Randomized Policy Optimization for Optimal Stopping

Xinyi Guan
UCLA Anderson School of Management, University of California, Los Angeles, California 90095, United States, xinyi.guan.phd@anderson.ucla.edu

Velibor V. Mišić
UCLA Anderson School of Management, University of California, Los Angeles, California 90095, United States, velibor.misic@anderson.ucla.edu

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 this paper, 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.

Key words: optimal stopping, approximate dynamic programming, randomization, non-convex optimization, option pricing.

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.

In this paper, we propose a new methodology for solving optimal stopping problems from data that is based on optimizing over the class of randomized linear policies. 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 paper is organized as follows. In Section 2, we review the relevant literature in optimal stopping, as well as other recent related work. In Section 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 4, we prove that our randomized policy SAA problem is consistent and develop our generalization guarantees. In Section 5, we show that our randomized policy SAA problem is NP-Hard, and present our backward optimization algorithm for solving it. In Section 6, we present the results of our numerical study on option pricing instances. Lastly, in Section 7, we conclude and discuss some potential directions for future research.

2. Literature Review

Our paper 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 (2021) 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 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 paper 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 paper 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 paper 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 paper, 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.

3. Problem Definition

In this section, we begin by defining our optimal stopping problem (Section 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 3.2) and given a sample of trajectories (Section 3.3). In Section 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 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.

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:

$$\tau_\pi=\min\{t\in[T]\mid\pi(t,\mathbf{x}(t))=\textbf{stop}\}. \tag{1}$$

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:

$$\operatorname*{supremum}_{\pi\in\Pi}\ \mathbb{E}[g(\tau_\pi,\mathbf{x}(\tau_\pi))]. \tag{2}$$

We make two important remarks regarding our optimal stopping problem (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 paper, 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 $t \in [T]$, $\mathbf{x} \in \mathcal{X}$, $0 \leq g(t, \mathbf{x}) \leq \bar{G}$.*

3.2. Deterministic linear policies

The optimal stopping problem (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:

$$\pi_{\mathbf{b}}(t, \mathbf{x}) = \begin{cases} \textbf{stop} & \text{if } \sum_{k=1}^{K} b_{t,k} \phi_k(\mathbf{x}(t)) > 0, \\ \textbf{continue} & \text{otherwise.} \end{cases} \tag{3}$$

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:

$$\operatorname*{supremum}_{\pi \in \Pi_{\mathcal{B}}} \mathbb{E}[g(\tau_\pi, \mathbf{x}(\tau_\pi))]. \tag{4}$$

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:

$$\operatorname*{supremum}_{\mathbf{b} \in \mathcal{B}} \mathbb{E}\left[\sum_{t=1}^{T} g(t, \mathbf{x}(t)) \cdot \prod_{t'=1}^{t-1} \mathbb{I}\{\mathbf{b}_{t'} \bullet \Phi(\mathbf{x}(t')) \leq 0\} \cdot \mathbb{I}\{\mathbf{b}_t \bullet \Phi(\mathbf{x}(t)) > 0\}\right], \tag{5}$$

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 (4). We also use $J_D(\mathbf{b})$ to denote the objective value of problem (5) at a fixed weight vector $\mathbf{b}$.

3.3. Data-driven optimization over deterministic linear policies

While problem (5) is a simplification of the general optimal stopping problem (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 (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

$$\operatorname*{supremum}_{\pi\in\Pi_{\mathcal{B}}} \frac{1}{\Omega}\sum_{\omega=1}^{\Omega} g(\tau_{\pi,\omega},\mathbf{x}(\omega,\tau_{\pi,\omega})). \tag{6}$$

Similarly to problem (5), we can re-write problem (6) as an optimization problem over $\mathbf{b}$ as follows:

$$\operatorname*{supremum}_{\mathbf{b}\in\mathcal{B}} \frac{1}{\Omega}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} g(t,\mathbf{x}(\omega,t))\cdot\prod_{t'=1}^{t-1}\mathbb{I}\{\mathbf{b}_{t'}\bullet\Phi(\mathbf{x}(\omega,t'))\le 0\}\cdot\mathbb{I}\{\mathbf{b}_t\bullet\Phi(\mathbf{x}(\omega,t))>0\}. \tag{7}$$

