSERTATION

Randomized Decision Making
in Stochastic Control and Revenue Management

by

Xinyi Guan
Doctor of Philosophy in Management
University of California, Los Angeles, 2024
Professor Velibor Mišić, Chair

Recent studies on randomized decision-making have uncovered the potential advantages of incorporating randomization into decision-making processes. In this Ph.D. dissertation, we consider randomized decisions in two specific problems in stochastic control and revenue management: optimal stopping and robust pricing.

Optimal stopping is the problem of determining when to stop a stochastic system in order to maximize reward, which is of practical importance in domains such as finance, operations management and healthcare. Existing methods for high-dimensional optimal stopping that are popular in practice produce deterministic linear policies – policies that deterministically stop based on the sign of a weighted sum of basis functions – but are not guaranteed to find the optimal policy within this policy class given a fixed basis function architecture. In Chapter 2, we propose a new methodology for optimal stopping based on *randomized* linear policies, which choose to stop with a probability that is determined by a weighted sum of basis functions. We motivate these policies by establishing that under mild conditions, given a fixed basis function architecture, optimizing over randomized linear policies is equivalent to

optimizing over deterministic linear policies. We formulate the problem of learning randomized linear policies from data as a smooth non-convex sample average approximation (SAA) problem. We theoretically prove the almost sure convergence of our randomized policy SAA problem and establish bounds on the out-of-sample performance of randomized policies obtained from our SAA problem based on Rademacher complexity. We also show that the SAA problem is in general NP-Hard, and consequently develop a practical heuristic for solving our randomized policy problem. Through numerical experiments on a benchmark family of option pricing problem instances, we show that our approach can substantially outperform state-of-the-art methods.

In Chapter 3, we consider the robust multi-product pricing problem. It is to determine the prices of a collection of products so as to maximize the worst-case revenue, where the worst case is taken over an uncertainty set of demand models that the firm expects could be realized in practice. A tacit assumption in this approach is that the pricing decision is a deterministic decision: the prices of the products are fixed and do not vary. In Chapter 3, we consider a *randomized* approach to robust pricing, where a decision maker specifies a distribution over potential price vectors so as to maximize its worst-case revenue over an uncertainty set of demand models. We formally define this problem – the *randomized robust price optimization* problem – and analyze when a randomized price scheme performs as well as a deterministic price vector, and identify cases in which it can yield a benefit. We also propose two solution methods for obtaining an optimal randomization scheme over a discrete set of candidate price vectors based on constraint generation and double column generation, respectively, and show how these methods are applicable for common demand models, such as the linear, semi-log and log-log demand models. We numerically compare the randomized approach against the deterministic approach on a variety of synthetic and real problem instances; on synthetic instances, we show that the improvement in worst-case revenue can be as much as 1300%, while on real data instances derived from a grocery retail scanner dataset, the improvement can be as high as 92%.

The dissertation of Xinyi Guan is approved.

Felipe Caro

Auyon Adnan Siddiq

Christopher Siu Tang

Velibor Mišić, Committee Chair

University of California, Los Angeles

2024

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First and foremost, I would like to express my heartfelt appreciation and gratitude to my esteemed advisor Professor Velibor Mišić for his unwavering support, guidance, and encouragement. Velibor has been guiding me since the very beginning of my research journey in operations, which traces back to my master's study at UCLA. He has consistently offered invaluable insights, suggestions, and encouragement during our discussions, which have played a crucial role in shaping my research endeavors and fostering scholarly growth. I learned from him how to capture insights from literature, identify important research questions, and develop data-driven methodologies. I am incredibly fortunate to have Velibor as my advisor and I profoundly appreciate his mentorship over the past five years.

I would like to extend my sincere thanks to Professors Felipe Caro, Auyon Siddiq and Christopher Tang for being my thesis committee members, providing insightful feedback on my research during both my practice job talk and oral defense. I am also grateful to Professors Fernanda Bravo, Francisco Castro and Scott Rodilitz for their invaluable support during my academic job market season.

I would also like to express my thanks to the fellow students at DOTM. I am thankful to my senior peers, especially Irem, Jingwei, Mirel, and Yi-Chun, for supporting me during the pandemic lockdowns and the job market season. In particular, I would like to express my special gratitude to Jingwei and Yi-Chun for sharing their valuable job market experience with me. I am also grateful to my cohort, Jian, Jingyuan, and Zach, for their help and support throughout my Ph.D. study.

Finally, I want to express my deepest gratitude to my parents, Jinhua Yu and Shouping Guan. Their steadfast love and unwavering support have been my enduring source of confidence and strength, enabling me to overcome setbacks in life. I am grateful for everything they have done for me from the bottom of my heart.

VITA

2017	B.S. (Chemistry), Peking University
2018	M.S. (Business Analytics), UCLA
2019	Anderson Fellowship, UCLA Anderson School of Management
2023	Finalist, INFORMS Finance Section Best Student Paper Competition
2023	Dissertation Year Fellowship, UCLA

PUBLICATIONS

X. Guan and V. V. Mišić. Randomized Policy Optimization for Optimal Stopping. Revise and Resubmit for *Management Science*, 2022.

X. Guan and V. V. Mišić. Randomized Robust Price Optimization. Major Revision for *Management Science*, 2023.

CHAPTER 1

Introduction

Many important problems involve making decisions in the presence of randomness or uncertainty. For example, in the field of stochastic control, a stochastic system evolves randomly and the goal is to find optimal control policies that optimize certain desired objective. For another example in revenue management, customer demand patterns are often uncertain and businesses aim to maximize revenue and improve profitability. Typically in these problems, one makes deterministic decisions. In the first example of stochastic control, the decision maker finds optimal policies that deterministically recommend a single best action for a given system state. In the second example of revenue management, the decision maker deterministically chooses fixed pricing or invenetory level for each period of the market.

In recent years, there has been a growing research interest in exploring randomized decision making, which involves selecting actions through a probabilistic method rather than solely relying on deterministic rules or algorithms. Studies have demonstrated that randomization can help the decision maker in either finding optimal solutions or achieving better performance. In this dissertation, motivated by the potential benefits of randomization in decision making, we explore randomized approaches for two specific problems in stochastic control and revenue management: optimal stopping and robust pricing.

The dissertation is organized as follows. In Chapter 2, we propose a new methodology for optimal stopping based on randomized linear policies. We apply our randomized policy approach to a standard option pricing problem and demonstrate its outperformance relative to existing state-of-the-art methods. In Chapter 3, we consider a randomized approach to

robust multi-product price optimization and propose tractable solution methods to obtain optimal randomized pricing schemes. We numerically compare our randomized robust pricing approach against the traditional deterministic robust pricing approach and highlight the substantial benefit of randomization. In the remainder of this section, we provide a high-level overview of our work in each chapter.

1.1 Randomized Policy Optimization for Optimal Stopping

Optimal stopping is the problem of deciding at what time to stop a stochastic system in order to maximize the expected reward. In each period, a decision maker must decide whether to stop the system, or allow it to continue for one more period. If the decision maker chooses to stop the system, she obtains a state-dependent reward; otherwise, she obtains no reward, but she may potentially stop the system at a later period for a higher reward.

In solving high-dimensional optimal stopping problems, existing approximate dynamic programming (ADP) methods that are popular in practice implement deterministic linear policies that deterministically stop based on the sign of a weighted sum of basis functions. The most prevalent method among these approaches is the least squares Monte Carlo (LSM) method introduced by Longstaff and Schwartz (2001). The LSM method uses least squares regression at each period to predict the continuation value based on the current state, using a sample of trajectories. Essentially, the linear weights in LSM policies are selected not with regard to policy optimization but rather to get better approximation of continuation values. This, however, is not guaranteed to find the optimal policy given a fixed basis function architecture. This is similar to the central issue in the growing contextual optimization literature, where the conventional "predict-then-optimize" approach that involves estimating a predictive model without knowledge of the downstream prescriptive problem is outperformed by approaches where the predictive model is estimated using a loss function that is tailored to the prescriptive problem (Elmachtoub and Grigas 2021).

This observation motivates us to consider directly maximizing the sample average estimate of the expected reward over the weights that define the deterministic linear policies. But such a sample average approximation (SAA) problem is a challenging discrete optimization problem. In view of this, we instead consider randomized linear policies that probabilistically choose to stop or continue at each period, where the stopping probability is characterized by a logistic function and the logit is a weighted sum of basis functions.

In Chapter 2, we formulate the problem of learning randomized linear policies from data as an SAA problem with a smooth non-convex objective function. We prove that under mild conditions, given a fixed basis function architecture, optimizing over randomized linear policies is equivalent to optimizing over deterministic linear policies. We theoretically show the almost sure convergence of our randomized policy SAA problem and establish bounds on the out-of-sample performance of randomized policies obtained from our SAA problem based on Rademacher complexity. We also show that our proposed SAA problem is in general NP-Hard, and consequently develop a practical heuristic for effectively solving our randomized policy problem. Our contributions are not only theoretical but also numerically grounded. Through numerical experiments on a benchmark family of option pricing problem instances, we show that the policies generated by our randomized policy approach in general are substantially better than policies produced by LSM, and are as good or better than policies produced by the pathwise optimization method (Desai et al. 2012), a state-of-the-art method based on martingale duality.

1.2 Randomized Robust Price Optimization

The challenge of demand uncertainty plays a pivotal role in price optimization. A firm sets the prices of its products by maximizing the revenue given a demand model. The estimation of the demand model is often not accurate due to limited data, which may lead to suboptimal revenues. Previous studies adopt robust optimization to handle this issue. The idea of it

is to select an uncertainty set of potential demand models and to maximize the worst-case revenue over all the demand models in the uncertainty set. Typically, robust price optimization aims to identify the single best pricing decision that optimizes the worst-case revenue. However, recent research (Delage et al. 2019) has revealed that with regard to the worst-case objective, it is possible to achieve better performance than the traditional deterministic robust optimization approach by randomizing over multiple solutions. Specifically, instead of optimizing over a single decision in some feasible set, one optimizes over a distribution supported on the feasible set that informs the decision maker how to randomize.

In Chapter 3, we propose a methodology for robust price optimization that is based on randomization. In particular, we propose solving a randomized robust price optimization (RRPO) problem, which outputs a probability distribution that specifies the frequency with which the firm should use different price vectors. We analyze when a randomized price scheme performs as well as a deterministic price vector, and identify cases in which it can yield a benefit. To tackle the RRPO problem over a discrete set of candidate price vectors, we propose tractable algorithms for different settings - whether the uncertainty set of demand function parameters is convex or finite. We show how these solution methods are applicable for common demand models, such as the linear, semi-log and log-log demand models. Notably, for semi-log and log-log demand models, we leverage the reformulation of nominal pricing problems into tractable mixed-integer exponential cone programs, thereby enabling efficient solutions to the RRPO problem.

To substantiate the practicality and effectiveness of randomized pricing, we conduct numerical experiments on different problem instances generated synthetically and problem instances calibrated with real data. The results from synthetic instances show that randomized pricing can improve worst-case revenues by as much as 1300% over deterministic pricing, while in our real data instances, the benefit can be as high as 92%. Additionally, we show that for instances of realistic size (up to 20 products), our algorithm can solve the RRPO problem in a reasonable amount of time (no more than four minutes on average).

CHAPTER 2

Randomized Policy Optimization for Optimal Stopping

2.1 Introduction

Optimal stopping is the problem of deciding at what time to stop a stochastic system in order to maximize the expected reward. Specifically, we are given a stochastic system, that starts at an initial state and transitions randomly from one state to another in discrete time, and a reward function, which maps each state at each time to a real value. In each period, we must decide whether to stop the system, or allow it to continue for one more period. If we choose to stop the system, we obtain the reward given by the reward function for the current state; otherwise, we obtain no reward, but we may potentially stop the system at a later period for a higher reward. Our goal is to find a policy, which is a mapping from the state at each period to the decision to stop or continue, so as to maximize the expected reward.

Optimal stopping problems are found in many application domains, such as finance, operations and healthcare. For example, in finance, an important application of optimal stopping is the problem of option pricing. In this problem, an option holder has the right to buy an asset (if it is a call option) or to sell an asset (if it is a put option) at some strike price. The stochastic system corresponds to the asset, and the system state corresponds to the asset's current price. The option holder's problem is to decide when to exercise the option, which is akin to stopping, so as to garner the greatest expected payoff. The price that an option writer should charge for the option is exactly the highest expected payoff that one

can obtain from an optimal exercise policy of the option. As another example, in operations management, consider a firm that needs to decide when to introduce a new product to a market. In this problem, the system corresponds to market conditions, and the system state would correspond to (say) the unit production cost and the predicted market share that the product would capture, which evolve stochastically over time as more and more competitors enter this market. At each period, the firm can decide to introduce the product into the market, which corresponds to stopping the system, and the reward corresponds to the profit obtained from this market. The problem is then to find a policy that determines whether to introduce the product or wait, so as to maximize the profit from introducing the product.

High-dimensional optimal stopping problems can in theory be solved exactly by dynamic programming. This approach involves obtaining the optimal value function, which maps the state at each period to the highest possible expected reward that can be attained conditional on starting at that state in that period, or the optimal continuation value function, which maps the state at each period to the highest possible expected reward that can be attained conditional on choosing to continue out of that state in that period. An optimal policy can then be found by considering the greedy policy with respect to the optimal value function or optimal continuation value function. However, this approach is untenable in practice for high-dimensional optimal stopping problems due to the curse of dimensionality.

As a result, a number of approaches based on approximate dynamic programming (ADP) have been proposed to solve high-dimensional optimal stopping problems, wherein one considers a policy that is greedy with respect to an approximate value function or continuation value function. Of these methods, the most prevalent ADP method is the least squares Monte Carlo (LSM) approach proposed by Longstaff and Schwartz (2001). This approach involves simulating a set of sample paths or trajectories of the system, and then iterating from the last period in the horizon to the first. At each period t, one uses least squares to obtain a regression model that predicts the continuation value based on the current state, using the sample of trajectories. One then compares the prediction with the reward from

stopping in the current period in each trajectory. If the reward from stopping is higher than the predicted continuation value, we choose to stop; otherwise, we choose to continue. Based on this decision, we update the continuation value, and we repeat the process again at period $t-1$. The algorithm continues in this way, until we reach the first period. The resulting policy is then to take the action that is greedy with respect to the approximate continuation value function.

From a theoretical standpoint, if one were given an infinite sample of trajectories and one could solve the least squares problem at each stage of the LSM algorithm over an unrestricted function class, then the regression model that one would obtain would exactly coincide with the optimal continuation value function. This is due to the fact that the conditional expectation function $m(x) = \mathbb{E}[Y \mid X = x]$ minimizes squared error, i.e., it solves the optimization problem $\min_m \mathbb{E}[(Y - m(X))^2]$. In such an idealized situation, the policy produced by LSM would indeed be optimal.

In practice, one must work with a finite sample of trajectories, and the regression function is constrained to be within the span of a finite collection of basis functions that are specified by the decision maker. Thus, the policy that is produced by LSM is a policy in which one decides to stop or continue by comparing the reward to a weighted sum of basis functions. This is significant for two reasons: (i) it is no longer the case that the policy produced by LSM is an optimal policy; and (ii) even when we restrict our focus to the corresponding policy class that LSM operates in – policies that stop if and only if the reward is greater than a weighted combination of basis functions – the policy produced by LSM may not be optimal within that class. This occurs because in LSM, the approximate continuation value function is obtained by minimizing squared loss, which does not account for the fact that this approximation will be used as part of a policy, and ultimately does not guarantee good out-of-sample policy performance.

This motivates the following question: *how can one obtain LSM-like policies that perform better than LSM*? The policy produced by LSM belongs to a broader family of policies that

we refer to as *deterministic linear policies*: policies that deterministically recommend to stop or continue at each period depending on whether a weighted sum of basis functions is positive or negative. (This class subsumes LSM policies if one includes the immediate reward at each period as a basis function.) Given a sample of trajectories, an immediate approach to obtaining a good policy from this class would be to formulate a sample average approximation (SAA) problem: optimize over the weights defining the deterministic linear policy, so as to maximize the sample average estimate of the expected reward of the policy. The drawback of this approach is that due to the discrete nature of how this family of policies works, the SAA problem is a challenging discrete optimization problem. Such a problem would be infeasible to solve for the sample sizes that are typically found in practical optimal stopping applications.

As an alternative to deterministic linear policies, one can also consider *randomized linear policies*. These are policies that probabilistically choose to stop or continue at each period, where the probability of stopping is given by a logistic probability and the logit that defines this probability is a weighted sum of basis functions. Just like the deterministic linear policy case, one can also formulate an SAA problem to maximize the sample average reward with respect to the weights that define this randomized policy. Although the resulting SAA problem is still a challenging non-convex problem, the objective function is now smooth and from a computational standpoint, one can now at least solve the problem heuristically using any of a number of practically successful gradient-based methods.

We make the following specific contributions:

1. **Model**: We propose the class of randomized linear policies for optimal stopping problems, and formulate the problem of learning such a policy from data as an SAA problem with a smooth, non-convex objective function. We prove that under mild conditions, solving the randomized linear policy SAA problem is equivalent to solving the deterministic linear policy SAA problem, in that the optimal objectives of the two problems are equivalent; under an additional condition, we also show that the true randomized

linear policy problem and the true deterministic linear policy problem, where sample averages are replaced by expectations, are also equivalent in objective value.

2. **Statistical guarantees**: We provide two statistical guarantees for our randomized policy SAA problem. First, we show that our learning problem is consistent: as the number of trajectories in our training sample grows, the optimal objective value and optimal solution converge almost surely to the optimal objective value and optimal solution set, respectively, of the true stochastic optimization problem, where sample averages are replaced with expectations. Second, we develop a generalization bound on the out-of-sample objective value of a randomized policy obtained from our SAA problem based on Rademacher complexity, and develop several different bounds on the Rademacher complexity for different choices of the set of feasible weights.

3. **Heuristic**: We prove that in general, our randomized policy SAA problem is NP-Hard, which follows from a reduction from the MAX-3SAT problem. Consequently, we propose a backward optimization algorithm for solving the problem heuristically, which optimizes the weights defining the randomized policy in stages, starting with the weights corresponding to the last period and working its way to the first stage.

4. **Numerical experiments**: Using a benchmark family of Bermudan max-call option pricing instances used in the recent literature, we show that our approach yields policies that in general are substantially better than policies produced by LSM, and are as good or better than policies produced by the pathwise optimization method (Desai et al. 2012), a state-of-the-art method based on martingale duality.

The rest of this chapter is organized as follows. In Section 2.2, we review the relevant literature in optimal stopping, as well as other recent related work. In Section 2.3, we formally define the optimal stopping problem, define the deterministic linear policy problem in its sample average and true stochastic forms, define the randomized linear policy problem in its sample average and true stochastic forms, and prove that the randomized linear policy

problem and deterministic linear problem are equivalent. In Section 2.4, we prove that our randomized policy SAA problem is consistent and develop our generalization guarantees. In Section 2.5, we show that our randomized policy SAA problem is NP-Hard, and present our backward optimization algorithm for solving it. In Section 2.6, we present the results of our numerical study on option pricing instances.

2.2 Literature Review

Our work is closely related to three streams of research: the optimal stopping and ADP literature; prediction-and-optimization literature; and non-convex optimization literature.

Optimal stopping and approximate dynamic programming (ADP). Optimal stopping problems have been extensively studied in many fields such as statistics, operations research and mathematical finance. In theory, optimal stopping problems can be solved by dynamic programming, but in practice, the curse of dimensionality renders this approach infeasible for all but the simplest optimal stopping problems. As a result, there has been much attention towards developing good approximate dynamic programming (ADP) methods for optimal stopping.

In the context of optimal stopping, the most popular family of ADP methods is that of simulation-regression. The idea of simulation-regression methods is to simulate a sample of trajectories of the system state and use least squares regression to approximate the optimal continuation value function (i.e., the optimal expected reward from choosing to continue for a given current state) at each step. The paper of Carriere (1996) was the first to introduce this type of approach for the valuation of American options, using non-parametric regression; later, Longstaff and Schwartz (2001) and Tsitsiklis and Van Roy (2001) independently considered this approach in the setting where the continuation value function is approximated as a linear combination of basis functions.

Besides simulation-regression, another important stream of ADP methods for optimal stopping is based on the idea of martingale duality. The main idea in this body of work is to relax the non-anticipativity of the policy, but to then penalize the use of future information through a martingale process. In doing so, one obtains an upper bound on the optimal reward, and in some cases one can also obtain policies that perform well. We refer the reader to Rogers (2002), Andersen and Broadie (2004), Haugh and Kogan (2004), Chen and Glasserman (2007), Brown et al. (2010), Desai et al. (2012) for salient examples of this methodology, and to the recent review paper of Brown and Smith (2022) for a detailed overview of this technique as it applies to stochastic dynamic programming more broadly.

Lastly, other recent research has considered approaches distinct from the above two streams. The paper of Ciocan and Mišić (2022) considers a method for directly obtaining optimal stopping policies from a sample of trajectories in the form of a binary tree. In a different direction, the paper of Sturt (2021a) proposes a method for obtaining threshold policies for low-dimensional optimal stopping problems using robust optimization.

Our methodology is most closely related to the simulation-regression approach and in particular, the least-squares Monte Carlo (LSM) approach of Longstaff and Schwartz (2001). There are several differences between our methodology and LSM. One difference is that our methodology involves the use of randomized policies, whereas the policy produced by LSM is deterministic. Aside from this, the key philosophical difference between our work and the LSM approach is that while LSM produces a policy in an indirect way – by approximating the continuation value function using least squares – our methodology involves formulating an SAA problem and obtaining a policy that *directly* maximizes an estimate of the expected reward obtained with respect to a sample of trajectories. In terms of algorithms, the backward algorithm for heuristically solving our SAA problem that we present in Section 2.5 is reminiscent of the LSM algorithm, but instead of solving a least squares problem, one solves a non-convex problem where the objective function is given by a weighted sum of logistic response functions.

Predict-then-optimize. Outside of optimal stopping, our work relates to the literature on combining prediction and optimization. In many analytics problems, the "predict-then-optimize" paradigm is often used: one first builds a predictive model by minimizing a loss function that measures predictive performance (for example, squared error), and then utilizes that predictive model in a subsequent optimization problem to obtain a decision. There are many papers that apply this type of approach (see, for example, Ferreira et al. 2016, Cohen et al. 2017, Bertsimas and Kallus 2020).

However, as pointed out in the recent paper of Elmachtoub and Grigas (2021), this type of predict-then-optimize paradigm can lead to suboptimal decisions, since the predictive model is trained using a loss function that does not account for how the predictive model will be used in the downstream optimization problem. The paper of Elmachtoub and Grigas (2021) proposes a Smart Predict-then-Optimize (SPO) framework, where the predictive model is estimated so as to minimize decision/prescriptive loss rather than predictive loss, and numerically shows that the SPO framework can result in significantly better out-of-sample performance.

Our work is partially inspired by the observation that the LSM algorithm bears a resemblance to the standard predict-then-optimize paradigm. In the LSM approach, one first predicts the continuation value based on squared error and then uses that prediction within a greedy policy. However, minimizing squared error does not necessarily translate into good prescriptive performance of the prediction model. Therefore, in order to find a good policy, we consider the problem of directly optimizing in-sample reward over the space of randomized linear policies.

Non-convex optimization. Lastly, our work is related to the growing literature on non-convex optimization. In the machine learning community, there has been considerable interest in how to solve non-convex optimization problems, since many learning tasks can be

naturally expressed as non-convex optimization problems. Since non-convex optimization problems are in general NP-Hard, a popular approach for tackling such problems is based on convex relaxation, where one relaxes the problem in some way to obtain a convex problem that is more tractable. However, as pointed out by Jain and Kar (2017), such convex relaxations generally change the problem drastically, and thus the solution of relaxation can perform poorly for the original problem. Because of this, there has been much recent work on directly solving the non-convex problems via approximate algorithms. Efficient techniques used in non-convex optimization approach include generalized projected gradient descent (Candes et al. 2015), generalized alternating minimization (Netrapalli et al. 2015), and stochastic optimization techniques (Ge et al. 2015). Although these approaches are not guaranteed to find the global optimum in general, it has been empirically observed that approximately optimal solutions to the true non-convex problem are often better than exactly optimal solutions to a convex relaxation of the problem (Jain and Kar 2017).

In our work, the optimal stopping problem of learning randomized policies from sample data is formulated as a non-convex optimization problem. We follow the spirit of non-convex optimization approaches and propose a backward optimization heuristic to directly work with this non-convex problem, which sequentially optimizes over the weights in each time period. In our implementation of this method, the weights in each time period are approximately optimized using the Adam algorithm (Kingma and Ba 2014), a first-order method that is widely used for non-convex optimization problems, particularly those arising in the training of deep neural networks. Although our heuristic is not guaranteed to find a globally optimal solution, we find numerically that the resulting policies can significantly outperform those obtained by LSM.

2.3 Problem Definition

In this section, we begin by defining our optimal stopping problem (Section 2.3.1). We then define the family of deterministic linear policies, and the problems of optimizing over deterministic linear policies given complete knowledge of the stochastic process (Section 2.3.2) and given a sample of trajectories (Section 2.3.3). In Section 2.3.4, we define the family of randomized linear policies and analogously to the deterministic linear policy case, we define the true stochastic optimization problem for this policy class and its finite sample counterpart. Finally, in Section 2.3.5, we state our main equivalence results, which assert that (i) the sample average approximation problems over deterministic and randomized linear policies are equivalent and (ii) the true stochastic optimization problems over deterministic and randomized linear policies are equivalent.

2.3.1 Optimal stopping problem

We consider a stochastic system that evolves over a discrete time horizon of T periods. Each period is denoted by t, and ranges in $[T]$, where we use the notation $[n]$ to denote the set $\{1, \ldots, n\}$ for any integer n. We use $\mathbf{x}$ to denote the state of the system, and $\mathbf{x}(t)$ to denote the state of the system in each period, which belongs to a state space $\mathcal{X}$. At each period, we can choose to stop the system or to continue for one more period. If we choose to stop, we receive a nonnegative reward $g(t, \mathbf{x})$ that is a function of the period t and the current state $\mathbf{x}$. If we continue, we do not receive a reward. The action space of the problem is therefore $\mathcal{A} = \{\textbf{stop}, \textbf{continue}\}$.

The decision maker has the ability to specify a deterministic policy $\pi : [T] \times \mathcal{X} \to \mathcal{A}$, which is a mapping from the current period and state we are in to one of the two actions. The policy π defines a stopping time τ_π, which is a random variable that represents the time in $[T]$ at which the decision maker stops:

$$\tau_\pi = \min\{t \in [T] \mid \pi(t, \mathbf{x}(t)) = \textbf{stop}\}. \tag{2.1}$$

We denote the case that the system is never stopped by $\tau_\pi = +\infty$, and we assume that the reward is zero in this case, i.e., $g(+\infty, \mathbf{x}) = 0$ for all $\mathbf{x} \in \mathcal{X}$.

Letting Π denote the set of all policies, the decision maker's goal is to specify the policy π that maximizes the expected discounted reward, which can be written as the following optimization problem:

$$\underset{\pi \in \Pi}{\text{supremum}}\ \mathbb{E}[g(\tau_\pi, \mathbf{x}(\tau_\pi))]. \tag{2.2}$$

We make two important remarks regarding our optimal stopping problem (2.2). First, we note that our formulation does not include a discount factor, which is common in the optimal stopping literature. Our motivation for this modeling choice was to simplify the mathematical exposition and to make certain expressions that appear later on less cumbersome. We also note that this is not a restrictive modeling choice, as the reward function g is time dependent, and so one can specify it so as to incorporate discounting. Second, for the entirety of the chapter, we shall assume that g is uniformly bounded, which we formalize in the following assumption.

Assumption 1 *There exists a finite upper bound $\bar{G}$ such that for any $t \in [T]$, $\mathbf{x} \in \mathcal{X}$, $0 \leq g(t, \mathbf{x}) \leq \bar{G}$.*

2.3.2 Deterministic linear policies

The optimal stopping problem (2.2) is a challenging problem to solve because the set of policies is unrestricted. Rather than working with the set of all policies, we will consider the set of policies that can be described using a linear combination of basis functions. Specifically, let us define $\phi_1, \ldots, \phi_K : \mathcal{X} \to \mathbb{R}$ to be a collection of basis functions, which map a state to a real number; for convenience, we will use $\Phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \ldots, \phi_K(\mathbf{x}))$ to denote the vector of basis functions. Let us also define $\mathbf{b}_t = (b_{t,1}, \ldots, b_{t,K}) \in \mathbb{R}^K$ to be a K-dimensional vector of weights corresponding to the policy at period $t \in [T]$, and additionally, let us use $\mathbf{b}$ to denote the collection of $\mathbf{b}_t$ vectors, i.e., $\mathbf{b} = (\mathbf{b}_1, \ldots, \mathbf{b}_T)$. We can then define the policy $\pi_{\mathbf{b}}$ as

the policy that recommends stopping whenever the weighted combination of basis functions, where the weights come from $\mathbf{b}$, is positive:

$$\pi_{\mathbf{b}}(t, \mathbf{x}) = \begin{cases} \textbf{stop} & \text{if } \sum_{k=1}^{K} b_{t,k}\phi_k(\mathbf{x}(t)) > 0, \\ \textbf{continue} & \text{otherwise.} \end{cases} \tag{2.3}$$

We let $\mathcal{B} \subseteq \mathbb{R}^{KT}$ be the set of feasible weight vectors, and let $\Pi_{\mathcal{B}}$ be the corresponding set of linear policies:

$$\Pi_{\mathcal{B}} = \{\pi_{\mathbf{b}} \mid \mathbf{b} \in \mathcal{B}\}.$$

The linear policy optimal stopping problem can then be written as:

$$\underset{\pi \in \Pi_B}{\text{supremum}}\ \mathbb{E}[g(\tau_\pi, \mathbf{x}(\tau_\pi))]. \tag{2.4}$$

Note that we can re-write this problem without the use of the stopping time τ_π, and to make the dependence on $\mathbf{b}$ more explicit, as follows:

$$\underset{b \in B}{\text{supremum}}\ \mathbb{E}\left[\sum_{t=1}^{T} g(t, \mathbf{x}(t)) \cdot \prod_{t'=1}^{t-1} \mathbb{I}\{\mathbf{b}_{t'} \bullet \Phi(\mathbf{x}(t')) \leq 0\} \cdot \mathbb{I}\{\mathbf{b}_t \bullet \Phi(\mathbf{x}(t)) > 0\}\right], \tag{2.5}$$

where we use $\mathbb{I}\{\cdot\}$ to denote the indicator function (i.e., $\mathbb{I}\{A\} = 1$ if A is true, and 0 if A is false), and for notational convenience, we use $\bullet$ to denote inner products, i.e., for $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$, $\mathbf{a} \bullet \mathbf{b} = \sum_{i=1}^{n} a_i b_i$. Note that the term $\prod_{t'=1}^{t-1} \mathbb{I}\{\mathbf{b}_{t'} \bullet \Phi(\mathbf{x}(t')) \leq 0\} \cdot \mathbb{I}\{\mathbf{b}_t \bullet \Phi(\mathbf{x}(t)) > 0\}$ is equal to 1 if and only if $\tau_\pi = t$; thus, this problem is equivalent to problem (2.4). We also use $J_D(\mathbf{b})$ to denote the objective value of problem (2.5) at a fixed weight vector $\mathbf{b}$.

2.3.3 Data-driven optimization over deterministic linear policies

While problem (2.5) is a simplification of the general optimal stopping problem (2.2), it is still challenging to solve as it requires one to compute expectations over the stochastic process $\{\mathbf{x}(t)\}_{t=1}^{T}$ exactly. More specifically, this problem is challenging because the stochastic process is sufficiently complicated that optimizing over the objective function of problem (2.5) is computationally difficult, or because the stochastic process itself is not known exactly.

Thus, rather than considering the exact version of the problem, one can consider solving a sample-average approximation (SAA) version of the problem, wherein one has access to a set of trajectories of the stochastic process.

To define this problem, we assume that we have access to a set of Ω trajectories and that each trajectory is indexed by ω, which ranges from 1 to Ω. Each trajectory ω corresponds to a sequence of states $\mathbf{x}(\omega, 1), \mathbf{x}(\omega, 2), \ldots, \mathbf{x}(\omega, t)$. Given a policy and a trajectory ω, we define the stopping time for policy π in trajectory ω as

$$\tau_{\pi,\omega} = \min\{t \in [T] \mid \pi(t, \mathbf{x}(\omega, t)) = \textbf{stop}\}.$$

Our SAA problem to determine the optimal linear policy is then

$$\underset{\pi \in \Pi_B}{\text{supremum}} \frac{1}{\Omega} \sum_{\omega=1}^{\Omega} g(\tau_{\pi,\omega}, \mathbf{x}(\omega, \tau_{\pi,\omega})). \tag{2.6}$$

Similarly to problem (2.5), we can re-write problem (2.6) as an optimization problem over $\mathbf{b}$ as follows:

$$\underset{b \in B}{\text{supremum}} \frac{1}{\Omega} \sum_{\omega=1}^{\Omega} \sum_{t=1}^{T} g(t, \mathbf{x}(\omega, t)) \cdot \prod_{t'=1}^{t-1} \mathbb{I}\{\mathbf{b}_{t'} \bullet \Phi(\mathbf{x}(\omega, t')) \leq 0\} \cdot \mathbb{I}\{\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega, t)) > 0\}. \tag{2.7}$$

Note that the term $\prod_{t'=1}^{t-1} \mathbb{I}\{\mathbf{b}_{t'} \bullet \Phi(\mathbf{x}(\omega, t')) \leq 0\} \cdot \mathbb{I}\{\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega, t)) > 0\}$ is equal to 1 if and only if $\tau_{\pi_{\mathbf{b}},\omega} = t$. Additionally, we use $\hat{J}_D(\mathbf{b})$ to denote the objective value of problem (2.7) at a fixed weight vector $\mathbf{b}$.

By re-writing problem (2.6) as problem (2.7), we can see that the deterministic policy SAA problem (2.7) can be regarded as a type of discrete optimization problem over the weight vector $\mathbf{b}$. (Note that the supremum in problem (2.7) is always attainable and can be replaced by a maximum, since the objective function $\hat{J}_D(\cdot)$ only takes finitely many values.) While this problem can be further re-formulated as a mixed-integer optimization problem, it is unlikely that one would be able to solve such a formulation to provable full or near optimality at a large scale (with tens of thousands or hundreds of thousands of trajectories). Moreover, the gradient of the objective function in problem (2.7), when it is defined, is always

zero due to the presence of the indicator function. This precludes the use of gradient-based methods, such as stochastic gradient descent, for solving the problem.

2.3.4 Randomized linear policies

Rather than solving problems (2.5) and (2.7), which optimize over deterministic linear policies, we can instead consider a problem where we optimize over randomized linear policies. In particular, given a collection of coefficients $\mathbf{b} = (\mathbf{b}_1, \ldots, \mathbf{b}_T)$ where $\mathbf{b}_1, \ldots, \mathbf{b}_T \in \mathbb{R}^K$ we consider randomized linear policies of the form

$$\tilde{\pi}_{\mathbf{b}}(t, \mathbf{x}) = \begin{cases} \textbf{stop} & \text{with probability } \sigma(\mathbf{b}_t \bullet \Phi(\mathbf{x})), \\ \textbf{continue} & \text{with probability } 1 - \sigma(\mathbf{b}_t \bullet \Phi(\mathbf{x})), \end{cases}$$

where $\sigma(u) = e^u/(1+e^u)$ corresponds to the logistic response function, and where the decision to stop in period t is independent of periods $1, \ldots, t-1$. Thus, given the coefficients in $\mathbf{b}$, the randomized policy $\tilde{\pi}_{\mathbf{b}}$ randomly chooses to stop with a logistic probability that depends on a weighted sum of basis functions.

The stopping time $\tau_{\tilde{\pi}}$ of a randomized policy $\tilde{\pi}$ is defined as follows. Conditional on a fixed trajectory $\{\mathbf{x}(t)\}_{t=1}^T$, the stopping time $\tau_{\tilde{\pi}}$ is a random variable, whose probability distribution is given by

$$\mathbb{P}(\tau_{\tilde{\pi}} = t \mid \mathbf{x}(1), \ldots, \mathbf{x}(T)) = \prod_{t'=1}^{t-1} (1 - \sigma(\mathbf{b}_{t'} \bullet \Phi(\mathbf{x}(t')))) \cdot \sigma(\mathbf{b}_t \bullet \Phi(\mathbf{x}(t))), \quad t = 1, \ldots, T,$$

$$\mathbb{P}(\tau_{\tilde{\pi}} = +\infty \mid \mathbf{x}(1), \ldots, \mathbf{x}(T)) = \prod_{t'=1}^{T} (1 - \sigma(\mathbf{b}_{t'} \bullet \Phi(\mathbf{x}(t')))).$$

With a slight abuse of notation, let $\mathcal{B} \subseteq \mathbb{R}^{KT}$ denote the set of feasible weight vectors for randomized policies, and define $\tilde{\Pi}_{\mathcal{B}}$ to be the set of feasible randomized policies:

$$\tilde{\Pi}_{\mathcal{B}} = \{\tilde{\pi}_{\mathbf{b}} \mid \mathbf{b} \in \mathcal{B}\}.$$

Thus, the expected reward of the randomized policy $\tilde{\pi}_{\mathbf{b}}$, where the expectation is taken over

both the stochastic process $\{\mathbf{x}(t)\}_{t=1}^{T}$ and the random stopping decisions can be written as

$$\underset{\tilde{\pi}\in\tilde{\Pi}_B}{\text{supremum}}\ \mathbb{E}[g(\tau_{\tilde{\pi}}, \mathbf{x}(\tau_{\tilde{\pi}}))], \tag{2.8}$$

or equivalently, as

$$\underset{b\in B}{\text{supremum}}\ \mathbb{E}\left[\sum_{t=1}^{T} g(t, \mathbf{x}(t)) \cdot \prod_{t'=1}^{t-1}(1 - \sigma(\mathbf{b}_{t'} \bullet \Phi(\mathbf{x}(t')))) \cdot \sigma(\mathbf{b}_t \bullet \Phi(\mathbf{x}(t)))\right], \tag{2.9}$$

where the expectation in problem (2.9) is now taken only over the stochastic process $\{\mathbf{x}(t)\}_{t=1}^{T}$. We shall use $J_R(\mathbf{b})$ to denote the objective function of problem (2.9) at a fixed $\mathbf{b} \in \mathcal{B}$.

Similarly to the deterministic problem, we can also consider a sample-average approximation of the true stochastic optimization problem (2.9). Given a sample of Ω trajectories as in Section 2.3.3, we can define the randomized policy SAA problem as

$$\underset{b\in B}{\text{supremum}}\ \frac{1}{\Omega}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} g(t, \mathbf{x}(\omega, t)) \cdot \prod_{t'=1}^{t-1}(1 - \sigma(\mathbf{b}_{t'} \bullet \Phi(\mathbf{x}(\omega, t')))) \cdot \sigma(\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega, t))). \tag{2.10}$$

In other words, we seek to find the coefficients $\mathbf{b} = (\mathbf{b}_1, \ldots, \mathbf{b}_T)$ so as to maximize the expected sample-average reward that arises from using these coefficients to effect randomized stopping decisions. We note that in problem (2.10), the optimization problem is formulated using the supremum. This is necessary, because although the objective function of (2.10) is continuous and bounded, the set $\mathcal{B}$ may not be compact, and therefore there may not have an attainable maximum. We shall use $\hat{J}_R(\mathbf{b})$ to denote the objective function of the randomized policy at a fixed weight vector $\mathbf{b} \in \mathcal{B}$.

2.3.5 Equivalence of deterministic and randomized policies

In this section, we investigate the connection between the deterministic policy problems laid out in Sections 2.3.2 and 2.3.3, and the randomized policy problems in Section 2.3.4. It turns out that under a small set of conditions, it is possible to show that the optimal objective values of the deterministic policy SAA problem (2.7) and the randomized policy SAA problem (2.10) are equivalent. With one additional assumption, it is also possible

to show that the optimal objective values of the deterministic and randomized policy true problems (problems (2.5) and (2.9) respectively) are also equivalent.

Recall that $J_D(\cdot)$, $\hat{J}_D(\cdot)$ $J_R(\cdot)$ and $\hat{J}_R(\cdot)$ are the respective objective functions of the deterministic policy true problem (2.5), the deterministic policy SAA problem (2.7), the randomized policy true problem (2.9) and the randomized policy SAA problem (2.10). For the purposes of the exposition of this section, we will use $\tilde{\mathbf{b}}$ to denote a vector of weights for the randomized policy problem, while $\mathbf{b}$ will be used to denote a vector of weights for the deterministic policy problem. We will also further disambiguate the sets of feasible weight vectors for the two problems by using $\mathcal{B}$ to denote the set of feasible weight vectors for the deterministic problem, and $\tilde{\mathcal{B}}$ the set of feasible weight vectors for the randomized problem.

Before stating our first result, we make two assumptions. Our first assumption is that the set of feasible weight vectors for the deterministic policy and randomized policy SAA problems are the same.

Assumption 2 $\mathcal{B} = \tilde{\mathcal{B}} = \mathbb{R}^{KT}$.

Our second assumption concerns the collection of basis functions.

Assumption 3 *The first basis function $\phi_1(\cdot)$ is the constant basis function, i.e., $\phi_1(\mathbf{x}) = 1$ for all $\mathbf{x} \in \mathcal{X}$.*

With these two assumptions, we state our first main result.

Theorem 1 *Under Assumptions 2 and 3 the objective values of problems* (2.7) *and* (2.10) *are equal, that is,*

$$\sup_{\mathbf{b}\in\mathcal{B}} \hat{J}_D(\mathbf{b}) = \sup_{\tilde{\mathbf{b}}\in\tilde{\mathcal{B}}} \hat{J}_R(\tilde{\mathbf{b}}).$$

The proof of Theorem 1 (see Appendix A.1.1) is based on two key ideas: (1) given a weight vector $\mathbf{b}$ of a deterministic policy, the same weight vector scaled by an arbitrarily

large positive constant α would result in the randomized policy behaving in the same (deterministic) way, since $\sigma(u) \to 1$ as $u \to \infty$ and $\sigma(u) \to 0$ as $u \to -\infty$; and (2) given a weight vector $\tilde{\mathbf{b}}$ of a randomized policy, one can view $\hat{J}_R(\tilde{\mathbf{b}})$ as the expectation of a deterministic policy with a particular basis function weight chosen randomly, so applying the probabilistic method implies the existence of a weight vector for a deterministic policy that performs at least as well as the randomized policy. With regard to the assumptions, Assumption 2 is a technical assumption that is necessary to be able to scale a deterministic weight vector into an appropriate randomized policy, as in idea (1), while Assumption 3 is a technical assumption that is necessary to avoid pathological cases where $\mathbf{b}_t \bullet \Phi(\mathbf{x}) = 0$ and to be able to appropriately apply the probabilistic method as in idea (2). From a practical perspective, Assumption 3 is not too restrictive, as it is common to use a constant basis function in implementations of ADP for optimal stopping.

Theorem 1 asserts that the SAA formulations of the two policy optimization problems are essentially equivalent. To establish equivalence of the true deterministic and randomized policy optimization problems (2.5) and (2.9), we need the following additional assumption, which concerns the stochastic process itself. We defer our discussion of this assumption until the statement of Theorem 2. To state this assumption, we let $\Phi_{2:K} : \mathcal{X} \to \mathbb{R}^{K-1}$ be defined as $\Phi_{2:K}(\mathbf{x}) = (\phi_2(\mathbf{x}), \ldots, \phi_K(\mathbf{x}))$, which is just the vector-valued mapping of the state $\mathbf{x}$ to the basis function values $\phi_2(\mathbf{x})$ through $\phi_K(\mathbf{x})$ (in other words, it is just the mapping Φ, only with the first basis function $\phi_1(\cdot)$ omitted).

Assumption 4 *For any hyperplane $A \subseteq \mathbb{R}^{K-1}$, i.e., a set of the form $A = \{\mathbf{y} \in \mathbb{R}^{K-1} \mid \mathbf{c} \bullet \mathbf{y} + d = 0\}$ for some $\mathbf{c} \in \mathbb{R}^{K-1}$, $d \in \mathbb{R}$, and any $t \in [T]$, $\mathbb{P}(\Phi_{2:K}(\mathbf{x}(t)) \in A) = 0$.*

We can now state our counterpart of Theorem 1 for the true stochastic optimization problems (2.9) and (2.5).

Theorem 2 *Under Assumptions 2, 3 and 4 the objective values of the randomized prob-*

lem (2.9) *and the deterministic problem* (2.5) *are equal, that is,*

$$\sup_{\mathbf{b}\in\mathcal{B}} J_D(\mathbf{b}) = \sup_{\tilde{\mathbf{b}}\in\tilde{\mathcal{B}}} J_R(\tilde{\mathbf{b}}).$$

The proof of Theorem 2 (see Appendix A.1.2) is similar to the proof of Theorem 1, but with several key differences. The most significant difference is that in the proof of Theorem 1, we show that a given deterministic linear policy can be approximated arbitrarily closely by a randomized policy. This is facilitated by Assumption 3, which allows one to avoid situations where the inner product of $\mathbf{b}_t$ and $\Phi(\mathbf{x}(\omega, t))$ is exactly zero in a given ω and t (since there are finitely many trajectories, one can perturb a given deterministic weight vector $\mathbf{b}$ into a new deterministic weight vector $\mathbf{b}'$ that has the same stopping behavior but never satisfies $\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega, t)) = 0$ for any ω and any t). In the true stochastic optimization problem setting, this is no longer possible. For this reason, we introduce Assumption 4, which requires that $\Phi_{2:K}(\mathbf{x}(t))$ has probability zero of being in any given hyperplane. This assumption allows us to avoid the aforementioned pathological cases where the stochastic process is such that, for a given non-zero weight vector $\mathbf{b}$ for the randomized policy problem, the inner product $\mathbf{b}_t \bullet \Phi(\mathbf{x}(t))$ may be exactly zero, which would mean the randomized policy would choose to stop or continue with equal probability.

With regard to Assumption 4, we note that this assumption holds for many, though not all, problem instances. For example, suppose that $\mathcal{X} \subseteq \mathbb{R}^n$ and $\phi_2(\mathbf{x}), \ldots, \phi_K(\mathbf{x})$ are polynomials of $\mathbf{x} \in \mathcal{X}$. In this case, the set $\{\mathbf{x} \in \mathcal{X} \mid \mathbf{c} \bullet \Phi_{2:K}(\mathbf{x}) + d = 0\}$ is the set of zeros of a polynomial function of $\mathbf{x}$, which is a measure zero set (Okamoto 1973). If we further assume that $\mathbf{x}(t)$ at each t has a bounded density, which is the case for many commonly used stochastic processes (e.g., geometric Brownian motion), then it immediately follows that $\mathbb{P}(\Phi_{2:K}(\mathbf{x}(t)) \in A) = 0$ for any hyperplane $A \subseteq \mathbb{R}^{K-1}$. As another example, suppose that $\mathcal{X} = \mathbb{R}^{K-1}$, and define E as $E = \Phi_{2:K}(\mathcal{X})$, the image of $\mathcal{X}$ under $\Phi_{2:K}(\cdot)$, which we assume to be an open subset of $\mathbb{R}^{K-1}$. Suppose also that the inverse function $\Phi_{2:K}^{-1}(\cdot)$

is defined on E and is continuously differentiable. Then the event $\Phi_{2:K}(\mathbf{x}(t)) \in A$ for a hyperplane $A \subseteq \mathbb{R}^{K-1}$ is equivalent to the event $\Phi_{2:K}(\mathbf{x}(t)) \in A \cap E$, which is equivalent to the event $\mathbf{x}(t) \in \Phi_{2:K}^{-1}(A \cap E)$. If A is a hyperplane in $\mathbb{R}^{K-1}$, it has measure zero, and so does $A \cap E$; and since $\Phi_{2:K}^{-1}(\cdot)$ is continuously differentiable, $\Phi_{2:K}^{-1}(A \cap E)$ is also a measure zero set in $\mathbb{R}^{K-1}$ (see Lemma 18.1 of Munkres 1991). If we again assume that each $\mathbf{x}(t)$ has a bounded density, then it again follows that $\mathbb{P}(\mathbf{x}(t) \in \Phi_{2:K}^{-1}(A \cap E)) = 0$ or equivalently, $\mathbb{P}(\Phi_{2:K}(\mathbf{x}(t)) \in A) = 0$. Where Assumption 4 could potentially fail is when the basis function mapping $\Phi_{2:K}(\cdot)$ collapses subsets of $\mathcal{X}$ to singletons, which could cause the probability of $\Phi_{2:K}(\mathbf{x}(t))$ being in certain hyperplanes to be non-zero.

We conclude this section by offering two remarks on Theorems 1 and 2. First, the significance of these two theorems is that in a certain sense, the problem of optimizing over deterministic policies and the problem of optimizing over randomized policies are the same. In the case of the true stochastic optimization problems, by solving the randomized problem (2.9), we can obtain a policy that performs as well as the one we would obtain by solving the deterministic problem (2.5). Similarly, in the case when we are working with a finite sample of trajectories, solving the randomized SAA problem (2.10) allows us to obtain a policy that performs as well as the one we would obtain by solving the deterministic SAA problem (2.7). From a practical perspective, the advantage of solving the randomized policy SAA problem (2.10), as opposed to the deterministic policy SAA problem (2.7), is that the objective function $\hat{J}_R(\cdot)$ is a differentiable function. Although $\hat{J}_R(\cdot)$ is non-convex due to the presence of the logistic response function $\sigma(\cdot)$, it is at least possible to approximately optimize $\hat{J}_R(\cdot)$ using gradient-based methods. The specific structure of $\hat{J}_R(\cdot)$ lends itself to an iterative algorithm that optimizes the weight vector $\tilde{\mathbf{b}}$ one period at a time, starting with the last period, that is reminiscent of the least-squares Monte Carlo (LSM) method; we defer our presentation of this algorithm to Section 2.5.2.

Second, we comment a little more on the motivation of our randomized policy optimization approach, in light of Theorems 1 and 2. Our interest in randomized linear policies

does not stem from some fundamental operational benefit that a randomized policy provides over a deterministic policy; stated differently, we do not wish to argue that in practice, a decision maker would want to make stopping decisions randomly as opposed to deterministically. Instead, our motivation for studying randomized policies is that the use of the logistic response function σ allows use to view the randomized policy true problem (2.9) and the SAA problem (2.10) as differentiable or "soft" counterparts to the deterministic policy problems (2.5) and (2.7), respectively, which are formulated using the indicator function $\mathbb{I}\{\cdot\}$ and involve making "hard" stopping decisions. Theorems 1 and 2 show that in general, this view is justified, as the deterministic and randomized problems are equal in objective value. As we will shortly see, the randomized policy SAA problem is amenable to an analysis of its convergence and generalization properties, and as we have already mentioned, is amenable to an intuitive heuristic for approximately solving it. Later, in Section 2.6, we will see numerically that using an approximate solution of the randomized policy SAA problem within a deterministic policy performs very well and can result in significant improvements over existing approaches.

2.4 Statistical properties

In this section, we investigate the statistical properties of the randomized policy SAA problem (2.10). In Section 2.4.1 we show that the objective value and optimal solution set of the randomized policy SAA problem converge almost surely to those of the true randomized policy problem. In Section 2.4.2, we establish guarantees on the out-of-sample performance of the solution obtained from the randomized policy SAA problem by characterizing the Rademacher complexity of the expected reward generated by a given set of weight vectors.

2.4.1 Convergence of randomized policy SAA problem

It is natural to expect that the optimal value and optimal solutions of the SAA problem (2.10) converge to their counterparts of the true optimization problem as the number of sample trajectories $\Omega \to \infty$. In this section, we provide Theorems 3 and 4 to establish these two convergence properties of our randomized policy SAA problem.

We first make the following two mild assumptions to facilitate the proofs of Theorems 3 and 4.

Assumption 5 *There exists a constant $Q > 0$ such that for any $\mathbf{x} \in \mathcal{X}$, $\|\Phi(\mathbf{x})\|_\infty \leq Q$.*

Assumption 6 *$\mathcal{B}$ is a compact subset of $\mathbb{R}^{KT}$.*

Note that we no longer carry Assumptions 2, 3 and 4. In particular, Assumption 2 is not relevant to Theorems 3 and 4, and Assumptions 3 and 4 are not required to establish our results here.

With these two assumptions, we can establish the following theorem which shows the almost sure uniform convergence of $\hat{J}_R(\cdot)$ to $J_R(\cdot)$ over the set $\mathcal{B}$.

Theorem 3 *Suppose that Assumptions 5 and 6 both hold. Then with probability one,*

$$\lim_{\Omega \to \infty} \sup_{\mathbf{b} \in \mathcal{B}} |\hat{J}_R(\mathbf{b}) - J_R(\mathbf{b})| = 0. \tag{2.11}$$

The proof of Theorem 3 is provided in Appendix A.1.3. It relies on the fact that the objective function $J_R(\cdot)$ in the true problem (2.9) and the objective function $\hat{J}_R(\cdot)$ in the SAA problem (2.10) have bounded Lipschitz constants, and the compactness of $\mathcal{B}$. Thus, we can use these two properties, together with the strong law of large numbers, to show uniform convergence.

Now, using Theorem 3, it is straightforward to derive the convergence of the SAA optimal objective value, which is stated in the following corollary.

Corollary 1 *Suppose that Assumptions 5 and 6 both hold. Then with probability one,*

$$\lim_{\Omega\to\infty} \sup_{\mathbf{b}\in\mathcal{B}} \hat{J}_R(\mathbf{b}) = \sup_{\mathbf{b}\in\mathcal{B}} J_R(\mathbf{b}). \tag{2.12}$$

For the convergence of the SAA optimal solutions, let us define the sets $\mathbf{B}^*$ and $\hat{\mathbf{B}}$ as

$$\mathbf{B}^* = \arg\max_{\mathbf{b}\in\mathcal{B}} J_R(\mathbf{b}),$$
$$\hat{\mathbf{B}} = \arg\max_{\mathbf{b}\in\mathcal{B}} \hat{J}_R(\mathbf{b}),$$

that is, $\mathbf{B}^*$ is the set of optimal solutions of the true stochastic problem (2.9) while $\hat{\mathbf{B}}$ is the set of optimal solutions to the SAA problem (2.10). In addition, let $\mathbb{D}(\hat{\mathbf{B}}, \mathbf{B}^*)$ be the *deviation* (see Chapter 7 of Shapiro et al. 2014) of the set $\hat{\mathbf{B}}$ from $\mathbf{B}^*$, that is,

$$\mathbb{D}(\hat{\mathbf{B}}, \mathbf{B}^*) = \sup_{\mathbf{b}\in\hat{\mathbf{B}}} \inf_{\mathbf{b}'\in\mathbf{B}^*} \|\mathbf{b} - \mathbf{b}'\|_2.$$

In the above definition, the inner infimum measures the distance between a given optimal solution $\mathbf{b}$ of the SAA problem (2.10) and the closest optimal solution of the true problem (2.9); the outer supremum then takes the largest such distance, over all optimal solutions of the SAA problem.

With these definitions, we can now apply Theorem 5.3 in Shapiro et al. (2014) to establish the following theorem:

Theorem 4 *Suppose that Assumptions 5 and 6 both hold. Then with probability one,* $\mathbb{D}(\hat{\mathbf{B}}, \mathbf{B}^*) \to 0$ *as* $\Omega \to \infty$.

Corollary 1 and Theorem 4 indicate that, given a sufficiently large sample size, the weight vector obtained by solving the SAA optimization problem (2.10) can be arbitrarily close to the optimal weight vector set of the true problem (2.9), and the corresponding optimal value of the SAA problem can be arbitrarily close to the optimal value of true problem.

2.4.2 Rademacher Complexity

In Section 2.4.1, we have seen from Theorem 3 that the optimal SAA objective value $\sup_{\mathbf{b}\in\mathcal{B}} \hat{J}_R(\mathbf{b})$ converges with probability one to the true optimal objective value $\sup_{\mathbf{b}\in\mathcal{B}} J_R(\mathbf{b})$ as the number of trajectories goes to infinity. However, in practice, we can only have access to a finite number of sample trajectories; in other words, there always exists some gap between $J_R(\mathbf{b})$ and $\hat{J}_R(\mathbf{b})$. Therefore, it is important to investigate how far $\hat{J}_R(\mathbf{b})$ could be away from $J_R(\mathbf{b})$ for a finite sample size and find good bounds on this gap. In this section, we will use a classical data-dependent complexity estimate of a function class, Rademacher complexity, to lower-bound the value of $J_R(\mathbf{b}) - \hat{J}_R(\mathbf{b})$, and provide three upper bounds on the Rademacher complexity term, corresponding to different choices of the weight vector set $\mathcal{B}$.

To establish this result, we require some additional definitions. We use Y to denote a system realization, which is a pair consisting of the sequence of states and the sequence of rewards, that is, $Y = (\{\mathbf{x}(t)\}_{t=1}^{T}, \{g(t, \mathbf{x}(t))\}_{t=1}^{T})$. We use $Y_1, \ldots, Y_\Omega$ to denote the sample of system realizations. We define the function $\Gamma : \mathbb{R}^T \times [0, \bar{G}]^T \to \mathbb{R}$ as

$$\Gamma(\mathbf{u}, \mathbf{v}) = \sum_{t=1}^{T} v_t \prod_{t'=1}^{t-1} (1 - \sigma(u_{t'}))\sigma(u_t), \tag{2.13}$$

where $\mathbf{u}, \mathbf{v} \in \mathbb{R}^T$. For a fixed weight vector $\mathbf{b} \in \mathcal{B}$, we define the function $\psi_{\mathbf{b}} : \mathcal{X}^T \times \mathbb{R}^T \to \mathbb{R}^{2T}$ which maps a system realization Y to a $2T$-dimensional vector as

$$\psi_{\mathbf{b}}(Y) = \begin{bmatrix} \mathbf{b}_1 \bullet \Phi(\mathbf{x}(1)) \\ \vdots \\ \mathbf{b}_T \bullet \Phi(\mathbf{x}(T)) \\ g(1, \mathbf{x}(1)) \\ \vdots \\ g(T, \mathbf{x}(T)) \end{bmatrix}. \tag{2.14}$$

We define $\mathcal{F} = \{\Gamma \circ \psi_{\mathbf{b}} \mid \mathbf{b} \in \mathcal{B}\}$ as the class of realization-to-reward functions. Note that for a fixed weight vector $\mathbf{b}$, the function value $(\Gamma \circ \psi_{\mathbf{b}})(Y)$ gives exactly the expected reward

of the randomized policy, where the expectation is taken over the stopping/continuation decisions, but conditional on the fixed system realization Y.

Lastly, we define the empirical Rademacher complexity $\hat{R}(\mathcal{F})$ as

$$\hat{R}(\mathcal{F}) = \frac{1}{\Omega}\mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{f\in\mathcal{F}}\sum_{\omega=1}^{\Omega}\epsilon_\omega f(Y_\omega)\right], \tag{2.15}$$

where $\epsilon_1, \ldots, \epsilon_\Omega$ are independent Rademacher random variables, that is, each ϵ_ω is equal to -1 or +1 with probability 1/2, and $\boldsymbol{\epsilon}$ is used to denote the vector of these random variables. We define the (ordinary) Rademacher complexity $R(\mathcal{F})$ as $R(\mathcal{F}) = \mathbb{E}_{Y_1,\ldots,Y_\Omega}[\hat{R}(\mathcal{F})]$.

Having set up the definitions of empirical Rademacher complexity and (ordinary/non-empirical) Rademacher complexity, Proposition 1 establishes the lower bounds of $J_R(\mathbf{b}) - \hat{J}_R(\mathbf{b})$ in terms of these two complexity terms.

Proposition 1 *Let $S = \{Y_1, \ldots, Y_\Omega\}$ be a collection of independent and identically distributed system realizations. For all $\delta > 0$, with probability at least $1-\delta$ over the sample S:*

$$J_R(\mathbf{b}) \geq \hat{J}_R(\mathbf{b}) - 2R(\mathcal{F}) - \bar{G}\sqrt{\frac{\log(1/\delta)}{2\Omega}}, \qquad \forall\ \mathbf{b}\in\mathcal{B} \tag{2.16}$$

$$J_R(\mathbf{b}) \geq \hat{J}_R(\mathbf{b}) - 2\hat{R}(\mathcal{F}) - 3\bar{G}\sqrt{\frac{\log(2/\delta)}{2\Omega}}, \qquad \forall\ \mathbf{b}\in\mathcal{B} \tag{2.17}$$

The proof of Proposition 1 is given in Appendix A.1.6; it follows the standard proof of generalization error bounds based on Rademacher complexity in statistical learning theory. We remark here that the generalization bounds established in Proposition 1 are different from those in classical statistical learning. Proposition 1 provides lower bounds on the true reward $J_R(\mathbf{b})$ in the form of the sample-based estimate $\hat{J}_R(\mathbf{b})$ minus a penalty term related to the complexity of our model; whereas in classical statistical learning problems, Rademacher complexity is used to upper-bound the true error in the form of the training error plus the complexity term. The reason for this difference is that our problem is to maximize the

expected reward, while the goal of classical statistical learning problem is to minimize some loss function.

The key quantities in Proposition 1 are the empirical and ordinary Rademacher complexities $R(\mathcal{F})$ and $\hat{R}(\mathcal{F})$. To understand how these quantities scale in the problem primitives and the structure of the admissible weight vector set $\mathcal{B}$, we have the following result, which provides deterministic bounds on $\hat{R}(\mathcal{F})$. (Note that since these bounds on $\hat{R}(\mathcal{F})$ hold almost surely, they are also valid bounds on $R(\mathcal{F})$.)

Theorem 5 *Suppose that Assumption 5 holds. Let $B \geq 0$. Then we have the following deterministic bounds for the empirical Rademacher complexity $\hat{R}(\mathcal{F})$:*

a) If $\mathcal{B} = \{\mathbf{b} \in \mathbb{R}^{KT} \mid \|\mathbf{b}\|_1 \leq B\}$, then $\hat{R}(\mathcal{F}) \leq \sqrt{2}(\bar{G}+1) \cdot \frac{BQ\sqrt{2\log(2KT)}}{\sqrt{\Omega}}$.

b) If $\mathcal{B} = \{\mathbf{b} \in \mathbb{R}^{KT} \mid \|\mathbf{b}\|_2 \leq B\}$, then $\hat{R}(\mathcal{F}) \leq \sqrt{2}(\bar{G}+1) \cdot \frac{BQ\sqrt{KT}}{\sqrt{\Omega}}$.

c) If $\mathcal{B} = \{\mathbf{b} \in \mathbb{R}^{KT} \mid \|\mathbf{b}\|_\infty \leq B\}$, then $\hat{R}(\mathcal{F}) \leq \sqrt{2}(\bar{G}+1) \cdot \frac{BQKT}{\sqrt{\Omega}}$.

The proof of Theorem 5 (see Appendix A.1.7) consists of two main steps. The first step is relating the Rademacher complexity of $\mathcal{F}$ to the Rademacher complexity of the class $\{\psi_{\mathbf{b}} \mid \mathbf{b} \in \mathcal{B}\}$. This involves the application of Maurer's vector contraction inequality (Maurer 2016), which is useful when a class of vector-valued functions is composed with a collection of scalar-valued Lipschitz functions, and can be used to relate the Rademacher complexity of the class of composite functions to the Rademacher complexity of the class of vector-valued functions. The outcome of this is that the Rademacher complexity of $\mathcal{F}$ can be written in terms of the Rademacher complexity of $\{\psi_{\mathbf{b}} \mid \mathbf{b} \in \mathcal{B}\}$; in the second step, we analyze the Rademacher complexity of this latter class by exploiting the structure of $\mathcal{B}$.

From this result, we can see that in all three cases, the Rademacher complexity scales gracefully with the problem dimension. In the worst case (when $\mathcal{B}$ is equal to the L_∞ norm ball; part c), it scales at most linearly with K and with T. This is partially driven by the fact that the function Γ is Lipschitz continuous (with respect to the L_2 norm) with

constant $\bar{G} + 1$. Importantly, this constant does not depend on T. This is not obvious, because the probability of stopping at period t is the product of t Lipschitz continuous and bounded functions, and so by standard properties of Lipschitz functions one should expect the Lipschitz constant to depend on T. It turns out that one can avoid a dependence on T because the products terms of the form $\prod_{t'=1}^{t-1}(1 - \sigma(\mathbf{b}_{t'} \bullet \Phi(\mathbf{x}(t'))))\sigma(\mathbf{b}_t \bullet \Phi(\mathbf{x}(t)))$ form a probability distribution. Consequently, the dependence on T in the bounds in Theorem 5 arises from the structure of the set $\mathcal{B}$, and not from the function Γ.

2.5 Solution Methodology

We now turn our attention to how one can actually solve the randomized policy SAA problem (2.10). In Section 2.5.1, we show that the randomized policy SAA problem is in general NP-Hard. Motivated by this, in Section 2.5.2 we propose an algorithm for approximately solving the SAA problem, based on backward induction. We conclude in Section 2.5.3 by comparing our proposed heuristic algorithm with the LSM algorithm.

2.5.1 Complexity of randomized policy SAA problem

Our main theoretical result on the solvability of the randomized policy SAA problem (2.10) is unfortunately a negative one.

Theorem 6 *The randomized policy SAA problem* (2.10) *is NP-Hard.*

We make a few remarks about this result. First, our proof of Theorem 6 (see Appendix A.1.8) involves considering the decision form of the randomized policy SAA problem (2.10), which asks whether there exists a weight vector $\mathbf{b}$ that achieves at least a certain target sample-average reward. By considering this decision problem, we show that for any instance of the decision form of the MAX-3SAT problem, a well-known NP-Complete problem, we can construct a corresponding instance of the randomized policy SAA problem such

that the answers to the two decision problems are identical. We note that the proof is not trivial, as the randomized policy SAA problem is in general a continuous problem, whereas MAX-3SAT is inherently discrete. In particular, showing that a positive answer to the SAA decision problem implies a positive answer to the MAX-3SAT problem involves viewing expressions involving $\sigma(\cdot)$ as expected values of expressions defined using a certain collection of i.i.d. random variables, and applying the probabilistic method to guarantee the existence of values for those random variables that can then be used to construct a solution to the MAX-3SAT problem. Most importantly, our proof does not achieve this equivalence by restricting the set of feasible weight vectors $\mathcal{B}$ to be a discrete set: the only restriction we place is to restrict the weight vectors be equal across time (i.e., $b_{t,k} = b_{t',k}$ for $t \neq t'$), which still results in $\mathcal{B}$ being uncountably infinite.

Second, we note that from an intuition standpoint, it is not reasonable to expect the randomized policy SAA problem (2.10) to be tractable. As alluded to before, this problem is a non-convex optimization problem, due to the presence of the function $\sigma(\cdot)$ that is neither convex nor concave. In addition, as $\sigma(u)$ can be viewed as a continuous approximation of the step function $\mathbb{I}\{u \geq 0\}$, one can expect the function $\hat{J}_R(\cdot)$ to have many local optima. In the next section, we consider a heuristic approach for solving the problem.

2.5.2 Backward optimization algorithm

Motivated by the fact that our randomized policy SAA problem (2.10) is theoretically intractable, we develop an iterative heuristic algorithm for solving the problem.

The high level idea of our heuristic is to solve problem (2.10) by optimizing over the weights one period at a time, starting from the last one. In particular, recall that $\mathbf{b} = (\mathbf{b}_1, \ldots, \mathbf{b}_T)$ and with a slight abuse of notation, let $\hat{J}_R(\mathbf{b}_1, \ldots, \mathbf{b}_T)$ denote the SAA objective value for the given collection of time-specific weight vectors. Assume also that that the set of feasible weight vectors is the Cartesian product of T period-wise weight vector sets, that is, $\mathcal{B} = \mathcal{B}_1 \times \ldots \times \mathcal{B}_T$, where $\mathcal{B}_1, \ldots, \mathcal{B}_T \subseteq \mathbb{R}^K$. The tth iteration of the algorithm involves

solving the single-period problem

$$\max_{\mathbf{b}'_t \in \mathcal{B}_t} \hat{J}_R(\mathbf{b}_1, \ldots, \mathbf{b}_{t-1}, \mathbf{b}'_t, \mathbf{b}_{t+1}, \ldots, \mathbf{b}_T) \tag{2.18}$$

and updating the tth weight vector in $\mathbf{b}$, which is $\mathbf{b}_t$, with the new solution $\mathbf{b}^*_t$. This process goes on from period $t = T$ all the way to $t = 1$; after the $t = 1$ iteration, the algorithm terminates. We formally define our procedure as Algorithm 1 below.

Algorithm 1 Backwards optimization algorithm for approximately solving the randomized policy SAA problem (2.10).

Initialize $\mathbf{b}_t \leftarrow \mathbf{0}$ for all $t \in [T]$.

Initialize $c_T(\omega) = 0$ for all $\omega \in [\Omega]$.

for $t = T, \ldots, 1$ **do**

Compute $p_t(\omega)$ as

$$p_t(\omega) = \prod_{t'=1}^{t-1} (1 - \sigma(\mathbf{b}_{t'} \bullet \Phi(\mathbf{x}(\omega, t')))). \tag{2.19}$$

Solve the problem

$$\max_{\mathbf{b}_t \in \mathcal{B}_t} \sum_{\omega=1}^{\Omega} \frac{1}{\Omega} \cdot p_t(\omega) \cdot [g(t, \mathbf{x}(\omega, t)) \cdot \sigma(\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega, t))) + c_t(\omega) \cdot (1 - \sigma(\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega, t))))] \tag{2.20}$$

to obtain an optimal solution $\mathbf{b}^*_t$.

Compute $c_{t-1}(\omega)$ as

$$c_{t-1}(\omega) = g(t, \mathbf{x}(\omega, t)) \cdot \sigma(\mathbf{b}^*_t \bullet \Phi(\mathbf{x}(\omega, t))) + c_t(\omega) \cdot (1 - \sigma(\mathbf{b}^*_t \bullet \Phi(\mathbf{x}(\omega, t)))). \tag{2.21}$$

end for

We pause to make several comments about Algorithm 1. First, observe that the period t problem solved in Algorithm 1, problem (2.20), is of a different form from problem (2.18). The two problems are equivalent in that problem (2.20) is a simplification of problem (2.18). In particular, $p_t(\omega)$ can be regarded as the probability, conditional on the weight vectors $\mathbf{b}_1, \ldots, \mathbf{b}_{t-1}$, of not having stopped by period t in trajectory ω. By using this term, we can

simplify the problem and remove the appearance of the weight vectors for periods prior to t. Similarly, $c_t(\omega)$ can be regarded as the expected continuation value at period t in trajectory ω, i.e., given that we have not stopped by period t, what is the expected reward (where the expectation is with respect to the randomness of the stopping decisions) from not stopping at period t, for the trajectory ω. Using both of these, and using the fact that $\hat{J}_R(\cdot)$ includes terms that only depend on $\mathbf{b}_{t'}$ for $t' < t$, we can boil problem (2.18) down to problem (2.20), which is of the form $\sum_\omega (c_\omega + d_\omega \cdot \sigma(\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega, t))))$.

Second, we note that problem (2.20) is still a challenging problem to solve, as the objective function is still non-convex. It is an instance of the sum-of-sigmoids problem (a sigmoid function being an S-shaped function, such as the logistic response function $\sigma(\cdot)$), which Udell and Boyd (2013) show to be NP-Hard in general. Similarly, Akçakuş and Mišić (2021) show that a related problem, of finding a binary product attribute vector that maximizes the expected market share under a mixture-of-logits model, is NP-Hard. However, problem (2.20) is more manageable to solve than the complete randomized policy SAA problem (2.10), as it involves only the weight variables for a single period (K variables) as opposed to all T periods (KT variables). In our implementation of Algorithm 1, we use the Adam algorithm (Kingma and Ba 2014) to approximately solve problem (2.20).

Lastly, we comment on how we use the solution $\mathbf{b}^* = (\mathbf{b}_1^*, \ldots, \mathbf{b}_T^*)$ produced by Algorithm 1. Although $\mathbf{b}^*$ corresponds to a randomized policy, in our numerical experiments we will focus on using $\mathbf{b}^*$ within a deterministic linear policy. In other words, we plug $\mathbf{b}^*$ into a policy of the form of equation (2.3). The reason for doing this is that in general, we have empirically observed that the deterministic policy defined with $\mathbf{b}^*$ performs better than the randomized policy defined with $\mathbf{b}^*$. To understand the intuition for this, let us consider problem (2.20). For this problem, a good weight vector $\mathbf{b}_t$ at time t would be one where, for most trajectories, $\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega, t))$ is very positive when $g(t, \mathbf{x}(\omega, t))$ is higher than $c_t(\omega)$, and where $\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega, t))$ is very negative when $c_t(\omega)$ is higher than $g(t, \mathbf{x}(\omega, t))$. When this is true for most trajectories, it is reasonable to expect that we could improve our objective

value by thresholding $\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega, t))$, i.e., rounding $\sigma(\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega, t)))$ to 0 or 1, which would have the effect of making the expression in the square brackets in problem (2.20) generally (i.e., for most trajectories) equal to $\max\{g(t, \mathbf{x}(\omega, t)), c_t(\omega)\}$, which is a higher quantity than $g(t, \mathbf{x}(\omega, t)) \cdot \sigma(\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega, t))) + c_t(\omega) \cdot (1 - \sigma(\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega, t))))$.

Besides this consideration, as discussed in Section 2.3.5, optimizing over randomized policies is equivalent to optimizing over deterministic policies, and our motivation for optimizing over randomized policies is to ultimately obtain good deterministic policies in a tractable manner. Lastly, we note that using $\mathbf{b}^*$ within a deterministic policy is similar to how in binary classification problems in machine learning, it is common to learn a probabilistic model whose natural output is a probability of a target class (for example, a logistic regression model), and to then threshold this probability to obtain a hard classification.

2.5.3 Comparison of backward optimization algorithm with least-squares Monte Carlo

Algorithm 1 shares some similarities with the least-squares Monte Carlo (LSM) algorithm of Longstaff and Schwartz (2001). For easier comparison, we state the basic LSM algorithm adapted to our problem setting as Algorithm 2 below.

In particular, the LSM algorithm also involves iterating backwards in time, and also involves updating the continuation value using the current policy. However, a key difference is that LSM involves solving a least-squares problem to obtain basis function weights $\mathbf{b}_t$, so as to predict the continuation value using those basis function weights. The stopping policy is then defined by comparing the current payoff to the predicted continuation value, where stopping is prescribed if and only if the current payoff is more than the predicted continuation value. In contrast, our algorithm involves directly optimizing over the stopping policy at a given period: in problem (2.20), we look for weights $\mathbf{b}_t$ for the stopping decision in the current period so that the expected reward, which accounts for both the current period's reward and the continuation value $c_t(\omega)$ that captures reward in future periods, is optimized.

Algorithm 2 Least-squares Monte Carlo (LSM) algorithm of Longstaff and Schwartz (2001).

Initialize $c_{T-1}(\omega) = g(T, \mathbf{x}(\omega, T))$ for all $\omega \in [\Omega]$.

for $t = T-1, \ldots, 1$ **do**

Solve the least-squares problem

$$\min_{\mathbf{b}_t \in \mathbb{R}^K} \frac{1}{2} \sum_{\omega=1}^{\Omega} (c_t(\omega) - \mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega, t)))^2 \tag{2.22}$$

to obtain an optimal solution $\mathbf{b}_t^*$.

Compute $c_{t-1}(\omega)$ as

$$c_{t-1}(\omega) = \begin{cases} c_t(\omega) & \text{if } \mathbf{b}_t^* \bullet \Phi(\mathbf{x}(\omega, t)) \geq g(t, \mathbf{x}(\omega, t)), \\ g(t, \mathbf{x}(\omega, t)) & \text{if } \mathbf{b}_t^* \bullet \Phi(\mathbf{x}(\omega, t)) < g(t, \mathbf{x}(\omega, t)). \end{cases} \tag{2.23}$$

end for

Besides this difference, it is also important to appreciate the higher level differences in the two approaches. In particular, LSM (Algorithm 2) produces a policy of the form

$$\pi(t, \mathbf{x}) = \begin{cases} \textbf{stop} & \text{if } g(t, \mathbf{x}) > \mathbf{b}_t \bullet \Phi(\mathbf{x}(t)), \\ \textbf{continue} & \text{if } g(t, \mathbf{x}) \leq \mathbf{b}_t \bullet \Phi(\mathbf{x}(t)). \end{cases}$$

Note that this policy can be made equivalent to a deterministic linear policy as we have defined it in Sections 2.3.2 and 2.3.3. Specifically, we can augment the state variable $\mathbf{x}(t)$ to include an additional coordinate that is equal to $g(t, \mathbf{x}(t))$ and then augment the basis function architecture $\Phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \ldots, \phi_K(\mathbf{x})))$ with a $K+1$th basis function $\phi_{K+1}(\cdot)$ that is exactly equal to this new coordinate. With these augmentations, the weight vector $\tilde{\mathbf{b}}_t = (-b_{t,1}, \ldots, -b_{t,K}, +1)$ is such that

$$g(t, \mathbf{x}) > \sum_{k=1}^{K} b_{t,k} \phi_k(\mathbf{x}(t)) \text{ if and only if } \sum_{k=1}^{K+1} \tilde{b}_{t,k} \phi_k(\mathbf{x}(t)) > 0,$$

i.e., the corresponding deterministic linear policy with the $K+1$ basis functions and the weight vectors $\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_T$ behaves identically to the LSM policy. Thus, LSM can be viewed

as a method for returning a solution to the deterministic policy SAA problem (2.7).

In light of this relationship, we note that, to our knowledge, there is no guarantee that the solution that LSM returns solves either the true deterministic policy problem (2.5) or the deterministic policy SAA problem (2.7). In contrast, Algorithm 1 is designed to directly (albeit approximately) solve the randomized policy SAA problem (2.10). Our results in Sections 2.3 and 2.4 provide theoretical justification for why this approach is desirable: under mild conditions, the true randomized policy problem (2.9) and its SAA counterpart (2.10) are equivalent to the true deterministic policy problem (2.5) and its SAA counterpart, respectively (guaranteed by our equivalence results, Theorems 1 and 2); as we accumulate more data, the optimal objective value and solution of the randomized policy SAA problem (2.10) converge to that of the true randomized policy problem (2.9) (guaranteed by our consistency results, Corollary 1 and Theorem 4); and optimizing the randomized policy SAA problem directly optimizes a lower bound on the out-of-sample reward that becomes tighter as one accumulates more data (guaranteed by our generalization bound and Rademacher complexity results, Proposition 1 and Theorem 5). Taken together, these results suggest that for a fixed basis function architecture, our method (Algorithm 1) has the potential to obtain policies that deliver better out-of-sample performance than LSM. In Section 2.6, we will showcase one family of benchmark problem instances where this is indeed the case.

2.6 Application to option pricing

In this section, we apply our randomized policy approach to a standard option pricing problem, previously considered in a number of papers (e.g., Desai et al. 2012, Ciocan and Mišić 2022). We define our option pricing problem in Section 2.6.1. In Section 2.6.2, we illustrate the difference between our randomized policy approach and prior approaches for obtaining deterministic linear policies using a simple option pricing problem involving a single asset. Then, in Section 2.6.3, we test our approach and compare it to prior approaches in a

higher dimensional setting with eight assets.

We implement our methods in the Julia programming language, version 0.6.4 (Bezanson et al. 2017). For the pathwise optimization method, we implement the pathwise linear program using the JuMP package (Lubin and Dunning 2015, Dunning et al. 2017) and solve it using Gurobi, version 9.5 (Gurobi Optimization, Inc. 2022). All our experiments are executed on Amazon Elastic Compute Cloud (EC2), on a single instance of type `r4.8xlarge` (Intel Xeon E5-2686 v4 processor with 32 virtual CPUs and 244 GBs of memory).

2.6.1 Background

The optimal stopping problem that we will focus on is pricing a Bermudan max-call option with a knock-out barrier, which was previously studied in Desai et al. (2012) and later in Ciocan and Mišić (2022). We consider the same family of problem instances used in those papers, and briefly review the details here.

In this family of problem instance, the option is dependent on n assets. The option is exercisable over a period of 3 calendar years with $T = 54$ equally spaced exercise times. The price of each underlying asset follows a geometric Brownian motion, with the drift set equal to the annualized risk-free rate r and the annualized volatility set to σ, and each asset is assumed to start at an initial price of $\bar{p}$. In all of the experiments that we will present, we shall assume $r = 5\%$ and $\sigma = 20\%$, as in Desai et al. (2012), and we will also assume the pairwise correlation between the assets to be zero. We use $p_i(t)$ denote the price of asset i at exercise time t.

The option has a strike price K and a knock-out barrier price B. The payoff of the option at any given time is determined by the strike price K, the knock-out barrier value B and the maximum price among the n underlying assets. If at time t the maximum price of the n underlying assets exceeds the barrier price B, the option is "knocked out" and the payoff becomes zero for all times $\tilde{t} \geq t$. We let $y(t)$ be an indicator variable that is 1 if the option

has not been knocked out by time t and zero otherwise:

$$y(t) = \mathbb{I}\left\{\max_{1\leq i\leq n, 1\leq t^{'}\leq t} p_i(t^{'}) < B\right\} \tag{2.24}$$

We let $g'(t)$ denote the (undiscounted) payoff from exercising the option at time t, which is defined as follows:

$$g'(t) = y(t) \cdot \max\left\{0, \max_{1\leq i\leq n} p_i(t) - K\right\}. \tag{2.25}$$

All payoffs are assumed to be discounted continuously according to the risk-free rate. This implies a discrete discount factor $\beta = \exp(-r \times 3/54) = 0.99723$. We can thus define the discounted reward $g(t)$ to be $g(t) = \beta^t \cdot g'(t)$, which can be thought of as the payoff denominated in dollars corresponding to time $t = 0$.

We compare three different methods: our randomized policy optimization (RPO) approach, the least-squares Monte Carlo (LSM) method of Longstaff and Schwartz (2001) and the pathwise optimization (PO) method of Desai et al. (2012). We test of each of these methods with a variety of basis functions. In our presentation of our results, we will denote the different sets of basis functions as follows:

- ONE: the constant basis function, equal to 1 for every state.
- PRICES: the price $p_i(t)$ of asset i for $i \in [n]$.
- PAYOFF: the undiscounted payoff $g'(t)$.
- KOIND: the knock-out (KO) indicator variable $y(t)$.
- PRICESKO: the KO adjusted prices $p_i(t) \cdot y(t)$ for $i \in [n]$.
- MAXPRICEKO and MAX2PRICEKO: the largest and second largest KO adjusted prices.
- PRICES2KO: the KO adjusted second-order price terms, $p_i(t) \cdot p_j(t) \cdot y(t)$ for $1 \leq i \leq j \leq n$.

In our implementation of the RPO approach, we use the backward algorithm, Algorithm 1. We use the coefficients obtained directly within a deterministic policy. We solve problem (2.20) using a custom implementation of Adam, a momentum-based first-order method (Kingma and Ba 2014, Goodfellow et al. 2016). We follow the parameter defaults in Kingma and Ba (2014), with the exception of the step size, for which we use 10^{-1}, as opposed to 10^{-3}. Additionally, we do not apply any minibatching, and compute the full gradient for the entire sample of Ω trajectories. For each solve of problem (2.20), we warm start the Adam algorithm using the coefficients obtained by LSM; we describe our warm starting scheme in more detail in Appendix A.2.1.

In our implementation of the pathwise optimization method, we follow Desai et al. (2012) in generating 500 inner samples.

2.6.2 Experiment #1: An illustrative example with $n = 1$

In our first experiment, to demonstrate the difference between our approach and incumbent approaches, we consider an instance of the option with $n = 1$ asset; thus, the undiscounted payoff and knock-out indicators can be written simply as

$$g'(t) = y(t) \cdot \max\{0, p_1(t) - K\}, \tag{2.26}$$

$$y(t) = \mathbb{I}\left\{\max_{1 \le t' \le t} p_1(t') < B\right\}. \tag{2.27}$$

We set $K = 100$ and $B = 150$, and vary $\bar{p}$ in the set $\{90, 100, 110\}$. For each initial price $\bar{p}$, we perform 10 replications, where in each replication we generate a set of $\Omega = 100,000$ trajectories to train each policy, and 100,000 trajectories for out-of-sample testing.

We test LSM with two basis function architectures: (i) ONE, and (ii) ONE and PAYOFF. Note that both of these basis function architectures imply an exercise policy that involves simply comparing the undiscounted payoff $g'(t)$ to a constant, state-independent threshold.

In particular, for (i), the exercise policy prescribes **stop** if and only if

$$\begin{aligned} g(t) &\geq b_{\text{ONE}} \cdot 1 \\ &= b_{\text{ONE}}, \end{aligned}$$

which is equivalent to

$$g'(t) \geq \beta^{-t} b_{\text{ONE}}.$$

For (ii), the exercise policy prescribes **stop** if and only if

$$g(t) \geq b_{\text{ONE}} \cdot 1 + b_{\text{PAYOFF}} \cdot g'(t).$$

Using the fact that $g(t) = \beta^t g'(t)$, we can re-arrange the above inequality into the following threshold rule in terms of the undiscounted payoff:

$$g'(t) \geq \frac{b_{\text{ONE}}}{\beta^t - b_{\text{PAYOFF}}},$$

which holds if $\beta^t - b_{\text{PAYOFF}} > 0$.

For the pathwise optimization method, we test it with the same two basis function architectures as LSM. Since the pathwise optimization-based policy is also a greedy policy based on an approximate continuation value function, one can again represent the policies obtained with the architectures (i) and (ii) as constant threshold policies. In addition to the policies, we also use the pathwise optimization solution to compute an upper bound on the optimal reward using an independent set of 100,000 trajectories (see Desai et al. 2012).

For the randomized policy approach, we test it with a single basis function architecture, consisting of ONE and PAYOFF. This results in an exercise policy where **stop** is recommended if and only if

$$b_{\text{ONE}} \times 1 + b_{\text{PAYOFF}} \times g'(t) > 0,$$

which is equivalent to the threshold rule

$$g'(t) > -\frac{b_{\text{ONE}}}{b_{\text{PAYOFF}}}$$

if $b_{\text{PAYOFF}} > 0$.

Table 2.1 shows the out-of-sample performance of the different methods under the different basis function architectures, as well as the pathwise optimization upper bounds. For each combination of a policy (a combination of one of the three methods – LSM, PO and RPO – and a basis function architecture) and an initial price $\bar{p}$, we report the average out-of-sample reward over the ten replications. We additionally report the standard error over those ten replications in parentheses.

From this table, we can see that even though the three methods – LSM, pathwise optimization and the randomized policy approach – produce policies within the same policy class, there are significant differences in performance. In particular, the policy produced by the randomized policy approach significantly outperforms LSM and pathwise optimization. Comparing to LSM with ONE, the randomized policy approach with ONE and PAYOFF attains an expected discounted reward that is as much as 89% higher. Comparing to LSM with ONE and PAYOFF, which in general performs better than LSM with ONE, the improvement by the randomized policy approach is as much as 7.7%. Comparing to PO with ONE and with ONE and PAYOFF, the randomized policy approach attains an improvement of up to 29% and 35%, respectively. In addition, the PO upper bounds are close to the performance of the randomized policy approach (for all three initial prices, the RPO lower bound is within 2.3% of the tightest PO upper bound). This suggests that for this problem setting, the policy is nearly optimal. This experiment highlights the fact that even for a simple problem instance involving only a single asset and the simplest possible policy class, the LSM method can return a policy that is substantially suboptimal.

It is also interesting to consider what the thresholds produced by the different methods look like. Figure 2.1 plots the thresholds for the five different policies at each period in the time horizon, for a single replication with $\bar{p} = 110$. We can see that there are substantial differences in the policies. The thresholds for the LSM policies are generally lower than those of the RPO policy, which implies that the LSM policies in general stop earlier in the time

Method	Basis functions	Initial price $\bar{p} = 90$	$\bar{p} = 100$	$\bar{p} = 110$
LSM	ONE	6.47 (0.010)	10.82 (0.011)	16.47 (0.008)
LSM	ONE, PAYOFF	11.37 (0.020)	16.64 (0.024)	22.01 (0.018)
PO	ONE	9.47 (0.017)	14.79 (0.017)	20.67 (0.014)
PO	ONE, PAYOFF	9.07 (0.032)	16.01 (0.029)	22.73 (0.023)
RPO	ONE, PAYOFF	**12.25 (0.018)**	**17.51 (0.023)**	**23.04 (0.018)**
PO-UB	ONE	18.26 (0.018)	25.47 (0.012)	32.49 (0.012)
PO-UB	ONE, PAYOFF	12.54 (0.009)	17.88 (0.009)	23.55 (0.005)

Table 2.1: Out-of-sample performance of different policies in $n = 1$ experiment.

horizon, when the reward will generally be lower. The PO policy with ONE also results in thresholds that are lower than the RPO policy. On the other hand, the PO policy with ONE and PAYOFF results in thresholds that are higher than those from RPO for roughly the first 40 periods; as a result, the PO policy may miss opportunities to stop earlier in the horizon. Interestingly, the thresholds for the LSM and PO policies begin rapidly decaying earlier in the time horizon than RPO (for LSM with ONE, LSM with ONE and PAYOFF, and PO with ONE, this starts right around the beginning of the horizon; for PO with ONE and PAYOFF, this starts at around $t = 34$). For RPO, there is a slow and steady decrease in the threshold until about $t = 48$, where the threshold begins to decrease much more quickly.

2.6.3 Experiment #2: multiple assets

In our second experiment, we consider instances of our option pricing problem with more than one asset. We specifically consider instances with n varying in $\{4, 8, 16\}$. As in the previous experiment, we vary $\bar{p}$ in $\{90, 100, 110\}$ and set the strike price $K = 100$. Following Desai et al. (2012), we set the barrier price $B = 170$. For each initial price $\bar{p}$ and each value

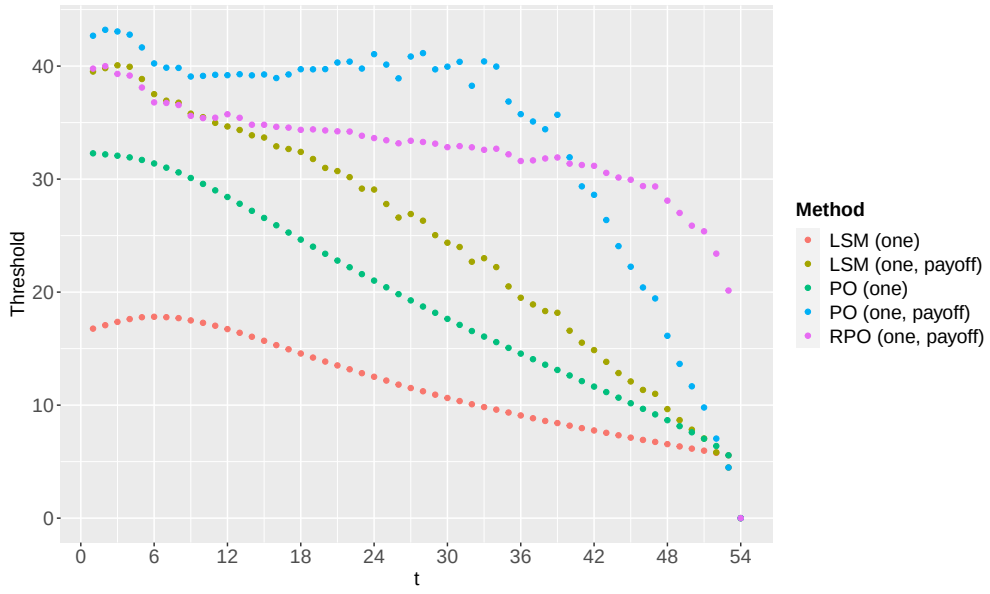

Figure 2.1: Plot of thresholds for policies in $n = 1$ experiment.

of n, we perform ten replications, where in each replication we generate a training set of $\Omega = 20,000$ trajectories, and a testing set of 100,000 trajectories. In what follows, we focus on the results for $n = 8$, and relegate the performance results for $n = 4$ and $n = 16$ to Appendix A.2.2.

We again test the LSM, PO and RPO methods with a variety of basis function architectures. We also obtain upper bounds from the PO method by reporting the objective value of the pathwise optimization linear program, which is a biased upper bound on the expected reward. We opt for this simpler approach over producing an unbiased upper bound (by generating an independent set of trajectories and the corresponding inner paths; see Desai et al. 2012) due to the significant computation time required in generating the inner paths. We note that this inexact approach has also been used in other work that has implemented the PO method (Ciocan and Mišić 2022).

Table 2.2 reports the out-of-sample performance of the LSM, PO and RPO methods, as well as the (biased) PO upper bound, for the different basis function architectures. Note

that the table is organized so that groups of policies corresponding to the same policy class are grouped together. (For example, LSM/PO with ONE and PRICES, LSM/PO with ONE, PRICES and PAYOFF, and RPO with ONE, PRICES and PAYOFF appear together.)

From this table, we can see that within each policy class, the RPO method in general outperforms the LSM method. In some cases the difference can be substantial: for example, with $\bar{p} = 90$ and the policy class corresponding to linear functions of KOIND and PAYOFF, the best LSM policy achieves a reward of 44.26 whereas RPO achieves a reward of 45.45, which is an improvement of 2.6%. Relative to the PO method, the performance of the RPO method in most cases is better, and in a few cases is slightly worse (for example, for $\bar{p} = 110$ and the PRICESKO, KOIND and PAYOFF policy class, the best PO policy attains a reward of 54.27 compared to 54.23 for the RPO policy).

In addition to the comparison of the methods within a fixed policy class, it is also insightful to compare the methods across policy classes, i.e., to think of what is the best attainable performance across any basis function architecture. In this regard, the highest rewards for all three initial prices are attained by the RPO method with KOIND and PAYOFF as the basis functions (45.45 for $\bar{p} = 90$, 51.37 for $\bar{p} = 100$, 54.50 for $\bar{p} = 110$). The best performance for the LSM method across any of the basis function architectures is substantially lower (44.26 for $\bar{p} = 90$, 50.07 for $\bar{p} = 100$, 53.46 for $\bar{p} = 110$). The best performance for the PO method is better, but still lower (44.79 for $\bar{p} = 90$, 50.91 for $\bar{p} = 100$, 54.35 for $\bar{p} = 110$).

Beside the performance, it is also useful to compare the methods in terms of computation time. Table 2.3 below shows the average computation time for each of the methods. For LSM, this is just the time to apply the LSM algorithm. For PO, this time includes the time to solve the PO linear program using Gurobi and the time to execute the regression, as well as the time to generate the inner paths and the time to formulate problem in JuMP. For RPO, this time is the time to apply the backward algorithm (Algorithm 1), which includes the time to solve the stage t problem (2.20) using Adam, but does not include the time to obtain the initial starting point using LSM.

Method	Basis function architecture	Initial price $\bar{p} = 90$	$\bar{p} = 100$	$\bar{p} = 110$
LSM	ONE	33.77 (0.023)	38.67 (0.010)	43.13 (0.013)
LSM	ONE, PAYOFF	41.18 (0.033)	43.21 (0.037)	45.00 (0.027)
PO	ONE	41.08 (0.015)	45.91 (0.021)	48.84 (0.016)
PO	ONE, PAYOFF	22.25 (0.177)	16.07 (0.144)	11.57 (0.119)
RPO	ONE, PAYOFF	**45.30 (0.022)**	**51.10 (0.012)**	**53.46 (0.053)**
PO-UB	ONE	52.19 (0.021)	57.45 (0.020)	60.35 (0.010)
PO-UB	ONE, PAYOFF	46.37 (0.024)	52.68 (0.051)	56.02 (0.047)
LSM	PRICES	33.81 (0.024)	38.54 (0.013)	43.02 (0.013)
LSM	PRICES, PAYOFF	39.56 (0.030)	41.74 (0.033)	44.12 (0.025)
PO	PRICES	40.93 (0.016)	44.83 (0.014)	47.49 (0.016)
PO	PRICES, PAYOFF	22.28 (0.124)	15.89 (0.116)	11.04 (0.091)
RPO	PRICES, PAYOFF	**44.49 (0.018)**	**49.77 (0.029)**	**52.23 (0.035)**
PO-UB	PRICES	51.40 (0.023)	57.20 (0.011)	60.32 (0.010)
PO-UB	PRICES, PAYOFF	46.36 (0.024)	52.64 (0.050)	55.94 (0.045)
LSM	PRICESKO	41.42 (0.017)	49.35 (0.017)	53.10 (0.009)
LSM	PRICESKO, PAYOFF	44.04 (0.017)	49.62 (0.012)	52.67 (0.006)
PO	PRICESKO	44.32 (0.017)	49.82 (0.015)	52.77 (0.018)
PO	PRICESKO, PAYOFF	44.18 (0.017)	50.06 (0.015)	53.19 (0.007)
RPO	PRICESKO, PAYOFF	**44.53 (0.019)**	**50.11 (0.013)**	**53.27 (0.010)**
PO-UB	PRICESKO	48.63 (0.015)	53.12 (0.010)	55.57 (0.011)
PO-UB	PRICESKO, PAYOFF	46.15 (0.023)	52.06 (0.034)	55.08 (0.024)
LSM	KOIND	39.37 (0.020)	48.09 (0.030)	53.26 (0.017)
LSM	KOIND, PAYOFF	44.26 (0.018)	50.07 (0.016)	53.19 (0.010)
PO	KOIND	43.87 (0.017)	50.85 (0.013)	54.35 (0.009)
PO	KOIND, PAYOFF	44.79 (0.025)	50.89 (0.013)	53.91 (0.008)
RPO	KOIND, PAYOFF	**45.45 (0.023)**	**51.37 (0.011)**	**54.50 (0.010)**
PO-UB	KOIND	49.29 (0.016)	53.47 (0.015)	55.69 (0.009)
PO-UB	KOIND, PAYOFF	46.15 (0.023)	52.07 (0.033)	55.05 (0.021)
LSM	PRICESKO, KOIND	41.84 (0.015)	49.37 (0.021)	53.46 (0.009)
LSM	PRICESKO, KOIND, PAYOFF	43.77 (0.019)	49.87 (0.018)	53.11 (0.007)
PO	PRICESKO, KOIND	44.01 (0.018)	**50.91 (0.013)**	**54.27 (0.008)**
PO	PRICESKO, KOIND, PAYOFF	43.98 (0.021)	50.69 (0.012)	53.84 (0.007)
RPO	PRICESKO, KOIND, PAYOFF	**44.08 (0.023)**	50.57 (0.031)	54.23 (0.010)
PO-UB	PRICESKO, KOIND	48.45 (0.020)	53.09 (0.011)	55.56 (0.010)
PO-UB	PRICESKO, KOIND, PAYOFF	46.14 (0.022)	52.05 (0.033)	55.04 (0.022)
LSM	PRICESKO, PRICES2KO, KOIND	43.32 (0.022)	49.86 (0.019)	53.26 (0.013)
LSM	PRICESKO, PRICES2KO, KOIND, PAYOFF	44.05 (0.022)	49.92 (0.019)	53.14 (0.012)
PO	PRICESKO, PRICES2KO, KOIND	44.33 (0.018)	**50.78 (0.014)**	53.93 (0.006)
PO	PRICESKO, PRICES2KO, KOIND, PAYOFF	**44.65 (0.018)**	50.65 (0.016)	53.77 (0.008)
RPO	PRICESKO, PRICES2KO, KOIND, PAYOFF	44.62 (0.015)	50.74 (0.021)	**54.03 (0.013)**
PO-UB	PRICESKO, PRICES2KO, KOIND	47.09 (0.016)	52.43 (0.019)	55.25 (0.010)
PO-UB	PRICESKO, PRICES2KO, KOIND, PAYOFF	46.09 (0.022)	51.98 (0.033)	55.00 (0.022)
LSM	PRICESKO, KOIND, MAXPRICEKO, MAX2PRICEKO	43.83 (0.018)	49.89 (0.023)	53.10 (0.008)
LSM	PRICESKO, KOIND, MAXPRICEKO, MAX2PRICEKO, PAYOFF	43.83 (0.017)	49.88 (0.022)	53.10 (0.008)
PO	PRICESKO, KOIND, MAXPRICEKO, MAX2PRICEKO	43.90 (0.026)	**50.66 (0.014)**	53.83 (0.008)
PO	PRICESKO, KOIND, MAXPRICEKO, MAX2PRICEKO, PAYOFF	44.04 (0.023)	50.65 (0.015)	53.82 (0.007)
RPO	PRICESKO, KOIND, MAXPRICEKO, MAX2PRICEKO, PAYOFF	**44.14 (0.016)**	50.55 (0.030)	**54.20 (0.010)**
PO-UB	PRICESKO, KOIND, MAXPRICEKO, MAX2PRICEKO	46.13 (0.017)	52.04 (0.033)	55.04 (0.022)
PO-UB	PRICESKO, KOIND, MAXPRICEKO, MAX2PRICEKO, PAYOFF	46.12 (0.023)	52.04 (0.033)	55.03 (0.021)

Table 2.2: Out-of-sample performance for different policies, for $n = 8$ assets.

Method	Basis function architecture	Initial price $\bar{p} = 90$	$\bar{p} = 100$	$\bar{p} = 110$
LSM	ONE	2.34 (0.248)	1.45 (0.020)	2.23 (0.245)
LSM	ONE, PAYOFF	2.22 (0.238)	2.21 (0.219)	2.32 (0.223)
PO	ONE	737.90 (49.467)	632.84 (39.330)	734.50 (55.495)
PO	ONE, PAYOFF	821.11 (20.357)	652.49 (22.014)	727.22 (25.883)
RPO	ONE, PAYOFF	80.65 (5.193)	332.82 (28.783)	305.96 (21.902)
LSM	KOIND	2.32 (0.159)	3.01 (0.207)	2.37 (0.267)
LSM	KOIND, PAYOFF	2.79 (0.318)	3.73 (0.357)	3.70 (0.335)
PO	KOIND	899.30 (21.038)	851.05 (18.996)	838.18 (27.319)
PO	KOIND, PAYOFF	944.45 (26.243)	819.51 (20.691)	822.24 (25.099)
RPO	KOIND, PAYOFF	4.87 (0.541)	16.86 (1.794)	20.41 (1.977)
LSM	PRICES	2.22 (0.190)	3.09 (0.286)	3.02 (0.218)
LSM	PRICES, PAYOFF	3.18 (0.294)	4.35 (0.291)	3.25 (0.203)
PO	PRICES	902.42 (27.296)	913.33 (25.858)	938.11 (18.843)
PO	PRICES, PAYOFF	1098.98 (21.250)	943.94 (33.928)	1070.94 (32.581)
RPO	PRICES, PAYOFF	174.12 (11.573)	565.27 (21.773)	584.55 (24.491)
LSM	PRICESKO	3.64 (0.334)	4.32 (0.493)	4.12 (0.383)
LSM	PRICESKO, PAYOFF	2.43 (0.248)	3.07 (0.367)	3.23 (0.303)
PO	PRICESKO	1163.28 (19.574)	1092.48 (13.862)	1093.83 (17.630)
PO	PRICESKO, PAYOFF	1269.60 (25.366)	1155.69 (22.082)	1158.41 (34.489)
RPO	PRICESKO, PAYOFF	6.55 (0.719)	14.52 (1.077)	14.56 (1.329)
LSM	PRICESKO, KOIND	3.90 (0.302)	5.05 (0.347)	4.62 (0.346)
LSM	PRICESKO, KOIND, PAYOFF	3.03 (0.454)	3.84 (0.165)	3.30 (0.371)
PO	PRICESKO, KOIND	1268.21 (35.401)	1099.33 (27.684)	1114.49 (23.758)
PO	PRICESKO, KOIND, PAYOFF	1395.44 (23.614)	1251.74 (35.740)	1202.02 (23.449)
RPO	PRICESKO, KOIND, PAYOFF	10.46 (1.729)	22.45 (1.831)	22.43 (2.184)
LSM	PRICESKO, PRICES2KO, KOIND	8.41 (0.353)	7.96 (0.429)	8.02 (0.552)
LSM	PRICESKO, PRICES2KO, KOIND, PAYOFF	6.61 (0.334)	11.39 (1.338)	8.59 (0.520)
PO	PRICESKO, PRICES2KO, KOIND	4712.40 (48.063)	4303.43 (186.668)	4824.68 (190.066)
PO	PRICESKO, PRICES2KO, KOIND, PAYOFF	3347.31 (21.754)	4884.33 (188.597)	4787.41 (150.816)
RPO	PRICESKO, PRICES2KO, KOIND, PAYOFF	38.18 (2.942)	87.08 (8.357)	66.91 (5.050)
LSM	PRICESKO, KOIND, MAXPRICEKO, MAX2PRICEKO	2.63 (0.136)	4.39 (0.388)	4.95 (0.431)
LSM	PRICESKO, KOIND, MAXPRICEKO, MAX2PRICEKO, PAYOFF	2.82 (0.176)	5.48 (0.480)	5.44 (0.513)
PO	PRICESKO, KOIND, MAXPRICEKO, MAX2PRICEKO	1026.37 (9.217)	1561.72 (50.908)	1534.24 (23.832)
PO	PRICESKO, KOIND, MAXPRICEKO, MAX2PRICEKO, PAYOFF	1012.27 (6.559)	1597.34 (37.282)	1491.44 (28.252)
RPO	PRICESKO, KOIND, MAXPRICEKO, MAX2PRICEKO, PAYOFF	12.37 (0.710)	37.34 (3.715)	33.59 (3.213)

Table 2.3: Computation time for different policies, for $n = 8$ assets.

From this table, we can see that LSM in general requires the least amount of computation time, requiring no more than 12 seconds on average. The RPO method requires more time, but in all cases its computation time is reasonable: in general, it requires no more than 585 seconds (approximately 10 minutes) on average. We note that the computation time for RPO is in general not monotonic in in the size of the basis function architecture: for example, RPO with ONE and PAYOFF (total of 2 basis functions) requires more time than RPO with PRICESKO, PRICES2KO, KOIND, PAYOFF (total of 46 basis functions). This is likely due to the non-convex nature of the objective function in the period t problem of the backward algorithm. In particular, with ONE and PAYOFF, the initial starting point produced by LSM could be further away from an approximately stationary point and Adam may require more iterations before termination, whereas with PRICESKO, PRICES2KO, KOIND and PAYOFF it may be closer and Adam may terminate more quickly.

Comparing to the PO method, we can see that the PO method requires a significantly larger amount of time than RPO, with the average computation time ranging from about 632 seconds ($\bar{p} = 100$, PO with ONE; just over 10 minutes) to 4884 seconds ($\bar{p} = 100$, PO with PRICESKO, PRICES2KO, KOIND, PAYOFF; roughly 80 minutes). The majority of this time comes from the generation of the inner paths, which is in general a computationally intensive task. Although RPO occasionally performs slightly worse than PO as we saw in Table 2.2, RPO may still be preferable to PO for obtaining good policies due to the significant computation times required by PO.

Lastly, we note that the computation time of the RPO method is sensitive to several implementation decisions. As alluded to above, the choice of starting point for problem (2.20), as well as the number of starting points used, will directly affect the time required for Adam to converge. Another decision is the step size used for Adam. In our experimentation, a smaller step size would lead to slower convergence, but would generally result in better solutions.

CHAPTER 3

Randomized Robust Price Optimization

3.1 Introduction

Price optimization is a key problem in modern business. The price optimization problem can be stated as follows: we are given a collection of products. We are given a demand model which tells us, for each product, what the expected demand for that product will be as a function of the price of that product as well as the price of the other products. Given this demand model, the price optimization problem is to decide a price vector – i.e., what price to set for each product – so as to maximize the total expected revenue arising from the collection of products.

The primary input to a price optimization approach is a demand model, which maps the price vector to the vector of expected demands of a product. However, in practice, the demand model is never known exactly, and must be estimated from data. This poses a challenge because data is typically limited, and thus a firm often faces uncertainty as to what the demand model is. This is problematic because a mismatch between the demand model used for price optimization – the nominal demand model – and the demand model that materializes in reality can lead to suboptimal revenues.

As a result, there has been much research in how to address demand model uncertainty in pricing. In the operations research community, a general framework for dealing with uncertainty is robust optimization. The idea of robust optimization is to select an uncertainty set, which is a set of values for the uncertain parameter that we believe could plausibly

occur, and to optimize the worst-case value of the objective function, where the worst-case is taken over the uncertainty set. In the price optimization context, one would construct an uncertainty set of potential demand models and determine the prices that maximize the worst-case expected revenue, where the worst-case is the minimum revenue over all of the demand models in the uncertainty set. In applying such a procedure, one can ensure that the performance of the chosen price vector is good under a multitude of demand models, and that one does not experience the deterioration of a price vector optimized from a single nominal demand model.

Typically in robust optimization, the robust optimization problem is to find the *single* best decision that optimizes the worst-case value of the objective function. Stated in a slightly different way, one *deterministically* implements a single decision. However, a recent line of research (Delage et al. 2019) has revealed that with regard to the worst-case objective, it is possible to obtain better performance than the traditional deterministic robust optimization approach by *randomizing* over multiple solutions. Specifically, instead of optimizing over a single decision in some feasible set that optimizes the worst-case objective, one optimizes over a *distribution* supported on the feasible set that informs the decision maker how to randomize.

In this chapter, we propose a methodology for robust price optimization that is based on randomization. In particular, we propose solving a randomized robust price optimization (RRPO) problem, which outputs a probability distribution that specifies the frequency with which the firm should use different price vectors. From a practical perspective, such a randomization scheme has the potential to be implemented in modern retailing as a strategy for mitigating demand uncertainty. In particular, in an ecommerce setting, randomization is already used for A/B testing, which involves randomly assigning some customers to one experimental condition and other customers to a different experimental condition. Thus, it is plausible that the same form of randomization could be used to display different price vectors with certain frequencies. In the brick-and-mortar setting, one can potentially implement

randomization by varying prices geographically or temporally. For example, if the RRPO solution is the three price vectors $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ with probabilities 0.2, 0.3, 0.5, then for a set of 50 regions, we would choose 10 ($= 50 \times 0.2$) to assign to price vector $\mathbf{p}_1$, 15 ($= 50 \times 0.3$) to assign to $\mathbf{p}_2$, and 25 ($= 50 \times 0.5$) to assign to $\mathbf{p}_3$. Similarly, if one were to implement the same randomization scheme temporally, then for a selling horizon of 20 weeks, one would use the price vector $\mathbf{p}_1$ for 4 ($= 20 \times 0.2$) weeks, $\mathbf{p}_2$ for 6 ($= 20 \times 0.3$) weeks and $\mathbf{p}_3$ for 10 ($= 20 \times 0.5$) weeks.

We make the following specific contributions:

1. **Benefits of randomization**. We formally define the RRPO problem and analyze it under different conditions to determine when the underlying robust price optimization problem is *randomization-receptive* – there is a benefit from implementing a randomized decision over a deterministic decision – versus when it is *randomization-proof* – the optimal randomized and deterministic decisions perform equally well. We show that the robust price optimization problem is randomization-proof in several interesting settings, which can be described roughly as follows: (1) when the set of feasible price vectors is convex and the set of uncertain revenue functions is concave; (2) when the set of feasible price vectors and the set of uncertain demand parameters are convex, and the revenue function obeys certain quasiconvexity and quasiconcavity properties with respect to the price vector and the uncertain parameter; and (3) when the set of feasible price vectors is finite, and a certain minimax property holds. We showcase a number of examples of special cases that satisfy the hypotheses of these results and consequently are randomization-proof. We also present several examples showing how these results can fail to hold when certain assumptions are relaxed and the problem thus becomes randomization-receptive.

2. **Tractable solution algorithms**. We propose algorithms for solving the RRPO problem in two different settings:

(a) In the first setting, we assume that the set of possible price vectors is a finite set and that the uncertainty set of demand function parameters is a convex set. In this setting, when the revenue function is quasiconvex in the uncertain parameters, we show that the RRPO problem can be solved via delayed constraint generation. The separation problem that is solved to determine which constraint to add is exactly the nominal pricing problem for a fixed uncertain parameter vector of the demand function. For the log-log and semi-log demand models, we show that this nominal pricing problem can be reformulated and solved to global optimality as a mixed-integer exponential cone program. We believe these reformulations are of independent interest as to the best of our knowledge, these are the first exact mixed-integer convex formulations for these problems in either the marketing or operations literatures, and they leverage recent advances in solution technology for mixed-integer conic programs (as exemplified in the conic solver Mosek).