Note that the term $\prod_{t'=1}^{t-1}\mathbb{I}\{\mathbf{b}_{t'}\bullet\Phi(\mathbf{x}(\omega,t'))\le 0\}\cdot\mathbb{I}\{\mathbf{b}_t\bullet\Phi(\mathbf{x}(\omega,t))>0\}$ 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 (7) at a fixed weight vector $\mathbf{b}$.

By re-writing problem (6) as problem (7), we can see that the deterministic policy SAA problem (7) can be regarded as a type of discrete optimization problem over the weight vector $\mathbf{b}$. (Note that the supremum in problem (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 (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.

3.4. Randomized linear policies

Rather than solving problems (5) and (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

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

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

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:

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

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

$$\operatorname*{supremum}_{\tilde{\pi} \in \tilde{\Pi}_{\mathcal{B}}} \mathbb{E}[g(\tau_{\tilde{\pi}}, \mathbf{x}(\tau_{\tilde{\pi}}))], \tag{8}$$

or equivalently, as

$$\operatorname*{supremum}_{\mathbf{b} \in \mathcal{B}} \mathbb{E}\left[\sum_{t=1}^{T} g(t, \mathbf{x}(t)) \cdot \prod_{t'=1}^{t-1} (1 - \sigma(\mathbf{b}_{t'} \bullet \Phi(\mathbf{x}(t')))) \cdot \sigma(\mathbf{b}_t \bullet \Phi(\mathbf{x}(t)))\right], \tag{9}$$

where the expectation in problem (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 (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 (9). Given a sample of Ω trajectories as in Section 3.3, we can define the randomized policy SAA problem as

$$\operatorname*{supremum}_{\mathbf{b} \in \mathcal{B}} \frac{1}{\Omega} \sum_{\omega=1}^{\Omega} \sum_{t=1}^{T} g(t, \mathbf{x}(\omega, t)) \cdot \prod_{t'=1}^{t-1} (1 - \sigma(\mathbf{b}_{t'} \bullet \Phi(\mathbf{x}(\omega, t')))) \cdot \sigma(\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega, t))). \tag{10}$$

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 (10), the optimization problem is formulated using the supremum. This is necessary, because although the objective function of (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 $\mathbf{b} \in \mathcal{B}$.

3.5. Equivalence of deterministic and randomized policies

In this section, we investigate the connection between the deterministic policy problems laid out in Sections 3.2 and 3.3, and the randomized policy problems in Section 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 (7) and the randomized policy SAA problem (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 (5) and (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 (5), the deterministic policy SAA problem (7), the randomized policy true problem (9) and the randomized policy SAA problem (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* (7) *and* (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 Section EC.1.1 of the ecompanion) 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 (5) and (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 (9) and (5).

THEOREM 2. *Under Assumptions 2, 3 and 4 the objective values of the randomized problem* (9) *and the deterministic problem* (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 Section EC.1.2 of the ecompanion) 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 (9), we can obtain a policy that performs as well as the one we would obtain by solving the deterministic problem (5). Similarly, in the case when we are working with a finite sample of trajectories, solving the randomized SAA problem (10) allows us to obtain a policy that performs as well as the one we would obtain by solving the deterministic SAA problem (7). From a practical perspective, the advantage of solving the randomized policy SAA problem (10), as opposed to the deterministic policy SAA problem (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 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 (9) and the SAA problem (10) as differentiable or "soft" counterparts to the deterministic policy problems (5) and (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 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.

4. Statistical properties

In this section, we investigate the statistical properties of the randomized policy SAA problem (10). In Section 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 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.