(b) In the second setting, we assume that both the price set and the uncertainty set are finite sets. In this setting, we show how the RRPO problem can be solved using a double column generation method, which involves iteratively generating new uncertainty realizations and price vectors by solving primal and dual separation problems, respectively. We show how the primal and dual separation problems can be solved exactly for the linear, semi-log and log-log demand models.

3. **Numerical evaluation with synthetic and real data**. We evaluate the effectiveness of randomized pricing on different problem instances generated synthetically and problem instances calibrated with real data. Using synthetic data instances, we show that randomized pricing can improve worst-case revenues by as much as 1300% over deterministic pricing, while in our real data instances, the benefit can be as high as 92%. Additionally, we show that for instances of realistic size (up to 20 products), our algorithm can solve the RRPO problem in a reasonable amount of time (no more than four minutes on average).

The rest of this chapter is organized as follows. Section 3.2 reviews the related literature on pricing, robust optimization and randomized robust optimization. Section 3.3 formally defines the nominal price optimization problem, the deterministic robust price optimization problem and the randomized robust price optimization problem. Section 3.4 analyzes the robust price optimization problem and provides conditions under which the price optimization problem is randomization-receptive and randomization-proof. Section 3.5 presents our constraint generation approach for solving the RRPO problem when the price set is a finite set and the uncertainty set is a convex set, and discusses how this approach can be adapted for different families of demand models. Section 3.6 provides a brief overview of our methodology for solving the RRPO problem when the price set and uncertainty sets are finite sets, with the details provided in Section B.3 of the companion. Section 3.7 presents our numerical experiments.

3.2 Literature review

Our work is closely related to three streams of research: pricing, robust optimization and the use of randomized strategies in optimization. We discuss each of these three streams below.

Pricing optimization and demand models. Optimal pricing has been extensively studied in many fields such as revenue management and marketing research; for a general overview of this research area, we refer readers to Soon (2011) and Gallego and Topaloglu (2019). An important stream of pricing literature is on static pricing, which involves setting a fixed price for a product. The most commonly considered demand models in the static pricing literature are the linear and log-log models. For example, the papers of Zenor (1994) and Bernstein and Federgruen (2003) assume linear demand functions in the study of pricing strategies. The papers of Reibstein and Gatignon (1984), and Montgomery and Bradlow (1999) use log-log (multiplicative) demand functions to represent aggregate demand. The paper of Kalyanam

(1996) considers a semi-log demand model. Besides linear, semi-log and log-log demand functions, another type of demand form that is extensively discussed in pricing literature is based on an underlying discrete choice model. For example, Hanson and Martin (1996), Aydin and Ryan (2000), and Hopp and Xu (2005) consider the product line pricing problem under the multinomial logit (MNL) model. Keller et al. (2014) consider attraction demand models which subsume MNL models. Li and Huh (2011) study the pricing problem with the nested logit (NL) models, and show the concavity of the profit function with respect to market share holds. Gallego and Wang (2014) characterize the optimal pricing structure under the general nested logit model with product-differentiated price sensitivities and arbitrary nest coefficients. The papers of Keller (2013) and Zhang et al. (2018) study the multiproduct pricing problem under the family of generalized extreme value (GEV) models which includes MNL and NL models as special cases. Our work differs from this prior work on multiproduct pricing in that the demand model is not assumed to be known, and that there is an uncertainty set of plausible demand models that could actually materialize. Correspondingly, the firm is concerned not with expected revenue under a single, nominal demand model, but with the worst-case revenue with respect to this uncertainty set of demand models.

Another significant stream of pricing literature is to consider multiple-period pricing decisions where the prices of products change over time and there is a fixed inventory of each product; we refer readers to McGill and van Ryzin (1999), Elmaghraby and Keskinocak (2003), Bitran and Caldentey (2003), and Talluri and van Ryzin (2004) for a comprehensive review on dynamic pricing strategies with inventory considerations. As in the static pricing literature, studies in dynamic pricing also vary in the types of demand models. Besbes and Zeevi (2015) assume linear demand in a multiperiod single product pricing problem, and show that the corresponding pricing policy can perform well even under model misspecification. Caro and Gallien (2012) consider multiplicative models where the demand rate and price discount have the logarithmic relationship, and consider a multiproduct clearance pricing optimization problem for the fast-fashion retailer Zara. Akçay et al. (2010) consider

dynamic pricing under MNL models for horizontally differentiated products and show that the profit function is unimodal in prices, while Song et al. (2021) and Dong et al. (2009) reformulate the MNL profit as a concave function of its market share rather than prices. Our work focuses on the static, single-period setting where there is no inventory consideration, and is thus not directly related to this stream of the pricing literature.

Robust Optimization. Within the literature mentioned above, either the probability distributions of demand are assumed to be known exactly or the demand models are estimated from historical data. However, in practice, the decision maker often has no access to the complete information of demand distributions. Also, in many real applications, the lack of sales data makes it hard to obtain a good estimation of demand models, which leads to model misspecification and thus suboptimal pricing decisions. In operations research, this type of challenge is most commonly addressed using the framework of robust optimization, where uncertain parameters are assumed to belong to some uncertainty set and one optimizes for the worst-case objective of parameters within the set. We refer readers to Ben-Tal et al. (2009), Bertsimas et al. (2011), Gabrel et al. (2014) and Bertsimas and den Hertog (2022) for a detailed overview of this approach. Robust optimization has been widely applied in various problem settings such as assortment optimization (Rusmevichientong and Topaloglu 2012, Bertsimas and Mišić 2017, Sturt 2021b), inventory management (Bertsimas and Thiele 2006, Govindarajan et al. 2021) and financial option pricing (Bandi and Bertsimas 2014, Sturt 2021a).

Within this literature, our work contributes to the substream that considers robust optimization for pricing. Thiele (2009) consider tractable robust counterparts to the deterministic multiproduct pricing problem with the budget-of-resource-consumption constraint in the case of additive demand uncertainty, and investigate the impact of uncertainty on the optimal prices of multiple products sharing capacitated resources. Mai and Jaillet (2019) consider robust multiproduct pricing optimization under the generalized extreme value (GEV)

choice model and characterize the robust optimal solutions for unconstrained and constrained pricing problems. Hamzeei et al. (2021) study the robust pricing problem with interval uncertainty of the price sensitivity parameters under the multi-product linear demand model. For robust dynamic pricing problems, Lim and Shanthikumar (2007) and Lim et al. (2008) use relative entropy to represent uncertainty in the demand rate and thus the demand uncertainty can be expressed through a constraint on relative entropy. Perakis and Sood (2006), Adida and Perakis (2010) and Chen and Chen (2018) all study robust dynamic pricing problems with demand uncertainty modeled by intervals. Harsha et al. (2019) study robust dynamic price optimization on an omnichannel network with cross-channel interactions in demand and supply where demand uncertainty is modeled through budget constraints. From a different perspective, Cohen et al. (2018) develop a data-driven framework for solving the robust dynamic pricing problem by directly using samples in the optimization. Specifically, the paper considers three types of robust objective (max-min, min-max regret and max-min ratio), and uses the given sampled scenarios to approximate the uncertainty set by a finite number of constraints.

Our work differs from the majority of this body of work that takes a robust optimization approach to pricing in that the decision we seek to make is no longer a deterministic decision, but a randomized one. Within this body of work, the papers closest to our work are Allouah et al. (2021) and Allouah et al. (2022). The paper of Allouah et al. (2021) considers a pricing problem under a valuation-based model of demand, where each customer has a valuation drawn from an unknown cumulative distribution function on the positive real line, and the firm only has one historical data point. The paper considers the pricing problem from a max-min ratio standpoint, where the firm seeks to find a pricing mechanism that maximizes the percentage attained of the true maximum revenue under the worst-case (minimum) valuation distribution consistent with the data point. The paper shows the approximation rate that is attainable given knowledge of different quantiles of the valuation distribution when the valuation distribution is a regular distribution or a monotone non-decreasing hazard rate

distribution. The mechanisms that are proposed in the paper, which are mappings from the data point to a price, include deterministic ones that offer a fixed price, as well as randomized ones that offer different prices probabilistically. The paper of Allouah et al. (2022) considers a similar setting, where instead of knowing a point on the valuation CDF exactly or to within an interval, one has access to an IID sample of valuations drawn from the unknown valuation CDF, and similarly proposes deterministic and randomized mechanisms for this setting.

With regard to Allouah et al. (2021) and Allouah et al. (2022), our setup differs in a number of ways. First, our methodology focuses on a max-min revenue objective, as opposed to a max-min ratio objective that considers performance relative to oracle-optimal revenue. Our methodology also does not start from a valuation model, but instead starts from an aggregate demand model, and additionally considers the multi-product case in the general setting. Additionally, we do not take data as a starting point, but instead assume that the aggregate demand model is uncertain. Lastly, the overarching goals are different: while Allouah et al. (2021) and Allouah et al. (2022) seek to understand the value of data, and how well one can do with limited data, our goal is to demonstrate that from the perspective of worst-case revenue performance, a randomized pricing strategy can be preferable over a deterministic fixed price, and to develop tractable computational methods for computing such strategies under commonly used demand models in the multi-product setting.

Randomized strategies in optimization under uncertainty. The conventional robust optimization problems mentioned above only consider deterministic solutions. In recent years, the benefit of using randomized strategies has received increasing attention in the literature on decision making under uncertainty and robust optimization. Mastin et al. (2015) study randomized strategies for min-max regret combinatorial optimization problems in the cases of interval uncertainty and uncertainty representable by discrete scenarios, and provide bounds on the gains from randomization for these two cases. Bertsimas et al. (2016b) consider randomness in a network interdiction min-max problem where the interdictor can

benefit from using a randomized strategy to select arcs to be removed.

The paper of Delage et al. (2019) considers the problem of making a decision whose payoff is uncertain and minimizing a risk measure of this payoff, and studies under what circumstances a randomized decision leads to lower risk than a deterministic decision. The paper characterizes the classes of randomization-receptive and randomization-proof risk measures in the absence of distributional ambiguity (i.e., classical stochastic programs), and discusses conditions under which problems with distributional ambiguity (i.e., distributionally robust problems) can benefit from randomized decisions.

Subsequently, the paper of Delage and Saif (2022) studies the value of randomized solutions for mixed-integer distributionally robust optimization problems. The paper develops bounds on the magnitude of improvement given by randomized solutions over deterministic solutions, and proposes a two-layer column generation method for solving single-stage and two-stage linear DRO problems with randomization. Our work relates to Delage and Saif (2022) in that we apply a similar two-layer column generation approach for solving the randomized robust pricing problem when the price set and the uncertainty set are both finite; we discuss this connection in more detail in our discussion of the paper of Wang et al. (2024) below. The paper of Sadana and Delage (2023) develops a randomization approach for solving a distributionally robust maximum flow network interdiction problem with a conditional-value-at-risk objective, which is also solved using a column generation approach.

Our work is most closely related to the excellent paper of Wang et al. (2024). The paper of Wang et al. (2024) introduces randomization into the robust assortment optimization and characterizes the conditions under which a randomized strategy strictly improves worst-case expected revenues over a deterministic strategy. The paper proposes several different solution methods for finding an optimal distribution over assortments for the MNL, Markov chain and ranking-based models. For the MNL model in particular, the paper adapts the two-layer column generation method of Delage and Saif (2022) to solve the randomized robust assortment optimization problem when the uncertainty set is discrete.

Our work shares a high-level viewpoint with the paper of Wang et al. (2024) in that revenue management decisions, such as assortment decisions and pricing decisions, are subject to uncertainty and from an operational point of view, have the potential to be randomized and to benefit from randomization. From a technical standpoint, several of our results on the benefit of randomization when the price set is discrete that are stated in Section 3.4.3 are generalizations of results in Wang et al. (2024) to the pricing setting that we study. In terms of methodology, the solution approach we apply when the price set and uncertainty set are discrete in Section 3.6 (described more fully in Section B.3) is related to the approach in Wang et al. (2024) for the MNL model, as we also use the two-layer column generation scheme of Delage and Saif (2022). The main difference between the method in Wang et al. (2024) and our method lies in the nature of the subproblems. In the paper of Wang et al. (2024), the primal subproblem is a binary sum of linear fractional functions problem, and the dual subproblem is essentially a mixture of multinomial logits assortment problem that can be reformulated as a mixed-integer linear program. In our work, the primal and dual subproblems that are used to generate new price vectors and uncertainty realizations comes from the underlying pricing problem and the structure of different demand models (linear, semi-log and log-log), which lead to different subproblems than in the assortment setting. In particular, in the semi-log and log-log cases, both the primal and dual subproblems can be formulated as mixed-integer exponential cone programs. In the semi-log and log-log cases specifically, the formulations of the dual subproblems, which are used to identify new price vectors to add, require a logarithmic transformation together with a biconjugate representation of the log-sum-exp function. This technique is also used to develop a constraint generation scheme for the randomized robust pricing problem when the price set is discrete and the uncertainty set is convex (Section 3.5); the subproblem in this case involves solving a nominal pricing problem under the semi-log or log-log model, which we are also able to reformulate exactly as a mixed-integer exponential cone program. As noted in the introduction, we believe these are the first exact formulations of these problems using mixed-integer

conic programming.

Other work on randomization. Lastly, we comment on several streams of work that use randomization but are unrelated to our work. Within the revenue management community, there are instances where randomization is an operational aspect of the algorithm. For example, in network revenue management, the heuristic of probabilistic allocation control involves using the primal variable values in the deterministic linear program (DLP) to decide how frequently requests should be accepted or rejected (Jasin and Kumar 2012). Here, randomization is used to ensure that the long run frequency with which different requests are accepted or rejected is as close as possible to the DLP solution, which corresponds to an idealized upper bound on expected revenue. As another example, randomization can be used for solving large-scale linear programs. For instance, Akchen and Mišić (2024) develop a randomized method which involves sampling a collection of columns and solving the corresponding restricted linear program. Here, randomization is a technique to avoid computational challenge due to column generation. Besides, randomization is often also a part of methods for problems that involve learning. For example, Ferreira et al. (2018) propose a method for network revenue management where there is uncertainty in demand rates based on Thompson sampling, which is a method from the bandit literature that involves taking a random sample from the posterior distribution of an uncertain parameter and taking the action that is optimal with respect to that sample. Here, randomization is a way of ensuring that the decision maker explores possibly suboptimal actions. In our work, the focus is not on using randomization to achieve better expected performance, using randomization to achieve low computational costs or using randomization to achieve a balance between exploration and exploitation, but rather to operationalize randomization to achieve better worst-case performance.

3.3 Problem definition

In this section, we begin by defining the nominal price optimization problem in Section 3.3.1. We subsequently define the deterministic robust price optimization problem in Section 3.3.2. Lastly, we define the randomized robust price optimization problem in Section 3.3.3.

3.3.1 Nominal price optimization problem

We assume that the firm offers I products, indexed from 1 to I. We let p_i denote the price of product $i \in [I]$, where we use the notation $[n] = \{1, \ldots, n\}$ for any positive integer n. We use $\mathbf{p} = (p_1, \ldots, p_I)$ to denote the vector of prices. We assume that the price vector $\mathbf{p}$ is constrained to lie in the set $\mathcal{P} \subseteq \mathbb{R}_+^I$, where $\mathbb{R}_+$ is the set of nonnegative real numbers.

We let d_i denote the demand function of product i, so that $d_i(\mathbf{p})$ denotes the demand of product i when the price vector $\mathbf{p}$ is chosen. The revenue function $R(\cdot)$ can then be written as $R(\mathbf{p}) = \sum_{i=1}^{I} p_i \cdot d_i(\mathbf{p})$.

The nominal price optimization (NPO) problem can be written simply as

$$\text{NPO}: \quad \max_{\mathbf{p} \in \mathcal{P}} R(\mathbf{p}).$$

There are numerous demand models that can be used in practice, which lead to different price optimization problems; we briefly review some of the more popular ones here.

1. *Linear demand model*: A linear demand model is defined by parameters $\boldsymbol{\alpha} \in \mathbb{R}^I$, $\boldsymbol{\beta} \in \mathbb{R}^I$, $\boldsymbol{\gamma} = (\gamma_{i,j})_{i,j \in [I], i \neq j} \in \mathbb{R}^{I \cdot (I-1)}$. The demand function $d_i(\cdot)$ of each product $i \in [I]$ has the form

$$d_i(\mathbf{p}) = \alpha_i - \beta_i p_i + \sum_{j \neq i} \gamma_{i,j} p_j, \tag{3.1}$$

where $\beta_i \geq 0$ is the own-price elasticity parameter of product i, which indicates how much demand for product i is affected by the price of product i, whereas $\gamma_{i,j}$ is a cross-price elasticity parameter that describes how much demand for product i is affected

by the price of a different product j. Note that $\gamma_{i,j}$ can be positive, which generally corresponds to products i and j being substitutes (i.e., when the price of product j increases, customers tend to switch to product i), or negative, which corresponds to products i and j being complements (i.e., products i and j tend to be purchased together, so when the price of product j increases, this causes a decrease in demand for product i). The corresponding revenue function $R(\cdot)$ is then

$$R(\mathbf{p}) = \sum_{i=1}^{I} p_i \cdot (\alpha_i - \beta_i p_i + \sum_{j \neq i} \gamma_{i,j} p_j). \tag{3.2}$$

2. *Semi-log demand model*: A semi-log demand model is defined by parameters $\boldsymbol{\alpha} \in \mathbb{R}^I$, $\boldsymbol{\beta} \in \mathbb{R}^I$, $\boldsymbol{\gamma} = (\gamma_{i,j})_{i,j \in [I], i \neq j} \in \mathbb{R}^{I \cdot (I-1)}$. In a semi-log demand model, the logarithm of the demand function $d_i(\cdot)$ of each product $i \in [I]$ has a linear form in the prices $p_1, \ldots, p_I$:

$$\log d_i(\mathbf{p}) = \alpha_i - \beta_i p_i + \sum_{j \neq i} \gamma_{i,j} p_j. \tag{3.3}$$

This implies that the demand function is

$$d_i(\mathbf{p}) = e^{\alpha_i - \beta_i p_i + \sum_{j \neq i} \gamma_{i,j} p_j}. \tag{3.4}$$

The corresponding revenue function $R(\cdot)$ is then

$$R(\mathbf{p}) = \sum_{i=1}^{I} p_i \cdot e^{\alpha_i - \beta_i p_i + \sum_{j \neq i} \gamma_{i,j} p_j}. \tag{3.5}$$

3. *Log-log demand model*: A log-log demand model is defined by parameters $\boldsymbol{\alpha} \in \mathbb{R}^I$, $\boldsymbol{\beta} \in \mathbb{R}^I$, $\boldsymbol{\gamma} = (\gamma_{i,j})_{i,j \in [I], i \neq j} \in \mathbb{R}^{I \cdot (I-1)}$. In a log-log demand model, the logarithm of the demand function $d_i(\cdot)$ of each product $i \in [I]$ has a linear form in the log-transformed prices $\log p_1, \ldots, \log p_I$:

$$\log d_i(\mathbf{p}) = \alpha_i - \beta_i \log p_i + \sum_{j \neq i} \gamma_{i,j} \log p_j. \tag{3.6}$$

This implies that the demand function for product i is

$$d_i(\mathbf{p}) = e^{\alpha_i - \beta_i \log p_i + \sum_{j \neq i} \gamma_{i,j} \log p_j} \tag{3.7}$$

$$= e^{\alpha_i} p_i^{-\beta_i} \cdot \prod_{j \neq i} p_j^{\gamma_{i,j}}, \tag{3.8}$$

and that the revenue function is therefore

$$R(\mathbf{p}) = \sum_{i=1}^{I} p_i \cdot e^{\alpha_i} p_i^{-\beta_i} \cdot \prod_{j \neq i} p_j^{\gamma_{i,j}} \tag{3.9}$$

$$= \sum_{i=1}^{I} e^{\alpha_i} \cdot p_i^{1-\beta_i} \cdot \prod_{j \neq i} p_j^{\gamma_{i,j}}. \tag{3.10}$$

3.3.2 Deterministic robust price optimization problem

We now define the deterministic robust price optimization (DRPO) problem. To define this problem abstractly, we let $\mathcal{R}$ denote an uncertainty set of possible revenue functions. The DRPO problem is to then maximize the worst-case revenue, where the worst-case is the minimum revenue of a given price vector taken over all revenue functions in $\mathcal{R}$. Mathematically, this problem can be written as

$$\text{DRPO}: \quad \max_{\mathbf{p} \in \mathcal{P}} \min_{R \in \mathcal{R}} R(\mathbf{p}).$$

We use Z^*_{DR} to denote the optimal objective value of the DRPO problem.

Although $\mathcal{R}$ can be defined in many different ways, we will now focus on one general case that we will assume for most of our subsequent results in Sections 3.4, 3.5 and 3.6. Suppose that we fix the demand model to a specific parametric family, such as a log-log demand model. Let $\mathbf{u}$ denote the vector of demand model parameters. For example, for log-log, $\mathbf{u}$ would be the tuple $\mathbf{u} = (\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$. Let $\mathcal{U}$ denote a set of possible values of $\mathbf{u}$; $\mathcal{U}$ is then an uncertainty set of model parameters. With a slight abuse of notation, let $d_i(\mathbf{p}, \mathbf{u})$ denote the demand for product i when the demand model parameters are specified by $\mathbf{u}$. Then $\mathcal{R}$ can be defined as:

$$\mathcal{R} = \{R(\cdot) \equiv \sum_{i=1}^{I} p_i \cdot d_i(\cdot, \mathbf{u}) \mid \mathbf{u} \in \mathcal{U}\}, \tag{3.11}$$

i.e., it is all of the possible revenue functions spanned by the uncertain parameter vector $\mathbf{u}$ in $\mathcal{U}$.

To express the DRPO problem in this setting more conveniently, we will abuse our notation slightly and use $R(\mathbf{p}, \mathbf{u})$ to denote the revenue function evaluated at a price vector $\mathbf{p}$ with a particular parameter vector $\mathbf{u}$ specified. With this abuse of notation, the DRPO problem can be written as

$$\text{DRPO}: \quad \max_{\mathbf{p}\in\mathcal{P}} \min_{\mathbf{u}\in\mathcal{U}} R(\mathbf{p}, \mathbf{u}).$$

3.3.3 Randomized robust price optimization problem

In the DRPO problem, we assume that the decision maker will deterministically implement a single price vector $\mathbf{p}$ in the face of uncertainty in the revenue function. In the RRPO problem, we instead assume that the decision maker will randomly select a price vector $\mathbf{p}$ according to some distribution F over the feasible price set $\mathcal{P}$. Under this assumption, we can write the RRPO problem as

$$\text{RRPO}: \quad \max_{F\in\mathcal{F}} \min_{R\in\mathcal{R}} \int_{\mathcal{P}} R(\mathbf{p})\, dF(\mathbf{p}),$$

where $\mathcal{F}$ is the set of all distributions supported on $\mathcal{P}$. We use Z^*_{RR} to denote the optimal objective value of the RRPO problem. Note that $Z^*_{\text{RR}} \geq Z^*_{\text{DR}}$. This is because for every $\mathbf{p}' \in \mathcal{P}$, the distribution $F(\cdot) = \delta_{\mathbf{p}'}(\cdot)$, where $\delta_{\mathbf{p}'}(\cdot)$ is the Dirac delta function at $\mathbf{p}'$, is contained in $\mathcal{F}$. For this distribution, $\min_{R\in\mathcal{R}} \int R(\mathbf{p})\, dF(\mathbf{p}) = \min_{R\in\mathcal{R}} R(\mathbf{p}')$, which is exactly the worst-case revenue of deterministically selecting $\mathbf{p}'$.

A special instance of this problem arises when $\mathcal{P}$ is a discrete, finite set. In this case, F is a discrete probability distribution, and one can re-write the inner problem as an optimization problem over a discrete probability distribution $\boldsymbol{\pi} = (\pi_{\mathbf{p}})_{\mathbf{p}\in\mathcal{P}}$ over $\mathcal{P}$:

$$\text{RRPO-D}: \quad \max_{\boldsymbol{\pi}\in\Delta_{\mathcal{P}}} \min_{R\in\mathcal{R}} \sum_{\mathbf{p}\in\mathcal{P}} R(\mathbf{p})\pi_{\mathbf{p}},$$

where we use Δ_S to denote the $(|S|-1)$-dimensional unit simplex, i.e., $\Delta_S = \{\boldsymbol{\pi} \in \mathbb{R}^S \mid$

$\sum_{i \in S} \pi_i = 1,\ \pi_i \geq 0\ \forall\ i \in S\}$.

Lastly, under the assumption that $\mathcal{R}$ is the set of revenue functions of a fixed demand model family whose parameter vector $\mathbf{u}$ belongs to a parameter uncertainty set $\mathcal{U}$, we can restate the RRPO problem when $\mathcal{P}$ is a generic set and when $\mathcal{P}$ is finite as

$$\text{RRPO}: \quad \max_{F \in \mathcal{F}} \min_{\mathbf{u} \in \mathcal{U}} \int_{\mathcal{P}} R(\mathbf{p}, \mathbf{u})\, dF(\mathbf{p}), \tag{3.12}$$

$$\text{RRPO-D}: \quad \max_{\boldsymbol{\pi} \in \Delta_{\mathcal{P}}} \min_{\mathbf{u} \in \mathcal{U}} \sum_{\mathbf{p} \in \mathcal{P}} R(\mathbf{p}, \mathbf{u}) \pi_{\mathbf{p}}. \tag{3.13}$$

3.4 Benefits of randomization

In this section, we analyze when randomization can be beneficial. To aid us, we introduce some additional nomenclature in this section, which follows the terminology established in the prior literature on randomized robust optimization (Delage et al. 2019, Delage and Saif 2022, Wang et al. 2024). We say that a robust price optimization (RPO) problem is *randomization-receptive* if $Z^*_{\text{RR}} > Z^*_{\text{DR}}$, that is, randomizing over price vectors leads to a higher worst-case revenue than deterministically selecting a single price vector. Otherwise, we say that a RPO problem is *randomization-proof* if $Z^*_{\text{RR}} = Z^*_{\text{DR}}$, that is, there is no benefit from randomizing over price vectors.

In the following three subsections, we derive three classes of results that establish when the RPO problem is randomization-proof. The first condition (Section 3.4.1) for randomization-proofness applies in the case when $\mathcal{P}$ is a convex set and $\mathcal{R}$ is an arbitrary set of revenue functions each of which is concave in $\mathbf{p}$. The second and third conditions apply to the case where $\mathcal{R}$ arises out of a single demand model, where the parameter vector $\mathbf{u}$ belongs to an uncertainty set. The second condition (Section 3.4.2) applies in the case when $\mathcal{P}$ and $\mathcal{U}$ are compact, convex sets and $R(\mathbf{p}, \mathbf{u})$ obeys certain quasiconvexity and quasiconcavity properties. The third condition (Section 3.4.3) is for the case when $\mathcal{P}$ is finite and involves a certain minimax condition being met; as corollaries, we show that randomization-proofness occurs if

the DRPO problem satisfies a strong duality property, and that randomization-receptiveness is essentially equivalent to the DRPO solution being different from the nominal price optimization problem solution at the worst-case $\mathbf{u}$. Along the way, we also give a number of examples where our results can be used to establish that a particular family of RPO problems is randomization-proof, and also highlight how the results fail to hold when certain hypotheses are relaxed.

The main takeaway from these results is that the set of RPO problems that are randomization-proof is small. As we will see, the conditions under which a RPO problem will be randomization-proof are delicate and quite restrictive, and are satisfied only for certain very special cases; in most other realistic cases, the RPO problem will be randomization-receptive. Consequently, in Sections 3.5 and 3.6, we will develop algorithms for solving the randomized robust when the candidate price vector $\mathcal{P}$ is finite, and in Sections 3.7 we will show a wide range of both synthetic and real data instances in which the RPO problem is randomization-receptive.

3.4.1 Concave revenue function uncertainty sets

Our first major result is for the case where $\mathcal{R}$ consists of concave revenue functions.

Theorem 7 *Suppose that $\mathcal{P}$ is a convex set and that $\mathcal{R}$ is such that every $R \in \mathcal{R}$ is a concave function of $\mathbf{p}$. Then the RPO problem is randomization-proof, that is, $Z^*_{\mathrm{RR}} = Z^*_{\mathrm{DR}}$.*

The proof of this results (see Appendix B.1.1) follows from a simple application of Jensen's inequality. We pause to make a few important comments about this result. First, one aspect of this result that is special is that $\mathcal{R}$ can be a very general set: it could be countable or uncountable, and it could consist of revenue functions corresponding to different families of demand models. This will not be the case for our later results in Section 3.4.2 and 3.4.3, which require that $\mathcal{R}$ is defined based on a single demand model family, and that the uncertainty set of parameter vectors for that family be a convex compact set.

Second, we remark that the condition that all functions in $\mathcal{R}$ be concave cannot be relaxed

in general. We illustrate this in the following example, where $\mathcal{R}$ consists of two functions and one of the two is non-concave.

EXAMPLE 1 Consider a single-product RPO problem, and suppose that the revenue function uncertainty set $\mathcal{R} = \{R_1, R_2\}$, where $R_1(\cdot)$ and $R_2(\cdot)$ are defined as

$$R_1(p) = p(10 - 2p),$$
$$R_2(p) = p \cdot 10p^{-2}.$$

Note that $R_1(p)$ is the revenue function corresponding to the linear demand function $d_1(p) = 10 - 2p$, while $R_2(p)$ is the revenue function corresponding to the log-log demand function $d_2(p) = \exp(\log(10) - 2\log(p)) = 10p^{-2}$. Note also that $R_1(\cdot)$ is concave, while $R_2(\cdot)$ is convex. Suppose additionally that $\mathcal{P} = [1, 4]$.

We first calculate the optimal value of the DRPO problem. Observe that in the interval $[1, 4]$, the only root of the equation $10 - 2p = 10p^{-2}$ is $p \approx p' = 1.137805\ldots$. For $p < p'$, $d_2(p) > d_1(p)$, and for $p > p'$, $d_1(p) > d_2(p)$. Therefore, the optimal value of the of the DRPO problem can be calculated as

$$\begin{aligned}
&\max_{p\in[1,4]} \min_{R\in\mathcal{R}} R(p) \\
&= \max\{ \max_{p\in[1,p']} \min_{R\in\mathcal{R}} R(p), \max_{p\in[p',4]} \min_{R\in\mathcal{R}} R(p)\} \\
&= \max\{ \max_{p\in[1,p']} p \cdot (10 - 2p), \max_{p\in[p',4]} p \cdot 10p^{-2}\} \\
&= 10p' - 2p'^2 \\
&= 8.78885
\end{aligned}$$

In the above, the first step follows because the best value of the worst-case revenue over $[1, 4]$ is equivalent to taking the higher of the best worst-case revenue over either $[1, p']$ or $[p', 4]$. The second step follows because for every $p \in [1, p']$, $d_1(p) < d_2(p)$, and so $R_1(p) = p \cdot d_1(p) < p \cdot d_2(p) = R_2(p)$; similarly, for every $p \in [p', 4]$, $d_1(p) > d_2(p)$, and so $R_2(p) < R_1(p)$. The third step follows by carrying out the maximization of each of the two functions from the prior step over its corresponding interval.

Now, let us lower bound the optimal value of the RRPO problem. Consider a distribution F that randomizes over prices in the following way:

$$p = \begin{cases} 1 & \text{with probability } 17/21, \\ 2.5 & \text{with probability } 4/21. \end{cases} \tag{3.14}$$

The worst-case revenue for this distribution is

$$\begin{aligned}
&\min_{R \in \mathcal{R}} \int_1^4 R(p)\, dF(p) \\
&= \min \left\{ \frac{17}{21} \cdot R_1(1) + \frac{4}{21} \cdot R_1(2.5), \frac{17}{21} \cdot R_2(1) + \frac{4}{21} \cdot R_2(2.5) \right\} \\
&= \min \left\{ \frac{17}{21} \cdot 8 + \frac{4}{21} \cdot 12.5, \frac{17}{21} \cdot 10 + \frac{4}{21} \cdot 4 \right\} \\
&= \min \left\{ \frac{62}{7}, \frac{62}{7} \right\} \\
&= \frac{62}{7} \\
&= 8.857143.
\end{aligned}$$

This implies that $Z^*_{\text{RR}} \geq 8.857143$, whereas $Z^*_{\text{DR}} = 8.78885$, and thus $Z^*_{\text{RR}} > Z^*_{\text{DR}}$. $\square$

Third, we note that the requirement that $\mathcal{P}$ be a convex set also cannot be relaxed in general. The following example illustrates how Theorem 7 can fail to hold when $\mathcal{P}$ is not a convex set.

Example 2 Consider again a single-product RPO problem. Suppose that $\mathcal{R} = \{R_1, R_2\}$, where $R_1(p) = p(10 - p)$, $R_2(p) = p(4 - 0.2p)$; R_1 and R_2 correspond to linear demand functions $d_1(p) = 10 - p$, $d_2(p) = 4 - 0.2p$. Suppose that $\mathcal{P} = \{p_1, p_2\}$, where $p_1 = 5$, $p_2 = 10$. From this data, observe that:

$$\begin{aligned}
R_1(p_1) &= 5(10 - 5) = 25, \\
R_1(p_2) &= 10(10 - 10) = 0, \\
R_2(p_1) &= 5(4 - 0.2(5)) = 15, \\
R_2(p_2) &= 10(4 - 0.2(10)) = 20.
\end{aligned}$$

We first calculate the optimal value of the DRPO problem:

$$\begin{aligned} Z^*_{\mathrm{DR}} &= \max_{p \in \{p_1, p_2\}} \min\{R_1(p), R_2(p)\} \\ &= \max\{\min\{25, 15\}, \min\{0, 20\}\} \\ &= \max\{15, 0\} \\ &= 15. \end{aligned}$$

For the RRPO problem, the optimal value is given by the following LP:

$$\begin{aligned} \underset{\eta, \boldsymbol{\pi}}{\text{maximize}} \quad & \eta && \text{(3.15a)} \\ \text{subject to} \quad & \eta \leq \pi_{p_1} \cdot p_1 \cdot (10 - p_1) + \pi_{p_2} \cdot p_2 \cdot (10 - p_2) && \text{(3.15b)} \\ & \eta \leq \pi_{p_1} \cdot p_1 \cdot (4 - 0.2p_1) + \pi_{p_2} \cdot p_2 \cdot (4 - 0.2p_2) && \text{(3.15c)} \\ & \pi_{p_1} + \pi_{p_2} = 1 && \text{(3.15d)} \\ & \pi_{p_1}, \pi_{p_2} \geq 0. && \text{(3.15e)} \end{aligned}$$

The optimal distribution over $\mathcal{P} = \{p_1, p_2\}$ is given by $\pi_{p_1} = 2/3$, $\pi_{p_2} = 1/3$, which leads to $Z^*_{\mathrm{RR}} = 50/3 = 16.6667$. Since this is higher than Z^*_{DR}, we conclude that this particular instance is randomization-receptive. □

Lastly, Theorem 7 has a number of implications for different classes of demand models.

Example 3 (Single-product pricing under linear demand). Suppose that $I = 1$, which corresponds to a single-product pricing problem. Let $\mathbf{u} = (\alpha, \beta) \in \mathbb{R}^2$ denote the vector of linear demand model parameters, and let $\mathcal{U} \subseteq \mathbb{R}^2$ be an uncertainty set of possible values of (α, β). Let $\mathcal{R} = \{R(\cdot, \mathbf{u}) \mid \mathbf{u} \in \mathcal{U}\}$ be the set of revenue functions that arise from the uncertainty set $\mathcal{U}$. Note that each revenue function is of the form $R(p) = \alpha p - \beta p^2$. Therefore, the condition that each $R \in \mathcal{R}$ is concave implies that $R''(p) = -2\beta \leq 0$. Thus, if $\mathcal{U}$ is such that $\inf\{\beta \mid (\alpha, \beta) \in \mathcal{U}\} \geq 0$, then the robust price optimization problem is randomization-proof. □

EXAMPLE 4 (Multi-product pricing under linear demand). In the more general multi-product pricing problem, let $\mathbf{u} = (\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) \in \mathbb{R}^I \times \mathbb{R}^I \times \mathbb{R}^{I(I-1)}$ denote the vector of linear demand model parameters, and let $\mathcal{U}$ be an arbitrary uncertainty set of these model parameter vectors. Let $\mathcal{R} = \{R(\cdot, \mathbf{u}) \mid \mathbf{u} \in \mathcal{U}\}$ be the set of revenue functions that arise from the uncertainty set $\mathcal{U}$. Observe that each revenue function $R(\cdot, \mathbf{u})$ is of the form

$$\begin{aligned} R(\mathbf{p}) &= \sum_{i=1}^{I} p_i(\alpha_i - \beta_i p_i + \sum_{j \neq i} \gamma_{i,j} p_j) \\ &= \boldsymbol{\alpha}^T \mathbf{p} - \mathbf{p}^T \mathbf{M}_{\boldsymbol{\beta},\boldsymbol{\gamma}} \mathbf{p}, \end{aligned}$$

where $\mathbf{M}_{\boldsymbol{\beta},\boldsymbol{\gamma}}$ is the matrix

$$\mathbf{M}_{\boldsymbol{\beta},\boldsymbol{\gamma}} = \begin{bmatrix} -\beta_1 & \gamma_{1,2} & \gamma_{1,3} & \cdots & \gamma_{1,I-1} & \gamma_{1,I} \\ \gamma_{2,1} & -\beta_2 & \gamma_{2,3} & \cdots & \gamma_{2,I-1} & \gamma_{2,I} \\ \vdots & \vdots & \ddots & & \vdots & \vdots \\ \gamma_{I,1} & \gamma_{I,2} & \gamma_{I,3} & \cdots & \gamma_{I,I-1} & -\beta_I \end{bmatrix}.$$

This implies that

$$\nabla^2 R(\mathbf{p}) = -2\mathbf{M}_{\boldsymbol{\beta},\boldsymbol{\gamma}}.$$

The function R is therefore concave if the matrix $\mathbf{M}_{\boldsymbol{\beta},\boldsymbol{\gamma}}$ is positive semidefinite. Therefore, if $\mathcal{U}$ is such that $\inf\{\lambda_{\min}(\mathbf{M}_{\boldsymbol{\beta},\boldsymbol{\gamma}}) \mid (\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) \in \mathcal{U}\} \geq 0$, where $\lambda_{\min}(\mathbf{A})$ denotes the minimum eigenvalue of a symmetric matrix $\mathbf{A}$, then the robust price optimization problem is randomization proof. □

EXAMPLE 5 (Single-product pricing under semi-log demand). For the single-product pricing problem under semi-log demand, $d(p) = \exp(\alpha - \beta p)$ is the demand function given the parameter vector $\mathbf{u} = (\alpha, \beta)$. Let $\mathcal{U} \subseteq \mathbb{R}^2$ be an uncertainty set of possible values of (α, β), and assume that β is bounded away from zero, that is, $\inf\{\beta \mid (\alpha, \beta) \in \mathcal{U}\} \geq 0$. $\mathcal{R} = \{R(\cdot, \mathbf{u}) \mid \mathbf{u} \in \mathcal{U}\}$ be the revenue function uncertainty set. For a given $R \in \mathcal{R}$, its second derivative is $R''(p) = R''(p) = \beta(\beta p - 2)e^{\alpha - \beta p}$. Thus, for $R''(p)$ to be nonpositive, we

need $\beta p - 2 \leq 0$ or equivalently $\beta p \leq 2$ (since β is assumed to be nonnegative) for all $p \in \mathcal{P}$ in order for $R(p)$ to be concave. Thus, if $\sup_{p \in \mathcal{P}} \sup_{(\alpha,\beta) \in \mathcal{U}} \{\beta p\} \leq 2$ and $\inf\{\beta \mid (\alpha, \beta) \in \mathcal{U}\} \geq 0$, then the RPO problem is randomization-proof. □

Example 6 (Single-product pricing under log-log demand). For the single-product pricing problem under log-log demand, $d(p) = \exp(\alpha - \beta \log p) = e^{\alpha} \cdot p^{-\beta}$ is the demand function and $\mathbf{u} = (\alpha, \beta) \in \mathbb{R}^2$ is the vector of uncertain demand model parameters. Let $\mathcal{U} \subseteq \mathbb{R}^2$ be an uncertainty set of possible values of (α, β), and assume that β is bounded away from zero from below, that is, $\inf\{\beta \mid (\alpha, \beta) \in \mathcal{U}\} \geq 0$. Let $\mathcal{R} = \{R(\cdot, \mathbf{u}) \mid \mathbf{u} \in \mathcal{U}\}$ be the revenue function uncertainty set. For a given $R \in \mathcal{R}$, its second derivative is $R''(p) = e^{\alpha} \cdot (\beta - 1)(\beta) \cdot p^{-\beta - 1}$. Thus, for $R''(p)$ to be nonpositive, we need $\beta - 1 \leq 0$, or equivalently $\beta \leq 1$. Thus, if $\sup_{(\alpha,\beta) \in \mathcal{U}} \beta \leq 1$ and $\inf_{(\alpha,\beta) \in \mathcal{U}} \beta \geq 0$, then the RPO problem is randomization-proof. □

3.4.2 Quasiconcavity in p and quasiconvexity in u

The second result we establish concerns the RRPO problem when there is a demand parameter uncertainty set $\mathcal{U}$. In this case, the RRPO and DRPO problems are

$$\text{RRPO}: \max_{F \in \mathcal{F}} \min_{\mathbf{u} \in \mathcal{U}} \int_{\mathcal{P}} R(\mathbf{p}, \mathbf{u}) \, dF(\mathbf{p}),$$

$$\text{DRPO}: \max_{\mathbf{p} \in \mathcal{P}} \min_{\mathbf{u} \in \mathcal{U}} R(\mathbf{p}, \mathbf{u}).$$

We make the following assumption about R.

Assumption 7 *R is a continuous function of $(\mathbf{p}, \mathbf{u})$.*

Under these assumptions, we obtain the following result.

Theorem 8 *Suppose that Assumptions 7 holds. Suppose that $\mathcal{P} \subseteq \mathbb{R}^I$ and $\mathcal{U} \subseteq \mathbb{R}^d$ are compact convex sets. Suppose that $\int_{\mathcal{P}} R(\mathbf{p}, \mathbf{u}) \, dF(\mathbf{p})$ is a quasiconvex function of $\mathbf{u}$ on $\mathcal{U}$ for any $F \in \mathcal{F}$. Suppose that $R(\mathbf{p}, \mathbf{u})$ is quasiconcave in $\mathbf{p}$ on $\mathcal{P}$ for any $\mathbf{u} \in \mathcal{U}$ and quasiconvex in $\mathbf{u}$ on $\mathcal{U}$ for any $\mathbf{p} \in \mathcal{P}$. Then, the robust price optimization problem is randomization-proof, that is, $Z^*_{\text{DR}} = Z^*_{\text{RR}}$.*

The proof of Theorem 8 (see Appendix B.1.2) follows from applying Sion's minimax theorem twice. This result allows us to show that a larger number of RPO problems are randomization-proof. We provide a few examples below.

EXAMPLE 7 Consider a single-product price optimization problem where the demand follows a semi-log model. The uncertain parameter is therefore $\mathbf{u} = (\alpha, \beta)$.

Observe that $R(p, \mathbf{u}) = pe^{\alpha - \beta p}$ is convex in $\mathbf{u}$. Thus, it is also quasiconvex in $\mathbf{u}$ for a fixed p. Additionally, for any distribution F over $\mathcal{P}$, we have that the function $\int_{\mathcal{P}} R(p, \mathbf{u})\, dF(p)$ is convex in $\mathbf{u}$ (it is a nonnegative weighted combination of the functions $\mathbf{u} = (\alpha, \beta) \mapsto pe^{\alpha - \beta p}$, each of which is convex), and is thus also quasiconvex in $\mathbf{u}$.

Note also that the function R is quasi-concave in p. To see this, observe that $\log R(p, \mathbf{u}) = \log p + \alpha - \beta p$, which is concave in p; this means that R is log-concave in p. Since any log-concave function is quasiconcave, it follows that R is quasiconcave in p.

Thus, if $\mathcal{P} \subseteq \mathbb{R}$ and $\mathcal{U} \subseteq \mathbb{R}^2$ are compact and convex, then Theorem 8 asserts that the RPO problem is randomization-proof. □

EXAMPLE 8 Consider a single-product price optimization problem where the demand follows a log-log model. The uncertain parameter is $\mathbf{u} = (\alpha, \beta)$, and $R(p, \mathbf{u}) = pe^{\alpha - \beta \log p}$. Assume that $\mathcal{P} \subseteq \mathbb{R}$ is a compact convex set, and that $\min\{p \mid p \in \mathcal{P}\} > 0$.

Observe that $R(p, \mathbf{u}) = p \cdot e^{\alpha - \beta \log p}$ is convex in $\mathbf{u}$, and therefore quasiconvex in $\mathbf{u}$ for a fixed p. Additionally, for any distribution F over $\mathcal{P}$, we have that the function $\int_{\mathcal{P}} R(p, \mathbf{u})\, dF(p)$ is convex in $\mathbf{u}$ and therefore also quasiconvex in $\mathbf{u}$ for a fixed F.

Lastly, with regard to quasiconcavity in p, observe that $\log R(p, \mathbf{u}) = \log p + \alpha - \beta \log p = (1 - \beta) \log p + \alpha$, which means that R is log-concave in p whenever $1 - \beta > 0$ or equivalently $\beta < 1$. Therefore, R will also be quasiconcave whenever $\beta < 1$.

Thus, if $\mathcal{P} \subseteq \mathbb{R}$ and $\mathcal{U} \subseteq \mathbb{R}^2$ are compact and convex, and $\max\{\beta \mid (\alpha, \beta) \in \mathcal{U}\} < 1$, then Theorem 8 guarantees that the RPO problem is randomization-proof. □

With regard to the above two examples, we note that in general, the revenue function

for a semi-log or a log-log demand model is not concave in p. Thus, Theorem 7 cannot be used in these cases, and we must use Theorem 8. Note, however, that the two examples above critically rely on the revenue function being log-concave and therefore quasiconcave, which is only the case for single product price optimization problems. Log-concavity and quasiconcavity are in general not preserved under addition (i.e., the sum of quasiconcave functions is not always quasiconcave, and the sum of log-concave functions is not always log-concave), and so Theorem 8 will in general not be applicable for multiproduct pricing problems involving the semi-log or log-log demand model.

3.4.3 Finite price set $\mathcal{P}$

In this section, we analyze randomization-receptiveness when $\mathcal{P}$ is a finite set. To study this setting, let us define the set $\mathcal{Q}$ as the set of all probability distributions supported on $\mathcal{U}$. We note that these results are adaptations of several results from Wang et al. (2024) to the pricing setting that we study, which develop analogous conditions for randomization-proofness for the robust assortment optimization problem.

Our first result establishes that randomization-proofness is equivalent to the existence of a distribution Q over $\mathcal{U}$ under which any price vector's expected performance is no better than the deterministic robust optimal value.

Theorem 9 *Suppose that $R(\mathbf{p}, \mathbf{u})$ is continuous in $\mathbf{u}$ for any fixed $\mathbf{p} \in \mathcal{P}$. A robust price optimization problem with finite $\mathcal{P}$ is randomization-proof if and only if there exists a distribution $Q \in \mathcal{Q}$ such that for all $\mathbf{p} \in \mathcal{P}$,*

$$\int_{\mathcal{U}} R(\mathbf{p}, \mathbf{u})\, dQ(\mathbf{u}) \leq Z^*_{\mathrm{DR}}.$$

To prove this result, we use Sion's minimax theorem to establish that

$$Z^*_{\mathrm{RR}} = \inf_{Q \in \mathcal{Q}} \max_{\mathbf{p} \in \mathcal{P}} \int_{\mathcal{U}} R(\mathbf{p}, \mathbf{u})\, dQ(\mathbf{u}); \tag{3.16}$$

with that result in hand, the condition in Theorem 9 is equivalent to establishing that

$$Z^*_{\text{DR}} \geq \inf_{Q \in \mathcal{Q}} \max_{\mathbf{p} \in \mathcal{P}} \int_{\mathcal{U}} R(\mathbf{p}, \mathbf{u})\, dQ(\mathbf{u}) = Z^*_{\text{RR}},$$

which, together with the inequality $Z^*_{\text{DR}} \leq Z^*_{\text{RR}}$ immediately yields randomization-proofness. We note that this result is analogous to Theorem 1 in Wang et al. (2024), which provides a similar necessary and sufficient condition for randomization-proofness in the context of robust assortment optimization. Our proof, which relies on Sion's minimax theorem, is perhaps slightly more direct than the proof of Theorem 1 in Wang et al. (2024), although this is a matter of taste.

Our next two results are consequences of this theorem. The first essentially states that a price optimization problem proof will be randomization-proof if the robust price optimization problem obeys strong duality. The second states that, under some conditions, a robust price optimization problem is randomization-receptive if and only if the deterministic robust price vector $\mathbf{p}^*_{\text{DR}}$ is not an optimal solution of the nominal price optimization problem under the worst-case $\mathbf{u}^*$ that attains the worst-case objective under $\mathbf{p}^*_{\text{DR}}$. We note that these results are both analogous to Corollaries 1 and 2 in Wang et al. (2024).

Corollary 2 *Suppose that $R(\mathbf{p}, \mathbf{u})$ is a continuous function of $\mathbf{u}$ for every $\mathbf{p} \in \mathcal{P}$. A robust price optimization problem with finite $\mathcal{P}$ is randomization-proof if and only if it satisfies strong duality:*

$$\max_{\mathbf{p} \in \mathcal{P}} \min_{\mathbf{u} \in \mathcal{U}} R(\mathbf{p}, \mathbf{u}) = \min_{\mathbf{u} \in \mathcal{U}} \max_{\mathbf{p} \in \mathcal{P}} R(\mathbf{p}, \mathbf{u}). \tag{3.17}$$

Corollary 3 *Suppose that $\mathcal{U}$ is a compact subset of $\mathbb{R}^d$, and that $R(\mathbf{p}, \mathbf{u})$ is a continuous function of $\mathbf{u}$ for every $\mathbf{p} \in \mathcal{P}$. Suppose that $\mathbf{p}^*_{\text{DR}} \in \arg\max_{\mathbf{p} \in \mathcal{P}} \min_{\mathbf{u} \in \mathcal{U}} R(\mathbf{p}, \mathbf{u})$ is an optimal solution of the deterministic robust price optimization problem, and suppose that $\min_{\mathbf{u} \in \mathcal{U}} R(\mathbf{p}^*_{\text{DR}}; \mathbf{u})$ has a unique solution $\mathbf{u}^*$. Then the robust price optimization is randomization-receptive if and only if $\mathbf{p}^*_{\text{DR}} \notin \arg\max_{\mathbf{p} \in \mathcal{P}} R(\mathbf{p}, \mathbf{u}^*)$.*

With regard to Corollary 3, we note that the uniqueness requirement for $\mathbf{u}^*$ cannot in general be relaxed. In Appendix B.1.6, we show an instance where $\min_{\mathbf{u} \in \mathcal{U}} R(\mathbf{p}^*_{\text{DR}}, \mathbf{u})$ has multi-

ple optimal solutions, $\mathbf{p}^*_{\text{DR}} \notin \arg\max_{\mathbf{p}\in\mathcal{P}} R(\mathbf{p}, \mathbf{u}')$ for every $\mathbf{u}'$ that solves $\min_{\mathbf{u}\in\mathcal{U}} R(\mathbf{p}^*_{\text{DR}}, \mathbf{u})$, and yet the problem is randomization-proof, i.e., $Z^*_{\text{DR}} = Z^*_{\text{RR}}$.

We remark that the necessary and sufficient conditions for randomization-proofness in Corollaries 2 and 3 are rather stringent and demanding. With regard to Corollary 2, strong duality is in general unlikely to hold given that $\mathcal{P}$ is a finite set. With regard to Corollary 3, we note that in general, the solution of the deterministic robust price optimization problem is unlikely to also be an optimal solution of an appropriately defined nominal price optimization problem; this is frequently not the case in many applications of robust optimization outside of pricing. Given this, these conditions are suggestive of the fact that most robust price optimization problems will be randomization-receptive. This motivates our study of solution algorithms for numerically solving the RRPO problem in the next two sections.

3.5 Solution algorithm for finite price set $\mathcal{P}$, convex uncertainty set $\mathcal{U}$

In this section, we describe a general solution algorithm for solving the RRPO problem when the price set $\mathcal{P}$ is a finite set, and the uncertainty set $\mathcal{U}$ is a general convex uncertainty set. Section 3.5.1 describes the general solution algorithm, which is a constraint generation algorithm that involves solving a nominal pricing problem over $\mathcal{P}$ as a subroutine. Sections 3.5.2, 3.5.3 and 3.5.4 describe how the solution algorithm specializes to the cases of the linear, semi-log and log-log demand models, respectively, and in particular, how the nominal pricing problem can be solved for each of these three cases; the formulations we present for the semi-log and log-log models here may be of independent interest as they are, to the best of our knowledge, the first exact mixed-integer convex formulations for the multi-product pricing problem under a finite price set for these demand models.

3.5.1 General solution approach

The first general solution scheme that we consider is when $\mathcal{P}$ is a discrete set and the uncertainty set $\mathcal{U}$ is a convex uncertainty set. In this case, if the revenue function $R(\mathbf{p}, \mathbf{u})$ is quasiconvex and continuous in $\mathbf{u} \in \mathcal{U}$, then the RRPO problem can be reformulated as follows:

$$\begin{aligned}
&\max_{\boldsymbol{\pi} \in \Delta_{\mathcal{P}}} \min_{\mathbf{u} \in \mathcal{U}} \sum_{\mathbf{p} \in \mathcal{P}} \pi_{\mathbf{p}} R(\mathbf{p}, \mathbf{u}) \\
&= \min_{\mathbf{u} \in \mathcal{U}} \max_{\boldsymbol{\pi} \in \Delta_{\mathcal{P}}} \sum_{\mathbf{p} \in \mathcal{P}} \pi_{\mathbf{p}} R(\mathbf{p}, \mathbf{u}) \\
&= \min_{\mathbf{u} \in \mathcal{U}} \max_{\mathbf{p} \in \mathcal{P}} R(\mathbf{p}, \mathbf{u}),
\end{aligned}$$

where the first equality follows by Sion's minimax theorem, and the second equality follows by the fact that the inner maximum is attained by setting $\pi_{\mathbf{p}} = 1$ for some $\mathbf{p}$ and setting $\pi_{\mathbf{p}'} = 0$ for all $\mathbf{p}' \neq \mathbf{p}$. This last problem can be written in epigraph form as

$$\begin{aligned}
\underset{u,t}{\text{minimize}} \quad & t && \text{(3.18a)} \\
\text{subject to} \quad & t \geq R(\mathbf{p}, \mathbf{u}), \quad \forall\ \mathbf{p} \in \mathcal{P}, && \text{(3.18b)} \\
& \mathbf{u} \in \mathcal{U}. && \text{(3.18c)}
\end{aligned}$$

Problem (3.18) can be solved using constraint generation. In such a scheme, we start with constraint (3.18b) enforced only at a subset $\hat{\mathcal{P}} \subset \mathcal{P}$, and solve problem (3.18) to obtain a solution $(\mathbf{u}, t)$. At this solution, we solve the problem $\max_{\mathbf{p} \in \mathcal{P}} R(\mathbf{p}, \mathbf{u})$, and compare this objective value to the current value of t. If it is less than or equal to t, we conclude that $(\mathbf{u}, t)$ satisfies constraint (3.18b) and terminate with $(\mathbf{u}, t)$ as the optimal solution. Otherwise, if it is greater than t, we have identified a $\mathbf{p}$ for which constraint (3.18b) and we add the new constraint to $\hat{\mathcal{P}}$. We then re-solve the problem to obtain a new solution $(\mathbf{u}, t)$ and repeat the process until we can no longer identify any violated constraints. To recover the optimal randomization scheme from the solution of this problem (i.e., the distribution $\boldsymbol{\pi}$), we simply consider the optimal dual variable of each constraint $t \leq R(\mathbf{p}, \mathbf{u})$.

The viability of this solution approach critically depends on our ability to solve the separation problem $\max_{\mathbf{p}\in\mathcal{P}} R(\mathbf{p},\mathbf{u})$ efficiently, and to solve the problem (3.18) efficiently for a fixed subset $\hat{\mathcal{P}} \subset \mathcal{P}$. In what follows, we shall demonstrate that this problem can actually be solved practically for the linear, semi-log and log-log problems.

To develop our approaches for the linear, semi-log and log-log models, we will make the following assumption about the price set $\mathcal{P}$, which simply states that $\mathcal{P}$ is a Cartesian product of finite sets of prices for each of the products.

Assumption 8 *$\mathcal{P} = \mathcal{P}_1 \times \ldots \times \mathcal{P}_I$, where $\mathcal{P}_i$ is a finite subset of $\mathbb{R}_+$ for each i.*

3.5.2 Linear demand model

We begin by showing how our solution approach for convex $\mathcal{U}$ applies to the linear demand model case. Recall that the linear model revenue function is

$$R(\mathbf{p},\mathbf{u}) = \sum_{i=1}^{I} p_i(\alpha_i - \beta_i p_i + \sum_{j\neq i} \gamma_{i,j} p_j). \tag{3.19}$$

For a fixed $\mathbf{p}$, the function $R(\mathbf{p},\mathbf{u})$ is linear and therefore convex and quasiconvex in $\mathbf{u} = (\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma})$. Thus, given a subset $\hat{\mathcal{P}} \subset \mathcal{P}$, the problem (3.18) should be easy to solve, assuming that $\mathcal{U}$ is also a sufficiently tractable convex set. For example, if $\mathcal{U}$ is a polyhedron, then since each constraint (3.18b) is linear in $\mathbf{u}$, problem (3.18) would be a linear program.

The separation problem for the linear demand model case is

$$\begin{aligned}
&\max_{\mathbf{p}\in\mathcal{P}} \sum_{i=1}^{I} p_i(\alpha_i - \beta_i p_i + \sum_{j\neq i} \gamma_{i,j} p_j) \\
&= \max_{\mathbf{p}\in\mathcal{P}} \sum_{i=1}^{I} p_i\alpha_i - \sum_{i=1}^{I} \beta_i p_i^2 + \sum_{i=1}^{I}\sum_{j\neq i} \gamma_{i,j} p_i p_j
\end{aligned}$$

Since $\mathcal{P} = \mathcal{P}_1 \times \ldots \times \mathcal{P}_I$, we can formulate this as a mixed-integer program. Let $x_{i,t}$ be a binary variable that is 1 if product i has price $t \in \mathcal{P}_i$, and 0 otherwise. Similarly, let y_{i,j,t_1,t_2}

be a binary decision variable that is 1 if product i is given price t_1 and product j is given price t_2 for $i \neq j$, and 0 otherwise. Then the separation problem can be straightforwardly written as

$$\underset{x,y}{\text{maximize}} \quad \sum_{i=1}^{I} \sum_{t \in \mathcal{P}_i} \alpha_i \cdot t \cdot x_{i,t} + \sum_{i=1}^{I} \sum_{t \in \mathcal{P}_i} t^2 \cdot \beta_i \cdot x_{i,t} + \sum_{i=1}^{I} \sum_{j \neq i} \sum_{t_1 \in \mathcal{P}_i} \sum_{t_2 \in \mathcal{P}_j} \gamma_{i,j} \cdot t_1 \cdot t_2 \cdot y_{i,j,t_1,t_2} \tag{3.20a}$$

$$\text{subject to} \quad \sum_{t \in \mathcal{P}_i} x_{i,t} = 1, \quad \forall\, i \in [I], \tag{3.20b}$$

$$\sum_{t_2 \in \mathcal{P}_j} y_{i,j,t_1,t_2} = x_{i,t_1}, \quad \forall\, i, j \in [I], j \neq i, t_1 \in \mathcal{P}_i, \tag{3.20c}$$

$$\sum_{t_1 \in \mathcal{P}_i} y_{i,j,t_1,t_2} = x_{i,t_2}, \quad \forall\, i, j \in [I], j \neq i, t_2 \in \mathcal{P}_j, \tag{3.20d}$$

$$x_{i,t} \in \{0,1\}, \quad \forall\, i \in [I],\ t \in \mathcal{P}_i, \tag{3.20e}$$

$$y_{i,j,t_1,t_2} \in \{0,1\}, \quad \forall\, i, j \in [I], i \neq j, t_1 \in \mathcal{P}_i, t_2 \in \mathcal{P}_j, \tag{3.20f}$$

where the first constraint simply enforces that exactly one price is chosen for each product, while the second and third constraints require that the y_{i,j,t_1,t_2} variables are essentially equal to $x_{i,t_1} \cdot x_{j,t_2}$.

3.5.3 Semi-log demand model

We will now show how the solution approach we have defined earlier applies to the semi-log demand model. Recall that the semi-log revenue function is

$$R(\mathbf{p}, \mathbf{u}) = \sum_{i=1}^{I} p_i \cdot e^{\alpha_i - \beta_i p_i + \sum_{j \neq i} \gamma_{i,j} p_j}. \tag{3.21}$$

Observe that for a fixed $\mathbf{p}$, the function $R(\mathbf{p}, \mathbf{u})$ is convex (and therefore quasiconvex) in $\mathbf{u}$, since it is the nonnegative weighted combination of exponentials of linear functions of $\mathbf{u} = (\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$. Thus, given a subset $\hat{\mathcal{P}} \subset \mathcal{P}$, solving problem (3.18) should again be "easy", assuming also that $\mathcal{U}$ is a sufficiently tractable convex set. (In particular, the function

$R(\mathbf{p}, \mathbf{u})$ can be represented using I exponential cones; assuming that $\mathcal{U}$ is also representable using conic constraints, problem (3.18) will thus be some type of continuous conic program.)

We now turn our attention to the separation problem, $\max_{\mathbf{p}\in\mathcal{P}} R(\mathbf{p}, \mathbf{u})$. Specifically, this problem is

$$\max_{\mathbf{p}\in\mathcal{P}} \sum_{i=1}^{I} p_i \cdot e^{\alpha_i - \beta_i p_i + \sum_{j\neq i} \gamma_{i,j} p_j}.$$

Observe that since the function $f(t) = \log(t)$ is monotonic, the set of optimal solutions remains unchanged if we consider the same problem with a log-transformed objective

$$\begin{aligned} &\max_{\mathbf{p}\in\mathcal{P}} \log \left(\sum_{i=1}^{I} p_i \cdot e^{\alpha_i - \beta_i p_i + \sum_{j\neq i} \gamma_{i,j} p_j} \right) \\ &= \max_{\mathbf{p}\in\mathcal{P}} \log \left(\sum_{i=1}^{I} e^{\alpha_i + \log p_i - \beta_i p_i + \sum_{j\neq i} \gamma_{i,j} p_j} \right). \end{aligned} \tag{3.22}$$

To now further re-formulate this problem, we observe that the objective function can be re-written using the function $g(\mathbf{y}) = \log(\sum_{i=1}^{I} e^{y_i})$. The function g is what is known as the *log-sum-exp* function, which is a convex function (Boyd and Vandenberghe 2004). More importantly, a standard result in convex analysis is that any proper, lower semi-continuous, convex function is equivalent to its *biconjugate* function, which is the convex conjugate of its convex conjugate (Rockafellar 1970). For the log-sum-exp function, this in particular means that $g(\mathbf{y})$ can be written as

$$g(\mathbf{y}) = \max_{\boldsymbol{\mu}\in\Delta_{[I]}} \{\boldsymbol{\mu}^T \mathbf{y} - \sum_{i=1}^{I} \mu_i \log \mu_i\}.$$

The function $h(x) = x \log x$ is the negative entropy function (Boyd and Vandenberghe 2004), and is a convex function; thus, the function inside the $\max\{\cdot\}$ is a linear function minus a sum of convex functions, and is a concave function.

For our problem, this means that (3.22) can be re-written as

$$\begin{aligned}
&\max_{\mathbf{p}\in\mathcal{P}} \log R(\mathbf{p},\mathbf{u}) \\
&= \max_{\mathbf{p}\in\mathcal{P}} \log\left(\sum_{i=1}^{I} e^{\alpha_i+\log p_i-\beta_i p_i+\sum_{j\neq i}\gamma_{i,j}p_j}\right) \\
&= \max_{\mathbf{p}\in\mathcal{P}} \max_{\boldsymbol{\mu}\in\Delta_{[I]}} \left\{\sum_{i=1}^{I}\mu_i(\alpha_i+\log p_i-\beta_i p_i+\sum_{j\neq i}\gamma_{i,j}p_j)-\sum_{i=1}^{I}\mu_i\log\mu_i\right\} \\
&= \max_{\mathbf{p}\in\mathcal{P},\ \boldsymbol{\mu}\in\Delta_{[I]}} \left\{\sum_{i=1}^{I}\mu_i(\alpha_i+\log p_i-\beta_i p_i+\sum_{j\neq i}\gamma_{i,j}p_j)-\sum_{i=1}^{I}\mu_i\log\mu_i\right\}. \qquad (3.23)
\end{aligned}$$

To further reformulate this problem, we now make use of Assumption 8, which states that $\mathcal{P}$ is the Cartesian product of finite sets. Let us introduce a new binary decision variable $x_{i,t}$ which is 1 if product i's price is set to price $t \in \mathcal{P}_i$, and 0 otherwise. Using this new decision variable, observe that we can replace p_i wherever it occurs with $\sum_{t\in\mathcal{P}_i} t\cdot x_{i,t}$. We can also similarly replace $\log p_i$ with $\sum_{t\in\mathcal{P}_i}\log t\cdot x_{i,t}$. Therefore, problem (3.23) can be further reformulated as

$$\underset{x,\boldsymbol{\mu}}{\text{maximize}} \quad \sum_{i=1}^{I}\mu_i\left(\alpha_i+\sum_{t\in\mathcal{P}_i}\log t\cdot x_{i,t}-\beta_i\cdot\sum_{t\in\mathcal{P}_i}t\cdot x_{i,t}+\sum_{j\neq i}\gamma_{i,j}\sum_{t\in\mathcal{P}_j}t\cdot x_{j,t}\right)-\sum_{i=1}^{I}\mu_i\log\mu_i \qquad (3.24a)$$

$$\text{subject to} \quad \sum_{i=1}^{I}\mu_i = 1, \qquad (3.24b)$$

$$\sum_{t\in\mathcal{P}_i} x_{i,t} = 1, \quad \forall\ i\in[I], \qquad (3.24c)$$

$$x_{i,t}\in\{0,1\}, \quad \forall\ i\in[I],\ t\in\mathcal{P}_i, \qquad (3.24d)$$

$$\mu_i \geq 0, \quad \forall\ i\in[I]. \qquad (3.24e)$$

This last problem is *almost* a mixed-integer convex program: as noted earlier, the expression $-\sum_{i=1}^{I}\mu_i\log\mu_i$ is concave in μ. The main wrinkle is the presence of the bilinear terms in the objective function, specifically terms of the form $\mu_i\cdot x_{j,t}$. Fortunately, we can circumvent this difficulty by introducing a new decision variable, $w_{i,j,t}$, which is the linearization of $\mu_i\cdot x_{j,t}$,

for each $i, j \in [I]$, $t \in \mathcal{P}_j$. By adding this new decision variable and additional constraints, we arrive at our final formulation, which is a mixed-integer convex program.

$$\begin{aligned}
\underset{\boldsymbol{\mu},w,x}{\text{maximize}} \quad & \sum_{i=1}^{I} \mu_i \alpha_i + \sum_{i=1}^{I} \sum_{t \in \mathcal{P}_i} \log t \cdot w_{i,i,t} - \sum_{i=1}^{I} \beta_i \cdot \sum_{t \in \mathcal{P}_i} t \cdot w_{i,i,t} \\
& \qquad + \sum_{i=1}^{I} \sum_{j \neq i} \gamma_{i,j} \cdot \Big(\sum_{t \in \mathcal{P}_j} t \cdot w_{i,j,t} \Big) - \sum_{i=1}^{I} \mu_i \log \mu_i && \text{(3.25a)} \\
\text{subject to} \quad & \sum_{t \in \mathcal{P}_j} w_{i,j,t} = \mu_i, \quad \forall\ i \in [I],\ j \in [I], && \text{(3.25b)} \\
& \sum_{i=1}^{I} w_{i,j,t} = x_{j,t}, \quad \forall\ j \in [I],\ t \in \mathcal{P}_j, && \text{(3.25c)} \\
& \sum_{i=1}^{I} \mu_i = 1, && \text{(3.25d)} \\
& \sum_{t \in \mathcal{P}_i} x_{i,t} = 1, \quad \forall\ i \in [I], && \text{(3.25e)} \\
& w_{i,j,t} \geq 0, \quad \forall\ i \in [I],\ j \in [I], t \in \mathcal{P}_j, && \text{(3.25f)} \\
& x_{i,t} \in \{0, 1\}, \quad \forall\ i \in [I],\ t \in \mathcal{P}_i, && \text{(3.25g)} \\
& \mu_i \geq 0, \quad \forall\ i \in [I]. && \text{(3.25h)}
\end{aligned}$$

There are a few important points to observe about this formulation. First, note that because the μ_i's sum to 1 over i, and the $x_{j,t}$'s are binary and sum to 1 over $t \in \mathcal{P}_j$ for any j, then ensuring that $w_{i,j,t} = \mu_i \cdot x_{j,t}$ can be done simply through constraints (3.25b) and (3.25c). This is different from the usual McCormick envelope-style linearization technique, which in this case would involve the four inequalities:

$$w_{i,j,t} \leq x_{j,t}, \tag{3.26}$$

$$w_{i,j,t} \leq \mu_i, \tag{3.27}$$

$$w_{i,j,t} \geq x_{j,t} + \mu_i - 1, \tag{3.28}$$

$$w_{i,j,t} \geq 0, \tag{3.29}$$

for every $i \in [I]$, $j \in [I]$, $t \in \mathcal{P}_j$. It is not difficult to show that these constraints are implied

by constraints (3.25b), (3.25c) and (3.25f).

Second, at the risk of belaboring the obvious, the optimal objective value of problem (3.25) is the value of $\max_{\mathbf{p}\in\mathcal{P}} \log R(\mathbf{p}, \mathbf{u})$, where R is the semi-log revenue function. Upon solving problem (3.25) to obtain the objective value Z', we can obtain the optimal objective value of the untransformed problem $\max_{\mathbf{p}\in\mathcal{P}} R(\mathbf{p}, \mathbf{u})$ as $e^{Z'}$.

Third, this formulation is notable because, to our knowledge, this is the first exact mixed-integer convex formulation of the nominal multi-product pricing problem under semi-log demand and a price set defined as the Cartesian product of finite sets. To date, virtually all research that has considered solving this type of problem in the marketing and operations management literatures has involved heuristics (see, for example, Section EC.3 of Mišić 2020, which solves log-log and semi-log multi-product pricing problems for a collection of stores using local search). From this perspective, although we developed this formulation as part of the overall solution approach for the RRPO problem, we believe it is of more general interest.

Building on the previous point, problem (3.25) can be formulated as a mixed-integer exponential cone program. Such problems are garnering increasing attention from the academic and industry sides. In particular, since 2019, the MOSEK solver (ApS 2022) supports the exponential cone and can solve mixed-integer conic programs that involve the exponential cone to global optimality. Although the solution technology for mixed-integer conic programs is not as developed as that of mixed-integer linear programs (as exemplified by state-of-the-art solvers such as Gurobi and CPLEX), it is reasonable to expect that these solvers will continue to improve and allow larger and larger problem instances to be solved to optimality in the future.

Lastly, we comment that the same reformulation technique used above – taking the logarithm, replacing the log-sum-exp function with its biconjugate, and then linearizing the products of the binary decision variables and the probability mass function values (the μ_i variables) that arise from the biconjugate – can also be used to derive an exact formulation

of the deterministic robust price optimization problem. By taking the same approach, one obtains a max-min-max problem, and one can use Sion's minimax theorem again to swap the inner maximization over μ with the minimization over $\mathbf{u}$ to obtain a robust counterpart that can then be further reformulated using duality or otherwise solved using delayed constraint generation. We provide the details of this derivation in Appendix B.2.1.

3.5.4 Log-log demand model

To now show how the solution scheme in Section 3.5.1 applies to the log-log approach, we again recall the form of the log-log revenue function:

$$R(\mathbf{p}, \mathbf{u}) = \sum_{i=1}^{I} p_i \cdot e^{\alpha_i - \beta_i \log p_i + \sum_{j \neq i} \gamma_{i,j} \log p_j} \tag{3.30}$$

$$= \sum_{i=1}^{I} e^{\alpha_i + \log p_i - \beta_i \log p_i + \sum_{j \neq i} \gamma_{i,j} \log p_j} \tag{3.31}$$

Using the same biconjugate trick as with the semi-log approach, we can show that

$$\begin{aligned}
&\max_{\mathbf{p} \in \mathcal{P}} \log R(\mathbf{p}, \mathbf{u}) \\
&= \max_{\mathbf{p} \in \mathcal{P}} \log \left(\sum_{i=1}^{I} e^{\alpha_i + \log p_i - \beta_i \log p_i + \sum_{j \neq i} \gamma_{i,j} \log p_j} \right) \\
&= \max_{\mathbf{p} \in \mathcal{P}} \max_{\boldsymbol{\mu} \in \Delta_{[I]}} \left\{ \sum_{i=1}^{I} \mu_i (\alpha_i + \log p_i - \beta_i \log p_i + \sum_{j \neq i} \gamma_{i,j} \log p_j) - \sum_{i=1}^{I} \mu_i \log \mu_i \right\}
\end{aligned} \tag{3.32}$$

If we now invoke Assumption 8, then we can introduce the same decision variables $x_{i,t}$ and $w_{i,j,t}$ as in problem (3.25) to obtain a mixed-integer convex formulation of the log-log price optimization problem, which has the same feasible region as the semi-log formulation (3.25):

$$\begin{aligned}
\underset{w,x,\boldsymbol{\mu}}{\text{maximize}} \quad & \sum_{i=1}^{I} \mu_i \alpha_i + \sum_{i=1}^{I} \sum_{t \in \mathcal{P}_i} \log t \cdot w_{i,i,t} - \sum_{i=1}^{I} \beta_i \cdot \sum_{t \in \mathcal{P}_i} \log t \cdot w_{i,i,t} \\
& \qquad + \sum_{i=1}^{I} \sum_{j \neq i} \gamma_{i,j} \sum_{t \in \mathcal{P}_j} \log t \cdot w_{i,j,t} - \sum_{i=1}^{I} \mu_i \log \mu_i
\end{aligned} \tag{3.33a}$$

$$\text{subject to} \quad \text{constraints (3.25b) – (3.25h).} \tag{3.33b}$$

While the feasible region of problem (3.33) is the same as that of (3.25), the objective function of (3.33) is different. Just like problem (3.25), problem (3.33) can be written as a mixed-integer exponential cone program, and similarly, to the best of our knowledge, this is the first exact mixed-integer convex formulation of the log-log multi-product price optimization problem under a Cartesian product price set. Lastly, just like the semi-log problem (3.25), one can easily modify the formulation to obtain an exact formulation of the deterministic robust price optimization problem under log-log demand (see Appendix B.2.2).

We note that the log-log separation problem has an interesting property, which is that there exist optimal solutions that are extreme, in the sense that each product's price is set to either its lowest or highest allowable price. This property is formalized in the following proposition (see Section B.1.7 for the proof).

Proposition 2 *Suppose that Assumption 8 holds. Let* $(\boldsymbol{\mu}, \mathbf{p})$ *be an optimal solution of problem* (3.32)*. Then there exists an optimal solution* $(\boldsymbol{\mu}, \mathbf{p}')$*, such that for each* $i \in [I]$*, either* $p'_i = \min \mathcal{P}_i$ *or* $p'_i = \max \mathcal{P}_i$*.*

3.6 Solution method for finite $\mathcal{P}$, finite $\mathcal{U}$

In addition to the case where $\mathcal{U}$ is convex, we also consider the case where $\mathcal{U}$ is a finite discrete set. Due to page limitations, our presentation of our solution method for this case is relegated to Appendix B.3. At a high level, the foundation of our approach is double column generation, which alternates between solving the primal version of the RRPO problem, which is $\max_{\boldsymbol{\pi} \in \Delta_{\mathcal{P}}} \min_{\mathbf{u} \in \mathcal{U}} \sum_{\mathbf{p} \in \mathcal{P}} \pi_{\mathbf{p}} R(\mathbf{p}, \mathbf{u})$, and the dual version of the RRPO problem, which is $\min_{\boldsymbol{\lambda} \in \Delta_{\mathcal{U}}} \max_{\mathbf{p} \in \mathcal{P}} \sum_{\mathbf{u} \in \mathcal{U}} \lambda_{\mathbf{u}} R(\mathbf{p}, \mathbf{u})$. In each iteration, we solve the primal problem with $\mathcal{P}$ replaced by a subset $\hat{\mathcal{P}} \subseteq \mathcal{P}$, where we use constraint generation to handle the inner minimization over $\mathbf{u} \in \mathcal{U}$; this gives rise to a finite set of uncertainty realizations $\hat{\mathcal{U}} \subseteq \mathcal{U}$. We then solve the dual problem with $\mathcal{U}$ replaced by $\hat{\mathcal{U}}$, where we use constraint generation to handle the inner maximization over $\mathbf{p} \in \mathcal{P}$, which gives rise to a finite set of price vectors

$\hat{\mathcal{P}} \subseteq \mathcal{P}$. At each step of the algorithm, the objective value of the primal problem restricted to $\hat{\mathcal{P}}$ is a lower bound on the true optimal objective, while the objective value of the dual problem restricted to $\hat{\mathcal{U}}$ is an upper bound on the optimal objective; the algorithm terminates when these two bounds are equal or are otherwise within a pre-specified tolerance.

To implement this approach for the demand models that we consider, one needs to be able to solve the primal separation problem (solve $\min_{\mathbf{u}\in\mathcal{U}} \sum_{\mathbf{p}\in\hat{\mathcal{P}}} \pi_{\mathbf{p}} R(\mathbf{p}, \mathbf{u})$) and the dual separation problem (solve $\max_{\mathbf{p}\in\mathcal{P}} \sum_{\mathbf{u}\in\hat{\mathcal{U}}} \lambda_{\mathbf{u}} \cdot R(\mathbf{p}, \mathbf{u})$). We show how both of these problems can be reformulated as mixed-integer exponential cone programs for the semi-log and log-log demand models, and as mixed-integer linear programs for the linear demand model.

3.7 Numerical experiments

In this section, we conduct several sets of experiments involving synthetic problem instances to understand the tractability of the RRPO approach and the improvement in worst-case revenue of the randomized robust pricing strategy over the deterministic robust pricing strategy. In Section 3.7.1, we consider instances involving the linear, semi-log and log-log models where the uncertainty set $\mathcal{U}$ is a convex set. In Section 3.7.2, we consider instances involving the linear, semi-log and log-log models where the uncertainty set $\mathcal{U}$ is a discrete set. Finally, in Section 3.7.3, we consider log-log and semi-log robust price optimization instances derived from a real data set on sales of orange juice products from a grocery store chain.

All of our code is implemented in the Julia programming language (Bezanson et al. 2017). All optimization models are implemented using the JuMP package (Lubin and Dunning 2015). All linear and mixed-integer linear programs are solved using Gurobi Gurobi Optimization, Inc. (2022) and all mixed-integer exponential cone programs are solved using Mosek (ApS 2022), with a maximum of 8 threads per program. All of our experiments are conducted on Amazon Elastic Compute Cloud (EC2), on a single instance of type `m6a.48xlarge`

(AMD EPYC 7R13 processor, with 192 virtual CPUs and 768 GB of memory).

3.7.1 Experiments with convex $\mathcal{U}$ and linear, log-log and semi-log demand models

In our first set of experiments, we consider the log-log and semi-log demand models, and specifically consider a $L1$-norm uncertainty set $\mathcal{U}$:

$$\mathcal{U} = \{\mathbf{u} = (\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) \mid \|\tilde{\mathbf{u}}\|_1 \leq \theta, [\tilde{\mathbf{u}}_k = \frac{\mathbf{u}_k - \mathbf{u}_{0k}}{\mathbf{u}_{0k}} \ \forall k \in \{1, .., I + I^2\}]\}, \tag{3.34}$$

where $\mathbf{u}_0$ is the vector of nominal values of the uncertain parameters $\mathbf{u} = (\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$, and θ is the budget of the uncertainty set.

For each of the three demand models (linear, semi-log and log-log), we vary the number of products I varies in $\{5, 10, 15, 20\}$. For each value of I, we generate 24 random instances, where the values of $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$ are independently randomly generated as follows:

1. *Linear demand.*

   Each $\alpha_i \sim$ Uniform$(200, 300)$, $\beta_i \sim$ Uniform$(5, 15)$, $\gamma_{i,j} \sim$ Uniform$(-0.1, +0.1)$.

2. *Semi-log demand.*

   Each $\alpha_i \sim$ Uniform$(4, 7)$, $\beta_i \sim$ Uniform$(1, 1.5)$, $\gamma_{i,j} \sim$ Uniform$(-0.4, +0.4)$.

3. *Log-log demand.*

   Each $\alpha_i \sim$ Uniform$(10, 14)$, $\beta_i \sim$ Uniform$(1, 2)$, $\gamma_{i,j} \sim$ Uniform$(-0.6, +0.6)$.

For each product $i \in [I]$, we set $\mathcal{P}_i = \{1, 2, 3, 4, 5\}$.

For the uncertainty set $\mathcal{U}$, the budget parameter θ varies in $\{0.1, 0, 5, 1, 1.5, 2\}$ for each instance.

For each instance, we solve the nominal problem, the DRPO problem and the RRPO problem. For DRPO and RRPO, we vary the budget parameter θ that defines the uncertainty within the set $\{0.1, 0.5, 1, 1.5, 2\}$. To solve the RRPO problem for each instance, we

execute the constraint generation solution algorithm described in Section 3.5. For instances with log-log demand, we take advantage of Proposition 2 and thus simplify the price set $\mathcal{P}$ to contain the highest and lowest price levels for each product only. To solve the DRPO problem, we formulate it as either a mixed-integer linear program (for linear demand) or a mixed-integer exponential cone program (for semi-log and log-log demand) via the log-sum-exp biconjugate-based technique described in Appendix B.2, and use standard LP duality techniques to reformulate the objective function of the resulting problem (formulation (B.7) and formulation (B.8) in Section B.2). Due to the prohibitive computation times that we encountered for the DRPO problem with log-log and semi-log demand, we impose a computation time limit of 20 minutes. From our experimentation with the DRPO problem for log-log and semi-log, it is often the case that an optimal or nearly optimal solution is found early on, and the bulk of the remaining computation time, which can be in the hours, is required by Mosek to prove optimality and close the gap. Finally, to solve the nominal problem for each instance, we also use the same biconjugate-based technique to formulate the nominal price optimization problem as a mixed-integer exponential cone program.

We present the objective value as well as the computation time of each RRPO, DRPO and nominal problem. We additionally compute several other metrics. We compute $\mathbb{E}[R(\mathbf{p}^*_{\mathrm{RR}}, \mathbf{u}_0)]$, which is the expected revenue of the randomized RPO solution assuming that the nominal parameter values are realized. We also compute $R(\mathbf{p}^*_{\mathrm{DR}}, \mathbf{u}_0)$, the nominal revenue of DRPO solution, and $Z_{\mathrm{N,WC}} = \min_{\mathbf{u}\in\mathcal{U}} R(\mathbf{p}^*_{\mathrm{N}}, \mathbf{u})$, the worst-case revenue of the nominal solution. We use the following metric to show the benefit of randomized strategy in robust price optimization:

$$\mathrm{RI} = (Z^*_{\mathrm{RR}} - Z^*_{\mathrm{DR}})/Z^*_{\mathrm{DR}} \times 100\% \tag{3.35}$$

For each metric, we compute its average over the 24 instances for each value of I and θ.

Tables 3.1, 3.2 and 3.3 shows the results for the linear, semi-log and log-log demand models, respectively. For linear demand, we find that the improvement by randomized robust pricing over deterministic robust pricing is modest; the largest average improvement

is 4.63% for $I = 5$, $\theta = 2$. We note that we experimented with other forms of uncertainty sets and choices of the nominal parameter values, but we generally did not encounter large improvements of the same size as we did for the other two demand models.

I	θ	t_{RR}	Z^*_{RR}	$\mathbb{E}[R(\mathbf{p}^*_{\mathrm{RR}}, \mathbf{u}_0)]$	t_{DR}	Z^*_{DR}	RI(%)	$R(\mathbf{p}^*_{\mathrm{DR}}, \mathbf{u}_0)$	t_{N}	Z^*_{N}	$Z_{\mathrm{N,WC}}$
5	0.1	7.77	4825.94	4967.01	0.40	4825.94	0.00	4967.01	0.34	4967.01	4825.94
5	0.5	0.06	4261.65	4967.01	0.03	4261.65	0.00	4967.01	–	–	4261.65
5	1.0	0.14	3574.50	4929.31	0.03	3557.12	0.52	4960.78	–	–	3556.29
5	1.5	0.22	2912.85	4816.66	0.03	2859.49	1.98	4899.71	–	–	2878.88
5	2.0	0.32	2293.09	4686.41	0.03	2195.89	4.63	4717.99	–	–	2201.48
10	0.1	0.17	9851.95	9998.29	0.10	9851.95	0.00	9998.29	0.08	9998.29	9851.95
10	0.5	0.18	9266.59	9998.29	0.10	9266.59	0.00	9998.29	–	–	9266.59
10	1.0	0.43	8552.00	9958.68	0.11	8534.89	0.20	9998.29	–	–	8534.89
10	1.5	0.62	7853.61	9911.18	0.11	7803.34	0.65	9990.94	–	–	7831.55
10	2.0	0.81	7173.52	9856.00	0.12	7077.66	1.37	9941.59	–	–	7128.21
15	0.1	0.34	14855.50	15002.93	0.21	14855.50	0.00	15002.93	0.16	15002.93	14855.50
15	0.5	0.34	14265.78	15002.93	0.21	14265.78	0.00	15002.93	–	–	14265.78
15	1.0	0.85	13539.69	14975.91	0.23	13528.63	0.08	15002.93	–	–	13528.63
15	1.5	1.20	12825.97	14931.12	0.24	12791.49	0.27	15002.93	–	–	12810.52
15	2.0	1.60	12126.11	14916.28	0.24	12054.34	0.61	15002.93	–	–	12092.41
20	0.1	0.57	19776.82	19923.37	0.36	19776.82	0.00	19923.37	0.28	19923.37	19776.82
20	0.5	0.55	19190.63	19923.37	0.37	19190.63	0.00	19923.37	–	–	19190.63
20	1.0	1.41	18464.77	19910.10	0.39	18457.88	0.04	19923.37	–	–	18457.88
20	1.5	2.08	17744.39	19895.41	0.42	17725.13	0.11	19923.37	–	–	17735.21
20	2.0	2.89	17030.17	19859.43	0.43	16992.38	0.22	19923.37	–	–	17012.55

Table 3.1: Results for linear instances with convex $\mathcal{U}$.

Besides linear demand, these results also show that for semi-log and log-log demand, there can be a very large difference between the randomized and deterministic robust pricing schemes. The benefit of randomization, quantified by the metric RI, ranges from about 3% to as much as 1320% for semi-log instances, and from about 7% to 243% for log-log

I	θ	t_{RR}	Z^*_{RR}	$\mathbb{E}[R(\mathbf{p}^*_{\mathrm{RR}}, \mathbf{u}_0)]$	t_{DR}	Z^*_{DR}	RI(%)	$R(\mathbf{p}^*_{\mathrm{DR}}, \mathbf{u}_0)$	t_{N}	Z^*_{N}	$Z_{\mathrm{N,WC}}$
5	0.1	12.04	2.60×10^3	4.53×10^3	0.45	2.55×10^3	2.86	4.56×10^3	0.37	4.56×10^3	2.54×10^3
5	0.5	0.28	5.40×10^2	2.92×10^3	0.18	4.26×10^2	30.89	3.67×10^3	–	–	3.25×10^2
5	1.0	0.34	2.14×10^2	2.33×10^3	0.19	1.52×10^2	42.14	1.81×10^3	–	–	66.06
5	1.5	0.38	1.09×10^2	1.77×10^3	0.21	72.97	49.46	9.42×10^2	–	–	27.85
5	2.0	0.39	60.06	1.67×10^3	0.22	38.76	53.11	9.17×10^2	–	–	14.54
10	0.1	0.80	2.29×10^5	4.05×10^5	0.45	2.27×10^5	5.14	4.08×10^5	0.20	4.08×10^5	2.27×10^5
10	0.5	1.55	5.44×10^4	2.44×10^5	1.77	2.83×10^4	86.68	2.84×10^5	–	–	2.44×10^4
10	1.0	2.35	1.93×10^4	2.00×10^5	4.78	6.22×10^3	178.14	1.41×10^5	–	–	3.24×10^3
10	1.5	3.36	8.37×10^3	1.59×10^5	12.76	2.19×10^3	242.84	8.94×10^4	–	–	1.20×10^3
10	2.0	5.03	4.62×10^3	1.29×10^5	21.73	1.07×10^3	273.72	1.42×10^5	–	–	6.58×10^2
15	0.1	1.83	7.26×10^6	1.33×10^7	1.93	7.24×10^6	2.79	1.34×10^7	0.75	1.34×10^7	7.24×10^6
15	0.5	4.50	1.18×10^6	9.18×10^6	11.57	7.12×10^5	91.22	1.29×10^7	–	–	6.63×10^5
15	1.0	10.18	3.41×10^5	6.11×10^6	123.07	9.89×10^4	302.39	6.11×10^6	–	–	5.26×10^4
15	1.5	20.74	1.47×10^5	5.46×10^6	499.02	3.64×10^4	406.11	6.59×10^6	–	–	2.40×10^4
15	2.0	32.79	7.67×10^4	4.79×10^6	799.69	1.66×10^4	502.85	7.32×10^6	–	–	1.21×10^4
20	0.1	6.65	2.17×10^8	4.13×10^8	5.74	2.16×10^8	3.87	4.13×10^8	1.66	4.13×10^8	2.16×10^8
20	0.5	25.98	2.33×10^7	1.96×10^8	204.50	1.65×10^7	178.27	4.11×10^8	–	–	1.63×10^7
20	1.0	36.70	6.93×10^6	1.83×10^8	795.72	1.11×10^6	715.40	7.17×10^7	–	–	7.42×10^5
20	1.5	59.76	3.01×10^6	1.30×10^8	1036.22	3.64×10^5	1000.01	3.66×10^8	–	–	3.1×10^5
20	2.0	75.66	1.56×10^6	2.05×10^8	1129.75	1.72×10^5	1320.44	2.52×10^8	–	–	1.43×10^5

Table 3.2: Results for semi-log instances with convex $\mathcal{U}$.

instances. Note that the magnitude of RI for the semi-log instances is larger than that for log-log, because the logarithm of demand in the semi-log model has a linear dependence on price which results in an exponential dependence of demand on price, but in log-log, the logarithm of demand is linear in the logarithm of price, resulting in a milder polynomial dependence of demand on price. For semi-log and log-log demand, both the worst-case revenue of RRPO solution and the worst-case revenue of DRPO solution decrease as the uncertainty set becomes larger, and the rate of reduction becomes less as the uncertainty budget θ is larger. In addition, for linear, semi-log and log-log demand, the RI generally increases as the uncertainty budget θ increases. Also, as we expect, $Z^*_{\mathrm{RR}} \geq Z^*_{\mathrm{DR}} \geq Z_{\mathrm{N,WC}}$.

I	θ	t_{RR}	Z^*_{RR}	$\mathbb{E}[R(\mathbf{p}^*_{\text{RR}}, \mathbf{u}_0)]$	t_{DR}	Z^*_{DR}	RI(%)	$R(\mathbf{p}^*_{\text{DR}}, \mathbf{u}_0)$	t_{N}	Z^*_{N}	$Z_{\text{N,WC}}$
5	0.5	11.93	3.47×10^5	2.01×10^6	0.39	3.30×10^5	6.90	1.95×10^6	0.55	2.85×10^6	2.34×10^5
5	1.0	0.16	9.74×10^4	1.94×10^6	0.14	9.23×10^4	7.24	1.88×10^6	–	–	6.03×10^4
5	1.5	0.17	2.87×10^4	1.94×10^6	0.14	2.70×10^4	7.93	1.83×10^6	–	–	1.76×10^4
5	2.0	0.17	8.52×10^3	1.94×10^6	0.14	7.99×10^3	7.93	1.83×10^6	–	–	5.20×10^3
10	0.5	1.53	2.78×10^6	1.20×10^7	25.53	2.07×10^6	34.28	1.07×10^7	0.55	2.24×10^7	1.37×10^6
10	1.0	2.21	1.15×10^6	9.94×10^6	44.96	8.27×10^5	38.84	7.59×10^6	–	–	5.04×10^5
10	1.5	2.94	5.67×10^5	9.20×10^6	54.57	4.03×10^5	41.13	8.15×10^6	–	–	2.41×10^5
10	2.0	2.93	3.00×10^5	8.74×10^6	61.97	2.11×10^5	43.68	7.21×10^6	–	–	1.24×10^5
15	0.5	11.42	1.18×10^7	5.42×10^7	934.31	7.45×10^6	70.56	5.77×10^7	4.63	1.28×10^8	4.45×10^6
15	1.0	22.91	5.32×10^6	3.77×10^7	1193.31	2.98×10^6	85.48	3.43×10^7	–	–	1.68×10^6
15	1.5	32.63	3.01×10^6	3.28×10^7	1200.57	1.58×10^6	93.71	2.51×10^7	–	–	8.68×10^5
15	2.0	38.84	1.83×10^6	3.06×10^7	1200.61	9.36×10^5	99.16	2.05×10^7	–	–	5.00×10^5
20	0.5	42.59	5.16×10^7	2.40×10^8	1200.50	1.89×10^7	178.79	1.41×10^8	21.44	6.77×10^8	9.07×10^6
20	1.0	115.78	2.37×10^7	1.62×10^8	1200.88	8.04×10^6	209.87	1.04×10^8	–	–	3.65×10^6
20	1.5	197.12	1.40×10^7	1.33×10^8	1200.97	4.40×10^6	229.16	8.81×10^7	–	–	2.04×10^6
20	2.0	257.13	8.95×10^6	1.21×10^8	1201.03	2.66×10^6	243.71	8.34×10^7	–	–	1.27×10^6

Table 3.3: Results for log-log instances with convex $\mathcal{U}$.

Interestingly, the randomized robust pricing scheme can achieve better performance than the deterministic robust scheme under the nominal demand model (for example, compare $\mathbb{E}[R(\mathbf{p}^*_{\text{RR}}, \mathbf{u}_0)]$ and $R(\mathbf{p}^*_{\text{DR}}, \mathbf{u}_0)$ for log-log demand with $I = 10$); this appears to be the case for almost all (I, θ) combinations for log-log, and for a smaller set of (I, θ) combinations for semi-log.

With regard to the computation time, we observe that the computation time generally grows with the number of products for both RRPO and DRPO. For linear demand, both RRPO and DRPO can be solved extremely quickly (no more than 3 seconds on average, even with $I = 20$ products). For log-log and semi-log, when the number of products is held constant, the amount of time required to solve either RRPO generally becomes larger as the uncertainty set becomes larger. However, what we find is that for both log-log and semi-log demand, RRPO generally requires much less time to solve to complete optimality than

DRPO; this is likely because the nominal problem (which is a key piece of the constraint generation method for RRPO when $\mathcal{U}$ is convex) can be solved rapidly, whereas the robust version of this mixed-integer exponential cone program is more challenging for Mosek.

3.7.2 Experiments with discrete $\mathcal{U}$ and and linear, log-log and semi-log demand models

In our second set of experiments, we consider linear, log-log and semi-log demand models, where uncertainty is modeled through a discrete uncertainty set. We specifically consider a discrete budget uncertainty set $\mathcal{U}$ here:

$$\mathcal{U} = \{\mathbf{u} = \mathbf{u}_0 - (\mathbf{u}_0 - \bar{\mathbf{u}}) \circ \xi - (\mathbf{u}_0 - \underline{\mathbf{u}}) \circ \eta \mid \mathbf{e}^\top \xi + \mathbf{e}^\top \eta \leq \Gamma, \xi + \eta \leq \mathbf{1}, \xi, \eta \in \{0,1\}^{I+I^2}\}, \quad (3.36)$$

where $\underline{\mathbf{u}}$ and $\bar{\mathbf{u}}$ are respectively the component-wise lower and upper bounds of $\mathbf{u}$, $\mathbf{u}_0$ is the nominal value of the uncertain parameter vector $\mathbf{u} = (\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$, Γ is the budget of uncertainty and $I + I + I(I-1) = I + I^2$ is the total number of demand model parameters. Under the budget uncertainty set $\mathcal{U}$, up to Γ parameters can attain their lower bounds or upper bounds, whereas the remaining parameters can only attain their nominal values. We shall assume that the lower bound vector $\underline{\mathbf{u}}$ and upper bound vector $\overline{\mathbf{u}}$ are defined as $\underline{\mathbf{u}} = 0.7\mathbf{u}_0$ and $\overline{\mathbf{u}} = 1.3\mathbf{u}_0$, where $\mathbf{u}_0$ is the vector of nominal parameters.

For each of the three demand models (linear, semi-log and log-log), we vary the number of products I in $\{5, 10, 15\}$. For each value of I, we generate 24 random instances, where the values of $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$ are independently randomly generated as follows:

1. *Linear demand.*

   Each $\alpha_i \sim \text{Uniform}(100, 200)$, $\beta_i \sim \text{Uniform}(5, 15)$, $\gamma_{i,j} \sim \text{Uniform}(-0.1, +0.1)$.

2. *Semi-log demand.*

   Each $\alpha_i \sim \text{Uniform}(8, 10)$, $\beta_i \sim \text{Uniform}(1.5, 2)$, $\gamma_{i,j} \sim \text{Uniform}(-0.5, +0.5)$.

3. *Log-log demand.*

Each $\alpha_i \sim \text{Uniform}(10, 14)$, $\beta_i \sim \text{Uniform}(1.5, 2)$, $\gamma_{i,j} \sim \text{Uniform}(-0.8, +0.8)$.

We set the price set of each $i \in [I]$ as $\mathcal{P}_i = \{1, 2, 3, 4, 5\}$.

For each instance, we solve the nominal problem, the DRPO problem and the RRPO problem. For both RRPO and DRPO, we test a different collection of Γ values for the uncertainty set depending on the value of I.

To solve the RRPO problem for each instance, we execute the double column generation algorithm described in Section B.3. In our preliminary experimentation with the restricted dual problem, we observed that exactly solving the dual separation problem (B.32) (for semi-log demand) or (B.38) (for log-log demand) via Mosek takes quite a long time. Therefore, to reduce the computation time of RRPO with discrete $\mathcal{U}$, we instead use a random improvement heuristic to obtain the solution of dual separation problem. Specifically, we randomly select a price vector $\mathbf{p}^0$ as a starting point. We start with changing the price of product $i = 1$ and keeping the prices of all other products unchanged, to search for a price vector $\mathbf{p}^1$ that makes the objective value of the dual separation problem the largest. Then based on the current price vector $\mathbf{p}^1$, we change the price of product $i = 2$ and keep the prices of all other products unchanged, to search for a better price vector $\mathbf{p}^2$. We repeat this for all of the products, yielding the price vector $\mathbf{p}^I$. We repeat this procedure with 100 random starting points, and retain the best solution over these 100 repetitions. Although this approximate method cannot guarantee that the overall double column generation procedure converges to a provably optimal solution, our preliminary experimentation with small instances suggests that it obtains the exact solution of RRPO that one would obtain if the dual separation problem were solved to provable optimality. For the linear demand model, we solve both primal and dual separation problems as mixed-integer programs in Gurobi.

With regard to the DRPO problem for each log-log and semi-log instance, we note that we do not have a solution algorithm or formulation to solve it exactly. Therefore, we again use the same random improvement heuristic to obtain an approximate solution of DRPO

with these demand models. We randomly pick a starting price vector, and change the price of one product at a time to improve the worst case objective value until we no longer get an improvement. We repeat this procedure 50 times and select the best resulting price vector from these 50 repetitions as the approximate solution of DRPO. We note that we use a smaller number of repetitions because each repetition involves solving worst-case problem over $\mathbf{u} \in \mathcal{U}$ repeatedly in order to evaluate the robust objective of each candidate price vector; this contributes to a large overall computation time for this approach. With regard to the DRPO problem for linear demand, we observe that the objective function of DRPO is linear in the uncertain parameter vector $\mathbf{u}$, and that the description of the set polyhedron (3.36) is integral (i.e., extreme points of this polyhedron naturally correspond to $\boldsymbol{\xi}, \boldsymbol{\eta} \in \{0,1\}^{2I+I^2}$). Therefore, DRPO can be solved exactly by relaxing the requirement $\boldsymbol{\xi}, \boldsymbol{\eta} \in \{0,1\}^{2I+I^2}$ in the uncertainty set (3.36), and reformulating the worst-case objective using LP duality, leading to a mixed-integer linear program.

Lastly, for the nominal problem for each instance, we use the biconjugate technique to formulate it as a mixed-integer exponential cone program.

We report the same metrics as in Section 3.7.1, with two minor modifications. We use $\hat{Z}_{\mathrm{DR}}$ and $\hat{\mathbf{p}}_{\mathrm{DR}}$ to denote the approximate objective value and solution of DRPO given by the random improvement heuristic. The approximate improvement percentage is then $\hat{\mathrm{RI}} = (Z^*_{\mathrm{RR}} - \hat{Z}_{\mathrm{DR}})/\hat{Z}_{\mathrm{DR}} \times 100\%$.

Table 3.4 shows the results for the linear demand model. Here, we interestingly find that the vast majority of instances are randomization-proof, i.e., the average RI is below 1%, if not exactly 0%. We note here that we tested other families of instances where $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$ and $\mathcal{P}_i$ are generated differently, but in virtually every case we found that the relative improvement of randomized over deterministic robust pricing was very small. These results, together with those for the convex $\mathcal{U}$ case, suggest that randomized pricing is of limited benefit compared to deterministic pricing for the uncertain linear demand model case.

Tables 3.5 and 3.6 show how the results vary for different values of discrete uncertainty

budget Γ for semi-log and log-log demand.[1] We can see that, in most of the cases we test, the randomized robust pricing strategy provides a substantial benefit over the deterministic robust price solution. The percentage improvement given by randomization ranges from 0% to as much as 488.59% for semi-log instances, and from 0% to 175.18% for log-log instances. Similar to the cases with convex $\mathcal{U}$, both Z^*_{RR} and $\hat{Z}_{\text{DR}}$ decrease as the uncertainty set becomes larger. While the RI metric generally decreases as Γ increases, in some instances it can be increasing in Γ at small values of Γ (this is visible in the average results metrics for $I = 10$ with semi-log demand). When Γ is large enough, the RI metric often becomes very small or even zero. This makes sense when interpreted through Corollary 3. Specifically, when nature is able to make a large number of demand model parameters take their worst values, it is likely that at the $\mathbf{u}^*$ at which the optimal objective of DRPO is attained is such that the price vector for the nominal problem with $\mathbf{u}^*$ coincides with the optimal price vector for DRPO. Thus, by Corollary 3, the problem will be randomization-proof.

With regard to the computation time, the computation time for both RRPO and DRPO increases with I. Interestingly, the computation time required by RRPO does not necessarily increase as the discrete uncertainty budget Γ increases; in some cases, when Γ is large, the RRPO solution degenerates to the DRPO solution, allowing the double column generation algorithm to terminate quickly. By comparing t_{RR} and t_{DR}, we can see that RRPO in general takes less time than DRPO. The computation time of the RRPO problem in semi-log instances is no more than approximately two minutes on average ($I = 15$, $\Gamma = 60$), while in log-log instances, solving RRPO requires no more than 1.5 minutes on average ($I = 15$, $\Gamma = 18$). Lastly, for linear demand, the computation time for RRPO is extremely small, requiring no more than a few seconds on average.

[1]We note here that for the log-log model, we encountered one instance ($I = 10$, $\Gamma = 44$) where $\hat{Z}_{\text{DR}}$ was higher than Z^*_{RR}; in general, Z^*_{RR} should be higher than Z^*_{DR}. We have verified that the reason for this anomaly was a numerical error in the solution of the worst-case subproblem in Mosek within the DRPO random improvement heuristic. This instance is omitted in our calculation of $\hat{Z}_{\text{DR}}$, RI and $\mathbb{E}[R(\mathbf{p}_{\text{DR}}, \mathbf{u}_0)]$, and the affected entries are indicated by * in Table 3.6.

3.7.3 Results using real data instances

In our last set of experiments, we evaluate the effectiveness of solution algorithms on problem instances calibrated with real data. For these experiments, we consider the `orangeJuice` data set from Montgomery (1997), which was accessed via the `bayesm` package in R (Rossi 2022). This data set contains price and sales data for $I = 11$ different orange juice brands at the Dominick's Finer Foods chain of grocery stores in the Chicago area. Each observation in the data set consists of: the store s; the week t; the log of the number of units sold $\log(q_{t,s,i})$ for brand i; the prices $p_{t,s,1}...p_{t,s,11}$ of the eleven orange juice brands; the dummy variable $d_{t,s,i}$ indicating whether brand i had any in-store displays at store s in week t; and the variable $f_{t,s,i}$ indicating if brand i was featured/advertised at store s in week t. We fit log-log and semi-log regression models for each brand i according to the following specifications:

$$\text{(semi-log)} \quad \log(q_{t,s,i}) = \alpha_i - \beta_i p_{t,s,i} + \sum_{j \neq i} \gamma_{ij} p_{t,s,j} + \theta_i d_{t,s,i} + \mu_i f_{t,s,i} + \epsilon_{t,s,i}, \tag{3.37}$$

$$\text{(log-log)} \quad \log(q_{t,s,i}) = \alpha_i - \beta_i \log(p_{t,s,i}) + \sum_{j \neq i} \gamma_{ij} \log(p_{t,s,j}) + \theta_i d_{t,s,i} + \mu_i f_{t,s,i} + \epsilon_{t,s,i}, \tag{3.38}$$

where $\{\epsilon_{t,s,i}\}_{t,s,i}$ is a collection of IID normally distributed error terms. The point estimates of the model parameters are provided in Appendix B.4.1. We note that prior work has considered the estimation of both of these types of models (see the examples in Rossi 2022; see also Montgomery 1997 and Mišić 2020).

We consider the problem of obtaining a price vector $\mathbf{p} = (p_1, p_2, \ldots, p_{11})$ for this collection of 11 products. To formulate the price vector set $\mathcal{P}$, we assume that each product i has five allowable prices, which are shown in Table 3.7. These prices correspond to the 0th (i.e., minimum), 25th, 50th, 75th and 100th (i.e., maximum) percentiles of the observed prices in the dataset.

For each type of demand model, we consider two forms of uncertainty set: a convex $L1$-norm uncertainty set (as in equation (3.34)) and a discrete budget uncertainty set (as

in equation (3.36)). We vary the budget θ of the $L1$-norm uncertainty set and present the results in Tables 3.8 and 3.9. We also vary the budget Γ of the discrete budget uncertainty set and present the results in tables B.3 and B.4. Specifically, for discrete budget uncertainty set, we assume that $\bar{\boldsymbol{\alpha}} = 1.2\boldsymbol{\alpha}$, $\underline{\boldsymbol{\alpha}} = 0.8\boldsymbol{\alpha}$, $\bar{\boldsymbol{\beta}} = 1.3\boldsymbol{\beta}$, $\underline{\boldsymbol{\beta}} = 0.7\boldsymbol{\beta}$, $\bar{\boldsymbol{\gamma}} = 1.4\boldsymbol{\gamma}$, and $\underline{\boldsymbol{\gamma}} = 0.6\boldsymbol{\gamma}$. Tables 3.8 and 3.9 below present the results under the convex $L1$-norm uncertainty set for the semi-log and log-log demand models, respectively. Due to page considerations, the results for the discrete $\mathcal{U}$ case are provided in Appendix B.4.2.

We can see from Tables 3.8 and 3.9 that the randomized pricing strategy performs significantly better than the deterministic pricing solution under the worst-case demand model, with the RI ranging from 17.86% to 47.81% for semi-log demand and from 27.71% to 92.31% for log-log demand. In addition, for the same demand type and uncertainty set, the computation time of RRPO is comparable to that of DRPO. With regard to the discrete uncertainty set case, the results shown in Section B.4.2 are qualitatively similar, with the randomized robust pricing strategy similarly outperforming the deterministic robust solution. We do also observe that under bothdemand models, solving RRPO with the discrete uncertainty set requires more time than solving it with convex uncertainty set, although the overall time is still reasonable (in the most extreme case, RRPO for the discrete uncertainty set can take up to approximately 300 seconds, and DRPO requires up to 600 second, compared to 60 seconds for both RRPO and DRPO for the $L1$-norm uncertainty set).

Lastly, it is also interesting to compare the randomized robust pricing strategy to the deterministic robust price vector. Taking the log-log demand model and the convex $L1$ uncertainty set with $\theta = 0.8$ as an example, the solution of the RRPO problem is the

following randomized pricing strategy:

$$\mathbf{p} = \begin{cases} (3.87, 5.82, 1.25, 0.99, 3.17, 5.09, 3.07, 0.91, 0.69, 2.69, 1.99) & \text{w.p. } 0.1628, \\ (1.29, 5.82, 3.35, 3.06, 0.88, 2.76, 3.07, 2.69, 3.08, 2.69, 4.99) & \text{w.p. } 0.1752, \\ (3.87, 2.86, 1.25, 3.06, 3.17, 5.09, 0.91, 2.69, 3.08, 0.52, 4.99) & \text{w.p. } 0.2658, \\ (3.87, 5.82, 3.35, 3.06, 0.88, 2.76, 3.07, 0.91, 3.08, 2.69, 4.99) & \text{w.p. } 0.0381, \\ (3.87, 2.86, 1.25, 3.06, 3.17, 2.76, 3.07, 2.69, 0.69, 2.69, 4.99) & \text{w.p. } 0.3258, \\ (3.87, 2.86, 1.25, 3.06, 3.17, 5.09, 0.91, 0.91, 3.08, 0.52, 4.99) & \text{w.p. } 0.0323. \end{cases} \tag{3.39}$$

Observe that in this randomized pricing strategy, each price vector is such that the product is set to either its lowest or highest allowable price. This is congruent with Proposition 2, which suggests that the nominal problem under the log-log demand model will always have a solution that involves setting each product to its highest or lowest price; since our solution algorithm is based on constraint generation using this nominal problem as a separation procedure, it makes sense that the randomized price vector will be supported on such extremal price vectors. On the other hand, the solution of the DRPO problem is the price vector $\mathbf{p}_{\text{DR}} = (3.87, 2.86, 1.25, 3.06, 3.17, 2.76, 0.91, 2.69, 0.69, 0.52, 4.99)$, for which we observe that the chosen prices are also either the lowest or highest for each product.