4.1. Convergence of randomized policy SAA problem

It is natural to expect that the optimal value and optimal solutions of the SAA problem (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,*

$$\lim_{\Omega \to \infty} \sup_{\mathbf{b} \in \mathcal{B}} |\hat{J}_R(\mathbf{b}) - J_R(\mathbf{b})| = 0. \tag{11}$$

The proof of Theorem 3 is provided in Section EC.1.3 of the ecompanion. It relies on the fact that the objective function $J_R(\cdot)$ in the true problem (9) and the objective function $\hat{J}_R(\cdot)$ in the SAA problem (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,*

$$\lim_{\Omega \to \infty} \sup_{\mathbf{b} \in \mathcal{B}} \hat{J}_R(\mathbf{b}) = \sup_{\mathbf{b} \in \mathcal{B}} J_R(\mathbf{b}). \tag{12}$$

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 (9) while $\hat{\mathbf{B}}$ is the set of optimal solutions to the SAA problem (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 (10) and the closest optimal solution of the true problem (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 (10) can be arbitrarily close to the optimal weight vector set of the true problem (9), and the corresponding optimal value of the SAA problem can be arbitrarily close to the optimal value of true problem.

4.2. Rademacher Complexity

In Section 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

$$\Gamma(\mathbf{u}, \mathbf{v}) = \sum_{t=1}^{T} v_t \prod_{t'=1}^{t-1} (1 - \sigma(u_{t'}))\sigma(u_t), \tag{13}$$

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

$$\psi_{\mathbf{b}}(Y) = \begin{bmatrix} \mathbf{b}_1 \bullet \Phi(\mathbf{x}(1)) \\ \vdots \\ \mathbf{b}_T \bullet \Phi(\mathbf{x}(T)) \\ g(1, \mathbf{x}(1)) \\ \vdots \\ g(T, \mathbf{x}(T)) \end{bmatrix}. \tag{14}$$

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

$$\hat{R}(\mathcal{F}) = \frac{1}{\Omega}\mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{f\in\mathcal{F}}\sum_{\omega=1}^{\Omega}\epsilon_\omega f(Y_\omega)\right], \tag{15}$$

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:*

$$J_R(\mathbf{b}) \geq \hat{J}_R(\mathbf{b}) - 2R(\mathcal{F}) - \bar{G}\sqrt{\frac{\log(1/\delta)}{2\Omega}}, \qquad \forall\ \mathbf{b}\in\mathcal{B} \tag{16}$$

$$J_R(\mathbf{b}) \geq \hat{J}_R(\mathbf{b}) - 2\hat{R}(\mathcal{F}) - 3\bar{G}\sqrt{\frac{\log(2/\delta)}{2\Omega}}, \qquad \forall\ \mathbf{b}\in\mathcal{B} \tag{17}$$

The proof of Proposition 1 is given in Section EC.1.6 of the ecompanion; 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 Section EC.1.7 of the ecompanion) 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.

5. Solution Methodology

We now turn our attention to how one can actually solve the randomized policy SAA problem (10). In Section 5.1, we show that the randomized policy SAA problem is in general NP-Hard. Motivated by this, in Section 5.2 we propose an algorithm for approximately solving the SAA problem, based on backward induction. We conclude in Section 5.3 by comparing our proposed heuristic algorithm with the LSM algorithm.

5.1. Complexity of randomized policy SAA problem

Our main theoretical result on the solvability of the randomized policy SAA problem (10) is unfortunately a negative one.

THEOREM 6. *The randomized policy SAA problem* (10) *is NP-Hard.*

We make a few remarks about this result. First, our proof of Theorem 6 (see Section EC.2 of the ecompanion) involves considering the decision form of the randomized policy SAA problem (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 (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.

5.2. Backward optimization algorithm

Motivated by the fact that our randomized policy SAA problem (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 (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

$$\max_{\mathbf{b}'_t \in \mathcal{B}_t} \hat{J}_R(\mathbf{b}_1, \ldots, \mathbf{b}_{t-1}, \mathbf{b}'_t, \mathbf{b}_{t+1}, \ldots, \mathbf{b}_T) \tag{18}$$

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.