I	Γ	t_{RR}	Z^*_{RR}	$\mathbb{E}[R(\mathbf{p}^*_{\mathrm{RR}}, \mathbf{u}_0)]$	t_{DR}	Z^*_{DR}	RI(%)	$R(\mathbf{p}^*_{\mathrm{DR}}, \mathbf{u}_0)$	t_{N}	Z^*_{N}	$Z_{\mathrm{N,WC}}$
5	3	8.76	1724.38	2458.17	0.43	1723.46	0.06	2462.66	0.32	2473.30	1719.29
5	6	0.27	1316.39	2382.10	0.04	1313.68	0.23	2383.35	–	–	1261.84
5	9	0.27	1159.33	2294.44	0.03	1157.98	0.12	2293.84	–	–	1039.24
5	12	0.19	1126.85	2259.05	0.03	1126.85	0.00	2259.05	–	–	987.29
5	18	0.20	1124.87	2256.23	0.03	1124.87	0.00	2256.23	–	–	984.10
5	24	0.17	1123.78	2256.23	0.03	1123.78	0.00	2256.23	–	–	982.18
10	5	0.53	3717.70	4990.89	0.11	3714.08	0.10	4997.68	0.09	5009.03	3711.50
10	7	0.59	3315.59	4972.42	0.11	3312.91	0.08	4973.49	–	–	3297.07
10	9	0.70	2982.02	4941.26	0.11	2979.68	0.08	4944.40	–	–	2942.80
10	14	1.35	2589.70	4782.85	0.12	2583.93	0.23	4791.83	–	–	2437.09
10	19	0.84	2372.37	4662.16	0.11	2371.21	0.05	4657.17	–	–	2133.38
10	26	0.56	2342.57	4636.61	0.10	2342.57	0.00	4636.61	–	–	2086.03
10	33	0.57	2339.18	4633.87	0.11	2339.18	0.00	4636.61	–	–	2081.46
10	44	0.57	2334.90	4627.78	0.10	2334.90	0.00	4627.78	–	–	2075.15
15	6	0.92	5920.62	7495.77	0.22	5918.51	0.04	7506.26	0.17	7513.95	5917.60
15	12	1.79	4706.04	7435.56	0.25	4701.16	0.11	7442.32	–	–	4668.21
15	18	3.16	4051.33	7250.57	0.27	4045.35	0.15	7248.00	–	–	3889.25
15	24	4.50	3722.40	7071.15	0.29	3713.86	0.23	7068.49	–	–	3428.59
15	36	1.12	3500.73	6889.94	0.21	3500.73	0.00	6889.47	–	–	3112.83

Table 3.4: Results for linear instances with discrete $\mathcal{U}$.

I	Γ	t_{RR}	Z^*_{RR}	$\mathbb{E}[R(\mathbf{p}^*_{\mathrm{RR}}, \mathbf{u}_0)]$	t_{DR}	$\hat{Z}_{\mathrm{DR}}$	RI(%)	$R(\hat{\mathbf{p}}_{\mathrm{DR}}, \mathbf{u}_0)$	t_{N}	Z^*_{N}	$Z_{\mathrm{N,WC}}$
5	3	13.23	8.65×10^3	1.57×10^5	20.35	5.56×10^3	63.01	2.17×10^5	0.93	2.27×10^5	5.04×10^3
5	6	0.43	3.06×10^3	1.81×10^5	20.52	1.92×10^3	66.89	2.20×10^5	–	–	1.78×10^3
5	9	0.55	1.84×10^3	2.09×10^5	22.96	1.64×10^3	18.42	2.22×10^5	–	–	1.63×10^3
5	12	0.63	1.65×10^3	2.22×10^5	25.53	1.60×10^3	4.39	2.25×10^5	–	–	1.60×10^3
5	18	0.27	1.59×10^3	2.27×10^5	23.87	1.59×10^3	0.11	2.27×10^5	–	–	1.59×10^3
5	24	0.08	1.59×10^3	2.27×10^5	13.53	1.59×10^3	0.00	2.27×10^5	–	–	1.59×10^3
10	5	1.59	1.21×10^6	5.30×10^7	117.69	4.95×10^5	173.47	6.73×10^7	0.15	7.16×10^7	4.48×10^5
10	7	2.59	6.50×10^5	4.10×10^7	119.28	1.87×10^5	233.11	5.69×10^7	–	–	1.88×10^5
10	9	3.49	4.32×10^5	3.64×10^7	121.43	1.09×10^5	250.12	5.41×10^7	–	–	1.05×10^5
10	14	6.44	1.80×10^5	3.67×10^7	180.63	6.69×10^4	161.08	6.96×10^7	–	–	6.78×10^4
10	19	10.23	9.63×10^4	5.29×10^7	362.29	6.36×10^4	67.80	7.15×10^7	–	–	6.36×10^4
10	26	12.14	6.69×10^4	6.40×10^7	559.63	6.32×10^4	19.70	7.15×10^7	–	–	6.33×10^4
10	33	11.37	6.37×10^4	7.11×10^7	699.06	6.33×10^4	6.47	7.15×10^7	–	–	6.33×10^4
10	44	7.84	6.33×10^4	7.15×10^7	799.81	6.33×10^4	0.09	7.16×10^7	–	–	6.33×10^4
15	6	37.63	3.55×10^8	3.22×10^9	331.13	1.59×10^7	488.59	4.52×10^9	0.44	5.05×10^9	1.42×10^7
15	12	18.86	8.69×10^6	3.24×10^9	344.94	2.15×10^6	471.14	3.51×10^9	–	–	1.77×10^6
15	18	33.08	3.11×10^6	3.08×10^9	653.64	1.18×10^6	323.61	4.91×10^9	–	–	1.16×10^6
15	24	48.52	1.75×10^6	3.67×10^9	1525.41	1.12×10^6	165.75	5.03×10^9	–	–	1.11×10^6
15	36	101.38	1.21×10^6	4.82×10^9	3400.17	1.1×10^6	38.58	5.02×10^9	–	–	1.10×10^6

Table 3.5: Results for semi-log instances with discrete $\mathcal{U}$.

I	Γ	t_{RR}	Z^*_{RR}	$\mathbb{E}[R(\mathbf{p}^*_{\text{RR}}, \mathbf{u}_0)]$	t_{DR}	$\hat{Z}_{\text{DR}}$	RI(%)	$R(\hat{\mathbf{p}}_{\text{DR}}, \mathbf{u}_0)$	t_{N}	Z^*_{N}	$Z_{\text{N,WC}}$
5	3	13.33	3.26×10^5	1.86×10^6	31.23	2.80×10^5	22.03	1.89×10^6	0.96	4.31×10^6	1.49×10^5
5	6	0.20	6.37×10^4	4.03×10^6	20.39	6.15×10^4	3.16	4.29×10^6	–	–	6.1×10^4
5	9	0.27	4.91×10^4	4.19×10^6	18.39	4.87×10^4	0.65	4.24×10^6	–	–	4.77×10^4
5	12	0.35	4.71×10^4	4.20×10^6	16.82	4.70×10^4	0.20	4.18×10^6	–	–	4.57×10^4
5	18	0.17	4.66×10^4	4.18×10^6	11.67	4.66×10^4	0.01	4.18×10^6	–	–	4.52×10^4
5	24	0.14	4.66×10^4	4.18×10^6	8.73	4.66×10^4	0.05	4.16×10^6	–	–	4.52×10^4
10	5	4.64	2.33×10^6	3.89×10^7	161.57	1.38×10^6	83.54	6.45×10^7	0.31	7.66×10^7	1.23×10^6
10	7	5.39	1.31×10^6	5.22×10^7	165.21	8.59×10^5	63.34	7.12×10^7	–	–	8.39×10^5
10	9	5.30	8.74×10^5	5.15×10^7	149.63	6.23×10^5	38.99	7.28×10^7	–	–	6.12×10^5
10	14	5.10	5.07×10^5	5.67×10^7	138.67	3.82×10^5	29.31	7.30×10^7	–	–	3.80×10^5
10	19	3.52	3.74×10^5	6.35×10^7	136.70	3.31×10^5	12.90	7.51×10^7	–	–	3.33×10^5
10	26	3.56	3.27×10^5	7.38×10^7	138.77	3.18×10^5	4.28	7.59×10^7	–	–	3.21×10^5
10	33	4.79	3.18×10^5	7.60×10^7	137.07	3.15×10^5	2.34	7.59×10^7	–	–	3.17×10^5
10	44	3.19	3.15×10^5	7.62×10^7	131.47	3.19×10^5*	1.19*	7.53×10^7*	–	–	3.14×10^5
15	6	35.68	1.71×10^7	4.33×10^8	552.64	7.75×10^6	175.18	7.83×10^8	1.20	8.38×10^8	7.20×10^6
15	12	62.70	5.55×10^6	4.93×10^8	582.20	2.94×10^6	102.06	7.78×10^8	–	–	2.76×10^6
15	18	76.60	2.92×10^6	4.55×10^8	561.96	1.92×10^6	65.73	8.24×10^8	–	–	1.88×10^6
15	24	54.30	2.03×10^6	6.79×10^8	572.08	1.71×10^6	32.76	8.19×10^8	–	–	1.71×10^6
15	36	48.34	1.69×10^6	8.11×10^8	600.30	1.63×10^6	8.07	8.32×10^8	–	–	1.63×10^6

Table 3.6: Results for log-log instances with discrete $\mathcal{U}$.

Product	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
	1.29	2.86	1.25	0.99	0.88	2.76	0.91	0.91	0.69	0.52	1.99
	2.49	4.19	2.69	1.99	1.99	3.67	1.99	1.99	1.79	1.58	2.99
	2.99	4.75	2.89	2.35	2.17	3.96	2.39	2.19	1.99	1.59	3.59
	3.19	4.99	3.12	2.49	2.49	4.49	2.56	2.39	2.36	1.99	3.99
	3.87	5.82	3.35	3.06	3.17	5.09	3.07	2.69	3.08	2.69	4.99

Table 3.7: Possible price levels for products in `orangeJuice` experiment instances.

θ	t_{RR}	Z^*_{RR}	$\mathbb{E}[R(\mathbf{p}^*_{\text{RR}}, \mathbf{u}_0)]$	t_{DR}	Z^*_{DR}	RI(%)	$R(\mathbf{p}^*_{\text{DR}}, \mathbf{u}_0)$	t_{N}	Z^*_{N}	$Z_{\text{N,WC}}$
0.10	16.86	342357.06	481125.25	15.98	290474.67	17.86	590546.51	0.83	590547.01	290474.76
0.50	41.42	197517.06	373973.40	47.65	147748.35	33.68	294483.28	–	–	96016.90
0.80	42.61	149709.04	352742.22	67.83	105734.14	41.59	294483.28	–	–	67924.78
1.00	67.47	125987.02	349644.33	74.03	86977.24	44.85	265173.68	–	–	55394.70
1.50	61.70	82880.96	348467.74	44.65	56474.64	46.76	265173.68	–	–	34864.43
2.00	65.21	54665.15	348466.26	52.10	37164.75	47.09	265173.68	–	–	22615.70

Table 3.8: Results for `orangeJuice` pricing problem with semi-log demand and convex $\mathcal{U}$.

θ	t_{RR}	Z^*_{RR}	$\mathbb{E}[R(\mathbf{p}^*_{\text{RR}}, \mathbf{u}_0)]$	t_{DR}	Z^*_{DR}	RI(%)	$R(\mathbf{p}^*_{\text{DR}}, \mathbf{u}_0)$	t_{N}	Z^*_{N}	$Z_{\text{N,WC}}$
0.10	0.71	722647.22	1051269.80	1.97	565866.71	27.71	922172.97	0.88	1112050.59	560812.30
0.50	1.42	342614.34	687632.84	4.99	233387.10	46.80	782893.68	–	–	152881.89
0.80	1.91	260049.66	672481.74	6.77	162276.97	60.25	782893.68	–	–	102893.20
1.00	2.08	217580.86	683576.99	9.41	128220.45	69.69	782893.68	–	–	81427.57
1.50	2.26	142307.66	670759.12	13.02	75897.66	87.50	599315.12	–	–	48983.56
2.00	2.39	94847.37	670758.38	13.49	49319.21	92.31	377932.71	–	–	31055.19

Table 3.9: Results for `orangeJuice` pricing problem with log-log demand and convex $\mathcal{U}$.

CHAPTER 4

Conclusions

In this thesis, we have studied how randomization helps to find optimal linear policies in optimal stopping and to achieve better worst-case performance in robust pricing. We briefly conclude each chapter and discuss potential future directions below.

In Chapter 2, we consider the problem of designing randomized policies for high-dimensional optimal stopping problems. We formulate the problem as an SAA problem, prove its convergence properties and establish generalization error bounds on the out-of-sample reward. Based on the NP-Hardness of the SAA problem, we develop a backward optimization heuristic for approximately solving the SAA problem. We show in the numerical experiments that our heuristic can achieve better performance than the LSM method and is better or comparable to the PO method.

There are at least two interesting future directions for randomized policies in high-dimensional optimal stopping. First, it would be interesting to further understand the behavior of the non-convex objective function of the randomized policy SAA problem and of the period t problem in the backward optimization heuristic, and to understand how one can obtain high quality solutions to both of these problems. In particular, our experimentation suggests that quality of the solution in the period t problem is fairly sensitive to the choice of starting point, so it would be interesting to explore other ways of selecting initial points, as well as other methods beside Adam for solving the period t problem. Second, it would be interesting to explore whether our methodology can be generalized to other stochastic dynamic programming problems outside of optimal stopping.

In Chapter 3, we considered the problem of designing randomized robust pricing strategies to maximize worst-case revenue. We presented idealized conditions under which such randomized pricing strategies fare no better than the deterministic robust pricing approach, and subsequently we developed solution methods for obtaining the randomized pricing strategies in different settings (when the price set is finite, and when the uncertainty set is either convex or discrete). We showed using both synthetic instances and real data instances that such randomized pricing strategies can lead to large improvements in worst-case revenue over deterministic robust price prescriptions.

With regard to future research in randomized robust pricing, an interesting direction is to consider a version of the robust price optimization problem that incorporates contextual information. For example, in the ecommerce setting, different customers who log onto a retailer's website will differ in characteristics (age, web browser, operating system, etc.). This information could be used to craft a richer uncertainty set, and to motivate randomization strategies that randomize differently based on user characteristics. More generally, we hope that this work, which was inspired by the paper of Wang et al. (2024), motivates further study in how randomization can be used to mitigate risk in revenue management applications.

APPENDIX A

Randomized Policy Optimization for Optimal Stopping

A.1 Omitted proofs

A.1.1 Proof of Theorem 1

We prove this result in two steps. We first show that $\max_{\mathbf{b}\in\mathcal{B}} \hat{J}_D(\mathbf{b}) \leq \sup_{\tilde{\mathbf{b}}\in\tilde{\mathcal{B}}} \hat{J}_R(\tilde{\mathbf{b}})$, and then show that $\sup_{\mathbf{b}\in\mathcal{B}} \hat{J}_D(\mathbf{b}) \geq \sup_{\tilde{\mathbf{b}}\in\tilde{\mathcal{B}}} \hat{J}_R(\tilde{\mathbf{b}})$.

Proof of $\sup_{\mathbf{b}\in\mathcal{B}} \hat{J}_D(\mathbf{b}) \leq \sup_{\tilde{\mathbf{b}}\in\tilde{\mathcal{B}}} \hat{J}_R(\tilde{\mathbf{b}})$*:* To establish this, fix any deterministic policy weight vector $\mathbf{b} \in \mathcal{B}$.

Without loss of generality, we can assume that $\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega, t))$ satisfies either $\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega, t)) > 0$ or $\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega, t)) < 0$ for each ω and t. (Stated differently, $\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega, t))$ cannot be exactly equal to zero.) If this is not the case, then using Assumption 3, we can modify the weight $b_{t,1}$ of the constant basis function $\phi_1(\mathbf{x}) = 1$ for any period t such that the condition is satisfied, and the sample-average reward $\hat{J}_D(\mathbf{b})$ remains unchanged.

Now, consider the randomized policy weight vector $\mathbf{b}'$ defined as $\mathbf{b}' = \alpha\mathbf{b}$, where $\alpha > 0$. Observe now that, since $\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega, t)) > 0$ or $\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega, t)) < 0$ for each ω and t, we have

that

$$\begin{aligned}\lim_{\alpha\to+\infty}\sigma(\mathbf{b}'_t\bullet\Phi(\mathbf{x}(\omega,t)))&=\lim_{\alpha\to+\infty}\sigma(\alpha\mathbf{b}_t\bullet\Phi(\mathbf{x}(\omega,t)))\\&=\begin{cases}+1 & \text{if } \mathbf{b}_t\bullet\Phi(\mathbf{x}(\omega,t))>0,\\ 0 & \text{if } \mathbf{b}_t\bullet\Phi(\mathbf{x}(\omega,t))\leq 0\end{cases}\\&=\mathbb{I}\{\mathbf{b}_t\bullet\Phi(\mathbf{x}(\omega,t))>0\}.\end{aligned}$$

Consequently, we have that

$$\begin{aligned}&\lim_{\alpha\to+\infty}\hat{J}_R(\mathbf{b}')\\&=\lim_{\alpha\to+\infty}\hat{J}_R(\alpha\mathbf{b})\\&=\lim_{\alpha\to+\infty}\frac{1}{\Omega}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T}g(t,\mathbf{x}(\omega,t))\cdot\prod_{t'=1}^{t-1}(1-\sigma(\alpha\mathbf{b}_{t'}\bullet\Phi(\mathbf{x}(\omega,t'))))\cdot\sigma(\alpha\mathbf{b}_t\bullet\Phi(\mathbf{x}(\omega,t)))\\&=\frac{1}{\Omega}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T}g(t,\mathbf{x}(\omega,t))\cdot\prod_{t'=1}^{t-1}(1-\mathbb{I}\{\mathbf{b}_{t'}\bullet\Phi(\mathbf{x}(\omega,t'))>0\})\cdot\mathbb{I}\{\mathbf{b}_t\bullet\Phi(\mathbf{x}(\omega,t))>0\}\\&=\hat{J}_D(\mathbf{b}).\end{aligned}$$

Since $\mathbf{b}'\in\tilde{\mathcal{B}}=\mathbb{R}^{KT}$, we have that $\hat{J}_R(\alpha\mathbf{b})\leq\sup_{\tilde{\mathbf{b}}\in\tilde{\mathcal{B}}}\hat{J}_R(\tilde{\mathbf{b}})$ for all $\alpha>0$; as a result, the limit of $\hat{J}_R(\alpha\mathbf{b})$ as $\alpha\to\infty$ must also be upper bounded by $\sup_{\tilde{\mathbf{b}}\in\tilde{\mathcal{B}}}\hat{J}_R(\tilde{\mathbf{b}})$. We thus have that $\sup_{\tilde{\mathbf{b}}\in\tilde{\mathcal{B}}}\hat{J}_R(\tilde{\mathbf{b}})$ is an upper bound on $\hat{J}_D(\mathbf{b})$ for any $\mathbf{b}\in\mathcal{B}$.

By the definition of the supremum, it therefore follows that

$$\sup_{\mathbf{b}\in\mathcal{B}}\hat{J}_D(\mathbf{b})\leq\sup_{\tilde{\mathbf{b}}\in\tilde{\mathcal{B}}}\hat{J}_R(\tilde{\mathbf{b}}). \tag{A.1}$$

Proof of $\sup_{\mathbf{b}\in\mathcal{B}}\hat{J}_D(\mathbf{b})\geq\sup_{\tilde{\mathbf{b}}\in\tilde{\mathcal{B}}}\hat{J}_R(\tilde{\mathbf{b}})$*:* To establish this inequality, fix a randomized policy weight vector $\tilde{\mathbf{b}}$ from $\tilde{\mathcal{B}}$. The key idea in the proof is that the logistic response function $\sigma(\cdot)$ can also be viewed as the cumulative distribution function (CDF) of a logistic random variable. Recall that a logistic random variable, $\xi\sim\text{Logistic}(\mu,s)$, where μ is the location parameter and s is the scale parameter, has CDF given by

$$\mathbb{P}(\xi<t)=\frac{e^{(t-\mu)/s}}{1+e^{(t-\mu)/s}}.$$

Thus, the logistic response function $\sigma(\cdot)$ corresponds to a Logistic$(0,1)$ random variable.

Armed with this insight, let us define T i.i.d. Logistic$(0,1)$ random variables, $\xi_1, \ldots, \xi_T$. Observe that we can write the reward of the randomized policy as

$$\begin{aligned}
&\hat{J}_R(\tilde{\mathbf{b}}) \\
&= \frac{1}{\Omega}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} g(t, \mathbf{x}(\omega,t)) \cdot \prod_{t'=1}^{t-1}(1 - \sigma(\tilde{\mathbf{b}}_{t'} \bullet \Phi(\mathbf{x}(\omega,t')))) \cdot \sigma(\tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(\omega,t))) \\
&= \frac{1}{\Omega}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} g(t, \mathbf{x}(\omega,t)) \cdot \prod_{t'=1}^{t-1}\mathbb{P}(\xi_{t'} \geq \tilde{\mathbf{b}}_{t'} \bullet \Phi(\mathbf{x}(\omega,t'))) \cdot \mathbb{P}(\xi_t < \tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(\omega,t))) \\
&= \frac{1}{\Omega}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} g(t, \mathbf{x}(\omega,t)) \cdot \prod_{t'=1}^{t-1}\mathbb{E}[\mathbb{I}\{\xi_{t'} \geq \tilde{\mathbf{b}}_{t'} \bullet \Phi(\mathbf{x}(\omega,t'))\}] \cdot \mathbb{E}[\mathbb{I}\{\xi_t < \tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(\omega,t))\}] \\
&= \frac{1}{\Omega}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} g(t, \mathbf{x}(\omega,t)) \cdot \mathbb{E}\left[\prod_{t'=1}^{t-1}\mathbb{I}\{\xi_{t'} \geq \tilde{\mathbf{b}}_{t'} \bullet \Phi(\mathbf{x}(\omega,t'))\} \cdot \mathbb{I}\{\xi_t < \tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(\omega,t))\}\right] \\
&= \mathbb{E}\left[\frac{1}{\Omega}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} g(t, \mathbf{x}(\omega,t)) \cdot \prod_{t'=1}^{t-1}\mathbb{I}\{\xi_{t'} \geq \tilde{\mathbf{b}}_{t'} \bullet \Phi(\mathbf{x}(\omega,t'))\} \cdot \mathbb{I}\{\xi_t < \tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(\omega,t))\}\right] \quad \text{(A.2)}
\end{aligned}$$

where the second equality follows by the definition of each ξ_t as a Logistic$(0,1)$ random variable; the third by the fact that $\mathbb{P}(A) = \mathbb{E}[\mathbb{I}\{A\}]$ for any event A; the fourth by the fact that $\xi_1, \ldots, \xi_T$ are independent; and the fifth by the linearity of expectation.

We now observe that there must exist values $\bar{\xi}_1, \ldots, \bar{\xi}_T$ for which the random variable in (A.2) is at least its expected value, i.e.,

$$\begin{aligned}
&\mathbb{E}\left[\frac{1}{\Omega}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} g(t, \mathbf{x}(\omega,t)) \cdot \prod_{t'=1}^{t-1}\mathbb{I}\{\xi_{t'} \geq \tilde{\mathbf{b}}_{t'} \bullet \Phi(\mathbf{x}(\omega,t'))\} \cdot \mathbb{I}\{\xi_t < \tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(\omega,t))\}\right] \\
&\leq \frac{1}{\Omega}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} g(t, \mathbf{x}(\omega,t)) \cdot \prod_{t'=1}^{t-1}\mathbb{I}\{\bar{\xi}_{t'} \geq \tilde{\mathbf{b}}_{t'} \bullet \Phi(\mathbf{x}(\omega,t'))\} \cdot \mathbb{I}\{\bar{\xi}_t < \tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(\omega,t))\}.
\end{aligned}$$

Finally, let us define a deterministic policy weight vector $\mathbf{b}$ as

$$b_{t,k} = \begin{cases} \tilde{b}_{t,k} - \bar{\xi}_t & \text{if } k = 1, \\ \tilde{b}_{t,k} & \text{if } k \neq 1, \end{cases}$$

for each t and k. In other words, we decrease the weight on the constant basis function exactly by $\bar{\xi}_t$, the realized value of the tth logistic random variable. (Note that this construction is

made possible by Assumption 3.) By constructing $\mathbf{b}$ in this way, we obtain that

$$\begin{aligned}
&\bar{\xi}_t < \tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(\omega, t)) \\
&\Leftrightarrow \tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(\omega, t)) - \bar{\xi}_t > 0 \\
&\Leftrightarrow \mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega, t)) > 0
\end{aligned}$$

for each ω and t. We thus have that

$$\begin{aligned}
&\hat{J}_R(\tilde{\mathbf{b}}) \\
&= \mathbb{E}\left[\frac{1}{\Omega}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} g(t, \mathbf{x}(\omega, t)) \cdot \prod_{t'=1}^{t-1} \mathbb{I}\{\xi_{t'} \geq \tilde{\mathbf{b}}_{t'} \bullet \Phi(\mathbf{x}(\omega, t'))\} \cdot \mathbb{I}\{\xi_t < \tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(\omega, t))\}\right] \\
&\leq \frac{1}{\Omega}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} g(t, \mathbf{x}(\omega, t)) \cdot \prod_{t'=1}^{t-1} \mathbb{I}\{\bar{\xi}_{t'} \geq \tilde{\mathbf{b}}_{t'} \bullet \Phi(\mathbf{x}(\omega, t'))\} \cdot \mathbb{I}\{\bar{\xi}_t < \tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(\omega, t))\} \\
&= \frac{1}{\Omega}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} g(t, \mathbf{x}(\omega, t)) \cdot \prod_{t'=1}^{t-1} \mathbb{I}\{\mathbf{b}_{t'} \bullet \Phi(\mathbf{x}(\omega, t')) \leq 0\} \cdot \mathbb{I}\{\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega, t)) > 0\} \\
&= \hat{J}_D(\mathbf{b})
\end{aligned}$$

As a result, the reward of a randomized policy weight vector $\tilde{\mathbf{b}}$ can be bounded by the reward of a deterministic policy weight vector $\mathbf{b}$. Thus, $\sup_{\mathbf{b}\in\mathcal{B}} \hat{J}_D(\mathbf{b})$ is a valid upper bound on $\hat{J}_R(\tilde{\mathbf{b}})$ for any $\tilde{\mathbf{b}} \in \mathcal{B}$. By the definition of the supremum as the least upper bound, we consequently have

$$\sup_{\tilde{\mathbf{b}}\in\tilde{\mathcal{B}}} \hat{J}_R(\tilde{\mathbf{b}}) \leq \sup_{\mathbf{b}\in\mathcal{B}} \hat{J}_D(\mathbf{b}). \tag{A.3}$$

Since we have shown both inequalities, it follows $\sup_{\tilde{\mathbf{b}}\in\tilde{\mathcal{B}}} \hat{J}_R(\tilde{\mathbf{b}}) = \sup_{\mathbf{b}\in\mathcal{B}} \hat{J}_D(\mathbf{b})$, as required. □

A.1.2 Proof of Theorem 2

We prove this in two steps: first, by showing that $\sup_{\mathbf{b}\in\mathcal{B}} J_D(\mathbf{b}) \leq \sup_{\tilde{\mathbf{b}}\in\tilde{\mathcal{B}}} J_R(\tilde{\mathbf{b}})$, and then by showing that $\sup_{\mathbf{b}\in\mathcal{B}} J_D(\mathbf{b}) \geq \sup_{\tilde{\mathbf{b}}\in\tilde{\mathcal{B}}} J_R(\tilde{\mathbf{b}})$.

Step 1: $\sup_{\mathbf{b}\in\mathcal{B}} J_D(\mathbf{b}) \leq \sup_{\tilde{\mathbf{b}}\in\tilde{\mathcal{B}}} J_R(\tilde{\mathbf{b}})$. Let $\mathbf{b} \in \mathcal{B}$. Let $\alpha > 0$ be a constant, and define $\tilde{\mathbf{b}}$ as follows:

$$\tilde{\mathbf{b}}_t = \begin{cases} \alpha \mathbf{b}_t & \text{if } \mathbf{b}_t \neq \mathbf{0}, \\ -\alpha \mathbf{e}_1 & \text{if } \mathbf{b}_t = \mathbf{0}, \end{cases}$$

where $\mathbf{0}$ is a K-dimensional vector of zeros and $\mathbf{e}_1 = (1, 0, \ldots, 0)$ is the first standard basis vector for $\mathbb{R}^K$.

Let $I = \{t \in [T] \mid \mathbf{b}_t \neq \mathbf{0}\}$, and for each $t \in I$, define the set Q_t as

$$Q_t = \{(y_2, \ldots, y_K) \in \mathbb{R}^{K-1} \mid b_{t,1} + \sum_{k=2}^{K} y_k b_{t,k} = 0\}. \tag{A.4}$$

Observe that Q_t is a hyperplane in $\mathbb{R}^{K-1}$, so by Assumption 4, we have that

$$\mathbb{P}(\Phi_{2:K}(\mathbf{x}(t)) \in Q_t) = 0. \tag{A.5}$$

We note that the event $\Phi_{2:K}(\mathbf{x}(t)) \in Q_t$ is exactly the event that the inner product of $\mathbf{b}_t$ and $\Phi(\mathbf{x}(t))$ is equal to zero (i.e., we are on the boundary between choosing to stop or to continue): in particular, we have that

$$\begin{aligned} & \Phi_{2:K}(\mathbf{x}(t)) \in Q_t \\ \Leftrightarrow\ & b_{t,1} + \sum_{k=2}^{K} \phi_k(\mathbf{x}(t)) b_{t,k} = 0 \\ \Leftrightarrow\ & \sum_{k=1}^{K} \phi_k(\mathbf{x}(t)) b_{t,k} = 0 \\ \Leftrightarrow\ & \mathbf{b}_t \bullet \Phi(\mathbf{x}(t)) = 0 \end{aligned}$$

where the third step follows because $\phi_1(\mathbf{x}) = 1$ for all $\mathbf{x} \in \mathcal{X}$ (this is Assumption 3).

Let E be the event defined as

$$E = \bigcup_{t\in I} \{\Phi_{2:K}(\mathbf{x}(t)) \in Q_t\}. \tag{A.6}$$

Observe that $\mathbb{P}(E) = 0$ since

$$\begin{aligned}\mathbb{P}(E) &= \mathbb{P}\left(\bigcup_{t\in I}\{\Phi_{2:K}(\mathbf{x}(t)) \in Q_t\}\right)\\ &\leq \sum_{t\in I}\mathbb{P}(\Phi_{2:K}(\mathbf{x}(t)) \in Q_t)\\ &= 0,\end{aligned}$$

where the inequality follows by the countable subadditivity of $\mathbb{P}$.

Observe also that for any $(\mathbf{x}(1), \ldots, \mathbf{x}(T)) \notin E$, we have the following behavior: if $\mathbf{b}_t \neq \mathbf{0}$, then

$$\begin{aligned}&\lim_{\alpha\to+\infty}\sigma(\tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(t)))\\ &= \lim_{\alpha\to+\infty}\sigma(\alpha\mathbf{b}_t \bullet \Phi(\mathbf{x}(t)))\\ &= \begin{cases} 1 & \text{if } \mathbf{b}_t \bullet \Phi(\mathbf{x}(t)) > 0,\\ 0 & \text{if } \mathbf{b}_t \bullet \Phi(\mathbf{x}(t)) \leq 0,\end{cases}\\ &= \mathbb{I}\{\mathbf{b}_t \bullet \Phi(\mathbf{x}(t)) > 0\}.\end{aligned}$$

Otherwise, if $\mathbf{b}_t = \mathbf{0}$, then

$$\begin{aligned}&\lim_{\alpha\to+\infty}\sigma(\tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(t)))\\ &= \lim_{\alpha\to+\infty}\sigma(-\alpha\mathbf{e}_1 \bullet \Phi(\mathbf{x}(t)))\\ &= \lim_{\alpha\to+\infty}\sigma(-\alpha)\\ &= 0\\ &= \mathbb{I}\{\mathbf{b}_t \bullet \Phi(\mathbf{x}(t)) > 0\}.\end{aligned}$$

Therefore, for any $(\mathbf{x}(1), \ldots, \mathbf{x}(T)) \notin E$, we have

$$\begin{aligned}&\lim_{\alpha\to+\infty}\sum_{t=1}^{T} g(t, \mathbf{x}(t)) \cdot \prod_{t'=1}^{t-1}(1 - \sigma(\tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(t')))) \cdot \sigma(\tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(t)))\\ &= \sum_{t=1}^{T} g(t, \mathbf{x}(t)) \cdot \prod_{t'=1}^{t-1}\mathbb{I}\{\mathbf{b}_{t'} \bullet \Phi(\mathbf{x}(t')) \leq 0\} \cdot \mathbb{I}\{\mathbf{b}_t \bullet \Phi(\mathbf{x}(t)) > 0\}.\end{aligned}$$

In addition, for all $(\mathbf{x}(1), \ldots, \mathbf{x}(T))$, the term in the limit obeys the bound

$$\begin{aligned}
&\sum_{t=1}^{T} g(t, \mathbf{x}(t)) \cdot \prod_{t'=1}^{t-1} (1 - \sigma(\tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(t')))) \cdot \sigma(\tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(t))) \\
&\leq \sum_{t=1}^{T} g(t, \mathbf{x}(t)) \\
&\leq T \cdot \bar{G},
\end{aligned}$$

where the first inequality holds because $0 \leq \sigma(u) \leq 1$ for any real u, and the second holds by Assumption 1.

Therefore, by applying the bounded convergence theorem, we can assert that

$$\begin{aligned}
&\lim_{\alpha \to +\infty} J_R(\tilde{\mathbf{b}}) \\
&= \lim_{\alpha \to +\infty} \mathbb{E}\left[\sum_{t=1}^{T} g(t, \mathbf{x}(t)) \cdot \prod_{t'=1}^{t-1} (1 - \sigma(\tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(t')))) \cdot \sigma(\tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(t))) \right] \qquad \text{(A.7)} \\
&= \mathbb{E}\left[\sum_{t=1}^{T} g(t, \mathbf{x}(t)) \cdot \prod_{t'=1}^{t-1} \mathbb{I}\{\mathbf{b}_{t'} \bullet \Phi(\mathbf{x}(t')) \leq 0\} \cdot \mathbb{I}\{\mathbf{b}_t \bullet \Phi(\mathbf{x}(t)) > 0\} \right] \qquad \text{(A.8)} \\
&= J_D(\mathbf{b}).
\end{aligned}$$

Note that in our application of the bounded convergence theorem, we are using the fact that the functions of $(\mathbf{x}(1), \ldots, \mathbf{x}(T))$ whose expectation defines $J_R(\tilde{\mathbf{b}})$ in (A.7) converge pointwise to the function of $(\mathbf{x}(1), \ldots, \mathbf{x}(T))$ whose expectation defines $J_D(\mathbf{b})$ in (A.8) almost everywhere with respect to the probability measure of $(\mathbf{x}(1), \ldots, \mathbf{x}(T))$. (The only set of values of $(\mathbf{x}(1), \ldots, \mathbf{x}(T))$ on which the pointwise convergence does not hold is E, for which we have already established that $\mathbb{P}(E) = 0$.)

Thus, $\lim_{\alpha \to +\infty} J_R(\tilde{\mathbf{b}}) = J_D(\mathbf{b})$. Since $J_R(\tilde{\mathbf{b}}) \leq \sup_{\mathbf{b}' \in \tilde{\mathcal{B}}} J_R(\mathbf{b}')$ by the definition of the supremum, it then follows that for any $\alpha > 0$,

$$\lim_{\alpha \to +\infty} J_R(\tilde{\mathbf{b}}) \leq \sup_{\mathbf{b}' \in \tilde{\mathcal{B}}} J_R(\mathbf{b}'),$$

which implies that

$$J_D(\mathbf{b}) \leq \sup_{\mathbf{b}' \in \tilde{\mathcal{B}}} J_R(\mathbf{b}').$$

Since $\mathbf{b}$ was arbitrary, we thus have that

$$\sup_{\mathbf{b} \in \mathcal{B}} J_D(\mathbf{b}) \leq \sup_{\mathbf{b}' \in \tilde{\mathcal{B}}} J_R(\mathbf{b}')$$

as required.

Step 2: $\sup_{\mathbf{b} \in \mathcal{B}} J_D(\mathbf{b}) \geq \sup_{\tilde{\mathbf{b}} \in \tilde{\mathcal{B}}} J_R(\tilde{\mathbf{b}})$. To show this, let $\tilde{\mathbf{b}}$ be any set of random policy weights in $\tilde{\mathcal{B}}$. As in the proof of Theorem 1, let us define random variables $\xi_1, \ldots, \xi_T$ that are i.i.d. standard logistic random variables, that is, for each $t \in [T]$, we have:

$$\mathbb{P}(\xi_t < s) = \sigma(s)$$

for all $s \in \mathbb{R}$. Then observe that for a fixed trajectory $\mathbf{x}(1), \ldots, \mathbf{x}(T)$, we can write the reward of the randomized policy with weights $\tilde{\mathbf{b}}$ as

$$\begin{aligned}
&= \sum_{t=1}^{T} g(t, \mathbf{x}(t)) \cdot \prod_{t'=1}^{t-1} (1 - \sigma(\tilde{\mathbf{b}}_{t'} \bullet \Phi(\mathbf{x}(t')))) \cdot \sigma(\tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(t))) \\
&= \sum_{t=1}^{T} g(t, \mathbf{x}(t)) \cdot \prod_{t'=1}^{t-1} \mathbb{P}(\xi_{t'} \geq \tilde{\mathbf{b}}_{t'} \bullet \Phi(\mathbf{x}(t'))) \cdot \mathbb{P}(\xi_t < \tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(t))) \\
&= \sum_{t=1}^{T} g(t, \mathbf{x}(t)) \cdot \prod_{t'=1}^{t-1} \mathbb{E}[\mathbb{I}\{\xi_{t'} \geq \tilde{\mathbf{b}}_{t'} \bullet \Phi(\mathbf{x}(t'))\}] \cdot \mathbb{E}[\mathbb{I}\{\xi_t < \tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(t))\}] \\
&= \sum_{t=1}^{T} g(t, \mathbf{x}(t)) \cdot \mathbb{E}_{\xi_1, \ldots, \xi_T} \left[\prod_{t'=1}^{t-1} \mathbb{I}\{\xi_{t'} \geq \tilde{\mathbf{b}}_{t'} \bullet \Phi(\mathbf{x}(t'))\} \cdot \mathbb{I}\{\xi_t < \tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(t))\} \right] \\
&= \mathbb{E}_{\xi_1, \ldots, \xi_T} \left[\sum_{t=1}^{T} g(t, \mathbf{x}(t)) \cdot \prod_{t'=1}^{t-1} \mathbb{I}\{\xi_{t'} \geq \tilde{\mathbf{b}}_{t'} \bullet \Phi(\mathbf{x}(t'))\} \cdot \mathbb{I}\{\xi_t < \tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(t))\} \right]. \qquad \text{(A.9)}
\end{aligned}$$

We thus have that

$$
\begin{aligned}
&J_R(\tilde{\mathbf{b}}) \\
&= \mathbb{E}_{\mathbf{x}(1),\ldots,\mathbf{x}(T)}\left[\sum_{t=1}^{T} g(t,\mathbf{x}(t)) \cdot \prod_{t'=1}^{t-1}(1-\sigma(\tilde{\mathbf{b}}_{t'} \bullet \Phi(\mathbf{x}(t')))) \cdot \sigma(\tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(t)))\right] \\
&= \mathbb{E}_{\mathbf{x}(1),\ldots,\mathbf{x}(T)}\left[\mathbb{E}_{\xi_1,\ldots,\xi_T}\left[\sum_{t=1}^{T} g(t,\mathbf{x}(t)) \cdot \prod_{t'=1}^{t-1} \mathbb{I}\{\xi_{t'} \geq \tilde{\mathbf{b}}_{t'} \bullet \Phi(\mathbf{x}(t'))\} \cdot \mathbb{I}\{\xi_t < \tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(t))\}\right]\right] \\
&= \mathbb{E}_{\xi_1,\ldots,\xi_T}\left[\mathbb{E}_{\mathbf{x}(1),\ldots,\mathbf{x}(T)}\left[\sum_{t=1}^{T} g(t,\mathbf{x}(t)) \cdot \prod_{t'=1}^{t-1} \mathbb{I}\{\xi_{t'} \geq \tilde{\mathbf{b}}_{t'} \bullet \Phi(\mathbf{x}(t'))\} \cdot \mathbb{I}\{\xi_t < \tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(t))\}\right]\right]
\end{aligned}
$$

where the interchange of expectations in the last step follows by Fubini's theorem, since the random variable (A.9) is always nonnegative.

By the definition of expected value, there must exist a realization $\xi'_1, \ldots, \xi'_T$ such that

$$
\begin{aligned}
&\mathbb{E}_{\xi_1,\ldots,\xi_T}\left[\mathbb{E}_{\mathbf{x}(1),\ldots,\mathbf{x}(T)}\left[\sum_{t=1}^{T} g(t,\mathbf{x}(t)) \cdot \prod_{t'=1}^{t-1} \mathbb{I}\{\xi_{t'} \geq \tilde{\mathbf{b}}_{t'} \bullet \Phi(\mathbf{x}(t'))\} \cdot \mathbb{I}\{\xi_t < \tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(t))\}\right]\right] \\
&\leq \mathbb{E}_{\mathbf{x}(1),\ldots,\mathbf{x}(T)}\left[\sum_{t=1}^{T} g(t,\mathbf{x}(t)) \cdot \prod_{t'=1}^{t-1} \mathbb{I}\{\xi'_{t'} \geq \tilde{\mathbf{b}}_{t'} \bullet \Phi(\mathbf{x}(t'))\} \cdot \mathbb{I}\{\xi'_t < \tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(t))\}\right].
\end{aligned}
$$

Now, let us define a weight vector $\mathbf{b}$ for the deterministic probelm as follows:

$$
b_{t,k} = \begin{cases} \tilde{b}_{t,k} & \text{if } k \neq 1, \\ \tilde{b}_{t,1} - \xi'_t & \text{if } k = 1, \end{cases} \tag{A.10}
$$

where we recall that the index $k = 1$ corresponds to the constant basis function $\phi_1(\cdot) = 1$. Observe that by the manner in which we have defined $\mathbf{b}$, we have that

$$
\begin{aligned}
&\mathbb{I}\{\xi_t \geq \tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(t))\} \\
&= \mathbb{I}\{0 \geq \tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(t)) - \xi_t\} \\
&= \mathbb{I}\{0 \geq \mathbf{b}_t \bullet \Phi(\mathbf{x}(t))\}.
\end{aligned}
$$

Thus, we have that

$$\begin{aligned} J_R(\tilde{\mathbf{b}}) &\leq \mathbb{E}_{\mathbf{x}(1),\ldots,\mathbf{x}(T)}\left[\sum_{t=1}^{T} g(t,\mathbf{x}(t)) \cdot \prod_{t'=1}^{t-1} \mathbb{I}\{\xi'_{t'} \geq \tilde{\mathbf{b}}_{t'} \bullet \Phi(\mathbf{x}(t'))\} \cdot \mathbb{I}\{\xi'_t < \tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(t))\}\right] \\ &= \mathbb{E}_{\mathbf{x}(1),\ldots,\mathbf{x}(T)}\left[\sum_{t=1}^{T} g(t,\mathbf{x}(t)) \cdot \prod_{t'=1}^{t-1} \mathbb{I}\{\mathbf{b}_{t'} \bullet \Phi(\mathbf{x}(t')) \leq 0\} \cdot \mathbb{I}\{\mathbf{b}_t \bullet \Phi(\mathbf{x}(t)) > 0\}\right] \\ &= J_D(\mathbf{b}) \\ &\leq \sup_{\mathbf{b}' \in \mathcal{B}} J_D(\mathbf{b}'). \end{aligned}$$

Since $\tilde{\mathbf{b}}$ was arbitrary, this implies that $\sup_{\mathbf{b}' \in \mathcal{B}} J_D(\mathbf{b}')$ is an upper bound on $J_R(\tilde{\mathbf{b}})$ for all $\tilde{\mathbf{b}} \in \tilde{\mathcal{B}}$, and thus that

$$\sup_{\tilde{\mathbf{b}} \in \tilde{\mathcal{B}}} J_R(\tilde{\mathbf{b}}) \leq \sup_{\mathbf{b} \in \mathcal{B}} J_D(\mathbf{b}), \tag{A.11}$$

as required. □

A.1.3 Proof of Theorem 3

To establish this result, we will show that the functions $\hat{J}_R(\cdot)$ and $J_R(\cdot)$ are Lipschitz continuous, and use this together with the compactness of $\mathcal{B}$ to establish uniform convergence of $\hat{J}_R(\cdot)$ to $J_R(\cdot)$. To establish that these two functions are Lipschitz continuous, we need three preliminary results. The first is a basic result that the product of bounded Lipschitz continuous functions is a Lipschitz continuous function. Note that for this result and all other results in this section of the Appendix, Lipschitz continuity is understood with respect to the L_1 norm, i.e., $f(\mathbf{b})$ is said to be Lipschitz continuous if there exists an $L > 0$ such that $|f(\mathbf{b}) - f(\mathbf{b}')| \leq L\|\mathbf{b} - \mathbf{b}'\|_1$ for all $\mathbf{b}, \mathbf{b}'$.

Lemma 1 *Suppose that $f, h : \mathcal{B} \to \mathbb{R}$ are Lipschitz continuous functions with Lipschitz constants L_f, and L_h, respectively, and are also uniformly bounded by constants K_f and K_h, i.e., $\sup_{\mathbf{b} \in \mathcal{B}} |f(\mathbf{b})| \leq K_f$, $\sup_{\mathbf{b} \in \mathcal{B}} |h(\mathbf{b})| \leq K_h$. Then the function $w : \mathcal{B} \to \mathbb{R}$ defined as $w(\mathbf{b}) = f(\mathbf{b})h(\mathbf{b})$ is also Lipschitz continuous with Lipschitz constant $L_w = K_f L_h + K_h L_f$.*

Proof of Lemma 1: Let $\mathbf{b}, \bar{\mathbf{b}} \in \mathcal{B}$ and consider $|w(\mathbf{b}) - w(\bar{\mathbf{b}})|$:

$$\begin{aligned}
|w(\mathbf{b}) - w(\bar{\mathbf{b}})| &= |f(\mathbf{b})h(\mathbf{b}) - f(\bar{\mathbf{b}})h(\bar{\mathbf{b}})| \\
&= |f(\mathbf{b})h(\mathbf{b}) - f(\mathbf{b})h(\bar{\mathbf{b}}) + f(\mathbf{b})h(\bar{\mathbf{b}}) - f(\bar{\mathbf{b}})h(\bar{\mathbf{b}})| \\
&\leq |f(\mathbf{b})|\cdot|h(\mathbf{b}) - h(\bar{\mathbf{b}})| + |f(\mathbf{b}) - f(\bar{\mathbf{b}})|\cdot|h(\bar{\mathbf{b}})| \\
&\leq K_f \cdot L_h \|\mathbf{b} - \bar{\mathbf{b}}\| + L_f \|\mathbf{b} - \bar{\mathbf{b}}\| \cdot K_h \\
&= (K_f L_h + L_f K_h)\|\mathbf{b} - \bar{\mathbf{b}}\|,
\end{aligned}$$

as required. □

The second result that we will use is that the probabilities of stopping and continuing at time t and at a state $\mathbf{x} \in \mathcal{X}$ in a randomized policy are Lipschitz continuous with respect to $\mathbf{b}$.

Lemma 2 *Suppose that Assumption 5 holds. For any $t \in [T]$ and $\mathbf{x} \in \mathcal{X}$, the functions f and h defined as*

$$\begin{aligned}
f(\mathbf{b}) &= \sigma(\mathbf{b}_t \bullet \Phi(\mathbf{x})), \\
h(\mathbf{b}) &= 1 - \sigma(\mathbf{b}_t \bullet \Phi(\mathbf{x})),
\end{aligned}$$

are Lipschitz continuous with Lipschitz constant Q.

Proof of Lemma 2: Observe that for f, the gradient of f satisfies

$$\begin{aligned}
\nabla_{\mathbf{b}_t} f(\mathbf{b}) &= \Phi(\mathbf{x})\sigma(\mathbf{b}_t \bullet \Phi(\mathbf{x})), \\
\nabla_{\mathbf{b}_{t'}} f(\mathbf{b}) &= 0, \quad \forall t' \neq t.
\end{aligned}$$

Therefore, by Assumption 5,

$$\|\nabla f(\mathbf{b})\|_\infty = \|\nabla_{\mathbf{b}_t} f(\mathbf{b})\|_\infty \leq \|\Phi(\mathbf{x})\|_\infty \leq Q.$$

Now, consider $\mathbf{b}$ and $\bar{\mathbf{b}}$ in $\mathcal{B}$. Since f is a differentiable function, it follows by the mean value theorem that there exists a $\mathbf{b}' \in \mathbb{R}^{KT}$ such that

$$f(\mathbf{b}) - f(\bar{\mathbf{b}}) = \nabla f(\mathbf{b}')^T(\mathbf{b} - \bar{\mathbf{b}}). \tag{A.12}$$

We thus have

$$|f(\mathbf{b}) - f(\bar{\mathbf{b}})| = |\nabla f(\mathbf{b}')^T(\mathbf{b} - \bar{\mathbf{b}})| \tag{A.13}$$

$$\leq \|\nabla f(\mathbf{b}')\|_\infty \|\mathbf{b} - \bar{\mathbf{b}}\|_1 \tag{A.14}$$

$$\leq Q\|\mathbf{b} - \bar{\mathbf{b}}\|_1, \tag{A.15}$$

where the first inequality follows by the Cauchy-Schwartz inequality, and the second by our earlier result that the norm of the gradient of f is bounded everywhere by Q. Thus, f is Lipschitz continuous with constant Q. The proof for h follows by an almost identical argument. □

Lemma 3 *Suppose Assumption 5 holds. Fix any* $(\mathbf{x}(1), \ldots, \mathbf{x}(T)) \in \mathcal{X}^T$*, and any* $t \in [T]$*. The function* $H_t(\cdot)$ *defined as*

$$H_t(\mathbf{b}) = \prod_{t'=1}^{t} (1 - \sigma(b_{t'} \bullet \Phi(\mathbf{x}(t'))))$$

is Lipschitz continuous with constant tQ*.*

Proof of Lemma 3: We will prove this by induction on t. The base case is when $t = 1$. In this case, $H_1(\mathbf{b}) = 1 - \sigma(\mathbf{b}_1 \bullet \Phi(\mathbf{x}(1)))$. By Lemma 2, this function is Lipschitz continuous with constant Q, as required.

To establish the claim for $t \geq 2$, suppose that $H_{t-1}(\cdot)$ is Lipschitz continuous with constant $(t-1)Q$. We now need to establish that $H_t(\cdot)$ is Lipschitz continuous with constant tQ.

To see this, observe that we can write $H_t(\mathbf{b}) = H_{t-1}(\mathbf{b}) \cdot (1 - \sigma(\mathbf{b}_t \bullet \Phi(\mathbf{x}(t))))$. The function $H_{t-1}(\cdot)$ and the function $h(\mathbf{b}) = 1 - \sigma(\mathbf{b}_t \bullet \Phi(\mathbf{x}(t)))$ are both bounded in absolute value by 1. Additionally, by Lemma 2, the function $h(\cdot)$ is Lipschitz continuous with constant Q. Together with the induction hypothesis that $H_{t-1}(\cdot)$ is Lipschitz continuous with constant $(t-1)Q$, we can invoke Lemma 1 to assert that $H_t(\cdot)$ is Lipschitz continuous with constant $(t-1)Q \cdot 1 + Q \cdot 1 = tQ$. □

Lemma 4 *Suppose Assumption 5 holds. The function $\hat{J}_R(\cdot)$ is Lipschitz continuous with Lipschitz constant $L = \bar{G}T^2Q$.*

Proof of Lemma 4: Let $\mathbf{b}, \bar{\mathbf{b}} \in \mathcal{B}$. We have

$$
\begin{aligned}
&|\hat{J}_R(\mathbf{b}) - \hat{J}_R(\bar{\mathbf{b}})| \\
&= |\frac{1}{\Omega}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} g(t, \mathbf{x}(\omega, t)) \prod_{t'=1}^{t-1}(1 - \sigma(\mathbf{b}_{t'} \bullet \Phi(\mathbf{x}(\omega, t'))))\sigma(\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega, t))) \\
&\quad - \frac{1}{\Omega}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} g(t, \mathbf{x}(\omega, t)) \prod_{t'=1}^{t-1}(1 - \sigma(\bar{\mathbf{b}}_{t'} \bullet \Phi(\mathbf{x}(\omega, t'))))\sigma(\bar{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(\omega, t)))| \\
&\leq \frac{1}{\Omega}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} g(t, \mathbf{x}(\omega, t))|\prod_{t'=1}^{t-1}(1 - \sigma(\mathbf{b}_{t'} \bullet \Phi(\mathbf{x}(\omega, t'))))\sigma(\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega, t))) \\
&\quad - \prod_{t'=1}^{t-1}(1 - \sigma(\bar{\mathbf{b}}_{t'} \bullet \Phi(\mathbf{x}(\omega, t'))))\sigma(\bar{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(\omega, t)))| \\
&\leq \frac{1}{\Omega}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} g(t, \mathbf{x}(\omega, t)) tQ\|\mathbf{b} - \bar{\mathbf{b}}\|_1 \\
&\leq \frac{1}{\Omega}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} \bar{G}TQ\|\mathbf{b} - \bar{\mathbf{b}}\|_1 \\
&= \frac{1}{\Omega} \cdot \Omega \cdot T \cdot \bar{G}TQ\|\mathbf{b} - \bar{\mathbf{b}}\|_1 \\
&= \bar{G}T^2Q\|\mathbf{b} - \bar{\mathbf{b}}\|_1
\end{aligned}
$$

where the first inequality is just the triangle inequality; the second inequality follows by applying Lemmas 3, 2 and 1 together; and the remaining steps follow by algebra and using the definition of $\bar{G}$ as a universal upper bound on $g(t, \mathbf{x})$ (Assumption 1). □

Lemma 5 *The function $J_R(\cdot)$ is Lipschitz continuous with Lipschitz constant $L = \bar{G}T^2Q$.*

Proof of Lemma 5: Let $\mathbf{b}, \bar{\mathbf{b}} \in \mathcal{B}$. Using similar logic as the proof of Lemma 4, we have

$$
\begin{aligned}
&|J_R(\mathbf{b}) - J_R(\bar{\mathbf{b}})| \\
&= \left| \mathbb{E}\left[\sum_{t=1}^{T} g(t, \mathbf{x}(t)) \prod_{t'=1}^{t-1} (1 - \sigma(\mathbf{b}_{t'} \bullet \Phi(\mathbf{x}(t')))) \sigma(\mathbf{b}_t \bullet \Phi(\mathbf{x}(t))) \right] \right. \\
&\qquad \left. - \mathbb{E}\left[\sum_{t=1}^{T} g(t, \mathbf{x}(t)) \prod_{t'=1}^{t-1} (1 - \sigma(\bar{\mathbf{b}}_{t'} \bullet \Phi(\mathbf{x}(t')))) \sigma(\bar{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(t))) \right] \right| \\
&\leq \mathbb{E}\left[\sum_{t=1}^{T} g(t, \mathbf{x}(t)) \cdot \left| \prod_{t'=1}^{t-1} (1 - \sigma(\mathbf{b}_{t'} \bullet \Phi(\mathbf{x}(t')))) \sigma(\mathbf{b}_t \bullet \Phi(\mathbf{x}(t))) \right. \right. \\
&\qquad \left. \left. - \prod_{t'=1}^{t-1} (1 - \sigma(\bar{\mathbf{b}}_{t'} \bullet \Phi(\mathbf{x}(t')))) \sigma(\bar{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(t))) \right| \right] \\
&\leq \mathbb{E}\left[\sum_{t=1}^{T} \bar{G} T Q \|\mathbf{b} - \bar{\mathbf{b}}\|_1 \right] \\
&= \bar{G} T^2 Q \|\mathbf{b} - \bar{\mathbf{b}}\|_1,
\end{aligned}
$$

as required. □

With Lemma 4 and 5, we can prove the following theorem, which will be the final stepping stone to Theorem 3.

Theorem 10 *Suppose that Assumptions 5 and 6 both hold. Fix any $\epsilon > 0$. With probability one, there exists a finite sample size N such that for all $\Omega \geq N$,*

$$
\sup_{\mathbf{b} \in \mathcal{B}} |J_R(\mathbf{b}) - \hat{J}_R(\mathbf{b})| \leq \epsilon. \tag{A.16}
$$

Proof of Theorem 10: For the given ϵ, set $\delta = \epsilon/(3L)$ where $L = \bar{G} T^2 Q$ is the Lipschitz constant of both $\hat{J}_R(\cdot)$ and $J_R(\cdot)$. Since $\mathcal{B}$ is compact (Assumption 6), there exist finitely many points $\mathbf{b}^1, \ldots, \mathbf{b}^M$ such that $\mathcal{B} \subseteq \bigcup_{m=1}^{M} B(\mathbf{b}^m, \delta)$, where $B(\mathbf{b}, r) = \{\mathbf{b}' \in \mathcal{B} \mid \|\mathbf{b}' - \mathbf{b}\|_1 < r\}$ is the open ball of radius r in the L_1 norm.

For each point $\mathbf{b}^m$, the strong law of large numbers guarantees that $\hat{J}_R(\mathbf{b}^m)$ converges to $J_R(\mathbf{b}^m)$ almost surely. Thus, almost surely, there exists an integer N_m such that for all $\Omega > N_m$, $|\hat{J}_R(\mathbf{b}^m) - J_R(\mathbf{b}^m)| < \epsilon/3$. Let $N = \max\{N_1, \ldots, N_M\}$. Then, almost surely, for all $\Omega > N$, it holds that $|\hat{J}_R(\mathbf{b}^m) - J_R(\mathbf{b}^m)| < \epsilon/3$ for all $m \in [M]$.

Now, consider any $\mathbf{b} \in \mathcal{B}$. By the definition of $\{\mathbf{b}^1, \ldots, \mathbf{b}^M\}$ as a δ-net of $\mathcal{B}$, there exists an m such that $\mathbf{b} \in B(\mathbf{b}^m, \delta)$. For all $\Omega > N$, we therefore have

$$\begin{aligned}
|\hat{J}_R(\mathbf{b}) - J_R(\mathbf{b})| &= |\hat{J}_R(\mathbf{b}) - \hat{J}_R(\mathbf{b}^m) + \hat{J}_R(\mathbf{b}^m) - J_R(\mathbf{b}^m) + J_R(\mathbf{b}^m) - J_R(\mathbf{b})| \\
&\leq |\hat{J}_R(\mathbf{b}) - \hat{J}_R(\mathbf{b}^m)| + |\hat{J}_R(\mathbf{b}^m) - J_R(\mathbf{b}^m)| + |J_R(\mathbf{b}^m) - J_R(\mathbf{b})| \\
&\leq L\|\mathbf{b} - \mathbf{b}^m\|_1 + \frac{\epsilon}{3} + L\|\mathbf{b} - \mathbf{b}^m\|_1 \\
&\leq L \cdot \frac{\epsilon}{3L} + \frac{\epsilon}{3} + L \cdot \frac{\epsilon}{3L} \\
&= \frac{\epsilon}{3} + \frac{\epsilon}{3} + \frac{\epsilon}{3} \\
&= \epsilon
\end{aligned}$$

where the second step follows by the triangle inequality; the third step follows by using the Lipschitz continuity of $\hat{J}_R(\cdot)$ and $J_R(\cdot)$ from Lemmas 4 and 5 respectively, as well as the almost sure convergence of $\hat{J}_R(\cdot)$ to $J_R(\cdot)$ at $\mathbf{b}^m$; the fourth step by our definition of $\mathbf{b}^m$ as the point in the δ-net containing $\mathbf{b}$; and the remaining steps by algebra.

Since $\mathbf{b}$ was arbitrary, it follows that almost surely, for all $\Omega > N$ and all $\mathbf{b} \in \mathcal{B}$, that $|\hat{J}_R(\mathbf{b}) - J_R(\mathbf{b})| < \epsilon$. This completes the proof. □

Using Theorem 10, we now finally prove Theorem 3.

Proof of Theorem 3: To show that $\sup_{\mathbf{b}\in\mathcal{B}} |\hat{J}_R(\mathbf{b}) - J_R(\mathbf{b})| \to 0$ as $\Omega \to \infty$ almost surely, we observe that this event can be written as

$$\bigcap_{\epsilon>0} \bigcup_{N=1}^{\infty} \bigcap_{\Omega>N} \left\{ \sup_{\mathbf{b}\in\mathcal{B}} |\hat{J}_R(\mathbf{b}) - J_R(\mathbf{b})| < \epsilon \right\},$$

which is equivalent to

$$\bigcap_{k=1}^{\infty}\bigcup_{N=1}^{\infty}\bigcap_{\Omega>N}\left\{\sup_{\mathbf{b}\in\mathcal{B}}|\hat{J}_R(\mathbf{b})-J_R(\mathbf{b})|<\frac{1}{2^k}\right\}. \tag{A.17}$$

The event in (A.17) is the countable intersection of events of the form $\bigcup_{N=1}^{\infty}\bigcap_{\Omega>N}\{|\hat{J}_R(\mathbf{b})-J_R(\mathbf{b})|<1/2^k\}$, each of which occurs with probability one by Theorem 10. Therefore, event (A.17) occurs with probability 1, which establishes the required result. □

A.1.4 Proof of Corollary 1

We will first show that if $\hat{J}_R(\cdot)$ converges uniformly to $J_R(\cdot)$ on $\mathcal{B}$, then it must be the case that $\sup_{\mathbf{b}\in\mathcal{B}}\hat{J}_R(\mathbf{b})$ converges to $\sup_{\mathbf{b}\in\mathcal{B}}J_R(\mathbf{b})$.

Let $\epsilon>0$. Then there exists an integer N such that for all $\Omega>N$, $\sup_{\mathbf{b}\in\mathcal{B}}|\hat{J}_R(\mathbf{b})-J_R(\mathbf{b})|<\epsilon/2$.

Let $\Omega>N$. Suppose without loss of generality that $\sup_{\mathbf{b}\in\mathcal{B}}\hat{J}_R(\mathbf{b})\leq\sup_{\mathbf{b}\in\mathcal{B}}J_R(\mathbf{b})$. Let $\tilde{\mathbf{b}}\in\mathcal{B}$ be a weight vector such that

$$J_R(\tilde{\mathbf{b}})\geq\sup_{\mathbf{b}\in\mathcal{B}}J_R(\mathbf{b})-\frac{\epsilon}{2},$$

or equivalently,

$$J_R(\tilde{\mathbf{b}})+\frac{\epsilon}{2}\geq\sup_{\mathbf{b}\in\mathcal{B}}J_R(\mathbf{b}).$$

Then we have

$$\begin{aligned}\left|\sup_{\mathbf{b}\in\mathcal{B}}J_R(\mathbf{b})-\sup_{\mathbf{b}\in\mathcal{B}}\hat{J}_R(\mathbf{b})\right| &= \sup_{\mathbf{b}\in\mathcal{B}}J_R(\mathbf{b})-\sup_{\mathbf{b}\in\mathcal{B}}\hat{J}_R(\mathbf{b})\\ &\leq J_R(\tilde{\mathbf{b}})+\frac{\epsilon}{2}-\hat{J}_R(\tilde{\mathbf{b}})\\ &\leq \sup_{\mathbf{b}\in\mathcal{B}}|\hat{J}_R(\mathbf{b})-J_R(\mathbf{b})|+\frac{\epsilon}{2}\\ &\leq \frac{\epsilon}{2}+\frac{\epsilon}{2}\\ &= \epsilon.\end{aligned}$$

(In the case that $\sup_{\mathbf{b}\in\mathcal{B}} \hat{J}_R(\mathbf{b}) \geq \sup_{\mathbf{b}\in\mathcal{B}} J_R(\mathbf{b})$, the same steps go through, with the modification that $\tilde{\mathbf{b}}$ is chosen to be within $\epsilon/2$ of $\sup \hat{J}_R(\mathbf{b})$, i.e., $\tilde{\mathbf{b}}$ satisfies $\hat{J}_R(\tilde{\mathbf{b}}) \geq \sup_{\mathbf{b}\in\mathcal{B}} \hat{J}_R(\mathbf{b}) - \epsilon/2$.)

Thus, we have shown that whenever $\sup_{\mathbf{b}\in\mathcal{B}}|\hat{J}_R(\mathbf{b}) - J_R(\mathbf{b})| \to 0$ as $\Omega \to \infty$, we also must have that $\sup_{\mathbf{b}\in\mathcal{B}} \hat{J}_R(\mathbf{b}) \to \sup_{\mathbf{b}\in\mathcal{B}} J_R(\mathbf{b})$ as $\Omega \to \infty$. Since the former occurs with probability one by Theorem 3, then it must be the case that $\lim_{\Omega\to\infty} \sup_{\mathbf{b}\in\mathcal{B}} \hat{J}_R(\mathbf{b}) = \sup_{\mathbf{b}\in\mathcal{B}} J_R(\mathbf{b})$ also occurs with probability one. $\square$

A.1.5 Proof of Theorem 4

By Theorem 5.3 from Shapiro et al. (2014), since (i) the set $\mathbf{B}^*$ of the optimal solutions of $\sup_{\mathbf{b}\in\mathcal{B}} J_R(\mathbf{b})$ is nonempty and $\mathbf{B}^* \subseteq \mathcal{B}$; (ii) $J_R(\cdot)$ is continuous on $\mathcal{B}$ as $J_R(\mathbf{b})$ is a Lipschitz continuous function of $\mathbf{b} \in \mathcal{B}$, and $J_R(\mathbf{b})$ is finite valued as we assume the reward $g(t, \mathbf{x})$ has a finite upper bound; (iii) $\hat{J}_R(\cdot)$ converges uniformly to $J_R(\cdot)$ with probability one by Theorem 3; and (iv) with probability one, for Ω large enough, the set $\hat{\mathbf{B}}_\Omega$ is nonempty and $\hat{\mathbf{B}} \subseteq \mathcal{B}$; then with probability one, $\mathbb{D}(\hat{\mathbf{B}}, \mathbf{B}^*) \to 0$ as $\Omega \to \infty$.$\square$

A.1.6 Proof of Proposition 1

Our proof of Proposition 1 follows the proof of Rademacher complexity-based generalization bounds in statistical learning (see for example Theorem 3.1 in Mohri et al. 2018). For completeness, we provide the proof here.

Given an i.i.d. sample of system realizations $S = (Y_1, \ldots, Y_\Omega)$, let $D(S)$ be the random variable defined as

$$D(S) = \sup_{f\in\mathcal{F}} \left(\frac{1}{\Omega} \sum_{\omega=1}^{\Omega} f(Y_\omega) - \mathbb{E}[f(Y)] \right),$$

where Y is a random variable that represents a single system realization. Our goal will be to obtain a high probability bound on $D(S)$. We will proceed in three steps: first, we will bound the deviation of $D(S)$ from its mean $\mathbb{E}[D(S)]$; second, we will bound $\mathbb{E}[D(S)]$;

and finally, we will put these two inequalities together, and show how they imply our main inequalities in terms of $J_R(\cdot)$ and $\hat{J}_R(\cdot)$.

Step 1. Let $S_i' = (Y_1, \ldots, Y_i', \ldots, Y_\Omega)$ be a sample of system realizations that differs from S only in the ith trajectory. It is straightforward to show that

$$D(S) - D(S_i') \leq \frac{\bar{G}}{\Omega},$$

and that by symmetry, $D(S_i') - D(S) \leq \bar{G}/\Omega$ as well. Together, these two inequalities imply that $D(S)$ satisfies the bounded differences property: for any $i \in \{1, \ldots, \Omega\}$, any S and any Y_i', we have $|D(S_i') - D(S)| \leq \bar{G}/\Omega$.

Thus, McDiarmid's inequality implies that with probability at least $1-\delta$ over the sample of system realizations S, the following inequality holds:

$$D(S) - \mathbb{E}[D(S)] \leq \bar{G}\sqrt{\frac{\log(1/\delta)}{2\Omega}}.$$

Step 2. We now bound $\mathbb{E}[D(S)]$. Let $S' = (Y_1', \ldots, Y_\Omega')$ be a second i.i.d. sample of Ω system realizations. We then have

$$\begin{aligned}
\mathbb{E}[D(S)] &= \mathbb{E}_S\left[\sup_{f\in\mathcal{F}}\left(\frac{1}{\Omega}\sum_{\omega=1}^{\Omega} f(Y_\omega) - \mathbb{E}[f(Y)]\right)\right] \\
&= \mathbb{E}_S\left[\sup_{f\in\mathcal{F}}\left(\frac{1}{\Omega}\sum_{\omega=1}^{\Omega} f(Y_\omega) - \mathbb{E}_{S'}[\frac{1}{\Omega}\sum_{\omega=1}^{\Omega} f(Y_\omega)]\right)\right] \\
&\leq \mathbb{E}_{S,S'}\left[\sup_{f\in\mathcal{F}}\left(\frac{1}{\Omega}\sum_{\omega=1}^{\Omega} f(Y_\omega) - \frac{1}{\Omega}\sum_{\omega=1}^{\Omega} f(Y_\omega')\right)\right] \\
&= \mathbb{E}_{S,S',\boldsymbol{\epsilon}}\left[\sup_{f\in\mathcal{F}}\frac{1}{\Omega}\sum_{\omega=1}^{\Omega} \epsilon_\omega(f(Y_\omega) - f(Y_\omega'))\right] \\
&\leq \mathbb{E}_{S,S',\boldsymbol{\epsilon}}\left[\sup_{f\in\mathcal{F}}\frac{1}{\Omega}\sum_{\omega=1}^{\Omega} \epsilon_\omega f(Y_\omega)\right] + \mathbb{E}_{S,S',\boldsymbol{\epsilon}}\left[\sup_{f\in\mathcal{F}}\frac{1}{\Omega}\sum_{\omega=1}^{\Omega} \epsilon_\omega f(Y_\omega')\right] \\
&= 2R(\mathcal{F}),
\end{aligned}$$

where $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_\Omega)$ denotes an i.i.d. set of Rademacher random variables, that is, each ϵ_ω satisfies $\mathbb{P}(\epsilon_\omega = +1) = 1/2$, $\mathbb{P}(\epsilon_\omega = -1) = 1/2$.

Step 3. Using results from Step 1 and 2, we have that $D(S) \leq 2R(\mathcal{F}) + \bar{G}\sqrt{\log(1/\delta)/(2\Omega)}$. By the definition of D, this implies that

$$\frac{1}{\Omega}\sum_{\omega=1}^{\Omega} f(Y_\omega) - \mathbb{E}[f(Y)] \leq 2R(\mathcal{F}) + \bar{G}\sqrt{\frac{\log(1/\delta)}{2\Omega}}, \quad \forall f \in \mathcal{F},$$

or equivalently,

$$\mathbb{E}[f(Y)] \geq \frac{1}{\Omega}\sum_{\omega=1}^{\Omega} f(Y_\omega) - 2R(\mathcal{F}) - \bar{G}\sqrt{\frac{\log(1/\delta)}{2\Omega}}, \quad \forall f \in \mathcal{F}. \tag{A.18}$$

Note that by the definition of $\mathcal{F}$, $f = \Gamma \circ \psi_{\mathbf{b}}$ for some $\mathbf{b} \in \mathcal{B}$, and thus

$$\begin{aligned}\mathbb{E}[f(Y)] &= \mathbb{E}[(\Gamma \circ \psi_{\mathbf{b}})(Y)] \\ &= J_R(\mathbf{b}),\end{aligned}$$

and

$$\begin{aligned}\frac{1}{\Omega}\sum_{\omega=1}^{\Omega} f(Y_\omega) &= \frac{1}{\Omega}\sum_{\omega=1}^{\Omega}(\Gamma \circ \psi_{\mathbf{b}})(Y_\omega) \\ &= \hat{J}_R(\mathbf{b}).\end{aligned}$$

Thus, (A.18) is equivalent to

$$J_R(\mathbf{b}) \geq \hat{J}_R(\mathbf{b}) - 2R(\mathcal{F}) - \bar{G}\sqrt{\frac{\log(1/\delta)}{2\Omega}}, \quad \forall \mathbf{b} \in \mathcal{B},$$

which is exactly inequality (2.16).

To establish inequality (2.17), let $\hat{R}_S(\mathcal{F})$ be the empirical Rademacher complexity with respect to a sample of system realizations S. It is straightforward to verify that $\hat{R}_S(\mathcal{F})$ satisfies the bounded differences property with the bound $\bar{G}/\Omega$: for any sample S_i' that differs

from S in only the ith trajectory, $|\hat{R}_S(\mathcal{F}) - \hat{R}_{S_i'}(\mathcal{F})| \leq \bar{G}/\Omega$. By then applying McDiarmid's inequality, we can bound the deviation of $\hat{R}_S(\mathcal{F})$ from $R(\mathcal{F})$: we have

$$R(\mathcal{F}) - \hat{R}_S(\mathcal{F}) \leq \bar{G}\sqrt{\frac{\log(1/\delta)}{2\Omega}}, \tag{A.19}$$

with probability at least $1 - \delta$ over the sample of trajectories S.

By now plugging in $\delta/2$ instead of δ in both inequality (A.18) and inequality (A.19) and combining them with the union bound, we obtain that with probability at least $1 - \delta$,

$$\mathbb{E}[f(Y)] \geq \frac{1}{\Omega}\sum_{\omega=1}^{\Omega} f(Y_\omega) - 2\hat{R}_S(\mathcal{F}) - 3\bar{G}\sqrt{\frac{\log(2/\delta)}{2\Omega}}, \quad \forall f \in \mathcal{F}. \tag{A.20}$$

This is equivalent to

$$J_R(\mathbf{b}) \geq \hat{J}_R(\mathbf{b}) - 2\hat{R}_S(\mathcal{F}) - 3\bar{G}\sqrt{\frac{\log(2/\delta)}{2\Omega}}, \quad \forall \mathbf{b} \in \mathcal{B}, \tag{A.21}$$

which is exactly inequality (2.17). □

A.1.7 Proof of Theorem 5

To prove Theorem 5, we need to first establish a number of auxiliary results. Our first result is that the function Γ, which maps the vector produced by $\psi_{\mathbf{b}}$ to an expected reward, is Lipschitz continuous with a particular constant. Note that for this result, Lipschitz continuity is understood with respect to the L_2 norm, as this will be needed later for the application of Maurer's contraction inequality.

Lemma 6 *The function* $\Gamma : \mathbb{R}^T \times [0, \bar{G}]^T \to \mathbb{R}$ *is Lipschitz continuous with Lipschitz constant* $\bar{G} + 1$.

Proof of Lemma 6: To prove this, we will show that the L_2 norm of the gradient of Γ can

be bounded by $\bar{G}+1$. To begin, let us consider the partial derivatives of Γ:

$$\frac{\partial}{\partial v_t}\Gamma = \prod_{t'=1}^{t-1}(1-\sigma(u_t))\sigma(u_t), \tag{A.22}$$

$$\begin{aligned}\frac{\partial}{\partial u_t}\Gamma = v_t\sigma(u_t)(1-\sigma(u_t))\prod_{t'=1}^{t-1}(1-\sigma(u_{t'})) \\ - \sum_{t'=t+1} v_{t'}\sigma(u_t)(1-\sigma(u_t))\prod_{t''=1}^{t-1}(1-\sigma(u_{t''}))\prod_{t''=t+1}^{t'-1}(1-\sigma(u_{t''}))\sigma(u_{t'})\end{aligned} \tag{A.23}$$

Observe that we can further re-arrange the partial derivative with respect to u_t as

$$\frac{\partial}{\partial u_t}\Gamma = \left[\prod_{t'=1}^{t-1}(1-\sigma(u_{t'}))\sigma(u_t)\right]\cdot\left[v_t - \sum_{t'=t}^{T} v_{t'}\prod_{t''=t}^{t'-1}(1-\sigma(u_{t''}))\sigma(u_{t'})\right].$$

For a fixed t, let us define A_t as

$$A_t = v_t - \sum_{t'=t}^{T} v_{t'}\prod_{t''=t}^{t'-1}(1-\sigma(u_{t''}))\sigma(u_{t'}), \tag{A.24}$$

and let us define $\tilde{p}_{t'}$ for each $t' \in \{t,\ldots,T\}$ as

$$\tilde{p}_{t'} = \prod_{t''=t}^{t'-1}(1-\sigma(u_{t''}))\sigma(u_{t'}). \tag{A.25}$$

We can thus re-write A_t as $A_t = v_t - \sum_{t'=t}^{T} v_{t'}\tilde{p}_{t'}$, which allows us to bound it from above as follows:

$$\begin{aligned}A_t &= v_t - \sum_{t'=t}^{T} v_{t'}\tilde{p}_{t'} \\ &\leq \bar{G} - \sum_{t'=t}^{T} 0\tilde{p}_{t'} \\ &= \bar{G},\end{aligned}$$

where we also use the fact that each v_t is bounded between 0 and $\bar{G}$.

We can also bound A_t from below as follows:

$$\begin{aligned} A_t &= v_t - \sum_{t'=t}^{T} v_{t'}\tilde{p}_{t'} \\ &\geq 0 - \sum_{t'=t}^{T} \bar{G}\tilde{p}_{t'} \\ &\geq -\bar{G}, \end{aligned}$$

where the first inequality follows because each v_t is bounded between 0 and $\bar{g}$, and the second inequality follows because each $\tilde{p}_{t'} \geq 0$ and $\sum_{t'=t}^{T} \tilde{p}_{t'} \leq 1$. (Each $\tilde{p}_{t'}$ can be thought of as the probability of stopping at t' according to the logits given in $\mathbf{u}$, conditional on starting from period t.) Thus, we have that $|A_t| \leq \bar{G}$.

Having defined and bounded A_t, let us additionally define p_t as

$$p_t = \prod_{t'=1}^{t-1} (1 - \sigma(u_{t'}))\sigma(u_t). \tag{A.26}$$

Similarly to the $\tilde{p}_{t'}$ values, it is straightforward to establish that $\sum_{t=1}^{T} p_t \leq 1$. With p_t now defined, we can write the partial derivatives of Γ more compactly as

$$\frac{\partial}{\partial v_t}\Gamma = p_t, \tag{A.27}$$

$$\frac{\partial}{\partial u_t}\Gamma = p_t A_t. \tag{A.28}$$

We can now proceed to bound the gradient of Γ. We have

$$\begin{aligned}
\|\nabla\Gamma\|_2 &= \left\| \begin{bmatrix} \nabla_{\mathbf{u}}\Gamma \\ \nabla_{\mathbf{v}}\Gamma \end{bmatrix} \right\|_2 \\
&\leq \|\nabla_{\mathbf{u}}\Gamma\|_2 + \|\nabla_{\mathbf{v}}\Gamma\|_2 \\
&= \left\| \begin{bmatrix} p_1 A_1 \\ \vdots \\ p_T A_T \end{bmatrix} \right\|_2 + \left\| \begin{bmatrix} p_1 \\ \dots \\ p_T \end{bmatrix} \right\|_2 \\
&= \sqrt{p_1^2 A_1^2 + \cdots + p_T^2 A_T^2} + \sqrt{p_1^2 + \dots p_T^2} \\
&\leq \sqrt{p_1 A_1^2 + \dots p_T A_T^2} + \sqrt{p_1 + \cdots + p_T} \\
&\leq \sqrt{p_1 \bar{G}^2 + \dots p_T \bar{G}^2} + \sqrt{p_1 + \cdots + p_T} \\
&\leq \sqrt{\bar{G}^2} + \sqrt{1} \\
&= \bar{G} + 1,
\end{aligned}$$

where the first inequality follows by the fact that $p_t^2 \leq p_t$ (since each $p_t \leq 1$); the second inequality follows by the fact that $|A_t| \leq \bar{G}$ for each t; and the last inequality follows by the fact that $\sum_{t=1}^{T} p_t \leq 1$.

Having established that $\|\nabla\Gamma\|_2 \leq \bar{G} + 1$, the fact that Γ is Lipschitz with constant $\bar{G} + 1$ follows by applying the mean value theorem and the Cauchy-Schwartz inequality. □

Armed with this result that Γ is Lipschitz, we can now relate the Rademacher complexity of $\mathcal{F}$ (the class of functions which map system realizations to rewards) to the Rademacher complexity of the weight vector set $\mathcal{B}$. We do so by using Maurer's vector contraction inequality (Maurer 2016), which is a result for analyzing the Rademacher complexity of a function class that arises from composing a vector-valued function with a Lipschitz function.

Lemma 7 *The empirical Rademacher complexity of $\mathcal{F}$ can be bounded as $\hat{R}(\mathcal{F}) \leq \sqrt{2}(\bar{G} + 1)\hat{R}(\mathcal{B})$, where the empirical Rademacher complexity $\hat{R}(\mathcal{B})$ of the set of feasible weight vectors*

is defined as

$$\hat{R}(\mathcal{B}) = \frac{1}{\Omega}\mathbb{E}\left[\sup_{\mathbf{b}\in\mathcal{B}}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T}\epsilon_{\omega,t}\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega,t))\right]. \tag{A.29}$$

Proof of Lemma 7: To establish this, we will use a specific form of the vector contraction inequality from Maurer (2016), which we re-state here:

Lemma 8 (Corollary 4 of Maurer (2016)) *Let $\mathcal{X}$ be any set, $(x_1,\ldots,x_n)\in\mathcal{X}^n$, let F be a class of functions $f:\mathcal{X}\to\ell_2$ and let $h_i:\ell_2\to\mathbb{R}$ have Lipschitz constant L. Then*

$$\mathbb{E}[\sup_{f\in F}\sum_{i=1}^{n}\epsilon_i h_i(f(x_i))] \leq \sqrt{2}L\mathbb{E}[\sup_{f\in F}\sum_{i,k}\epsilon_{i,k}f_k(x_i)], \tag{A.30}$$

where ℓ_2 is the set of square summable sequences of real numbers, $\{\epsilon_i\}$ is a collection of independent Rademacher variables, $\{\epsilon_{i,k}\}$ is a collection of independent (doubly indexed) Rademacher variables, and $f_k(x_i)$ is the kth component of $f(x_i)$.

With this result in mind, we bound the empirical Rademacher complexity as follows:

$$\begin{aligned}
\hat{R}(\mathcal{F}) &= \frac{1}{\Omega}\mathbb{E}\left[\sup_{f\in\mathcal{F}}\sum_{\omega=1}^{\Omega}\epsilon_\omega f(Y_\omega)\right] \\
&= \frac{1}{\Omega}\mathbb{E}\left[\sup_{\mathbf{b}\in\mathcal{B}}\sum_{\omega=1}^{\Omega}\epsilon_\omega(\Gamma\circ\psi_{\mathbf{b}})(Y_\omega)\right] \\
&\leq \frac{1}{\Omega}\sqrt{2}(\bar{G}+1)\mathbb{E}\left[\sup_{\mathbf{b}\in\mathcal{B}}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{2T}\epsilon_{\omega,t}\psi_{\mathbf{b},t}(Y_\omega)\right] \\
&\leq \frac{1}{\Omega}\sqrt{2}(\bar{G}+1)\mathbb{E}\left[\sup_{\mathbf{b}\in\mathcal{B}}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T}\epsilon_{\omega,t}\psi_{\mathbf{b},t}(Y_\omega)\right] \\
&\quad + \frac{1}{\Omega}\sqrt{2}(\bar{G}+1)\mathbb{E}\left[\sup_{\mathbf{b}\in\mathcal{B}}\sum_{\omega=1}^{\Omega}\sum_{t=T+1}^{2T}\epsilon_{\omega,t}\psi_{\mathbf{b},t}(Y_\omega)\right] \\
&= \frac{1}{\Omega}\sqrt{2}(\bar{G}+1)\mathbb{E}\left[\sup_{\mathbf{b}\in\mathcal{B}}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T}\epsilon_{\omega,T+t}\mathbf{b}_t\bullet\Phi(\mathbf{x}(\omega,t))\right] \\
&\quad + \frac{1}{\Omega}\sqrt{2}(\bar{G}+1)\mathbb{E}\left[\sup_{\mathbf{b}\in\mathcal{B}}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T}\epsilon_{\omega,t}g(t,\mathbf{x}(\omega,t))\right] \\
&= \frac{1}{\Omega}\sqrt{2}(\bar{G}+1)\mathbb{E}\left[\sup_{\mathbf{b}\in\mathcal{B}}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T}\epsilon_{\omega,t}\mathbf{b}_t\bullet\Phi(\mathbf{x}(\omega,t))\right] \\
&= \sqrt{2}(\bar{G}+1)\hat{R}(\mathcal{B}),
\end{aligned}$$

where the first inequality follows by Lemma 6 and Maurer's vector contraction inequality (note that $\psi_{\mathbf{b},t}(Y)$ is used to denote the tth coordinate of $\psi_{\mathbf{b}}(Y)$); the second inequality follows by basic properties of suprema and by linearity of expectation; the third equality follows by the definition of $\psi_{\mathbf{b}}(\cdot)$; and the fourth equality follows because the last T coordinates of $\psi_{\mathbf{b}}(\cdot)$ do not depend on $\mathbf{b}$, and thus the expectation of the weighted sum of the Rademacher random variables works out to zero. □

We are now in a position to prove Theorem 5.

Proof of Theorem 5: To establish each of the three statements, we first bound $\hat{R}(\mathcal{B})$; combining this bound with Lemma 7 then establishes the result. We note that the proofs of

part (a) and part (b) follow standard arguments for obtaining the Rademacher complexity of hypothesis classes defined by norm balls (for example, see the proofs of Theorem 11 and 12 in Liang 2018).

Proof of Part (a): For this result, observe that $\mathcal{B}$ is equal to the L_1 ball of radius B, and is a bounded polyhedron. Therefore, letting $\mathcal{B}^{ext}$ denote the set of extreme points of $\mathcal{B}$, we can write $\mathcal{B}$ as $\mathcal{B} = \text{conv}(\mathcal{B}^{ext})$. By a standard property of Rademacher complexity, we thus have $\hat{R}(\mathcal{B}) = \hat{R}(\mathcal{B}^{\text{ext}})$.

Each extreme point $\mathbf{b} \in \mathcal{B}^{ext}$ is either of the form $\mathbf{b} = +B\mathbf{e}^{t',k'}$ or $\mathbf{b} = -B\mathbf{e}^{t',k'}$, where $\mathbf{e}^{t,k}$ is the standard unit vector with a one at the (t,k) position, and zeros everywhere else. Thus, given $\mathbf{b} = \pm B\mathbf{e}^{t',k'}$, and given $\omega \in [\Omega]$ and $t \in [T]$, we will have

$$\begin{aligned} |\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega,t))| &= |B\mathbf{e}^{t',k'} \bullet \Phi(\mathbf{x}(\omega,t))| \\ &= B|\phi_{k'}(\mathbf{x}(\omega,t))| \end{aligned}$$

if $t = t'$, and $|\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega,t))| = 0$ if $t \neq t'$.

Thus, given $\mathbf{b} \in \mathcal{B}^{ext}$, the vector $\mathbf{w} = [w_{\omega,t}]_{\omega,t}$ where $w_{\omega,t} = \mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega,t))$, has L_2 norm of

$$\begin{aligned} \|\mathbf{w}\|_2 &= \sqrt{\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} w_{\omega,t}^2} \\ &= \sqrt{\sum_{\omega=1}^{\Omega} w_{\omega,t'}^2} \\ &\leq \sqrt{\sum_{\omega=1}^{\Omega} B^2Q^2} \\ &= \sqrt{\Omega}BQ. \end{aligned}$$

We now recall Massart's finite lemma (see Theorem 3.3 in Mohri et al. 2018):

Lemma 9 (Massart's Finite Lemma) *Let $A \subset \mathbb{R}^m$ be a finite set, with $r = \max_{\mathbf{x}\in A}\|\mathbf{x}\|_2$. Then we have*

$$\mathbb{E}[\sup_{\mathbf{x}\in A}\sum_{i=1}^{m} x_i\epsilon_i] \leq r\sqrt{2\log|A|},$$

where $\epsilon_1, \ldots, \epsilon_m$ are i.i.d. Rademacher variables.

Let W consist of vectors $\mathbf{w}$ constructed in the manner described above for each extreme point in $\mathcal{B}^{ext}$. We clearly have that $|W| = |\mathcal{B}^{ext}| = 2KT$. We therefore have

$$\begin{aligned}
\hat{R}(\mathcal{B}) &= \frac{1}{\Omega}\mathbb{E}\left[\sup_{\mathbf{b}\in\mathcal{B}}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T}\epsilon_{\omega,t}\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega,t))\right] \\
&= \frac{1}{\Omega}\mathbb{E}\left[\sup_{\mathbf{b}\in\mathcal{B}^{ext}}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T}\epsilon_{\omega,t}\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega,t))\right] \\
&= \frac{1}{\Omega}\mathbb{E}\left[\sup_{\mathbf{w}\in W}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T}\epsilon_{\omega,t}w_{\omega,t}\right] \\
&\leq \frac{1}{\Omega}\cdot\sqrt{\Omega}BQ\cdot\sqrt{2\log(2KT)} \\
&= \frac{BQ\sqrt{2\log(2KT)}}{\sqrt{\Omega}},
\end{aligned}$$

where the inequality follows by Massart's finite lemma.

Proof of Part (b): For this case, observe that we can write

$$\mathbb{E}\left[\sup_{\mathbf{b}\in\mathcal{B}}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T}\epsilon_{\omega,t}\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega,t))\right] \tag{A.31}$$

$$= \mathbb{E}\left[\sup_{\mathbf{b}\in\mathcal{B}}\sum_{t=1}^{T}\mathbf{b}_t \bullet \left[\sum_{\omega=1}^{\Omega}\epsilon_{\omega,t}\Phi(\mathbf{x}(\omega,t))\right]\right]$$

$$= \mathbb{E}\left[\sup_{\mathbf{b}\in\mathcal{B}}\mathbf{b}\bullet\mathbf{V}\right] \tag{A.32}$$

where $\mathbf{V}$ is defined as

$$\mathbf{V} = \begin{bmatrix} \sum_{\omega=1}^{\Omega} \epsilon_{\omega,1} \Phi(\mathbf{x}(\omega,1)) \\ \vdots \\ \sum_{\omega=1}^{\Omega} \epsilon_{\omega,T} \Phi(\mathbf{x}(\omega,T)) \end{bmatrix}$$
$$= \sum_{\omega=1}^{\Omega} \epsilon_{\omega,1} \begin{bmatrix} \Phi(\mathbf{x}(\omega,1)) \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} + \cdots + \sum_{\omega=1}^{\Omega} \epsilon_{\omega,T} \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \Phi(\mathbf{x}(\omega,T)) \end{bmatrix}.$$

For convenience let us define the vectors $\mathbf{V}_{\omega,1}, \ldots, \mathbf{V}_{\omega,T} \in \mathbb{R}^{KT}$ as

$$\mathbf{V}_{\omega,1} = \begin{bmatrix} \Phi(\mathbf{x}(\omega,1)) \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}, \quad \cdots \quad , \mathbf{V}_{\omega,T} = \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \Phi(\mathbf{x}(\omega,T)) \end{bmatrix},$$

so that $\mathbf{V} = \sum_{\omega=1}^{\Omega} \sum_{t=1}^{T} \epsilon_{\omega,t} \mathbf{V}_{\omega,t}$.

Let us now proceed with bounding (A.32):

$$\begin{aligned} \mathbb{E}\left[\sup_{\mathbf{b}\in\mathcal{B}} \mathbf{b} \bullet \mathbf{V}\right] &= B\mathbb{E}[\|\mathbf{V}\|_2] \\ &\leq B\sqrt{\mathbb{E}[\|\mathbf{V}\|_2^2]} \\ &= B\sqrt{\mathbb{E}\left[\|\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} \epsilon_{\omega,t}\mathbf{V}_{\omega,t}\|_2^2\right]} \\ &= B\sqrt{\mathbb{E}\left[\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} \epsilon_{\omega,t}^2\|\mathbf{V}_{\omega,t}\|_2^2\right]} \\ &= B\sqrt{\mathbb{E}\left[\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} \|\mathbf{V}_{\omega,t}\|_2^2\right]} \\ &= B\sqrt{\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} \|\mathbf{V}_{\omega,t}\|_2^2]} \end{aligned}$$

where the first step follows because the maximizing $\mathbf{b} \in \mathcal{B}$ is equal to $\mathbf{b} = B\mathbf{V}/\|\mathbf{V}\|_2$; the second step follows by the concavity of $f(x) = \sqrt{x}$ and Jensen's inequality; the third step follows by the definition of the $\mathbf{V}_{\omega,t}$'s; the fourth step follows by expanding the square of the norm, and then using the independence of the $\epsilon_{\omega,t}$ to eliminate the cross-terms; and the last step by recognizing that the $\mathbf{V}_{\omega,t}$ vectors are not random.

At this juncture, we observe that the square 2-norm of the $\mathbf{V}_{\omega,t}$'s can be bounded as follows:

$$\begin{aligned}
\|\mathbf{V}_{\omega,t}\|_2^2 &= \left\| \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \Phi(\mathbf{x}(\omega,t)) \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \right\|_2^2 \\
&= \phi_1(\mathbf{x}(\omega,t))^2 + \cdots + \phi_K(\mathbf{x}(\omega,t))^2 \\
&\leq KQ^2.
\end{aligned}$$

Thus, returning to our bound, we have

$$\begin{aligned}
\mathbb{E}[\sup_{\mathbf{b}\in\mathcal{B}} \mathbf{b} \bullet \mathbf{V}] &\leq B\sqrt{\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T}\|\mathbf{V}_{\omega,t}\|_2^2]} \\
&\leq B\sqrt{\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} KQ^2} \\
&= B\sqrt{\Omega T K Q^2} \\
&= BQ\sqrt{\Omega K T}.
\end{aligned}$$

This implies that the empirical Rademacher complexity can be bounded as

$$\begin{aligned}\hat{R}(\mathcal{B}) &= \frac{1}{\Omega}\mathbb{E}[\sup_{\mathbf{b}\in\mathcal{B}}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T}\epsilon_{\omega,t}\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega,t))] \\ &\leq \frac{1}{\Omega}\cdot BQ\sqrt{\Omega KT}. \\ &= \frac{BQ\sqrt{KT}}{\sqrt{\Omega}}.\end{aligned}$$

Proof of Part (c): Using the same definition of the vector $\mathbf{V}$ as in the proof of part (b), we can write

$$\begin{aligned}&\mathbb{E}\left[\sup_{\mathbf{b}\in\mathcal{B}}\sum_{t=1}^{T}\sum_{\omega=1}^{\Omega}\epsilon_{\omega,t}\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega,t))\right] \\ &= \mathbb{E}\left[\sup_{\mathbf{b}\in\mathcal{B}}\sum_{t=1}^{T}\mathbf{b}_t \bullet \left[\sum_{\omega=1}^{\Omega}\epsilon_{\omega,t}\Phi(\mathbf{x}(\omega,t))\right]\right] \\ &= \mathbb{E}\left[\sup_{\mathbf{b}\in\mathcal{B}}\mathbf{b}\bullet\mathbf{V}\right] \qquad \text{(A.33)}\end{aligned}$$

We now observe that for an arbitrary vector $\mathbf{a}\in\mathbb{R}^n$, the optimal solution to $\max_{\mathbf{x}\in\mathbb{R}^n:\|\mathbf{x}\|_\infty\leq B}\mathbf{a}\bullet\mathbf{x}$ is given by $\mathbf{x} = B\text{sign}(\mathbf{a})$, where $\text{sign}(\mathbf{a})$ is an n-dimensional vector with each entry carrying the sign of the corresponding coordinate of $\mathbf{a}$. The objective value

is given by $B\text{sign}(\mathbf{a}) \bullet \mathbf{a} = B\|\mathbf{a}\|_1$. Thus, we can bound (A.33) as follows:

$$\begin{aligned}
\mathbb{E}[\sup_{\mathbf{b}\in\mathcal{B}} \mathbf{b} \bullet \mathbf{V}] &= B\mathbb{E}[\|\mathbf{V}\|_1] \\
&= B\mathbb{E}\left[\sum_{t=1}^{T}\sum_{k=1}^{K}\left|\sum_{\omega=1}^{\Omega}\epsilon_{\omega,t}\phi_k(\mathbf{x}(\omega,t))\right|\right] \\
&= B\sum_{t=1}^{T}\sum_{k=1}^{K}\mathbb{E}\left[\left|\sum_{\omega=1}^{\Omega}\epsilon_{\omega,t}\phi_k(\mathbf{x}(\omega,t))\right|\right] \\
&\leq B\sum_{t=1}^{T}\sum_{k=1}^{K}\sqrt{\mathbb{E}\left[(\sum_{\omega=1}^{\Omega}\epsilon_{\omega,t}\phi_k(\mathbf{x}(\omega,t)))^2\right]} \\
&\leq B\sum_{t=1}^{T}\sum_{k=1}^{K}\sqrt{\mathbb{E}\left[\sum_{\omega=1}^{\Omega}\epsilon_{\omega,t}^2\phi_k(\mathbf{x}(\omega,t))^2\right]} \\
&\leq B\sum_{t=1}^{T}\sum_{k=1}^{K}\sqrt{\Omega Q^2} \\
&= BQKT\sqrt{\Omega},
\end{aligned}$$

where the second step follows by the definition of $\mathbf{V}$; the third step follows by the linearity of expectation; the fourth step follows by the concavity of the square root function and Jensen's inequality; the fifth step by expanding the square of the weighted sum of the $\epsilon_{\omega,t}$'s, and using the independence of the $\epsilon_{\omega,t}$'s to eliminate cross terms; the sixth step by using the definition of Q and the fact that $\epsilon_{\omega,t}^2 = 1$; and the remaining steps by algebra.

We now bound the Rademacher complexity as

$$\begin{aligned}
\hat{R}(\mathcal{B}) &= \frac{1}{\Omega}\mathbb{E}\left[\sup_{\mathbf{b}\in\mathcal{B}}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T}\epsilon_{\omega,t}\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega,t))\right] \\
&\leq \frac{1}{\Omega}\cdot BKT\sqrt{\Omega}Q \\
&= \frac{BQKT}{\sqrt{\Omega}},
\end{aligned}$$

as required. □

A.1.8 Proof of Theorem 6

We will show that the problem is NP-Hard by showing that the decision version of the MAX-3SAT problem is equivalent to decision version of the randomized policy SAA problem.

The MAX-3SAT problem is a well-known NP-Complete problem, which can be defined as follows. We are given N binary variables, denoted by $y_1, \ldots, y_N$. We also have M clauses, $c_1, \ldots, c_M$, where each clause is a disjunction involving three literals (one of the binary variables or its negation). As an example, a clause could be $y_1 \vee y_4 \vee \neg y_5$, which is satisfied if $y_1 = 1$, $y_4 = 1$ or $y_5 = 0$. The optimization form of the MAX-3SAT problem is to find values for the binary variables $y_1, \ldots, y_N$ that maximizes the number of satisfied clauses. For our purposes, it will be easier to work with the decision form of the problem, which we state below.

MAX-3SAT

Inputs:

- Integers N, M;
- Clauses $c_1, \ldots, c_M$ of three literals;
- Target number of satisfied clauses W.

Question: Do there exist binary values $y_1, \ldots, y_N$ such that the number of satisfied literals $c_1, \ldots, c_M$ is at least W?

We similarly define the decision form of the randomized policy SAA problem.

Randomized Policy SAA

Inputs:

- Integers Ω, K, T;
- State space $\mathcal{X}$;
- Basis function mapping $\Phi(\cdot)$;
- Reward function $g(\cdot, \cdot)$;
- Sample of trajectories $\mathbf{x}(1, \cdot), \ldots, \mathbf{x}(\Omega, \cdot)$;
- Set of feasible weight vectors $\mathcal{B} \subseteq \mathbb{R}^{KT}$;
- Target expected reward θ.

Question: Does there exist a weight vector $\mathbf{b} \in \mathcal{B}$ such that the reward $\hat{J}_R(\mathbf{b}) \geq \theta$? That is, is the inequality

$$\frac{1}{\Omega} \sum_{\omega=1}^{\Omega} \sum_{t=1}^{T} g(t, \mathbf{x}(\omega, t)) \prod_{t'=1}^{t-1} (1 - \sigma(\mathbf{b}_{t'} \bullet \Phi(\mathbf{x}(\omega, t')))) \sigma(\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega, t))) \geq \theta$$

satisfied?

We now show how, for any arbitrary instance of the MAX-3SAT decision problem, we can construct a corresponding instance of the randomized policy SAA decision problem such that the two decision problems are equivalent (the answer to the MAX-3SAT decision problem is yes if and only if the answer to the randomized policy SAA decision problem is yes). We begin by constructing the instance, and then show the equivalence.

Construction of instance: Given a MAX-3SAT decision problem instance, let $\mathcal{X} = \mathbb{R}^N$, and let the basis function mapping Φ be just equal to the identity mapping, i.e., $\Phi(\mathbf{x}) = \mathbf{x}$ for any $\mathbf{x} \in \mathcal{X}$. Thus, the dimension of the basis function vector K is equal to N.

For the trajectories, we will construct $\Omega = M$ trajectories of $T = 3$ periods. For each

clause $m \in [M]$, let $i_{m,1}, i_{m,2}, i_{m,3}$ be the indices of the binary variables that participate in the clause, and let $a_{m,1}, a_{m,2}, a_{m,3}$ be equal to +1 or -1 if the literal is the binary variable itself or its negation, respectively. For example, if the clause were $y_3 \vee \neg y_4 \vee y_7$, then $i_{m,1} = 3$, $i_{m,2} = 4$, $i_{m,3} = 7$, and $a_{m,1} = +1$, $a_{m,2} = -1$, $a_{m,3} = +1$. With these definitions, let us define the trajectories as follows, for each $\omega \in [M]$, each $t \in \{1, 2, 3\}$:

$$x_i(\omega, t) = \begin{cases} a_{m,t} & \text{if } i = i_{m,t}, \\ 0 & \text{otherwise.} \end{cases}$$

For example, for the previous clause, assuming $N = 8$, then the trajectory would be:

$$\mathbf{x}(m, \cdot) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ +1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & +1 \\ 0 & 0 & 0 \end{bmatrix}.$$

For the set of feasible weight vectors, we will define $\mathcal{B}$ as

$$\mathcal{B} = \{\mathbf{b} \in \mathbb{R}^{KT} \mid b_{k,1} = b_{k,2} = b_{k,3} \text{ for all } k \in [K]\}.$$

In words, the weight vector set $\mathcal{B}$ is such that the weight of basis function k is the same in all three periods. For notational convenience, we will drop the time subscript, and just use the subscript k to refer to the weight of basis function k, e.g., b_k instead of $b_{k,1}$.

For the reward function $g(\cdot, \cdot)$, we simply set it as $g(t, \mathbf{x}) = \Omega$ for all $t \in \{1, 2, 3\}$ and $\mathbf{x} \in \mathcal{X}$.

Lastly, for the target objective value θ, we set it equal to $W - 1/2$.

To understand the strategy of our construction, let us write out the expected reward:

$$\begin{aligned}
&\hat{J}_R(\mathbf{b}) \\
&= \frac{1}{\Omega}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} g(t, \mathbf{x}(\omega, t)) \prod_{t'=1}^{t-1}(1 - \sigma(\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega, t'))))\sigma(\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega, t))) \\
&= \frac{1}{\Omega}\sum_{\omega=1}^{\Omega} \Omega[\sigma(a_{\omega,1}b_{i_{\omega,1}}) + (1 - \sigma(a_{\omega,1}b_{i_{\omega,1}}))\sigma(a_{\omega,2}b_{i_{\omega,2}}) \\
&\quad + (1 - \sigma(a_{\omega,1}b_{i_{\omega,1}}))(1 - \sigma(a_{\omega,2}b_{i_{\omega,2}}))\sigma(a_{\omega,3}b_{i_{\omega,3}})] \\
&= \sum_{m=1}^{M}[\sigma(a_{m,1}b_{i_{m,1}}) + (1 - \sigma(a_{m,1}b_{i_{m,1}}))\sigma(a_{m,2}b_{i_{m,2}}) \\
&\quad + (1 - \sigma(a_{m,1}b_{i_{m,1}}))(1 - \sigma(a_{m,2}b_{i_{m,2}}))\sigma(a_{m,3}b_{i_{m,3}})]. \qquad \text{(A.34)}
\end{aligned}$$

To gain some intuition for how this last expression will correspond to the number of satisfied clauses, we make a couple of remarks here.

First, we will see shortly that b_i will correspond to the binary variable y_i in the MAX-3SAT problem. The weight b_i can be thought of as a "soft" / "continuous", real-valued counterpart of the binary variable y_i; we want to use very large positive values of b_i to correspond to the variable y_i being equal to 1, and very small negative values of b_i to correspond to the variable y_i being equal to 0.

Second, to understand how the expression in the square brackets corresponds to a clause evaluating to 1 or 0, observe that we can write a disjunction as the sum of products of the literals. For example, the clause $y_3 \vee \neg y_4 \vee y_7$ we could write as

$$\begin{aligned}
&y_3 + (\neg y_3) \cdot (\neg y_4) + (\neg y_3) \cdot (\neg\neg y_4) \cdot y_7 \\
&= y_3 + (1 - y_3)(1 - y_4) + (1 - y_3)(y_4)y_7. \qquad \text{(A.35)}
\end{aligned}$$

In the above expression, observe that if $y_3 = 1$, then the first term evaluates to 1, and the rest evaluate to 0; otherwise, if $y_3 = 0$ and $y_4 = 0$, then the first term evaluates to 0, the second to 1, and the third to 0; otherwise, if $y_3 = 0$, $y_4 = 1$ and $y_7 = 1$, then the first and second terms evaluate to 0, while the last evaluates to 1. Thus, the two expressions – the

original clause $y_3 \vee \neg y_4 \vee y_7$ and the expression (A.35) – are equivalent. The term in the square brackets in (A.34) has this same form, and we will see shortly that we can use this to establish our needed equivalence. With a slight abuse of terminology, we will refer to the term in the square brackets in (A.34) as the reward of a single trajectory m.

We now proceed with showing the equivalence of the MAX-3SAT decision problem and the randomized policy SAA decision problem with the structure described above.

MAX-3SAT answer is yes $\Rightarrow$ *randomized policy SAA answer is yes*: If the MAX-3SAT decision problem answer is yes, then let $y_1, \ldots, y_N$ be an assignment with objective at least W. Let $\alpha > 0$ be a positive constant, and define a weight vector $\mathbf{b}$ for the randomized policy SAA problem as follows:

$$b_i = \begin{cases} +\alpha & \text{if } y_i = 1, \\ -\alpha & \text{if } y_i = 0. \end{cases} \tag{A.36}$$

Observe now that for a given clause/trajectory m, taking the limit as $\alpha \to \infty$ of

$\sigma(a_{m,t}b_{i_{m,t}})$ gives us the following:

$$\begin{aligned}
&\lim_{\alpha\to+\infty}\sigma(a_{m,t}b_{i_{m,t}}) \\
&= \begin{cases} \lim_{\alpha\to+\infty}\sigma(\alpha) & \text{if } a_{m,t}=+1, y_{i_{m,t}}=1, \\ \lim_{\alpha\to+\infty}\sigma(-\alpha) & \text{if } a_{m,t}=-1, y_{i_{m,t}}=1, \\ \lim_{\alpha\to+\infty}\sigma(-\alpha) & \text{if } a_{m,t}=+1, y_{i_{m,t}}=0, \\ \lim_{\alpha\to+\infty}\sigma(+\alpha) & \text{if } a_{m,t}=-1, y_{i_{m,t}}=0 \end{cases} \\
&= \begin{cases} 1 & \text{if } a_{m,t}=+1, y_{i_{m,t}}=1, \\ 0 & \text{if } a_{m,t}=-1, y_{i_{m,t}}=1, \\ 0 & \text{if } a_{m,t}=+1, y_{i_{m,t}}=0, \\ 1 & \text{if } a_{m,t}=-1, y_{i_{m,t}}=0 \end{cases} \\
&= \begin{cases} y_{i_{m,t}} & \text{if } a_{m,t}=+1, \\ \neg y_{i_{m,t}} & \text{if } a_{m,t}=-1 \end{cases}
\end{aligned}$$

In other words, as $\alpha\to\infty$, $\sigma(a_{m,t}b_{i_{m,t}})$ evaluates to exactly the tth literal of clause m. By our aforementioned equivalence of a disjunction and a sum of products of binary variables (as in the example in equation (A.35)), it follows that

$$\begin{aligned}
\lim_{\alpha\to+\infty}\hat{J}_R(\mathbf{b}) &= \lim_{\alpha\to+\infty}\sum_{m=1}^{M}[\sigma(a_{m,1}b_{i_{m,1}}) + (1-\sigma(a_{m,1}b_{i_{m,1}}))\sigma(a_{m,2}b_{i_{m,2}}) \\
&\qquad + (1-\sigma(a_{m,1}b_{i_{m,1}}))(1-\sigma(a_{m,2}b_{i_{m,2}}))\sigma(a_{m,3}b_{i_{m,3}})] \\
&= \sum_{m=1}^{M} c_m,
\end{aligned}$$

i.e., the limit as α goes to infinity is exactly equal to the number of satisfied clauses in the MAX-3SAT solution $y_1,\ldots,y_N$. Since the answer to the MAX-3SAT decision problem is yes, we know that $\sum_{m=1}^{M} c_m \geq W$, so that the limit $\lim_{\alpha\to+\infty}\hat{J}_R(\mathbf{b}) \geq W$ as well. Since the limit is at least W, it follows that there must exist an α, and thus a corresponding $\mathbf{b}$ (as defined in (A.36)) such that $\hat{J}_R(\mathbf{b}) \geq W - 1/2$.

Randomized policy SAA answer is yes ⇒ MAX-3SAT answer is yes: To show the other direction of the equivalence, let us suppose we have a solution $\mathbf{b}$ for the randomized policy SAA problem with objective value $\hat{J}_R(\mathbf{b}) \geq W - 1/2$. We now need to construct a solution for the MAX-3SAT decision problem with objective value at least W.

Let us use $c_m(y_1, \ldots, y_N)$ to denote the value of clause m as a function of the binary variables $y_1, ..., y_N$. We claim that

$$\hat{J}_R(\mathbf{b}) = \mathbb{E}\left[\sum_{m=1}^{M} c_m(\mathbb{I}\{\xi_1 \leq b_1\}, \ldots, \mathbb{I}\{\xi_N \leq b_N\})\right], \tag{A.37}$$

where $\xi_1, \ldots, \xi_N$ are i.i.d. standard logistic random variables (i.e., $\mathbb{P}(\xi_i \leq t) = \sigma(t)$ for all the variables i). Once we show this, we can use the probabilistic method to assert the existence of $y_1, \ldots, y_N$ that give an affirmative answer to the MAX-3SAT problem.

To show the equivalence (A.37), we argue that for any clause m,

$$\begin{aligned}
&\mathbb{E}[c_m(\mathbb{I}\{\xi_1 \leq b_1\}, \ldots, \mathbb{I}\{\xi_N \leq b_N\})] \\
&= \sigma(a_{m,1} b_{i_{m,1}}) + (1 - \sigma(a_{m,1} b_{i_{m,1}}))\sigma(a_{m,2} b_{i_{m,2}}) \\
&\quad + (1 - \sigma(a_{m,1} b_{i_{m,1}}))(1 - \sigma(a_{m,2} b_{i_{m,2}}))\sigma(a_{m,3} b_{i_{m,3}}).
\end{aligned} \tag{A.38}$$

To see why this must be true, we argue by way of an example. Consider again the example clause $y_3 \vee \neg y_4 \vee y_7$. Consider the right-hand side of (A.38), which is the reward of the corresponding trajectory, after we substitute in the values of the $a_{m,t}$'s. This right hand side works out to

$$\sigma(b_3) + (1 - \sigma(b_3))\sigma(-b_4) + (1 - \sigma(b_3)(1 - \sigma(-b_4))\sigma(b_7).$$

We now use an important property of the logistic response function σ, which is that for any real u, $\sigma(u) = 1 - \sigma(-u)$. Therefore, we can readily modify the above expression so that the coefficient of any b_i is always $+1$:

$$\sigma(b_3) + (1 - \sigma(b_3))(1 - \sigma(b_4)) + (1 - \sigma(b_3)\sigma(b_4)\sigma(b_7).$$

Letting $\xi_1, \ldots, \xi_N$ denote i.i.d. standard logistic random variables, the above can be equivalently written as

$$\mathbb{P}(\xi_3 \leq b_3) + (1 - \mathbb{P}(\xi_3 \leq b_3))(1 - \mathbb{P}(\xi_4 \leq b_4)) + (1 - \mathbb{P}(\xi_3 \leq b_3)) \cdot \mathbb{P}(\xi_4 \leq b_4) \cdot \mathbb{P}(\xi_7 \leq b_7) \tag{A.39}$$

$$\begin{aligned} &= \mathbb{E}[\mathbb{I}\{\xi_3 \leq b_3\}] + \mathbb{E}[1 - \mathbb{I}\{\xi_3 \leq b_3\}]\mathbb{E}[1 - \mathbb{I}\{\xi_4 \leq b_4\}] \\ &\qquad + \mathbb{E}[1 - \mathbb{I}\{\xi_3 \leq b_3\}]\mathbb{E}[\mathbb{I}\{\xi_4 \leq b_4\}]\mathbb{E}[\mathbb{I}\{\xi_7 \leq b_7\}] \\ &= \mathbb{E}[\mathbb{I}\{\xi_3 \leq b_3\} + (1 - \mathbb{I}\{\xi_3 \leq b_3\})(1 - \mathbb{I}\{\xi_4 \leq b_4\}) + (1 - \mathbb{I}\{\xi_3 \leq b_3\})\mathbb{I}\{\xi_4 \leq b_4\}\mathbb{I}\{\xi_7 \leq b_7\}], \end{aligned} \tag{A.40}$$

where the equality on the final line follows by the independence of the ξ's and the linearity of expectation. Now, let $y_3 = \mathbb{I}\{\xi_3 \leq b_3\}$, $y_4 = \mathbb{I}\{\xi_4 \leq b_4\}$ and $y_7 = \mathbb{I}\{\xi_7 \leq b_7\}$. Observe that the expression inside the expectation in (A.40) can be written as

$$y_3 + (1 - y_3)(1 - y_4) + (1 - y_3)y_4y_7$$

which is logically identical to $y_3 \vee \neg y_4 \vee y_7$. Thus, in this example, it follows that equation (A.38) holds. Note that there is nothing special in the particular clause that we chose; the same procedure, which involves using the identity $\sigma(-u) = 1 - \sigma(u)$ to eliminate any term of the form $\sigma(-b_i)$ that appears in the right-hand side of (A.38), can be used to turn the right-hand side of (A.38) into the expected value of the clause function $c_m(y_1, \ldots, y_N)$ when one replaces each y_i with $\mathbb{I}\{\xi_i \leq b_i\}$.