We pause to make several comments about Algorithm 1. First, observe that the period t problem solved in Algorithm 1, problem (20), is of a different form from problem (18). The two problems

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

$$p_t(\omega) = \prod_{t'=1}^{t-1} (1 - \sigma(\mathbf{b}_{t'} \bullet \Phi(\mathbf{x}(\omega, t')))). \tag{19}$$

Solve the problem

$$\max_{\mathbf{b}_t \in \mathcal{B}_t} \sum_{\omega=1}^{\Omega} \frac{1}{\Omega} \cdot p_t(\omega) \cdot [g(t, \mathbf{x}(\omega, t)) \cdot \sigma(\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega, t))) + c_t(\omega) \cdot (1 - \sigma(\mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega, t))))] \tag{20}$$

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

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

$$c_{t-1}(\omega) = g(t, \mathbf{x}(\omega, t)) \cdot \sigma(\mathbf{b}_t^* \bullet \Phi(\mathbf{x}(\omega, t))) + c_t(\omega) \cdot (1 - \sigma(\mathbf{b}_t^* \bullet \Phi(\mathbf{x}(\omega, t)))). \tag{21}$$

end for

are equivalent in that problem (20) is a simplification of problem (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 (18) down to problem (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 (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 (20) is more manageable to solve than the complete randomized policy SAA problem (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 (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 (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 (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 (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 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.

5.3. Comparison of Algorithm 1 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 (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

$$\min_{\mathbf{b}_t \in \mathbb{R}^K} \frac{1}{2} \sum_{\omega=1}^{\Omega} (c_t(\omega) - \mathbf{b}_t \bullet \Phi(\mathbf{x}(\omega, t)))^2 \tag{22}$$

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

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

$$c_{t-1}(\omega) = \begin{cases} c_t(\omega) & \text{if } \mathbf{b}_t^* \bullet \Phi(\mathbf{x}(\omega, t)) \geq g(t, \mathbf{x}(\omega, t)), \\ g(t, \mathbf{x}(\omega, t)) & \text{if } \mathbf{b}_t^* \bullet \Phi(\mathbf{x}(\omega, t)) < g(t, \mathbf{x}(\omega, t)). \end{cases} \tag{23}$$

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 3.2 and 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 (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 (5) or the deterministic policy SAA problem (7). In contrast, Algorithm 1 is designed to directly (albeit approximately) solve the randomized policy SAA problem (10). Our results in Sections 3 and 4 provide theoretical justification for why this approach is desirable: under mild conditions, the true randomized policy problem (9) and its SAA counterpart (10) are equivalent to the true deterministic policy problem (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 (10) converge to that of the true randomized policy problem (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 6, we will showcase one family of benchmark problem instances where this is indeed the case.

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 6.1. In Section 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 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).

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:

$$y(t) = \mathbb{I}\left\{\max_{1\leq i\leq n, 1\leq t^{'}\leq t} p_i(t^{'}) < B\right\} \tag{24}$$

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

$$g'(t) = y(t) \cdot \max\left\{0, \max_{1\leq i\leq n} p_i(t) - K\right\}. \tag{25}$$

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 (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 (20), we warm start the Adam algorithm using the coefficients obtained by LSM; we describe our warm starting scheme in more detail in Section EC.3.1 of the ecompanion.