Since (A.38) holds, by linearity of expectation it must be the case that (A.37) also holds. Consequently, there must exist values $\xi'_1, \ldots, \xi'_N$ of the random variables $\xi_1, \ldots, \xi_N$ which satisfy the following:

$$\begin{aligned} &\mathbb{E}[\sum_{m=1}^{M} c_m(\mathbb{I}\{\xi_1 \leq b_1\}, \ldots, \mathbb{I}\{\xi_N \leq b_N\})] \\ &\leq \sum_{m=1}^{M} c_m(\mathbb{I}\{\xi'_1 \leq b_1\}, \ldots, \mathbb{I}\{\xi'_N \leq b_N\}). \end{aligned} \tag{A.41}$$

Define now a candidate solution to the MAX-3SAT problem $y_1, \ldots, y_N$ as $y_i = \mathbb{I}\{\xi_i' \leq b_i\}$ for each i. By (A.41) and (A.37), we have

$$\sum_{m=1}^{M} c_m(y_1, \ldots, y_N) \geq \hat{J}_R(\mathbf{b}).$$

Recall that $\hat{J}_R(\mathbf{b}) \geq W - 1/2$, so we further have that

$$\sum_{m=1}^{M} c_m(y_1, \ldots, y_N) \geq W - 1/2.$$

Since W is an integer, and the number of satisfied clauses must also be an integer, the above is equivalent to

$$\sum_{m=1}^{M} c_m(y_1, \ldots, y_N) \geq W,$$

which shows that the answer to the MAX-3SAT decision problem is yes.

We have shown that the MAX-3SAT decision problem and randomized policy SAA decision problem are equivalent for the constructed instance of the randomized policy SAA problem. Since the particular instance of the randomized policy SAA decision problem can be constructed in polynomial time, and since the MAX-3SAT problem is NP-Complete (Garey and Johnson 1979), it follows that the randomized policy SAA decision problem is NP-Hard. □

A.2 Additional numerical results

A.2.1 Warm starting of RPO method using LSM

In this section, we briefly describe how we use the LSM solution to warm start each solve of problem (2.20). Suppose that the basis function set contains PAYOFF, i.e., the undiscounted payoff $g'(t)$ is a basis function. Let $\mathbf{b}_t = (b_{t,1}, \ldots, b_{t,K})$ be the vector of weights for the LSM algorithm, as we have defined it in Section 2.5.3 (Algorithm 2). The LSM policy stops at

time t if and only if

$$g(t) > \sum_{k=1}^{K} b_{t,k}\phi_k(\mathbf{x}(t)).$$

Using the fact that $g(t) = \beta^t g'(t) = \beta^t \phi_K(\mathbf{x}(t))$, we can re-write this as

$$\begin{aligned} g(t) - \sum_{k=1}^{K} b_{t,k}\phi_k(\mathbf{x}(t)) > 0 \\ \Rightarrow \beta^t \phi_K(\mathbf{x}(t)) - \sum_{k=1}^{K} b_{t,k}\phi_k(\mathbf{x}(t)) > 0 \\ \Rightarrow \sum_{k=1}^{K} b'_{t,k}\phi_k(\mathbf{x}(t)) > 0, \end{aligned}$$

where the vector $\mathbf{b}'_t$ is defined as $\mathbf{b}'_t = (-\beta^{-t}b_{t,1}, \ldots, -\beta^{-t}b_{t,K-1}, 1 - \beta^{-t}b_{t,K})$.

Observe that, as discussed in Section 2.5.3, $\mathbf{b}'_t$ can be viewed as a weight vector defining a deterministic linear policy at time t, that would behave identically to the LSM policy at time t. At the same time, one can also treat $\mathbf{b}'_t$ as a candidate weight vector for a randomized policy at time t. Thus, our warm starting strategy is to simply use $\mathbf{b}'_t$ as the initial solution to problem (2.20).

A.2.2 Additional policy performance results for Section 2.6.3

Table A.1 displays the results comparing LSM, PO and RPO for instances with $n = 4$ assets, while Table A.2 displays analogous results for $n = 16$ assets. Note that for $n = 16$ assets, we omit the results for PO for the basis function architecture containing the second-order price basis functions (PRICES2KO) due to the significant computational effort required for the PO method in this case.

Method	Basis function architecture	Initial price $\bar{p} = 90$	$\bar{p} = 100$	$\bar{p} = 110$
LSM	ONE	24.68 (0.019)	31.78 (0.016)	37.45 (0.038)
LSM	ONE, PAYOFF	32.84 (0.030)	40.02 (0.047)	43.16 (0.043)
PO	ONE	30.84 (0.024)	38.97 (0.019)	44.57 (0.027)
PO	ONE, PAYOFF	22.67 (0.167)	20.77 (0.126)	16.53 (0.127)
RPO	ONE, PAYOFF	**34.48 (0.020)**	**42.92 (0.020)**	**49.16 (0.020)**
PO-UB	ONE	43.23 (0.032)	51.11 (0.024)	56.46 (0.022)
PO-UB	ONE, PAYOFF	35.11 (0.023)	43.94 (0.034)	50.55 (0.032)
LSM	PRICES	25.74 (0.025)	32.08 (0.025)	37.38 (0.040)
LSM	PRICES, PAYOFF	32.34 (0.021)	38.14 (0.040)	40.74 (0.030)
PO	PRICES	31.40 (0.023)	38.92 (0.015)	43.42 (0.017)
PO	PRICES, PAYOFF	23.04 (0.138)	19.94 (0.099)	15.63 (0.095)
RPO	PRICES, PAYOFF	**33.96 (0.018)**	**42.03 (0.013)**	**47.89 (0.020)**
PO-UB	PRICES	40.57 (0.022)	49.27 (0.011)	55.62 (0.018)
PO-UB	PRICES, PAYOFF	35.11 (0.023)	43.94 (0.034)	50.53 (0.032)
LSM	PRICESKO	28.53 (0.029)	38.34 (0.018)	46.55 (0.034)
LSM	PRICESKO, PAYOFF	33.45 (0.018)	41.71 (0.019)	47.73 (0.016)
PO	PRICESKO	32.68 (0.024)	41.84 (0.016)	47.78 (0.018)
PO	PRICESKO, PAYOFF	32.67 (0.027)	41.52 (0.020)	48.02 (0.019)
RPO	PRICESKO, PAYOFF	**33.98 (0.020)**	**42.14 (0.017)**	**48.17 (0.016)**
PO-UB	PRICESKO	39.52 (0.020)	46.89 (0.012)	51.89 (0.012)
PO-UB	PRICESKO, PAYOFF	35.07 (0.020)	43.79 (0.030)	50.17 (0.026)
LSM	KOIND	26.19 (0.027)	35.61 (0.020)	44.02 (0.048)
LSM	KOIND, PAYOFF	33.39 (0.028)	41.89 (0.028)	48.06 (0.022)
PO	KOIND	31.51 (0.025)	41.04 (0.018)	48.43 (0.024)
PO	KOIND, PAYOFF	32.22 (0.047)	42.28 (0.029)	49.01 (0.016)
RPO	KOIND, PAYOFF	**34.53 (0.020)**	**43.07 (0.020)**	**49.39 (0.019)**
PO-UB	KOIND	41.46 (0.028)	48.38 (0.022)	52.83 (0.018)
PO-UB	KOIND, PAYOFF	35.08 (0.021)	43.79 (0.031)	50.18 (0.027)
LSM	PRICESKO, KOIND	30.23 (0.030)	39.07 (0.015)	46.59 (0.029)
LSM	PRICESKO, KOIND, PAYOFF	32.72 (0.023)	41.24 (0.023)	47.74 (0.025)
PO	PRICESKO, KOIND	31.88 (0.019)	40.61 (0.027)	48.41 (0.025)
PO	PRICESKO, KOIND, PAYOFF	31.40 (0.030)	40.59 (0.019)	**48.45 (0.020)**
RPO	PRICESKO, KOIND, PAYOFF	**32.95 (0.023)**	**41.42 (0.025)**	48.09 (0.036)
PO-UB	PRICESKO, KOIND	38.82 (0.016)	46.45 (0.016)	51.75 (0.014)
PO-UB	PRICESKO, KOIND, PAYOFF	35.07 (0.021)	43.78 (0.030)	50.16 (0.026)
LSM	PRICESKO, PRICES2KO, KOIND	31.92 (0.032)	40.93 (0.014)	47.74 (0.019)
LSM	PRICESKO, PRICES2KO, KOIND, PAYOFF	33.41 (0.023)	41.82 (0.021)	48.02 (0.021)
PO	PRICESKO, PRICES2KO, KOIND	32.18 (0.028)	41.88 (0.017)	48.73 (0.015)
PO	PRICESKO, PRICES2KO, KOIND, PAYOFF	33.66 (0.021)	42.48 (0.017)	48.78 (0.015)
RPO	PRICESKO, PRICES2KO, KOIND, PAYOFF	**33.97 (0.026)**	**42.59 (0.021)**	**48.93 (0.022)**
PO-UB	PRICESKO, PRICES2KO, KOIND	36.30 (0.010)	44.56 (0.011)	50.51 (0.011)
PO-UB	PRICESKO, PRICES2KO, KOIND, PAYOFF	35.07 (0.021)	43.74 (0.025)	50.08 (0.023)
LSM	PRICESKO, KOIND, MAXPRICEKO, MAX2PRICEKO	32.93 (0.023)	41.37 (0.020)	47.81 (0.025)
LSM	PRICESKO, KOIND, MAXPRICEKO, MAX2PRICEKO, PAYOFF	32.99 (0.025)	41.38 (0.018)	47.79 (0.024)
PO	PRICESKO, KOIND, MAXPRICEKO, MAX2PRICEKO	32.52 (0.024)	40.92 (0.020)	48.48 (0.019)
PO	PRICESKO, KOIND, MAXPRICEKO, MAX2PRICEKO, PAYOFF	32.23 (0.027)	41.12 (0.020)	**48.49 (0.018)**
RPO	PRICESKO, KOIND, MAXPRICEKO, MAX2PRICEKO, PAYOFF	**33.23 (0.024)**	**41.59 (0.022)**	48.16 (0.035)
PO-UB	PRICESKO, KOIND, MAXPRICEKO, MAX2PRICEKO	35.38 (0.020)	43.84 (0.029)	50.17 (0.025)
PO-UB	PRICESKO, KOIND, MAXPRICEKO, MAX2PRICEKO, PAYOFF	35.06 (0.022)	43.77 (0.030)	50.16 (0.025)

Table A.1: Out-of-sample performance for different policies, for $n = 4$ assets.

Method	Basis function architecture	Initial price $\bar{p} = 90$	$\bar{p} = 100$	$\bar{p} = 110$
LSM	ONE	39.08 (0.015)	43.20 (0.016)	47.14 (0.017)
LSM	ONE, PAYOFF	43.15 (0.033)	45.15 (0.016)	47.47 (0.020)
PO	ONE	46.29 (0.018)	48.93 (0.014)	51.07 (0.009)
PO	ONE, PAYOFF	18.10 (0.142)	15.89 (0.277)	34.50 (0.257)
RPO	ONE, PAYOFF	**51.52 (0.028)**	**52.73 (0.040)**	**53.60 (0.028)**
PO-UB	ONE	57.57 (0.008)	60.29 (0.011)	61.87 (0.007)
PO-UB	ONE, PAYOFF	53.21 (0.035)	56.11 (0.037)	57.40 (0.039)
LSM	PRICES	38.97 (0.019)	43.12 (0.017)	47.06 (0.018)
LSM	PRICES, PAYOFF	42.22 (0.026)	44.55 (0.019)	47.13 (0.021)
PO	PRICES	45.57 (0.016)	48.05 (0.013)	50.37 (0.007)
PO	PRICES, PAYOFF	18.14 (0.098)	16.35 (0.242)	34.82 (0.081)
RPO	PRICES, PAYOFF	**50.00 (0.033)**	**52.07 (0.028)**	**53.57 (0.032)**
PO-UB	PRICES	57.47 (0.004)	60.27 (0.011)	61.84 (0.008)
PO-UB	PRICES, PAYOFF	53.16 (0.033)	56.03 (0.034)	57.31 (0.037)
LSM	PRICESKO	50.31 (0.009)	53.39 (0.011)	54.70 (0.008)
LSM	PRICESKO, PAYOFF	50.28 (0.011)	52.93 (0.010)	54.46 (0.009)
PO	PRICESKO	50.84 (0.010)	53.44 (0.011)	55.03 (0.008)
PO	PRICESKO, PAYOFF	50.83 (0.009)	53.45 (0.008)	54.95 (0.006)
RPO	PRICESKO, PAYOFF	**50.92 (0.010)**	**53.60 (0.010)**	**55.22 (0.010)**
PO-UB	PRICESKO	53.31 (0.007)	55.44 (0.007)	56.70 (0.006)
PO-UB	PRICESKO, PAYOFF	52.49 (0.022)	55.07 (0.017)	56.41 (0.016)
LSM	KOIND	49.83 (0.015)	53.79 (0.012)	55.15 (0.007)
LSM	KOIND, PAYOFF	50.66 (0.015)	53.36 (0.008)	54.84 (0.008)
PO	KOIND	51.59 (0.012)	54.46 (0.012)	55.73 (0.006)
PO	KOIND, PAYOFF	51.38 (0.012)	53.96 (0.008)	55.31 (0.007)
RPO	KOIND, PAYOFF	**51.93 (0.011)**	**54.58 (0.013)**	**55.97 (0.007)**
PO-UB	KOIND	53.47 (0.009)	55.49 (0.007)	56.74 (0.006)
PO-UB	KOIND, PAYOFF	52.50 (0.021)	55.06 (0.014)	56.40 (0.015)
LSM	PRICESKO, KOIND	50.38 (0.011)	53.70 (0.010)	54.99 (0.009)
LSM	PRICESKO, KOIND, PAYOFF	50.50 (0.013)	53.28 (0.010)	54.79 (0.009)
PO	PRICESKO, KOIND	**51.60 (0.011)**	54.34 (0.010)	55.55 (0.005)
PO	PRICESKO, KOIND, PAYOFF	51.27 (0.011)	53.91 (0.008)	55.29 (0.008)
RPO	PRICESKO, KOIND, PAYOFF	51.41 (0.013)	**54.38 (0.014)**	**55.87 (0.008)**
PO-UB	PRICESKO, KOIND	53.30 (0.008)	55.43 (0.005)	56.69 (0.005)
PO-UB	PRICESKO, KOIND, PAYOFF	52.48 (0.021)	55.04 (0.014)	56.38 (0.015)
LSM	PRICESKO, KOIND, MAXPRICEKO, MAX2PRICEKO	50.50 (0.008)	53.26 (0.013)	54.79 (0.014)
LSM	PRICESKO, KOIND, MAXPRICEKO, MAX2PRICEKO, PAYOFF	50.49 (0.009)	53.26 (0.013)	54.79 (0.014)
PO	PRICESKO, KOIND, MAXPRICEKO, MAX2PRICEKO	51.23 (0.010)	53.89 (0.012)	55.28 (0.011)
PO	PRICESKO, KOIND, MAXPRICEKO, MAX2PRICEKO, PAYOFF	51.23 (0.011)	53.89 (0.011)	55.28 (0.011)
RPO	PRICESKO, KOIND, MAXPRICEKO, MAX2PRICEKO, PAYOFF	**51.39 (0.017)**	**54.37 (0.016)**	**55.84 (0.012)**
PO-UB	PRICESKO, KOIND, MAXPRICEKO, MAX2PRICEKO	52.48 (0.027)	55.04 (0.019)	56.38 (0.018)
PO-UB	PRICESKO, KOIND, MAXPRICEKO, MAX2PRICEKO, PAYOFF	52.40 (0.039)	54.90 (0.082)	56.38 (0.019)
LSM	PRICESKO, PRICES2KO, KOIND	50.32 (0.014)	53.19 (0.010)	54.61 (0.008)
LSM	PRICESKO, PRICES2KO, KOIND, PAYOFF	50.25 (0.016)	53.05 (0.010)	54.60 (0.008)
RPO	PRICESKO, PRICES2KO, KOIND, PAYOFF	**50.94 (0.021)**	**53.78 (0.019)**	**55.24 (0.033)**

Table A.2: Out-of-sample performance for different policies, for $n = 16$ assets.

APPENDIX B

Randomized Robust Price Optimization

B.1 Omitted proofs

B.1.1 Proof of Theorem 7

To prove this, we prove that $Z^*_{\mathrm{DR}} \geq Z^*_{\mathrm{RR}}$. For any $R \in \mathcal{R}$, and any distribution F supported on $\mathcal{P}$, we have

$$\int_{\mathcal{P}} R(\mathbf{p})\, dF(\mathbf{p}) \leq R\left(\int_{\mathcal{P}} \mathbf{p}\, dF(\mathbf{p})\right), \tag{B.1}$$

which follows by Jensen's inequality and the concavity of R. This implies that for any $F \in \mathcal{F}$,

$$\inf_{R\in\mathcal{R}} \int_{\mathcal{P}} R(\mathbf{p})\, dF(\mathbf{p}) \leq \inf_{R\in\mathcal{R}} R\left(\int_{\mathcal{P}} \mathbf{p}\, dF(\mathbf{p})\right). \tag{B.2}$$

Therefore, we have that

$$\begin{aligned} Z^*_{\mathrm{RR}} &= \max_{F\in\mathcal{F}} \left\{ \inf_{R\in\mathcal{R}} \int_{\mathcal{P}} R(\mathbf{p})\, dF(\mathbf{p}) \right\} \\ &\leq \max_{F\in\mathcal{F}} \left\{ \inf_{R\in\mathcal{R}} R\left(\int_{\mathcal{P}} \mathbf{p}\, dF(\mathbf{p})\right) \right\} \\ &\leq \max_{\mathbf{p}\in\mathcal{P}} \left\{ \inf_{R\in\mathcal{R}} R\left(\mathbf{p}\right)) \right\} \\ &= Z^*_{\mathrm{DR}}, \end{aligned}$$

where the second inequality follows because $\mathcal{P}$ is assumed to be convex, and thus for any $F \in \mathcal{F}$, $\int_{\mathcal{P}} \mathbf{p}\, dF(\mathbf{p})$ is contained in $\mathcal{P}$. □

B.1.2 Proof of Theorem 8

Before we begin, we recall Sion's minimax theorem:

Theorem 11 (Sion's minimax theorem (Corollary 3.3 of Sion 1958)) *Let M and N be convex spaces, with at least one of the two spaces being compact Let $f : M \times N \to \mathbb{R}$ be a function such that $f(\mu, \nu)$ is quasiconcave and upper semi-continuous in μ for any fixed ν, and quasiconvex and lower semi-continuous in ν for any fixed μ. Then* $\sup_{\mu \in M} \inf_{\nu \in N} f(\mu, \nu) = \inf_{\nu \in N} \sup_{\mu \in M} f(\mu, \nu)$.

We have:

$$\begin{aligned} Z^*_{\mathrm{RR}} &= \max_{F \in \mathcal{F}} \min_{\mathbf{u} \in \mathcal{U}} \int_{\mathcal{P}} R(\mathbf{p}, \mathbf{u})\, dF(\mathbf{p}) \\ &= \min_{\mathbf{u} \in \mathcal{U}} \max_{F \in \mathcal{F}} \int_{\mathcal{P}} R(\mathbf{p}, \mathbf{u})\, dF(\mathbf{p}) \\ &= \min_{\mathbf{u} \in \mathcal{U}} \max_{\mathbf{p} \in \mathcal{P}} R(\mathbf{p}, \mathbf{u}) \\ &= \max_{\mathbf{p} \in \mathcal{P}} \min_{\mathbf{u} \in \mathcal{U}} R(\mathbf{p}, \mathbf{u}) \\ &= Z^*_{\mathrm{DR}}. \end{aligned}$$

In the above, the steps are justified as follows. The first step follows by the definition of Z^*_{RR}.

The second step follows by applying Sion's minimax theorem to interchange the order of minimization over $\mathcal{U}$ and maximization over $\mathcal{F}$. The justification for applying Sion's minimax theorem here is that (1) the set $\mathcal{F}$ of distributions supported on $\mathcal{P}$ is a convex set; (2) $\int_{\mathcal{P}} R(\mathbf{p}, \mathbf{u})\, dF(\mathbf{p})$ is linear in F when $\mathbf{u}$ is fixed; and (3) $\int_{\mathcal{P}} R(\mathbf{p}, \mathbf{u}) dF(\mathbf{p})$ is quasiconvex in $\mathbf{u}$ when F is fixed by the hypotheses of the theorem. Note that $\int_{\mathcal{P}} R(\mathbf{p}, \mathbf{u})\, dF(\mathbf{p})$ is continuous in F if the set of measures $\mathcal{F}$ is endowed with the topology of weak convergence. Additionally, note that $\int_{\mathcal{P}} R(\mathbf{p}, \mathbf{u})\, dF(\mathbf{p})$ is continuous in $\mathbf{u}$. This is guaranteed because, by compactness of $\mathcal{P}$ and $\mathcal{U}$ and continuity of R in $(\mathbf{p}, \mathbf{u})$ from Assumption 7, there exists a constant C such that $|R(\mathbf{p}, \mathbf{u})| < C$ for all $\mathbf{p} \in \mathcal{P}$ and $\mathbf{u} \in \mathcal{U}$; thus, by the bounded convergence

theorem, for any sequence $(\mathbf{u}_k)_{k=1}^{\infty}$ such that $\mathbf{u}_k \to \mathbf{u}$, we shall also have $\int_{\mathcal{P}} R(\mathbf{p}, \mathbf{u}_k)\, dF(\mathbf{p}) \to \int_{\mathcal{P}} R(\mathbf{p}, \mathbf{u})\, dF(\mathbf{p})$.

The third step follows by the fact that $\max_{F \in \mathcal{F}} \int_{\mathcal{P}} R(\mathbf{p}, \mathbf{u})\, dF(\mathbf{p}) = \max_{\mathbf{p} \in \mathcal{P}} R(\mathbf{p}, \mathbf{u})$, since $\mathcal{F}$ includes the distribution the Dirac delta distribution $\delta_{\mathbf{p}'}$ that places unit probability mass on $\mathbf{p}'$, for every $\mathbf{p}' \in \mathcal{P}$.

The fourth step follows by applying Sion's minimax theorem again, using the hypotheses that $R(\mathbf{p}, \mathbf{u})$ is quasiconvex in $\mathbf{u}$ and quasiconcave in $\mathbf{p}$, and additionally that R is continuous in both $\mathbf{u}$ and $\mathbf{p}$ (Assumption 7). □

B.1.3 Proof of Theorem 9

To establish this result, observe that

$$\begin{aligned}
&\inf_{Q \in \mathcal{Q}} \max_{\mathbf{p} \in \mathcal{P}} \int_{\mathcal{U}} R(\mathbf{p}, \mathbf{u})\, dQ(\mathbf{u}) \\
&= \inf_{Q \in \mathcal{Q}} \max_{\boldsymbol{\pi} \in \Delta_{\mathcal{P}}} \sum_{\mathbf{p} \in \mathcal{P}} \pi(\mathbf{p}) \cdot \int_{\mathcal{U}} R(\mathbf{p}, \mathbf{u})\, dQ(\mathbf{u}) \\
&= \max_{\boldsymbol{\pi} \in \Delta_{\mathcal{P}}} \inf_{Q \in \mathcal{Q}} \sum_{\mathbf{p} \in \mathcal{P}} \pi(\mathbf{p}) \cdot \int_{\mathcal{U}} R(\mathbf{p}, \mathbf{u})\, dQ(\mathbf{u}) \\
&= \max_{\boldsymbol{\pi} \in \Delta_{\mathcal{P}}} \inf_{Q \in \mathcal{Q}} \int_{\mathcal{U}} \sum_{\mathbf{p} \in \mathcal{P}} \pi(\mathbf{p}) \cdot R(\mathbf{p}, \mathbf{u})\, dQ(\mathbf{u}) \\
&= \max_{\boldsymbol{\pi} \in \Delta_{\mathcal{P}}} \min_{\mathbf{u} \in \mathcal{U}} \sum_{\mathbf{p} \in \mathcal{P}} \pi(\mathbf{p}) \cdot R(\mathbf{p}, \mathbf{u}) \\
&= Z^*_{\mathrm{RR}}.
\end{aligned}$$

In the above, the steps are justified as follows. The first step follows because maximizing a function of $\mathbf{p}$ over the finite set $\mathcal{P}$ is the same as maximizing the expected value of that same function with respect to all probability mass functions $\boldsymbol{\pi}$ supported on $\mathcal{P}$.

The second step follows by Sion's minimax theorem, because the quantity $\sum_{\mathbf{p} \in \mathcal{P}} \pi(\mathbf{p}) \cdot \int_{\mathcal{U}} R(\mathbf{p}, \mathbf{u})\, dQ(\mathbf{u})$ is linear and therefore quasiconcave in $\boldsymbol{\pi}$ for a fixed Q, and is linear and therefore quasiconvex in Q for a fixed $\boldsymbol{\pi}$; additionally, the set $\Delta_{\mathcal{P}} = \{\boldsymbol{\pi} \in \mathbb{R}^{|\mathcal{P}|} \mid$

$\mathbf{1}^T\boldsymbol{\pi} = 1, \boldsymbol{\pi} \geq \mathbf{0}\}$ is a compact convex set, and $\mathcal{Q}$ is a convex set. Additionally, note that $\sum_{\mathbf{p}\in\mathcal{P}} \pi(\mathbf{p}) \cdot \int_{\mathcal{U}} R(\mathbf{p}, \mathbf{u})\, dQ(\mathbf{u})$ is clearly continuous in $\boldsymbol{\pi}$. It is also continuous in Q, because each term $\int_{\mathcal{U}} R(\mathbf{p}, \mathbf{u})\, dQ(\mathbf{u})$ is continuous in Q when $\mathcal{Q}$ is endowed with the topology of weak convergence, and there are finitely many such terms.

The third step follows by the linearity of integration. The fourth step follows by the fact that $\mathcal{Q}$ contains the Dirac delta distribution that places unit mass on $\mathbf{u}$, for every $\mathbf{u} \in \mathcal{U}$. The final step just follows from the definition of Z^*_{RR}.

With this result in hand, observe that the existence of a $Q \in \mathcal{Q}$ such that for all $\mathbf{p} \in \mathcal{P}$, $\int_{\mathcal{U}} R(\mathbf{p}, \mathbf{u}) dQ(\mathbf{u}) \leq Z^*_{\mathrm{DR}}$ is equivalent to the existence of $Q \in \mathcal{Q}$ such that

$$\max_{\mathbf{p}\in\mathcal{P}} \int_{\mathcal{U}} R(\mathbf{p}, \mathbf{u}) dQ(\mathbf{u}) \leq Z^*_{\mathrm{DR}},$$

which is equivalent to

$$\inf_{Q\in\mathcal{Q}} \max_{\mathbf{p}\in\mathcal{P}} \int_{\mathcal{U}} R(\mathbf{p}, \mathbf{u}) dQ(\mathbf{u}) \leq Z^*_{\mathrm{DR}}.$$

Since the left-hand side of this inequality is equal to Z^*_{RR}, the existence of the distribution $Q \in \mathcal{Q}$ as in the theorem statement is equivalent to $Z^*_{\mathrm{RR}} \leq Z^*_{\mathrm{DR}}$; since it is always the case that $Z^*_{\mathrm{RR}} \geq Z^*_{\mathrm{DR}}$, this is equivalent to the problem being randomization-proof. □

B.1.4 Proof of Corollary 2

Observe that since $\max_{\mathbf{p}\in\mathcal{P}} \min_{\mathbf{u}\in\mathcal{U}} R(\mathbf{p}, \mathbf{u}) \geq \min_{\mathbf{u}\in\mathcal{U}} \max_{\mathbf{p}\in\mathcal{P}} R(\mathbf{p}, \mathbf{u})$ always holds, equation (3.17) is equivalent to

$$\max_{\mathbf{p}\in\mathcal{P}} \min_{\mathbf{u}\in\mathcal{U}} R(\mathbf{p}, \mathbf{u}) \leq \min_{\mathbf{u}\in\mathcal{U}} \max_{\mathbf{p}\in\mathcal{P}} R(\mathbf{p}, \mathbf{u}),$$

or equivalently,

$$Z^*_{\mathrm{DR}} \geq \min_{\mathbf{u}\in\mathcal{U}} \max_{\mathbf{p}\in\mathcal{P}} R(\mathbf{p}, \mathbf{u}).$$

Observe that the condition $\min_{\mathbf{u}\in\mathcal{U}} \max_{\mathbf{p}\in\mathcal{P}} R(\mathbf{p}, \mathbf{u}) \leq Z^*_{\mathrm{DR}}$ is exactly equivalent to the condition that there exists a $\mathbf{u} \in \mathcal{U}$ such that for all $\mathbf{p} \in \mathcal{P}$, $R(\mathbf{p}, \mathbf{u}) \leq Z^*_{\mathrm{DR}}$.

To connect this to Theorem 9, consider $Q = \delta_{\mathbf{u}}$, where $\delta_{\mathbf{u}}$ is the Dirac delta distribution centered at $\mathbf{u}$. For any $\mathbf{p} \in \mathcal{P}$, $R(\mathbf{p}, \mathbf{u}) = \int_{\mathcal{U}} R(\mathbf{p}, \mathbf{u}')\, dQ(\mathbf{u}')$. Thus, for this choice of Q, it is the case that for all $\mathbf{p} \in \mathcal{P}$, $\int_{\mathcal{U}} R(\mathbf{p}, \mathbf{u}')\, dQ(\mathbf{u}') \leq Z^*_{\text{DR}}$. By Theorem 9, this is equivalent to randomization-proofness. Thus, it follows that the strong duality condition (3.17) is equivalent to the RPO problem being randomization-proof. □

B.1.5 Proof of Corollary 3

To prove the $\Rightarrow$ direction of the equivalence, suppose that the robust price optimization problem is randomization-receptive. From Theorem 9, a robust price optimization problem is randomization-proof if and only if there exists a distribution $Q \in \mathcal{Q}$ such that for all $\mathbf{p} \in \mathcal{P}$, $\int_{\mathcal{U}} R(\mathbf{p}, \mathbf{u})\ dQ(\mathbf{u}) \leq Z^*_{\text{DR}}$. The negation of this latter statement is the following statement:

$$\forall Q \in \mathcal{Q},\ \exists \mathbf{p} \in \mathcal{P} \text{ such that } \int_{\mathcal{U}} R(\mathbf{p}, \mathbf{u})\ dQ(\mathbf{u}) > Z^*_{\text{DR}}. \tag{B.3}$$

We need to show that $\mathbf{p}^*_{\text{DR}} \notin \arg\max_{\mathbf{p} \in \mathcal{P}} R(\mathbf{p}, \mathbf{u}^*)$. To establish this, it is sufficient to show that there exists a $\tilde{\mathbf{p}} \in \mathcal{P}$ such that $R(\tilde{\mathbf{p}}, \mathbf{u}^*) > R(\mathbf{p}^*_{\text{DR}}, \mathbf{u}^*)$. Let $\tilde{Q} = \delta_{\mathbf{u}^*}$, where $\delta_{\mathbf{u}^*}$ denotes the Dirac delta distribution centered at $\mathbf{u}^*$. By invoking (B.3), we are assured of the existence of a price vector $\tilde{\mathbf{p}}$ such that $\int_{\mathcal{U}} R(\tilde{\mathbf{p}}, \mathbf{u})\ d\tilde{Q}(\mathbf{u}) > Z^*_{\text{DR}}$. Since $\tilde{Q} = \delta_{\mathbf{u}^*}$, we have that $\int_{\mathcal{U}} R(\tilde{\mathbf{p}}, \mathbf{u})\ d\tilde{Q}(\mathbf{u}) = R(\tilde{\mathbf{p}}, \mathbf{u}^*)$, and thus we have that

$$R(\tilde{\mathbf{p}}, \mathbf{u}^*) > R(\tilde{\mathbf{p}}_{\text{DR}}, \mathbf{u}^*),$$

exactly as needed. Thus, it follows that $\mathbf{p}^*_{\text{DR}} \notin \arg\max_{\mathbf{p} \in \mathcal{P}} R(\mathbf{p}, \mathbf{u}^*)$.

To prove the $\Leftarrow$ direction of the equivalence, let $\tilde{\mathbf{p}} \in \mathcal{P}$ be a price vector for which $R(\tilde{\mathbf{p}}, \mathbf{u}^*) > R(\mathbf{p}^*_{\text{DR}}, \mathbf{u}^*)$. To establish that the problem is randomization-receptive, we shall again use the condition (B.3).

Let $Q \in \mathcal{Q}$ be an arbitrary distribution. We need to show that there exists a $\mathbf{p} \in \mathcal{P}$ that satisfies $\int_{\mathcal{U}} R(\mathbf{p}, \mathbf{u})\ dQ(\mathbf{u}) > Z^*_{\text{DR}}$. There are two mutually exclusive and collectively

exhaustive cases to consider:

Case 1: There exists a closed set $B \subseteq \mathcal{U}$ such that $\mathbf{u}^* \notin B$ and $Q(B) > 0$. The candidate price vector we will consider in this case is $\mathbf{p}^*_{\text{DR}}$. In this case, observe that since $\mathcal{U}$ is compact, then B is also compact, and together with the extreme value theorem we can assert that $\min_{\mathbf{u}\in B} R(\mathbf{p}^*_{\text{DR}}, \mathbf{u}) = R(\mathbf{p}^*_{\text{DR}}, \tilde{\mathbf{u}})$ for some $\tilde{\mathbf{u}} \in B$. Additionally, since $\min_{\mathbf{u}\in\mathcal{U}} R(\mathbf{p}^*_{\text{DR}}, \mathbf{u})$ has a unique solution $\mathbf{u}^*$ and $\mathbf{u}^* \notin B$, we are assured that $R(\mathbf{p}^*_{\text{DR}}, \tilde{\mathbf{u}}) > R(\mathbf{p}^*_{\text{DR}}, \mathbf{u}^*)$. Armed with these facts, we have that

$$
\begin{aligned}
&\int_{\mathcal{U}} R(\mathbf{p}^*_{\text{DR}}, \mathbf{u})\, dQ(\mathbf{u}) \\
&= \int_{B} R(\mathbf{p}^*_{\text{DR}}, \mathbf{u})\, dQ(\mathbf{u}) + \int_{\mathcal{U}\setminus B} R(\mathbf{p}^*_{\text{DR}}, \mathbf{u})\, dQ(\mathbf{u}) \\
&\geq R(\mathbf{p}^*_{\text{DR}}, \tilde{\mathbf{u}}) \cdot Q(B) + R(\mathbf{p}^*_{\text{DR}}, \mathbf{u}^*) \cdot (1 - Q(B)) \\
&> R(\mathbf{p}^*_{\text{DR}}, \mathbf{u}^*) \cdot (Q(B) + 1 - Q(B)) \\
&= R(\mathbf{p}^*_{\text{DR}}, \mathbf{u}^*) \\
&= Z^*_{\text{DR}},
\end{aligned}
$$

which establishes that condition (B.3) holds in Case 1.

Case 2: For every closed set $B \subseteq \mathcal{U}$, either $\mathbf{u}^* \in B$ or $Q(B) = 0$. The candidate price vector in this case will be $\tilde{\mathbf{p}}$.

To establish condition (B.3) in this case, let ϵ be any number such that $0 < \epsilon < R(\tilde{\mathbf{p}}, \mathbf{u}^*) - R(\mathbf{p}^*_{\text{DR}}, \mathbf{u}^*)$. The assumption about the continuity of $R(\mathbf{p}, \mathbf{u})$ in $\mathbf{u}$ implies that there must exist a $\delta > 0$ such that for any $\mathbf{u} \in \mathcal{U}$ with $\|\mathbf{u} - \mathbf{u}^*\| < \delta$, $R(\tilde{\mathbf{p}}, \mathbf{u}) > R(\mathbf{p}^*_{\text{DR}}, \mathbf{u}^*) + \epsilon$.

Let $C = \{\mathbf{u} \in \mathcal{U} \mid \|\mathbf{u} - \mathbf{u}^*\| < \delta\}$, which is an open set. Additionally, let $B = \mathcal{U} \setminus C = \{\mathbf{u} \in \mathcal{U} \mid \|\mathbf{u} - \mathbf{u}^*\| \geq \delta\}$ be the complement of C, which must be a closed set. By the assumption of Case 2, any closed subset of $\mathcal{U}$ must be such that either $\mathbf{u}^*$ is inside that set, or the measure of that set under Q is zero. Here, by construction, B cannot contain $\mathbf{u}^*$; therefore, we must have that $Q(B) = 0$. Since C and B are complements, it must also be the case that $Q(C) = 1$.

Armed with these facts, we now have that

$$\begin{aligned}
&\int_{\mathcal{U}} R(\tilde{\mathbf{p}}, \mathbf{u})\, dQ(\mathbf{u}) \\
&= \int_{C} R(\tilde{\mathbf{p}}, \mathbf{u})\, dQ(\mathbf{u}) + \int_{B} R(\tilde{\mathbf{p}}, \mathbf{u})\, dQ(\mathbf{u}) \\
&\geq [R(\mathbf{p}^*_{\mathrm{DR}}, \mathbf{u}^*) + \epsilon] \cdot Q(C) + 0 \\
&= R(\mathbf{p}^*_{\mathrm{DR}}, \mathbf{u}^*) + \epsilon \\
&> R(\mathbf{p}^*_{\mathrm{DR}}, \mathbf{u}^*) \\
&= Z^*_{\mathrm{DR}},
\end{aligned}$$

which again establishes that condition (B.3) holds.

Since we have shown that condition (B.3) holds in these two mutually exclusive and collectively exhaustive cases, it follows that the problem is randomization-receptive, as required. □

B.1.6 Example of necessity of uniqueness assumption in Corollary 3

Consider a single product pricing instance, i.e., $I = 1$, which we define as follows. Let $\mathcal{P} = \{p_1, p_2, p_3\}$ where $p_1 = 5$, $p_2 = 8$, $p_3 = 9$. Let the demand model d be a linear demand model, so that the uncertain parameter $\mathbf{u} = (\alpha, \beta)$ and $d(p, \mathbf{u}) = \alpha - \beta p$. Finally, let $\mathcal{U} = \{(\alpha_1, \beta_1), (\alpha_2, \beta_2), (\alpha_3, \beta_3)\}$, where $(\alpha_1, \beta_1) = (10, 1)$, $(\alpha_2, \beta_2) = (3, 0.1)$, $(\alpha_3, \beta_3) = (3.6, 0.2)$.

We first calculate $\min_{\mathbf{u}\in\mathcal{U}} R(p, \mathbf{u})$ for each $p \in \mathcal{P}$. We have:

- For $p_1 = 5$: $p_1(\alpha_2 - \beta_2 p_1) = 12.5 < p_1(\alpha_3 - \beta_3 p_1) = 13 < p_1(\alpha_1, \beta_1 p_1) = 25$. Hence, $\min_{\mathbf{u}\in\mathcal{U}} R(p_1, \mathbf{u}) = \min\{12.5, 13, 25\} = 12.5$.
- For $p_2 = 8$: $p_2(\alpha_1 - \beta_1 p_2) = p_2(\alpha_3 - \beta_3 p_2) = 16 < p_2(\alpha_2 - \beta_2 p_2) = 17.6$. Hence, $\min_{\mathbf{u}\in\mathcal{U}} R(p_2, \mathbf{u}) = \min\{16, 16, 17.6\} = 16$, and note also that the minimizing $\mathbf{u}$ is not unique (the minimum is attained at both (α_1, β_1) and (α_3, β_3)).

- For $p_3 = 9$: $p_3(\alpha_1 - \beta_1 p_3) = 9 < p_3(\alpha_3 - \beta_3 p_3) = 16.2 < p_3(\alpha_2 - \beta_2 p_3) = 18.9$. Hence, $\min_{\mathbf{u}\in\mathcal{U}} R(p_3, \mathbf{u}) = \min\{9, 16.2, 18.9\} = 9$.

From this, we can see that the optimal deterministic robust price is $p^*_{\text{DR}} = p_2 = 8$ and the optimal deterministic robust objective value is $Z^*_{\text{DR}} = 16$. At $p = 8$, we can see that $\arg\min_{\mathbf{u}\in\mathcal{U}} R(p_2, \mathbf{u}) = \{(\alpha_1, \beta_1), (\alpha_3, \beta_3)\}$.

Let us now consider the RRPO problem. When we write the problem $\max_{\boldsymbol{\pi}\in\Delta_{\mathcal{P}}} \min_{\mathbf{u}\in\mathcal{U}} \sum_{\mathbf{p}\in\mathcal{P}} \pi_{\mathbf{p}} R(\mathbf{p}, \mathbf{u})$ as a linear program, we get the following problem:

$$\begin{aligned}
\underset{\boldsymbol{\pi},t}{\text{maximize}} \quad & t && \text{(B.4a)}\\
\text{subject to} \quad & t \le \pi_{p_1} \cdot p_1(\alpha_1 - \beta_1 p_1) + \pi_{p_2} \cdot p_2(\alpha_1 - \beta_1 p_2) + \pi_{p_3} \cdot p_3(\alpha_1 - \beta_1 p_3), && \text{(B.4b)}\\
& t \le \pi_{p_1} \cdot p_1(\alpha_2 - \beta_2 p_1) + \pi_{p_2} \cdot p_2(\alpha_2 - \beta_2 p_2) + \pi_{p_3} \cdot p_3(\alpha_2 - \beta_2 p_3), && \text{(B.4c)}\\
& t \le \pi_{p_1} \cdot p_1(\alpha_3 - \beta_3 p_1) + \pi_{p_2} \cdot p_2(\alpha_3 - \beta_3 p_2) + \pi_{p_3} \cdot p_3(\alpha_3 - \beta_3 p_3), && \text{(B.4d)}\\
& \pi_{p_1} + \pi_{p_2} + \pi_{p_3} = 1, && \text{(B.4e)}\\
& \pi_{p_1}, \pi_{p_2}, \pi_{p_3} \ge 0, && \text{(B.4f)}
\end{aligned}$$

or equivalently,

$$\begin{aligned}
\underset{\boldsymbol{\pi},t}{\text{maximize}} \quad & t && \text{(B.5a)}\\
\text{subject to} \quad & t \le 25\pi_{p_1} + 16\pi_{p_2} + 9\pi_{p_3} && \text{(B.5b)}\\
& t \le 12.5\pi_{p_1} + 17.6\pi_{p_2} + 18.9\pi_{p_3}, && \text{(B.5c)}\\
& t \le 13\pi_{p_1} + 16\pi_{p_2} + 16.2\pi_{p_3}, && \text{(B.5d)}\\
& \pi_{p_1} + \pi_{p_2} + \pi_{p_3} = 1, && \text{(B.5e)}\\
& \pi_{p_1}, \pi_{p_2}, \pi_{p_3} \ge 0, && \text{(B.5f)}
\end{aligned}$$

for which the optimal objective value is $Z^*_{\text{RR}} = 16$, which is the same as Z^*_{DR}. Thus, if the uniqueness condition on $\min_{\mathbf{u}\in\mathcal{U}} R(\mathbf{p}^*_{\text{DR}}, \mathbf{u})$ is relaxed, then it is possible for the problem to be randomization proof.

B.1.7 Proof of Proposition 2

Observe that the objective function in (3.32) can be re-arranged as

$$\begin{aligned}
&\max_{\boldsymbol{\mu}\in\Delta_{[I]},\mathbf{p}\in\mathcal{P}} \left\{ \sum_{i=1}^{I} \mu_i(\alpha_i + \log p_i - \beta_i \log p_i + \sum_{j\neq i} \gamma_{i,j} \log p_j) - \sum_{i=1}^{I} \mu_i \log \mu_i \right\} \\
&= \max_{\boldsymbol{\mu}\in\Delta_{[I]}} \max_{\mathbf{p}\in\mathcal{P}} \left\{ \sum_{i=1}^{I} \mu_i \cdot \alpha_i + \sum_{i=1}^{I} \mu_i \cdot \left[1 - \beta_i + \sum_{j\neq i} \gamma_{j,i} \right] \cdot \log p_i - \sum_{i=1}^{I} \mu_i \log \mu_i \right\} \\
&= \max_{\boldsymbol{\mu}\in\Delta_{[I]}} \left[\sum_{i=1}^{I} \mu_i \cdot \alpha_i + \sum_{i=1}^{I} \max_{p_i\in\mathcal{P}_i} \left\{ \mu_i \cdot \left[1 - \beta_i + \sum_{j\neq i} \gamma_{j,i} \right] \cdot \log p_i \right\} - \sum_{i=1}^{I} \mu_i \log \mu_i \right]
\end{aligned}$$

where the first step follows by algebra, and the second by the separability of the objective in $p_1, \ldots, p_I$ and Assumption 8 (since the price set is a Cartesian product and the objective is separable, each product's price can be optimized independently). Thus, when $\boldsymbol{\mu}$ is fixed, the optimal value of p_i for the above objective depends on the sign of $(1 - \beta_i + \sum_{j\neq i} \gamma_{j,i})$. If this coefficient is positive, then since $\log p_i$ is increasing in p_i, it is optimal to set $p_i' = \max \mathcal{P}_i$. If this coefficient is negative, then it is optimal to set $p_i' = \min \mathcal{P}_i$. It thus follows that for any $\boldsymbol{\mu}$ for which we can find a price vector $\mathbf{p}$ such that $(\boldsymbol{\mu}, \mathbf{p})$ is optimal, it will be the case that $(\boldsymbol{\mu}, \mathbf{p}')$ will also be optimal. □

B.2 Deterministic robust price optimization for finite $\mathcal{P}$, convex $\mathcal{U}$ under the semi-log and log-log demand models

In this section, we describe how to formulate the DRPO problem as a mixed-integer exponential cone program for the semi-log and log-log demand models. In both cases, we assume that $\mathcal{U}$ is a convex uncertainty set, and that Assumption 8 on the structure of $\mathcal{P}$ holds.

B.2.1 Semi-log model

For the semi-log demand model, we can write the DRPO problem as

$$\begin{aligned}&\max_{\mathbf{p}\in\mathcal{P}}\min_{\mathbf{u}\in\mathcal{U}} R(\mathbf{p},\mathbf{u})\\&=\max_{\mathbf{p}\in\mathcal{P}}\min_{\mathbf{u}\in\mathcal{U}}\sum_{i=1}^{I} p_i\cdot e^{\alpha_i-\beta_i p_i+\sum_{j\neq i}\gamma_{i,j}p_j}.\end{aligned}\tag{B.6}$$

To accomplish our reformulation, we will make use of the fact that the optimal solution set of the DRPO problem is unchanged upon log-transformation, that is,

$$\arg\max_{\mathbf{p}\in\mathcal{P}}\min_{\mathbf{u}\in\mathcal{U}} R(\mathbf{p},\mathbf{u}) = \arg\max_{\mathbf{p}\in\mathcal{P}}\min_{\mathbf{u}\in\mathcal{U}} \log R(\mathbf{p},\mathbf{u}).$$

Thus, instead of problem (B.6), we can focus on the following problem:

$$\begin{aligned}&\max_{\mathbf{p}\in\mathcal{P}}\min_{\mathbf{u}\in\mathcal{U}}\log\left(\sum_{i=1}^{I} p_i\cdot e^{\alpha_i-\beta_i p_i+\sum_{j\neq i}\gamma_{i,j}p_j}\right)\\&=\max_{\mathbf{p}\in\mathcal{P}}\min_{\mathbf{u}\in\mathcal{U}}\log\left(\sum_{i=1}^{I} e^{\log p_i+\alpha_i-\beta_i p_i+\sum_{j\neq i}\gamma_{i,j}p_j}\right)\end{aligned}$$

Here, we can again use the log-sum-exp biconjugate technique to reformulate the objective function in the following way:

$$\begin{aligned}&\log\left(\sum_{i=1}^{I} e^{\log p_i+\alpha_i-\beta_i p_i+\sum_{j\neq i}\gamma_{i,j}p_j}\right)\\&=\max_{\boldsymbol{\mu}\in\Delta_{[I]}}\left\{\sum_{i=1}^{I}\mu_i\cdot(\log p_i+\alpha_i-\beta_i p_i+\sum_{j\neq i}\gamma_{i,j}p_j)-\sum_{i=1}^{I}\mu_i\log\mu_i\right\}.\end{aligned}$$

Thus, the overall problem becomes the following max-min-max problem:

$$\max_{\mathbf{p}\in\mathcal{P}}\min_{\mathbf{u}\in\mathcal{U}}\max_{\boldsymbol{\mu}\in\Delta_{[I]}}\left\{\sum_{i=1}^{I}\mu_i\cdot(\log p_i+\alpha_i-\beta_i p_i+\sum_{j\neq i}\gamma_{i,j}p_j)-\sum_{i=1}^{I}\mu_i\log\mu_i\right\}.$$

Here, we observe that the objective function is linear in $\mathbf{u}=(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma})$, and is concave in $\boldsymbol{\mu}$; additionally, the feasible region of $\mathbf{u}$ is assumed to be convex, and the feasible region of $\boldsymbol{\mu}$ is convex and compact (being just the $(|I|-1)$-dimensional unit simplex). Therefore, we can

use Sion's minimax theorem to interchange the minimization over $\mathbf{u}$ and the maximization over $\boldsymbol{\mu}$, which gives us

$$\begin{aligned}
&\max_{\mathbf{p}\in\mathcal{P}} \min_{\mathbf{u}\in\mathcal{U}} \max_{\boldsymbol{\mu}\in\Delta_{[I]}} \left\{ \sum_{i=1}^{I} \mu_i \cdot (\log p_i + \alpha_i - \beta_i p_i + \sum_{j\neq i} \gamma_{i,j} p_j) - \sum_{i=1}^{I} \mu_i \log \mu_i \right\} \\
&= \max_{\mathbf{p}\in\mathcal{P}} \max_{\boldsymbol{\mu}\in\Delta_{[I]}} \min_{\mathbf{u}\in\mathcal{U}} \left\{ \sum_{i=1}^{I} \mu_i \cdot (\log p_i + \alpha_i - \beta_i p_i + \sum_{j\neq i} \gamma_{i,j} p_j) - \sum_{i=1}^{I} \mu_i \log \mu_i \right\} \\
&= \max_{\mathbf{p}\in\mathcal{P}, \boldsymbol{\mu}\in\Delta_{[I]}} \min_{\mathbf{u}\in\mathcal{U}} \left\{ \sum_{i=1}^{I} \mu_i \cdot (\log p_i + \alpha_i - \beta_i p_i + \sum_{j\neq i} \gamma_{i,j} p_j) - \sum_{i=1}^{I} \mu_i \log \mu_i \right\}
\end{aligned}$$

Under Assumption 8, this final problem can then be reformulated as robust mixed-integer exponential cone program, just as in Section 3.5.3. We introduce the same binary decision variable $x_{i,t}$ which is 1 if product i is offered at price $t \in \mathcal{P}_i$, and 0 otherwise, and we use $w_{i,j,t}$ to denote the linearization of $\mu_i \cdot x_{j,t}$ for $i, j \in [I]$, $t \in \mathcal{P}_j$. This gives rise to the following program:

$$\begin{aligned}
\underset{\boldsymbol{\mu},w,x}{\text{maximize}} \quad & \min_{\mathbf{u}\in\mathcal{U}} \left\{ \sum_{i=1}^{I} \mu_i \alpha_i + \sum_{i=1}^{I} \left(\sum_{t\in\mathcal{P}_i} \log t \cdot w_{i,i,t} - \beta_i \sum_{t\in\mathcal{P}_i} t \cdot w_{i,i,t} + \sum_{j\neq i} \gamma_{i,j} \sum_{t\in\mathcal{P}_j} t \cdot w_{i,j,t} \right) \right. \\
& \qquad \left. - \sum_{i=1}^{I} \mu_i \log \mu_i \right\} && \text{(B.7a)} \\
\text{subject to} \quad & \sum_{t\in\mathcal{P}_j} w_{i,j,t} = \mu_i, \quad \forall\ i \in [I],\ j \in [I], && \text{(B.7b)} \\
& \sum_{i=1}^{I} w_{i,j,t} = x_{j,t}, \quad \forall\ j \in [I],\ t \in \mathcal{P}_j, && \text{(B.7c)} \\
& \sum_{i=1}^{I} \mu_i = 1, && \text{(B.7d)} \\
& \sum_{t\in\mathcal{P}_i} x_{i,t} = 1, \quad \forall\ i \in [I], && \text{(B.7e)} \\
& w_{i,j,t} \geq 0, \quad \forall\ i \in [I],\ j \in [I], t \in \mathcal{P}_j, && \text{(B.7f)} \\
& x_{i,t} \in \{0,1\}, \quad \forall\ i \in [I],\ t \in \mathcal{P}_i, && \text{(B.7g)} \\
& \mu_i \geq 0, \quad \forall\ i \in [I]. && \text{(B.7h)}
\end{aligned}$$

Note that the feasible region of this problem is identical to that of problem (3.25), which appeared in our discussion of the separation problem for the RRPO problem when $\mathcal{U}$ is convex and $\mathcal{P}$ is finite. The difference here is that the objective is now a robust objective; it is the worst-case value of the objective of problem (3.25), taken over the convex uncertainty set $\mathcal{U}$. Depending on the structure of $\mathcal{U}$, the overall problem can remain in the mixed-integer convex program problem class. For example, if $\mathcal{U}$ is a polyhedron, then one can use LP duality to reformulate the robust problem exactly by introducing additional variables and constraints, as is normally done in robust optimization (Bertsimas and Sim 2004, Ben-Tal and Nemirovski 2000, Bertsimas et al. 2011). Similarly, if $\mathcal{U}$ is a second-order cone representable set, then one can again use conic duality to reformulate the problem. Alternatively, one can also consider solving the problem using a cutting plane method/delayed constraint generation approach, whereby one reformulates the program in epigraph form and then solves the inner minimization over $\mathbf{u}$ to identify new cuts to add (Bertsimas et al. 2016a).