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

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

$$g'(t) = y(t) \cdot \max\{0, p_1(t) - K\}, \tag{26}$$

$$y(t) = \mathbb{I}\left\{\max_{1 \le t' \le t} p_1(t') < B\right\}. \tag{27}$$

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

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

which is equivalent to

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

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

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

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:

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

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 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 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 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

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 1 **Out-of-sample performance of different policies in $n=1$ experiment.**

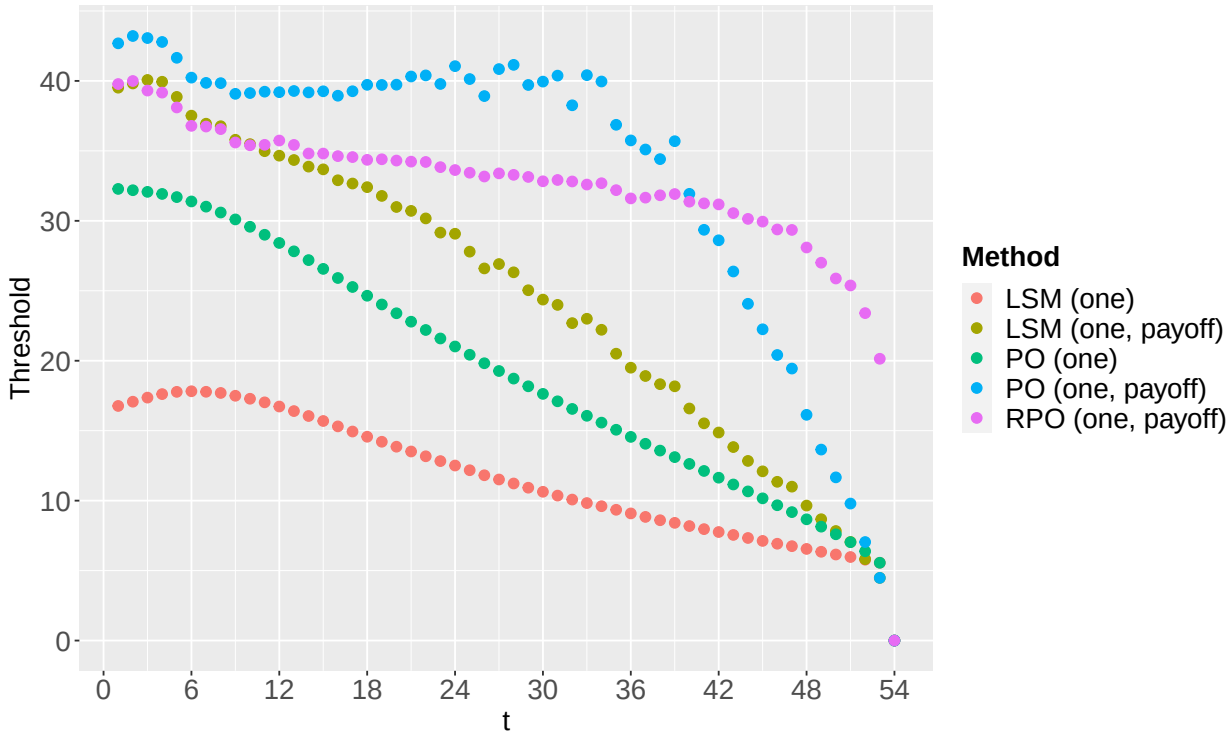

Figure 1 **Plot of thresholds for policies in $n=1$ experiment.**

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.

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 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 Section EC.3.2 of the ecompanion.

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 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 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

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 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 3 Computation time for different policies, for $n=8$ assets.

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 (20) using Adam, but does not include the time to obtain the initial starting point using LSM.

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, 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 (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.

7. Conclusion

In this paper, 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 directions for future work. 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.

References

İ. Akçakuş and V. V. Mišić. Exact logit-based product design. *Available at SSRN 3875986*, 2021.

L. Andersen and M. Broadie. Primal-dual simulation algorithm for pricing multidimensional American options. *Management Science*, 50(9):1222–1234, 2004.

D. Bertsimas and N. Kallus. From predictive to prescriptive analytics. *Management Science*, 66(3):1025–1044, 2020.

J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 2017.

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.

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.

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.

V. V. Desai, V. F. Farias, and C. C. Moallemi. Pathwise optimization for optimal stopping problems. *Management Science*, 58(12):2292–2308, 2012.

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.

K. J. Ferreira, B. H. A. Lee, and D. Simchi-Levi. Analytics for an online retailer: Demand forecasting and price optimization. *Manufacturing & Service Operations Management*, 18(1):69–88, 2016.

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 pointsonline 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.

Gurobi Optimization, Inc. Gurobi Optimizer Reference Manual, 2022. URL `http://www.gurobi.com`.

M. B. Haugh and L. Kogan. Pricing American options: a duality approach. *Operations Research*, 52(2): 258–270, 2004.

P. Jain and P. Kar. Non-convex optimization for machine learning. *Foundations and Trends® in Machine Learning*, 10(3-4):142–336, 2017.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

P. Liang. CS229T/STAT231: Statistical Learning Theory (Winter 2016) Lecture Notes, 2018. URL `https://github.com/percyliang/cs229t/blob/master/lectures/notes.pdf`.

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.

A. Maurer. A vector-contraction inequality for rademacher complexities. In *International Conference on Algorithmic Learning Theory*, pages 3–17. Springer, 2016.

M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning*. MIT press, 2018.

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.

L. C. G. Rogers. Monte Carlo valuation of American options. *Mathematical Finance*, 12(3):271–286, 2002.

A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on stochastic programming: modeling and theory*. SIAM, 2014.

B. Sturt. A nonparametric algorithm for optimal stopping based on robust optimization. *arXiv preprint arXiv:2103.03300*, 2021.

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.

Electronic companion for "Randomized Policy Optimization for Optimal Stopping"

EC.1. Proofs

EC.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

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

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

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

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 (EC.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

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

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. □

EC.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

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

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

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

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

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

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

$$\lim_{\alpha \to +\infty} \sigma(\tilde{\mathbf{b}}_t \bullet \Phi(\mathbf{x}(t)))$$

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

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

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

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 (EC.7) converge pointwise to the function of $(\mathbf{x}(1), \ldots, \mathbf{x}(T))$ whose expectation defines $J_D(\mathbf{b})$ in (EC.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

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

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 (EC.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:

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

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

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

as required. □

EC.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 ecompanion, 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 EC.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 $L_w=K_fL_h+K_hL_f$.*

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

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

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 EC.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 EC.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

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

We thus have

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

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

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

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 EC.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 EC.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 EC.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 EC.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 EC.1 to assert that $H_t(\cdot)$ is Lipschitz continuous with constant $(t-1)Q \cdot 1 + Q \cdot 1 = tQ$. □

LEMMA EC.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 EC.4: Let $\mathbf{b}, \bar{\mathbf{b}} \in \mathcal{B}$. We have

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

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

where the first inequality is just the triangle inequality; the second inequality follows by applying Lemmas EC.3, EC.2 and EC.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 EC.5. *The function $J_R(\cdot)$ is Lipschitz continuous with Lipschitz constant $L = \bar{G}T^2Q$.*

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

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

as required. □

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

THEOREM EC.1. *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 $\Omega \geq N$,*

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

Proof of Theorem EC.1: For the given ϵ, set $\delta = \epsilon/(3L)$ where $L = \bar{G}T^2Q$ 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 EC.4 and EC.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 EC.1, 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

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

The event in (EC.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 EC.1. Therefore, event (EC.17) occurs with probability 1, which establishes the required result. □

EC.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. □

EC.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 $\Omega \to \infty$.□

EC.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

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

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

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 the results from Step 1 and Step 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,

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

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, (EC.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 (16).