B.2.2 Log-log model

For the log-log demand model, we can write the DRPO problem as

$$\begin{aligned}
&\max_{\mathbf{p}\in\mathcal{P}} \min_{\mathbf{u}\in\mathcal{U}} R(\mathbf{p},\mathbf{u}) \\
&= \max_{\mathbf{p}\in\mathcal{P}} \min_{\mathbf{u}\in\mathcal{U}} \sum_{i=1}^{I} p_i \cdot e^{\alpha_i - \beta_i \log p_i + \sum_{j\neq i} \gamma_{i,j} \log p_j} \\
&= \max_{\mathbf{p}\in\mathcal{P}} \min_{\mathbf{u}\in\mathcal{U}} \sum_{i=1}^{I} e^{\alpha_i + (1-\beta_i) \log p_i + \sum_{j\neq i} \gamma_{i,j} \log p_j}
\end{aligned}$$

Again, as with the semi-log model, solving the above problem is equivalent to solving the same problem with a log-transformed objective. Taking this log-transformed problem as our starting point, replacing the log-sum-exp function with its biconjugate and applying Sion's

minimax theorem gives us:

$$\begin{aligned}
&\max_{\mathbf{p}\in\mathcal{P}} \min_{\mathbf{u}\in\mathcal{U}} \log\left(\sum_{i=1}^{I} e^{\alpha_i+(1-\beta_i)\log p_i+\sum_{j\neq i}\gamma_{i,j}\log p_j}\right) \\
&= \max_{\mathbf{p}\in\mathcal{P}} \min_{\mathbf{u}\in\mathcal{U}} \max_{\boldsymbol{\mu}\in\Delta_{[I]}} \left\{\sum_{i=1}^{I}\left[\alpha_i\mu_i + \sum_{i=1}^{I}(1-\beta_i)\mu_i\cdot\log p_i + \sum_{j\neq i}\gamma_{i,j}\mu_i\cdot\log p_j\right] - \sum_{i=1}^{I}\mu_i\log\mu_i\right\} \\
&= \max_{\mathbf{p}\in\mathcal{P},\boldsymbol{\mu}\in\Delta_{[I]}} \min_{\mathbf{u}\in\mathcal{U}} \left\{\sum_{i=1}^{I}\left[\alpha_i\mu_i + \sum_{i=1}^{I}(1-\beta_i)\mu_i\cdot\log p_i + \sum_{j\neq i}\gamma_{i,j}\mu_i\cdot\log p_j\right] - \sum_{i=1}^{I}\mu_i\log\mu_i\right\}.
\end{aligned}$$

Under Assumption 8, this last problem can be re-written as the following robust version of problem (3.33), with the decision variables defined identically:

$$\begin{aligned}
\underset{\boldsymbol{\mu},w,x}{\text{maximize}} \quad & \min_{\mathbf{u}\in\mathcal{U}} \left\{\sum_{i=1}^{I}\mu_i\alpha_i + \sum_{i=1}^{I}\left(\sum_{t\in\mathcal{P}_i}\log t\cdot w_{i,i,t} - \beta_i\cdot\sum_{t\in\mathcal{P}_i}\log t\cdot w_{i,i,t}\right.\right. \\
&\qquad\left.\left. + \sum_{j\neq i}\gamma_{i,j}\sum_{t\in\mathcal{P}_j}\log t\cdot w_{i,j,t}\right) - \sum_{i=1}^{I}\mu_i\log\mu_i\right\} \qquad \text{(B.8a)} \\
\text{subject to} \quad & \text{constraints (3.25b) – (3.25h).} \qquad \text{(B.8b)}
\end{aligned}$$

Again, this problem has exactly the same feasible region as the log-log separation problem (3.33) and the semi-log separation problem (3.25). Additionally, just as with the deterministic robust problem (B.7) for the semi-log model, this problem can be further reformulated by exploiting the structure of $\mathcal{U}$, or otherwise one can design a cutting plane method that generates violated uncertain parameter vectors $\mathbf{u}\in\mathcal{U}$ on the fly.

B.3 Solution method for finite $\mathcal{P}$, finite $\mathcal{U}$

The second solution approach we consider is for the case where both $\mathcal{P}$ and $\mathcal{U}$ are finite sets. In particular, we assume that the uncertainty set $\mathcal{U}$ is a binary representable set. For fixed positive integers m and n, we let $\mathcal{U}$ be defined as

$$\mathcal{U} = \{\mathbf{u} = \mathbf{F}\mathbf{z} \mid \mathbf{A}\mathbf{z}\leq\mathbf{b}, \mathbf{z}\in\{0,1\}^n\}, \qquad \text{(B.9)}$$

where $\mathbf{b}$ is a m dimensional real vector, $\mathbf{A}$ is a m-by-n real matrix and $\mathbf{F}$ is a d-by-n real matrix, where d is the dimension of the uncertain parameter vector $\mathbf{u}$.

Recall that when $\mathcal{P}$ is finite, then the RRPO problem is

$$Z^*_{\text{RR}} = \max_{\boldsymbol{\pi} \in \Delta_{\mathcal{P}}} \min_{\mathbf{u} \in \mathcal{U}} \sum_{\mathbf{p} \in \mathcal{P}} \pi_{\mathbf{p}} R(\mathbf{p}, \mathbf{u}). \tag{B.10}$$

We can transform this problem into a dual problem where the outer problem is to optimize a distribution over uncertain parameter vectors, and the inner problem is to optimize over the price vector, as follows:

$$\begin{aligned} Z^*_{\text{RR}} &= \max_{\boldsymbol{\pi} \in \Delta_{\mathcal{P}}} \min_{\mathbf{u} \in \mathcal{U}} \sum_{\mathbf{p} \in \mathcal{P}} \pi_{\mathbf{p}} R(\mathbf{p}, \mathbf{u}) && \text{(B.11)} \\ &= \max_{\boldsymbol{\pi} \in \Delta_{\mathcal{P}}} \min_{\boldsymbol{\lambda} \in \Delta_{\mathcal{U}}} \sum_{\mathbf{p} \in \mathcal{P}} \sum_{\mathbf{u} \in \mathcal{U}} \pi_{\mathbf{p}} \lambda_{\mathbf{u}} R(\mathbf{p}, \mathbf{u}) && \text{(B.12)} \\ &= \min_{\boldsymbol{\lambda} \in \Delta_{\mathcal{U}}} \max_{\boldsymbol{\pi} \in \Delta_{\mathcal{P}}} \sum_{\mathbf{p} \in \mathcal{P}} \sum_{\mathbf{u} \in \mathcal{U}} \pi_{\mathbf{p}} \lambda_{\mathbf{u}} R(\mathbf{p}, \mathbf{u}) && \text{(B.13)} \\ &= \min_{\boldsymbol{\lambda} \in \Delta_{\mathcal{U}}} \max_{\mathbf{p} \in \mathcal{P}} \sum_{\mathbf{u} \in \mathcal{U}} \lambda_{\mathbf{u}} R(\mathbf{p}, \mathbf{u}), && \text{(B.14)} \end{aligned}$$

where the first equality follows because minimization of a function of $\mathbf{u}$ over the finite set $\mathcal{U}$ is the same as minimizing the expected value of that function over all probability mass functions supported on $\mathcal{U}$; the second equality follows by linear programming duality; and the final equality follows because maximization of a function of $\mathbf{p}$ over $\mathcal{P}$ is the same as maximizing the expected value of that function over all probability mass functions supported on $\mathcal{P}$. We refer to problem (B.10) as the *primal* problem and (B.14) as the *dual* problem.

Consider now the *restricted primal problem*, where we replace $\mathcal{P}$ with a subset $\hat{\mathcal{P}} \subseteq \mathcal{P}$ in problem (B.10), and the *restricted dual problem*, where we replace $\mathcal{U}$ with a subset $\hat{\mathcal{U}} \subseteq \mathcal{U}$ in problem (B.14). Let us denote the objective values of these two problems with $Z_{P,\hat{\mathcal{P}}}$ and

$Z_{D,\hat{\mathcal{U}}}$, respectively. These two problems are:

$$Z_{P,\hat{\mathcal{P}}} = \max_{\boldsymbol{\pi} \in \Delta_{\hat{\mathcal{P}}}} \min_{\mathbf{u} \in \mathcal{U}} \sum_{\mathbf{p} \in \hat{\mathcal{P}}} \pi_{\mathbf{p}} R(\mathbf{p}, \mathbf{u}), \tag{B.15}$$

$$Z_{D,\hat{\mathcal{U}}} = \min_{\boldsymbol{\lambda} \in \Delta_{\hat{\mathcal{U}}}} \max_{\mathbf{p} \in \mathcal{P}} \sum_{\mathbf{u} \in \hat{\mathcal{U}}} \lambda_{\mathbf{u}} R(\mathbf{p}, \mathbf{u}). \tag{B.16}$$

Observe that $Z_{P,\hat{\mathcal{P}}}$ and $Z_{D,\hat{\mathcal{U}}}$ bound Z^*_{RR} from below and above, that is,

$$Z_{P,\hat{\mathcal{P}}} \leq Z^*_{\text{RR}} \leq Z_{D,\hat{\mathcal{U}}}.$$

In the above, the justification for the first inequality is because maximizing over distributions supported on the smaller set of price vectors $\hat{\mathcal{P}}$ cannot result in a higher worst-case objective than solving the full primal problem with $\mathcal{P}$, which gives the value Z^*_{RR}. The second inequality similarly follows because minimizing over distributions supported on the smaller set of uncertainty realizations $\hat{\mathcal{U}}$ cannot result in a lower worst-case objective than solving the full dual problem with $\mathcal{U}$, which gives Z^*_{RR}.

The idea of double column generation is as follows. Let us pick some subset of price vectors $\hat{\mathcal{P}} \subseteq \mathcal{P}$ and some subset of uncertainty realizations $\hat{\mathcal{U}} \subseteq \mathcal{U}$. Observe that the restricted primal problem (B.15) for $\hat{\mathcal{P}}$ can be written in epigraph form as

$$\begin{aligned}
\underset{\boldsymbol{\pi}, t}{\text{maximize}} \quad & t && \text{(B.17a)}\\
\text{subject to} \quad & t \leq \sum_{\mathbf{p} \in \hat{\mathcal{P}}} \pi_{\mathbf{p}} R(\mathbf{p}, \mathbf{u}), \quad \forall\, \mathbf{u} \in \mathcal{U}, && \text{(B.17b)}\\
& \sum_{\mathbf{p} \in \hat{\mathcal{P}}} \pi_{\mathbf{p}} = 1, && \text{(B.17c)}\\
& \pi_{\mathbf{p}} \geq 0, \quad \forall\, \mathbf{p} \in \hat{\mathcal{P}}. && \text{(B.17d)}
\end{aligned}$$

This problem has a huge number of constraints (one for each $\mathbf{u} \in \mathcal{U}$). However, we can solve it using delayed constraint generation, starting from the set $\hat{\mathcal{U}}$. Upon solving it in this way, at termination we will have a subset $\mathcal{U}'$ of uncertainty realizations from $\mathcal{U}$ that were found during the constraint generation process. We update $\hat{\mathcal{U}}$ to be equal to $\mathcal{U}'$.

With this (updated) subset $\hat{\mathcal{U}}$ in hand, we now solve the restricted dual problem (B.16) for $\hat{\mathcal{U}}$. This problem can be written in epigraph form as

$$\begin{aligned}
\underset{\boldsymbol{\lambda},\rho}{\text{minimize}} \quad & \rho && \text{(B.18a)}\\
\text{subject to} \quad & \rho \geq \sum_{\mathbf{u}\in\hat{\mathcal{U}}} \lambda_{\mathbf{u}} R(\mathbf{p},\mathbf{u}), \quad \forall\ \mathbf{p}\in\mathcal{P}, && \text{(B.18b)}\\
& \sum_{\mathbf{u}\in\hat{\mathcal{U}}} \lambda_{\mathbf{u}} = 1, && \text{(B.18c)}\\
& \lambda_{\mathbf{u}} \geq 0, \quad \forall\ \mathbf{u}\in\hat{\mathcal{U}}. && \text{(B.18d)}
\end{aligned}$$

This problem also has a huge number of constraints, but again we can solve it using delayed constraint generation, with the initial subset of price vectors set to $\hat{\mathcal{P}}$. At termination, we will have a new subset $\mathcal{P}'$ of price vectors, which will contain the original set of price vectors in $\hat{\mathcal{P}}$. We then update $\hat{\mathcal{P}}$ to $\mathcal{P}'$, and go back to solving the restricted primal problem. The process then repeats: after solving the restricted primal, we will have a new (bigger) $\hat{\mathcal{U}}$; we then solve the restricted dual, after which we have a new (bigger) $\hat{\mathcal{P}}$; we then go back to the restricted primal, and so on. After each iteration of solving the restricted primal and restricted dual, the set $\hat{\mathcal{P}}$ expands and the set $\hat{\mathcal{U}}$ expands. Thus, the bounds $Z_{P,\hat{\mathcal{P}}}$ and $Z_{D,\hat{\mathcal{U}}}$ get closer and closer to Z^*_{RR}. The algorithm can then be terminated either when $Z_{P,\hat{\mathcal{P}}} = Z_{D,\hat{\mathcal{U}}}$, which would imply that both restricted primal and restricted dual objective values exactly coincide with Z^*_{RR}; or otherwise, one can terminate when $Z_{D,\hat{\mathcal{U}}} - Z_{P,\hat{\mathcal{P}}} < \epsilon$, where $\epsilon > 0$ is a user specified tolerance.

The overall algorithmic approach is formalized as Algorithm 3. This algorithm invokes two procedures, PrimalCG (Algorithm 4) and DualCG (Algorithm 5), which are delayed constraint generation algorithms for solving the restricted primal and dual problems respectively. We note that Algorithm 3 is an adaptation of the double column generation algorithm of Wang et al. (2024) for the randomized robust assortment optimization problem, which is itself adapted from the double column generation algorithm of Delage and Saif (2022) for solving mixed-integer distributionally robust optimization problems. The proof of

correctness of this procedure follows similarly to Delage and Saif (2022), and is omitted for brevity. The novelty in our approach lies in how we handle the separation problems which are at the heart of PrimalCG and DualCG, which we discuss next.

Algorithm 3 Double column generation method for solving the finite $\mathcal{P}$, finite $\mathcal{U}$ RRPO problem.

1: Initialize $\hat{\mathcal{P}}$ to be a non-empty subset of $\mathcal{P}$, and $\hat{\mathcal{U}}$ to be a non-empty subset of $\mathcal{U}$.
2: Set LB $\leftarrow -\infty$, UB $\leftarrow +\infty$
3: **repeat**
4: Run PrimalCG($\hat{\mathcal{P}}$,$\hat{\mathcal{U}}$) to solve the restricted primal problem with $\hat{\mathcal{P}}$ and with $\hat{\mathcal{U}}$ as the initial uncertainty set. Let the objective value be $Z_{P,\hat{\mathcal{P}}}$ and the new uncertainty set be $\mathcal{U}'$.
5: Set $\hat{\mathcal{U}} \leftarrow \mathcal{U}'$.
6: Set LB $\leftarrow Z_{P,\hat{\mathcal{P}}}$.
7: Run DualCG($\hat{\mathcal{P}}$,$\hat{\mathcal{U}}$) to solve the restricted dual problem with $\hat{\mathcal{U}}$ and with $\hat{\mathcal{P}}$ as the initial price vector set. Let the objective value be $Z_{D,\hat{\mathcal{P}}}$ and the new price vector set be $\mathcal{P}'$.
8: Set $\hat{\mathcal{P}} \leftarrow \mathcal{P}'$.
9: Set UB $\leftarrow Z_{D,\hat{\mathcal{U}}}$.
10: **until** UB $-$ LB $< \epsilon$

Note that the doubly restricted primal and dual problems (B.19) and (B.21) solved in PrimalCG and DualCG are both linear programs, and can be thus be solved easily. The principal difficulty in these procedures comes from the primal and dual separation problems (B.20) and (B.22), which require optimizing over a price vector $\mathbf{p} \in \mathcal{P}$ and an uncertain parameter vector $\mathbf{u} \in \mathcal{U}$ respectively. In the following sections, we discuss how these two separation problems can be tackled for the linear, semi-log and log-log demand models. Note that in all three sections, we continue to make Assumption 8, which states that $\mathcal{P}$ can be written as the Cartesian product of finite sets of prices for each product, i.e.,

Algorithm 4 PRIMALCG procedure.

1: Initialize $\mathcal{U}' \leftarrow \hat{\mathcal{U}}$

2: **repeat**

3: Solve the doubly restricted primal problem:

$$\underset{\boldsymbol{\pi},t}{\text{maximize}} \quad t \tag{B.19a}$$

$$\text{subject to} \quad t \leq \sum_{\mathbf{p}\in\hat{\mathcal{P}}} \pi_{\mathbf{p}} R(\mathbf{p},\mathbf{u}), \quad \forall \mathbf{u} \in \mathcal{U}', \tag{B.19b}$$

$$\sum_{\mathbf{p}\in\hat{\mathcal{P}}} \pi_{\mathbf{p}} = 1, \tag{B.19c}$$

$$\pi_{\mathbf{p}} \geq 0, \quad \forall\ \mathbf{p} \in \hat{\mathcal{P}}. \tag{B.19d}$$

Let $(\boldsymbol{\pi}, t^*)$ be the optimal solution of the doubly restricted problem.

4: Solve the primal separation problem:

$$\min_{\mathbf{u}\in\mathcal{U}} \sum_{\mathbf{p}\in\hat{\mathcal{P}}} \pi_{\mathbf{p}} \cdot R(\mathbf{p},\mathbf{u}). \tag{B.20}$$

Let t' and $\mathbf{u}^*$ be the optimal objective value and solution of this separation problem.

5: **if** $t^* > t'$ **then**

6: Set $\mathcal{U}' \leftarrow \mathcal{U}' \cup \{\mathbf{u}^*\}$

7: **end if**

8: **until** $t^* \leq t'$

9: Set $Z_{P,\hat{\mathcal{P}}} \leftarrow t^*$

10: **return** $(Z_{P,\hat{\mathcal{P}}}, \mathcal{U}')$.

Algorithm 5 DUALCG procedure.

1: Initialize $\mathcal{P}' \leftarrow \hat{\mathcal{P}}$

2: **repeat**

3: Solve the doubly restricted dual problem:

$$\underset{\boldsymbol{\lambda},\rho}{\text{minimize}} \quad \rho \tag{B.21a}$$

$$\text{subject to} \quad \rho \geq \sum_{\mathbf{u}\in\hat{\mathcal{U}}} \lambda_{\mathbf{u}} R(\mathbf{p},\mathbf{u}), \quad \forall \mathbf{p} \in \mathcal{P}', \tag{B.21b}$$

$$\sum_{\mathbf{u}\in\hat{\mathcal{U}}} \lambda_{\mathbf{u}} = 1, \tag{B.21c}$$

$$\lambda_{\mathbf{u}} \geq 0, \quad \forall\ \mathbf{u} \in \hat{\mathcal{U}}. \tag{B.21d}$$

Let $(\boldsymbol{\lambda}, \rho^*)$ be the optimal solution of the doubly restricted problem.

4: Solve the dual separation problem:

$$\max_{\mathbf{p}\in\mathcal{P}} \sum_{\mathbf{u}\in\hat{\mathcal{U}}} \lambda_{\mathbf{u}} \cdot R(\mathbf{p},\mathbf{u}) \tag{B.22}$$

Let ρ' and $\mathbf{p}^*$ be the optimal objective value and solution of this separation problem.

5: **if** $\rho^* < \rho'$ **then**

6: Set $\mathcal{P}' \leftarrow \mathcal{P}' \cup \{\mathbf{p}^*\}$

7: **end if**

8: **until** $\rho^* \geq \rho'$

9: Set $Z_{D,\hat{\mathcal{U}}} \leftarrow \rho^*$

10: **return** $(Z_{D,\hat{\mathcal{U}}}, \mathcal{P}')$.

$\mathcal{P} = \mathcal{P}_1 \times \ldots \times \mathcal{P}_I$, where $\mathcal{P}_1, \ldots, \mathcal{P}_I$ are finite sets.

B.3.1 Primal and dual subproblems for linear demand model

For the linear demand model, the primal separation problem is

$$\min_{\mathbf{u}\in\mathcal{U}} \sum_{\mathbf{p}\in\hat{\mathcal{P}}} \pi_{\mathbf{p}} \cdot \left[\sum_{i=1}^{I} p_i \cdot (\alpha_i - \beta_i p_i + \sum_{j\neq i} \gamma_{i,j} p_j) \right]. \tag{B.23}$$

Note that this objective function is linear in $\mathbf{u} = (\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$. Therefore, the whole problem can be expressed as

$$\begin{aligned}
\text{minimize} \quad & \sum_{\mathbf{p}\in\hat{\mathcal{P}}} \pi_{\mathbf{p}} \cdot \left[\sum_{i=1}^{I} p_i \cdot (\alpha_i - \beta_i p_i + \sum_{j\neq i} \gamma_{i,j} p_j) \right] && \text{(B.24a)} \\
\text{subject to} \quad & \mathbf{u} = \mathbf{F}\mathbf{z}, && \text{(B.24b)} \\
& \mathbf{A}\mathbf{z} \leq \mathbf{z}, && \text{(B.24c)} \\
& \mathbf{z} \in \{0,1\}^n, && \text{(B.24d)}
\end{aligned}$$

which is a mixed-integer linear program.

The dual separation problem is

$$\max_{\mathbf{p}\in\mathcal{P}} \sum_{\mathbf{u}\in\hat{\mathcal{U}}} \lambda_{\mathbf{u}} \cdot \left[\sum_{i=1}^{I} p_i(\alpha_i - \beta_i p_i + \sum_{j\neq i} \gamma_{i,j} p_j) \right]. \tag{B.25}$$

By introducing the same binary variables as in the separation problem (3.20) (the linear demand separation problem for the convex $\mathcal{U}$ setting), we obtain the following mixed-integer

linear program:

$$\underset{x,y}{\text{maximize}} \quad \sum_{\mathbf{u}\in\hat{\mathcal{U}}} \lambda_{\mathbf{u}} \cdot \left[\sum_{i=1}^{I} \sum_{t\in\mathcal{P}_i} \alpha_i \cdot t \cdot x_{i,t} + \sum_{i=1}^{I} \sum_{t\in\mathcal{P}_i} t \cdot \beta_i \cdot x_{i,t} + \sum_{i=1}^{I} \sum_{j\neq i} \sum_{t_1\in\mathcal{P}_i} \sum_{t_2\in\mathcal{P}_j} \gamma_{i,j} \cdot t_1 \cdot t_2 \cdot y_{i,j,t_1,t_2} \right] \tag{B.26a}$$

$$\text{subject to} \quad \sum_{t\in\mathcal{P}_i} x_{i,t} = 1, \quad \forall\ i \in [I], \tag{B.26b}$$

$$\sum_{t_2\in\mathcal{P}_j} y_{i,j,t_1,t_2} = x_{i,t_1}, \quad \forall\ i,j \in [I], j \neq i, t_2 \in \mathcal{P}_j, \tag{B.26c}$$

$$\sum_{t_1\in\mathcal{P}_i} y_{i,j,t_1,t_2} = x_{i,t_1}, \quad \forall\ i,j \in [I], j \neq i, t_1 \in \mathcal{P}_i, \tag{B.26d}$$

$$x_{i,t} \in \{0,1\}, \quad \forall\ i \in [I],\ t \in \mathcal{P}_i, \tag{B.26e}$$

$$x_{i,j,t_1,t_2} \in \{0,1\}, \quad \forall\ i,j \in [I], i \neq j, t_1 \in \mathcal{P}_i, t_2 \in \mathcal{P}_j. \tag{B.26f}$$

Importantly, note that the size of this problem does not scale with the number of uncertainty realizations inside $\hat{\mathcal{U}}$; the form of this problem is equivalent to problem (3.20) where $\mathbf{u}$ is replaced with $\sum_{\mathbf{u}\in\hat{\mathcal{U}}} \lambda_{\mathbf{u}} \cdot \mathbf{u}$ (the "average" uncertain demand parameter). As we will see in the next couple of sections, the same will not be true for the semi-log and log-log demand models.

B.3.2 Primal and dual subproblems for semi-log demand model

For the semi-log demand model, the primal separation problem is

$$\min_{\mathbf{u}\in\mathcal{U}} \sum_{\mathbf{p}\in\hat{\mathcal{P}}} \pi_{\mathbf{p}} \cdot \left[\sum_{i=1}^{I} p_i \cdot e^{\alpha_i - \beta_i p_i + \sum_{j\neq i} \gamma_{i,j} p_j} \right]. \tag{B.27}$$

Note that this objective function is convex in $\mathbf{u} = (\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$, because the weights $\pi_{\mathbf{p}}$ and p_i for a given $\mathbf{p} \in \mathcal{P}$ and $i \in [I]$ are nonnegative, and because the function $e^{\alpha_i - \beta_i p_i + \sum_{j\neq i} \gamma_{i,j} p_j}$ is

convex in $\mathbf{u} = (\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$. Thus, the whole problem can be expressed as

$$\underset{u,z}{\text{minimize}} \quad \sum_{\mathbf{p}\in\hat{\mathcal{P}}} \pi_{\mathbf{p}} \cdot \left[\sum_{i=1}^{I} p_i \cdot e^{\alpha_i - \beta_i p_i + \sum_{j\neq i} \gamma_{i,j} p_j} \right] \tag{B.28a}$$

$$\text{subject to} \quad \mathbf{u} = \mathbf{F}\mathbf{z}, \tag{B.28b}$$

$$\mathbf{A}\mathbf{z} \leq \mathbf{b}, \tag{B.28c}$$

$$\mathbf{z} \in \{0,1\}^n, \tag{B.28d}$$

which can be re-written as a mixed-integer exponential cone program.

The dual separation problem is

$$\max_{\mathbf{p}\in\mathcal{P}} \sum_{\mathbf{u}\in\hat{\mathcal{U}}} \lambda_{\mathbf{u}} \cdot \left[\sum_{i=1}^{I} p_i \cdot e^{\alpha_i - \beta_i p_i + \sum_{j\neq i} \gamma_{i,j} p_j} \right]. \tag{B.29}$$

The objective function of this problem is in general not concave in $\mathbf{p}$. However, just as in Section 3.5.3, the related problem of optimizing the logarithm of this objective, which is

$$\max_{\mathbf{p}\in\mathcal{P}} \log \left[\sum_{\mathbf{u}\in\hat{\mathcal{U}}} \lambda_{\mathbf{u}} \cdot \left[\sum_{i=1}^{I} p_i \cdot e^{\alpha_i - \beta_i p_i + \sum_{j\neq i} \gamma_{i,j} p_j} \right] \right] \tag{B.30}$$

$$= \max_{\mathbf{p}\in\mathcal{P}} \log \left[\sum_{\mathbf{u}\in\hat{\mathcal{U}}} \sum_{i=1}^{I} e^{\log \lambda_{\mathbf{u}} + \log p_i + \alpha_i - \beta_i p_i + \sum_{j\neq i} \gamma_{i,j} p_j} \right] \tag{B.31}$$

can be reformulated as a mixed-integer exponential cone program using the same biconjugate-based technique in Section 3.5.3. In particular, when Assumption 8 holds, then problem (B.31) is equivalent to

$$
\begin{aligned}
\underset{w,x,\boldsymbol{\mu}}{\text{maximize}} \quad & \sum_{\mathbf{u}\in\hat{\mathcal{U}}}\sum_{i=1}^{I}\mu_{\mathbf{u},i}\cdot(\log\lambda_{\mathbf{u}}+\alpha_i) \\
& +\sum_{\mathbf{u}\in\hat{\mathcal{U}}}\sum_{i=1}^{I}\sum_{t\in\mathcal{P}_i}\log t w_{\mathbf{u},i,i,t} \\
& +\sum_{\mathbf{u}\in\hat{\mathcal{U}}}\sum_{i=1}^{I}\sum_{t\in\mathcal{P}_i}(-\beta_i)\cdot t\cdot w_{\mathbf{u},i,i,t} \\
& +\sum_{\mathbf{u}\in\hat{\mathcal{U}}}\sum_{i=1}^{I}\sum_{j\neq i}\gamma_{i,j}\cdot\sum_{t\in\mathcal{P}_j} t\cdot w_{\mathbf{u},i,j,t} \\
& -\sum_{\mathbf{u}\in\hat{\mathcal{U}}}\sum_{i=1}^{I}\mu_{\mathbf{u},i}\log\mu_{\mathbf{u},i} && \text{(B.32a)} \\
\text{subject to} \quad & \sum_{\mathbf{u}\in\hat{\mathcal{U}}}\sum_{i=1}^{I}\mu_{\mathbf{u},i}=1, && \text{(B.32b)} \\
& \sum_{t\in\mathcal{P}_j} w_{\mathbf{u},i,j,t}=\mu_{\mathbf{u},i}, \quad \forall\ \mathbf{u}\in\hat{\mathcal{U}},\ i,j\in[I], && \text{(B.32c)} \\
& \sum_{\mathbf{u}\in\hat{\mathcal{U}}}\sum_{i=1}^{I} w_{\mathbf{u},i,j,t}=x_{j,t}, \quad \forall\ j\in[I],\ t\in\mathcal{P}_j, && \text{(B.32d)} \\
& \sum_{t\in\mathcal{P}_j} x_{j,t}=1, \quad \forall j\in[I], && \text{(B.32e)} \\
& w_{\mathbf{u},i,j,t}\geq 0, \quad \forall\ \mathbf{u}\in\hat{\mathcal{U}},\ i,j\in[I], t\in\mathcal{P}_j, && \text{(B.32f)} \\
& \mu_{\mathbf{u},i}\geq 0, \quad \forall\ \mathbf{u}\in\hat{\mathcal{U}},\ i\in[I], && \text{(B.32g)} \\
& x_{j,t}\in\{0,1\}, \quad \forall\ j\in[I],\ t\in\mathcal{P}_j, && \text{(B.32h)}
\end{aligned}
$$

where $x_{j,t}$ is a binary decision variable that is 1 if product j is offered at price $t\in\mathcal{P}_j$, and 0 otherwise; $\mu_{\mathbf{u},i}$ is a nonnegative decision variable introduced as part of the biconjugate-based reformulation; and $w_{\mathbf{u},i,j,t}$ is a decision variable that represents the linearization of $\mu_{\mathbf{u},i}\cdot x_{j,t}$ for all $\mathbf{u}\in\hat{\mathcal{U}}$, $i,j\in[I]$, and $t\in\mathcal{P}_j$.

As with problem (3.25), this problem can be expressed as a mixed-integer exponential cone program. One notable difference between formulation (B.32) and formulation (3.25)

from earlier is that the number of decision variables and constraints is larger because the decision variable $\mu_{\mathbf{u},i}$ is introduced for every combination of an uncertainty realization in $\hat{\mathcal{U}}$ and each product i; thus, $\boldsymbol{\mu}$ represents a probability mass function over the set $\hat{\mathcal{U}} \times [I]$.

B.3.3 Primal and dual subproblems for log-log demand model

For the log-log demand model, the primal separation problem is

$$\min_{\mathbf{u}\in\mathcal{U}} \sum_{\mathbf{p}\in\hat{\mathcal{P}}} \pi_{\mathbf{p}} \cdot \sum_{i=1}^{I} p_i \cdot e^{\alpha_i - \beta_i \log p_i + \sum_{j\neq i} \gamma_{i,j} \log p_j}. \tag{B.33}$$

Note that the objective function is convex in $\mathbf{u} = (\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$; it is the nonnegative weighted combination of terms of the form $e^{\alpha_i - \beta_i \log p_i + \sum_{j\neq i} \gamma_{i,j} \log p_j}$, each of which are convex in $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$. Thus, the overall problem, which can be stated as

$$\begin{aligned}
\underset{z}{\text{minimize}} \quad & \sum_{\mathbf{p}\in\hat{\mathcal{P}}} \pi_{\mathbf{p}} \cdot \sum_{i=1}^{I} p_i \cdot e^{\alpha_i - \beta_i \log p_i + \sum_{j\neq i} \gamma_{i,j} \log p_j} && \text{(B.34a)} \\
\text{subject to} \quad & \mathbf{u} = \mathbf{F}\mathbf{z}, && \text{(B.34b)} \\
& \mathbf{A}\mathbf{z} \leq \mathbf{b}, && \text{(B.34c)} \\
& \mathbf{z} \in \{0,1\}^n, && \text{(B.34d)}
\end{aligned}$$

is a mixed-integer convex program, and can be expressed as a mixed-integer exponential cone program.

The dual separation problem is

$$\max_{\mathbf{p}\in\mathcal{P}} \sum_{\mathbf{u}\in\hat{\mathcal{U}}} \lambda_{\mathbf{u}} \cdot \left[\sum_{i=1}^{I} p_i \cdot e^{\alpha_i - \beta_i \log p_i + \sum_{j\neq i} \gamma_{i,j} \log p_j} \right]. \tag{B.35}$$

The objective function of this problem is in general not concave in $\mathbf{p}$. However, following the same method as in Section 3.5.4, we can reformulate the related problem of maximizing

the logarithm, which is

$$\max_{\mathbf{p}\in\mathcal{P}} \log \left(\sum_{\mathbf{u}\in\hat{\mathcal{U}}} \lambda_{\mathbf{u}} \cdot \left[\sum_{i=1}^{I} p_i \cdot e^{\alpha_i - \beta_i \log p_i + \sum_{j\neq i} \gamma_{i,j} \log p_j} \right] \right) \tag{B.36}$$

$$= \max_{\mathbf{p}\in\mathcal{P}} \log \left(\sum_{\mathbf{u}\in\hat{\mathcal{U}}} \sum_{i=1}^{I} e^{\log \lambda_{\mathbf{u}} + \log p_i + \alpha_i - \beta_i \log p_i + \sum_{j\neq i} \gamma_{i,j} \log p_j} \right) \tag{B.37}$$

as a mixed-integer exponential cone program. Under Assumption 8, the resulting formulation is

$$\begin{aligned}
\underset{x,w,\boldsymbol{\mu}}{\text{maximize}} \quad & \sum_{\mathbf{u}\in\hat{\mathcal{U}}} \sum_{i=1}^{I} \mu_{\mathbf{u},i} \cdot (\log \lambda_{\mathbf{u}} + \alpha_i) \\
& + \sum_{\mathbf{u}\in\hat{\mathcal{U}}} \sum_{i=1}^{I} \sum_{t\in\mathcal{P}_i} (1-\beta_i) \log t \cdot w_{\mathbf{u},i,i,t} \\
& + \sum_{\mathbf{u}\in\hat{\mathcal{U}}} \sum_{i=1}^{I} \sum_{j\neq i} \gamma_{i,j} \cdot \sum_{t\in\mathcal{P}_j} \log t \cdot w_{\mathbf{u},i,j,t} \\
& - \sum_{\mathbf{u}\in\hat{\mathcal{U}}} \sum_{i=1}^{I} \mu_{\mathbf{u},i} \log \mu_{\mathbf{u},i} && \text{(B.38a)} \\
\text{subject to} \quad & \sum_{\mathbf{u}\in\hat{\mathcal{U}}} \sum_{i=1}^{I} \mu_{\mathbf{u},i} = 1, && \text{(B.38b)} \\
& \sum_{t\in\mathcal{P}_j} w_{\mathbf{u},i,j,t} = \mu_{\mathbf{u},i}, \quad \forall\ \mathbf{u}\in\hat{\mathcal{U}},\ i,j\in[I], && \text{(B.38c)} \\
& \sum_{\mathbf{u}\in\hat{\mathcal{U}}} \sum_{i=1}^{I} w_{\mathbf{u},i,j,t} = x_{j,t}, \quad \forall\ j\in[I],\ t\in\mathcal{P}_j, && \text{(B.38d)} \\
& \sum_{t\in\mathcal{P}_j} x_{j,t} = 1, \quad \forall j\in[I], && \text{(B.38e)} \\
& w_{\mathbf{u},i,j,t} \geq 0, \quad \forall\ \mathbf{u}\in\hat{\mathcal{U}},\ i,j\in[I], t\in\mathcal{P}_j, && \text{(B.38f)} \\
& \mu_{\mathbf{u},i} \geq 0, \quad \forall\ \mathbf{u}\in\hat{\mathcal{U}},\ i\in[I], && \text{(B.38g)} \\
& x_{j,t}\in\{0,1\}, \quad \forall\ j\in[I],\ t\in\mathcal{P}_j, && \text{(B.38h)}
\end{aligned}$$

where the decision variables have the same meaning as those in formulation (B.32).

B.4 Additional numerical results

B.4.1 Estimation results for `orangeJuice` data set

Tables B.1 and B.2 display the point estimates of $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ for the semi-log and log-log demand models for the `orangeJuice` data set.

	Product										
Parameters	1	2	3	4	5	6	7	8	9	10	11
$\boldsymbol{\alpha}_{\text{semi-log}}$	9.873	9.829	8.598	9.504	9.024	9.828	8.582	7.901	7.152	11.161	10.896
$\boldsymbol{\beta}_{\text{semi-log}}$	1.0222	0.4581	1.2735	1.7888	1.3354	0.6507	1.6491	1.3945	2.0809	1.6290	0.0383
$\boldsymbol{\alpha}_{\text{log-log}}$	10.140	10.956	8.266	8.421	9.045	10.613	7.832	7.127	6.563	11.326	11.198
$\boldsymbol{\beta}_{\text{log-log}}$	2.7195	2.0410	3.3037	3.8855	2.9357	2.6101	3.6063	2.8209	3.9717	2.7942	0.1542

Table B.1: Estimation results for $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$.

B.4.2 Performance results for `orangeJuice` data set

Tables B.3 and B.4 below compare the performance of the nominal, deterministic robust and randomized robust pricing solutions under a discrete budget uncertainty set for the `orangeJuice` data set.

$\boldsymbol{\gamma}_{\text{semi-log}}$	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$	$j=6$	$j=7$	$j=8$	$j=9$	$j=10$	$j=11$
$i=1$	–	0.0571	0.0813	0.0966	0.0193	-0.0232	0.1305	0.1904	0.1490	0.0582	0.0815
$i=2$	0.1384	–	0.0041	0.0009	0.0204	0.0153	0.0090	0.1040	-0.0023	0.0491	0.0394
$i=3$	0.3386	0.0916	–	0.1943	0.0702	-0.0062	0.0051	0.0950	-0.0310	0.0690	0.0950
$i=4$	0.4313	0.0976	-0.1112	–	0.4089	0.3518	0.2085	-0.0777	-0.0352	0.0383	-0.2290
$i=5$	0.1916	0.0490	0.3026	0.2966	–	-0.1538	0.1547	-0.0314	0.1034	0.3338	0.0370
$i=6$	0.0211	0.0493	-0.0194	-0.0018	0.0888	–	0.0340	0.0472	-0.0167	0.0297	0.1119
$i=7$	0.2007	0.0388	0.0706	0.0672	0.3233	0.0837	–	0.0377	0.2216	-0.0504	0.1405
$i=8$	0.0117	0.0119	0.0932	0.0757	0.1023	-0.0160	0.1345	–	0.1372	0.2143	0.2699
$i=9$	0.0955	0.0373	-0.0211	0.3651	0.4176	0.0358	0.2127	0.1462	–	0.2337	0.1627
$i=10$	0.0412	-0.3941	0.0764	0.4867	0.4810	0.0109	-0.0814	-0.1047	0.0878	–	0.0274
$i=11$	-0.0893	-0.1587	-0.1358	-0.0252	-0.0690	0.0079	-0.0574	-0.1117	-0.1271	0.0809	–
$\boldsymbol{\gamma}_{\text{log-log}}$	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$	$j=6$	$j=7$	$j=8$	$j=9$	$j=10$	$j=11$
$i=1$	–	0.2196	0.1631	0.2129	0.0646	-0.0577	0.2576	0.3338	0.2494	0.0621	0.2939
$i=2$	0.3474	–	0.0403	0.0004	0.0338	0.0492	0.0193	0.1879	-0.0042	0.0739	0.1257
$i=3$	0.8673	0.5123	–	0.4400	0.1482	-0.0338	0.0527	0.1480	-0.1001	0.1267	0.3683
$i=4$	1.1581	0.3822	-0.2283	–	0.8367	1.2659	0.4495	-0.1569	-0.0115	0.1003	-0.7321
$i=5$	0.4624	0.2241	0.8344	0.6406	–	-0.6800	0.3223	0.0646	0.1426	0.5815	0.1782
$i=6$	0.0462	0.2424	-0.0343	-0.0173	0.2086	–	0.0975	0.1187	-0.0364	0.0561	0.4159
$i=7$	0.4644	0.2531	0.0971	0.1204	0.6997	0.2946	–	0.1728	0.4912	-0.0564	0.4497
$i=8$	0.0652	0.1430	0.1980	0.1587	0.2705	-0.0988	0.3198	–	0.3034	0.2992	0.9436
$i=9$	0.2971	-0.1190	-0.0216	0.7986	0.8825	0.3045	0.5869	0.1706	0.0	0.3171	0.4223
$i=10$	0.1406	-1.7987	0.1061	1.0453	1.0852	0.0811	-0.1213	-0.1760	0.0454	–	-0.0371
$i=11$	-0.2246	-0.7519	-0.3177	-0.0483	-0.1635	-0.0570	-0.1100	-0.1780	-0.2646	0.1341	–

Table B.2: Estimation results for $\boldsymbol{\gamma}$ for `orangeJuice` data set.

Γ	t_{RR}	Z^*_{RR}	$\mathbb{E}[R(\mathbf{p}^*_{\text{RR}}, \mathbf{u}_0)]$	t_{DR}	$\hat{Z}_{\text{DR}}$	RI(%)	$R(\hat{\mathbf{p}}_{\text{DR}}, \mathbf{u}_0)$	t_{N}	Z^*_{N}	$Z_{\text{N,WC}}$
5	14.64	162753.97	290939.28	225.36	102626.41	58.59	260321.17	0.81	590547.01	85304.36
10	6.29	70401.48	404458.22	208.42	47969.46	46.76	350396.25	–	–	38815.61
15	3.93	39567.50	349936.30	209.14	32757.43	20.79	334211.84	–	–	22798.44
20	16.70	31438.76	328664.19	197.37	25348.77	24.02	299970.20	–	–	15940.16

Table B.3: Results for `orangeJuice` pricing problem with semi-log demand and discrete $\mathcal{U}$.

Γ	t_{RR}	Z^*_{RR}	$\mathbb{E}[R(\mathbf{p}^*_{\text{RR}}, \mathbf{u}_0)]$	t_{DR}	$\hat{Z}_{\text{DR}}$	RI(%)	$R(\hat{\mathbf{p}}_{\text{DR}}, \mathbf{u}_0)$	t_{N}	Z^*_{N}	$Z_{\text{N,WC}}$
5	12.85	272399.89	605265.00	306.63	174478.12	56.12	811254.69	0.87	1110000.00	117186.58
10	10.41	135761.31	750084.92	260.96	77297.42	75.63	896972.12	–	–	50458.15
15	15.59	72930.45	761785.89	193.13	44914.20	62.38	896972.12	–	–	27389.15
20	8.56	45153.74	770505.32	190.76	27675.07	63.16	409330.70	–	–	17502.32

Table B.4: Results for `orangeJuice` pricing problem with log-log demand and discrete $\mathcal{U}$

Bibliography

E. Adida and G. Perakis. Dynamic pricing and inventory control: robust vs. stochastic uncertainty models—a computational study. *Annals of Operations Research*, 181(1):125–157, 2010.

İ. Akçakuş and V. V. Mišić. Exact logit-based product design. *Available at SSRN 3875986*, 2021.

Y. Akçay, H. P. Natarajan, and S. H. Xu. Joint dynamic pricing of multiple perishable products under consumer choice. *Management Science*, 56(8):1345–1361, 2010.

Yi-Chun Akchen and Velibor V Mišić. Column-randomized linear programs: Performance guarantees and applications. *Operations Research*, 2024.

A. Allouah, A. Bahamou, and O. Besbes. Optimal pricing with a single point. *arXiv preprint arXiv:2103.05611*, 2021.

A. Allouah, A. Bahamou, and O. Besbes. Pricing with samples. *Operations Research*, 70(2): 1088–1104, 2022.

L. Andersen and M. Broadie. Primal-dual simulation algorithm for pricing multidimensional American options. *Management Science*, 50(9):1222–1234, 2004.

MOSEK ApS. *The MOSEK optimization toolbox for C manual. Version 10.0.*, 2022. URL `http://docs.mosek.com/10.0/capi/index.html`.

G. Aydin and J. K. Ryan. Product line selection and pricing under the multinomial logit choice model. In *Proceedings of the 2000 MSOM conference*. Citeseer, 2000.

C. Bandi and D. Bertsimas. Robust option pricing. *European Journal of Operational Research*, 239 (3):842–853, 2014.

A. Ben-Tal and A. Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical programming*, 88(3):411–424, 2000.

A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust optimization*, volume 28. Princeton university press, 2009.

F. Bernstein and A. Federgruen. Pricing and replenishment strategies in a distribution system with competing retailers. *Operations Research*, 51(3):409–426, 2003.

D. Bertsimas and D. den Hertog. *Robust and adaptive optimization*. Dynamic Ideas LLC, 2022.

D. Bertsimas and N. Kallus. From predictive to prescriptive analytics. *Management Science*, 66 (3):1025–1044, 2020.

D. Bertsimas and V. V. Mišić. Robust product line design. *Operations Research*, 65(1):19–37, 2017.

D. Bertsimas and M. Sim. The price of robustness. *Operations research*, 52(1):35–53, 2004.

D. Bertsimas and A. Thiele. A robust optimization approach to inventory theory. *Operations research*, 54(1):150–168, 2006.

D. Bertsimas, D. B. Brown, and C. Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, 2011.

D. Bertsimas, I. Dunning, and M. Lubin. Reformulation versus cutting-planes for robust optimization: A computational study. *Computational Management Science*, 13:195–217, 2016a.

D. Bertsimas, E. Nasrabadi, and J. B. Orlin. On the power of randomization in network interdiction. *Operations Research Letters*, 44(1):114–120, 2016b.

O. Besbes and A. Zeevi. On the (surprising) sufficiency of linear models for dynamic pricing with demand learning. *Management Science*, 61(4):723–739, 2015.

J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 2017.

G. Bitran and R. Caldentey. An overview of pricing models for revenue management. *Manufacturing & Service Operations Management*, 5(3):203–229, 2003.

S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

D. B. Brown and J. E. Smith. Information relaxations and duality in stochastic dynamic programs: A review and tutorial. *Working paper*, 2022.

D. B. Brown, J. E. Smith, and P. Sun. Information relaxations and duality in stochastic dynamic programs. *Operations research*, 58(4-part-1):785–801, 2010.

E. J. Candes, X. Li, and M. Soltanolkotabi. Phase retrieval via wirtinger flow: Theory and algorithms. *IEEE Transactions on Information Theory*, 61(4):1985–2007, 2015.

F. Caro and J. Gallien. Clearance pricing optimization for a fast-fashion retailer. *Operations research*, 60(6):1404–1422, 2012.

J. F. Carriere. Valuation of the early-exercise price for derivative securities using simulations and splines. *Insurance: Mathematics and Economics*, 19(1):19–30, 1996.

M. Chen and Z.-L. Chen. Robust dynamic pricing with two substitutable products. *Manufacturing & Service Operations Management*, 20(2):249–268, 2018.

N. Chen and P. Glasserman. Additive and multiplicative duals for American option pricing. *Finance and Stochastics*, 11(2):153–179, 2007.

D. F. Ciocan and V. V. Mišić. Interpretable optimal stopping. *Management Science*, 68(3):1616–1638, 2022.

M. C. Cohen, N.-H. Z. Leung, K. Panchamgam, G. Perakis, and A. Smith. The impact of linear optimization on promotion planning. *Operations Research*, 65(2):446–468, 2017.

M. C. Cohen, R. Lobel, and G. Perakis. Dynamic pricing through data sampling. *Production and Operations Management*, 27(6):1074–1088, 2018.

E. Delage and A. Saif. The value of randomized solutions in mixed-integer distributionally robust optimization problems. *INFORMS Journal on Computing*, 34(1):333–353, 2022.

E. Delage, D. Kuhn, and W. Wiesemann. "dice"-sion–making under uncertainty: When can a random decision reduce risk? *Management Science*, 65(7):3282–3301, 2019.

V. V. Desai, V. F. Farias, and C. C. Moallemi. Pathwise optimization for optimal stopping problems. *Management Science*, 58(12):2292–2308, 2012.

L. Dong, P. Kouvelis, and Z. Tian. Dynamic pricing and inventory control of substitute products. *Manufacturing & Service Operations Management*, 11(2):317–339, 2009.

I. Dunning, J. Huchette, and M. Lubin. JuMP: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320, 2017.

A. N. Elmachtoub and P. Grigas. Smart "predict, then optimize". *Management Science*, 2021.

W. Elmaghraby and P. Keskinocak. Dynamic pricing in the presence of inventory considerations: Research overview, current practices, and future directions. *Management science*, 49(10): 1287–1309, 2003.

K. J. Ferreira, B. H. A. Lee, and D. Simchi-Levi. Analytics for an online retailer: Demand forecast-

ing and price optimization. *Manufacturing & Service Operations Management*, 18(1):69–88, 2016.

K. J. Ferreira, D. Simchi-Levi, and H. Wang. Online network revenue management using thompson sampling. *Operations research*, 66(6):1586–1602, 2018.

V. Gabrel, C. Murat, and A. Thiele. Recent advances in robust optimization: An overview. *European journal of operational research*, 235(3):471–483, 2014.

G. Gallego and H. Topaloglu. *Revenue management and pricing analytics*, volume 209. Springer, 2019.

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

M. R. Garey and D. S. Johnson. *Computers and intractability*. W. H. Freeman New York, 1979.

R. Ge, F. Huang, C. Jin, and Y. Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conference on learning theory*, pages 797–842. PMLR, 2015.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

A. Govindarajan, A. Sinha, and J. Uichanco. Distribution-free inventory risk pooling in a multilocation newsvendor. *Management Science*, 67(4):2272–2291, 2021.

Gurobi Optimization, Inc. Gurobi Optimizer Reference Manual, 2022. URL `http://www.gurobi.com`.

M. Hamzeei, A. Lim, and J. Xu. Robust price optimization of multiple products under interval uncertainties. *Journal of Revenue and Pricing Management*, pages 1–13, 2021.

W. Hanson and K. Martin. Optimizing multinomial logit profit functions. *Management Science*, 42(7):992–1003, 1996.

P. Harsha, S. Subramanian, and J. Uichanco. Dynamic pricing of omnichannel inventories. *Manufacturing & Service Operations Management*, 21(1):47–65, 2019.

M. B. Haugh and L. Kogan. Pricing American options: a duality approach. *Operations Research*, 52(2):258–270, 2004.

W. J. Hopp and X. Xu. Product line selection and pricing with modularity in design. *Manufacturing & Service Operations Management*, 7(3):172–187, 2005.

P. Jain and P. Kar. Non-convex optimization for machine learning. *Foundations and Trends® in Machine Learning*, 10(3-4):142–336, 2017.

S. Jasin and S. Kumar. A re-solving heuristic with bounded revenue loss for network revenue management with customer choice. *Mathematics of Operations Research*, 37(2):313–345, 2012.

K. Kalyanam. Pricing decisions under demand uncertainty: A bayesian mixture model approach. *Marketing Science*, 15(3):207–221, 1996.

P. W. Keller. *Tractable multi-product pricing under discrete choice models*. PhD thesis, Massachusetts Institute of Technology, 2013.

P. W. Keller, R. Levi, and G. Perakis. Efficient formulations for pricing under attraction demand models. *Mathematical Programming*, 145(1):223–261, 2014.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

P. Liang. CS229T/STAT231: Statistical Learning Theory (Winter 2016) Lecture Notes, 2018. URL `https://github.com/percyliang/cs229t/blob/master/lectures/notes.pdf`.

A. Lim, J. G. Shanthikumar, and T. Watewai. Robust multi-product pricing. *Available at SSRN 1078012*, 2008.

A. E. B. Lim and J. G. Shanthikumar. Relative entropy, exponential utility, and robust dynamic pricing. *Operations Research*, 55(2):198–214, 2007.

F. A. Longstaff and E. S. Schwartz. Valuing American options by simulation: a simple least-squares approach. *The Review of Financial Studies*, 14(1):113–147, 2001.

M. Lubin and I. Dunning. Computing in operations research using Julia. *INFORMS Journal on Computing*, 27(2):238–248, 2015.

T. Mai and P. Jaillet. Robust multi-product pricing under general extreme value models. *arXiv preprint arXiv:1912.09552*, 2019.

A. Mastin, P. Jaillet, and S. Chin. Randomized minmax regret for combinatorial optimization under uncertainty. In *International Symposium on Algorithms and Computation*, pages 491–501. Springer, 2015.

A. Maurer. A vector-contraction inequality for rademacher complexities. In *International Conference on Algorithmic Learning Theory*, pages 3–17. Springer, 2016.

J. I. McGill and G. J. van Ryzin. Revenue management: Research overview and prospects. *Transportation science*, 33(2):233–256, 1999.

V. V. Mišić. Optimization of tree ensembles. *Operations Research*, 68(5):1605–1624, 2020.

M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning*. MIT press, 2018.

A. L. Montgomery. Creating micro-marketing pricing strategies using supermarket scanner data. *Marketing science*, 16(4):315–337, 1997.

A. L. Montgomery and E. T. Bradlow. Why analyst overconfidence about the functional form of demand models can lead to overpricing. *Marketing Science*, 18(4):569–583, 1999.

J. R. Munkres. *Analysis on manifolds*. Addison-Wesley Publishing Company, 1991.

P. Netrapalli, P. Jain, and S. Sanghavi. Phase retrieval using alternating minimization. *IEEE Transactions on Signal Processing*, 63(18):4814–4826, 2015.

M. Okamoto. Distinctness of the eigenvalues of a quadratic form in a multivariate sample. *The Annals of Statistics*, pages 763–765, 1973.

G. Perakis and A. Sood. Competitive multi-period pricing for perishable products: A robust optimization approach. *Mathematical Programming*, 107(1):295–335, 2006.

D. J. Reibstein and H. Gatignon. Optimal product line pricing: The influence of elasticities and cross-elasticities. *Journal of marketing research*, 21(3):259–267, 1984.

R. T. Rockafellar. *Convex analysis*, volume 18. Princeton university press, 1970.

L. C. G. Rogers. Monte Carlo valuation of American options. *Mathematical Finance*, 12(3):271–286, 2002.

P. E. Rossi. `bayesm`: Bayesian inference for marketing/micro-econometrics, 2022. URL `http://CRAN.R-project.org/package=bayesm`. R package version 3.1-5.

P. Rusmevichientong and H. Topaloglu. Robust assortment optimization in revenue management under the multinomial logit choice model. *Operations research*, 60(4):865–882, 2012.

U. Sadana and E. Delage. The value of randomized strategies in distributionally robust risk-averse network interdiction problems. *INFORMS Journal on Computing*, 35(1):216–232, 2023.

A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on stochastic programming: modeling and theory*. SIAM, 2014.

M. Sion. On general minimax theorems. *Pacific Journal of Mathematics*, 8(4):171–176, 1958.

J.-S. J. Song, Z. X. Song, and X. Shen. Demand management and inventory control for substitutable products. *Available at SSRN 3866775*, 2021.

W. Soon. A review of multi-product pricing models. *Applied mathematics and computation*, 217 (21):8149–8165, 2011.

B. Sturt. A nonparametric algorithm for optimal stopping based on robust optimization. *arXiv preprint arXiv:2103.03300*, 2021a.

B. Sturt. The value of robust assortment optimization under ranking-based choice models. *Available at SSRN 3981736*, 2021b.

K. T. Talluri and G. J. van Ryzin. *The Theory and Practice of Revenue Management*. Kluwer Academic Publishers, 2004.

A. Thiele. Multi-product pricing via robust optimisation. *Journal of Revenue and Pricing Management*, 8(1):67–80, 2009.

J. N. Tsitsiklis and B. Van Roy. Regression methods for pricing complex American-style options. *IEEE Transactions on Neural Networks*, 12(4):694–703, 2001.

M. Udell and S. Boyd. Maximizing a sum of sigmoids. *Optimization and Engineering*, pages 1–25, 2013.

Zhengchao Wang, Heikki Peura, and Wolfram Wiesemann. Randomized assortment optimization. *Operations Research*, 2024.

M. J. Zenor. The profit benefits of category management. *Journal of Marketing Research*, 31(2): 202–213, 1994.

H. Zhang, P. Rusmevichientong, and H. Topaloglu. Multiproduct pricing under the generalized extreme value models with homogeneous price sensitivity parameters. *Operations Research*, 66(6):1559–1570, 2018.