To establish inequality (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

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

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

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

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

This is equivalent to

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

which is exactly inequality (17). □

EC.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 EC.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 EC.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:

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

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

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

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

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

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

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

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

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

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

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

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

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

$$
\begin{aligned}
&\leq \sqrt{p_1 A_1^2 + \ldots p_T A_T^2} + \sqrt{p_1 + \cdots + p_T} \\
&\leq \sqrt{p_1 \bar{G}^2 + \ldots p_T \bar{G}^2} + \sqrt{p_1 + \cdots + p_T} \\
&\leq \sqrt{\bar{G}^2} + \sqrt{1} \\
&= \bar{G} + 1,
\end{aligned}
$$

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 EC.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*

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

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

LEMMA EC.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*

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

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:

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

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

where the first inequality follows by Lemma EC.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. $\square$

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 EC.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

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

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

LEMMA EC.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

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

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

where $\mathbf{V}$ is defined as

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

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

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

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 (EC.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

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

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

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

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 (EC.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. □

EC.2. 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 $t \in \{1, 2, 3\}$:

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

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:

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

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

$$\begin{aligned} & y_3 + (\neg y_3) \cdot (\neg y_4) + (\neg y_3) \cdot (\neg\neg y_4) \cdot y_7 \\ & = y_3 + (1 - y_3)(1 - y_4) + (1 - y_3)(y_4) y_7. \end{aligned} \tag{EC.35}$$

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 (EC.35) – are equivalent. The term in the square brackets in (EC.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 (EC.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 ⇒ 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:

$$b_i = \begin{cases} +\alpha \text{ if } y_i = 1, \\ -\alpha \text{ if } y_i = 0. \end{cases} \tag{EC.36}$$

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 (EC.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 (EC.36)) such that $\hat{J}_R(\mathbf{b}) \geq W - 1/2$.

Randomized policy SAA answer is yes $\Rightarrow$ *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

$$\hat{J}_R(\mathbf{b}) = \mathbb{E}\left[\sum_{m=1}^{M} c_m(\mathbb{I}\{\xi_1 \leq b_1\}, \ldots, \mathbb{I}\{\xi_N \leq b_N\})\right], \tag{EC.37}$$

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 (EC.37), we argue that for any clause m,

$$\begin{aligned}&\mathbb{E}[c_m(\mathbb{I}\{\xi_1 \leq b_1\}, \ldots, \mathbb{I}\{\xi_N \leq b_N\})] \\ &= \sigma(a_{m,1}b_{i_{m,1}}) + (1-\sigma(a_{m,1}b_{i_{m,1}}))\sigma(a_{m,2}b_{i_{m,2}}) + (1-\sigma(a_{m,1}b_{i_{m,1}}))(1-\sigma(a_{m,2}b_{i_{m,2}}))\sigma(a_{m,3}b_{i_{m,3}}).\end{aligned} \tag{EC.38}$$

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 (EC.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

$$\mathbb{P}(\xi_3 \leq b_3) + (1 - \mathbb{P}(\xi_3 \leq b_3))(1 - \mathbb{P}(\xi_4 \leq b_4)) + (1 - \mathbb{P}(\xi_3 \leq b_3)) \cdot \mathbb{P}(\xi_4 \leq b_4) \cdot \mathbb{P}(\xi_7 \leq b_7) \tag{EC.39}$$

$$= \mathbb{E}[\mathbb{I}\{\xi_3 \leq b_3\}] + \mathbb{E}[1 - \mathbb{I}\{\xi_3 \leq b_3\}]\mathbb{E}[1 - \mathbb{I}\{\xi_4 \leq b_4\}] + \mathbb{E}[1 - \mathbb{I}\{\xi_3 \leq b_3\}]\mathbb{E}[\mathbb{I}\{\xi_4 \leq b_4\}]\mathbb{E}[\mathbb{I}\{\xi_7 \leq b_7\}]$$

$$= \mathbb{E}[\mathbb{I}\{\xi_3 \leq b_3\} + (1 - \mathbb{I}\{\xi_3 \leq b_3\})(1 - \mathbb{I}\{\xi_4 \leq b_4\}) + (1 - \mathbb{I}\{\xi_3 \leq b_3\})\mathbb{I}\{\xi_4 \leq b_4\}\mathbb{I}\{\xi_7 \leq b_7\}], \tag{EC.40}$$

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 (EC.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 (EC.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 (EC.38), can be used to turn the right-hand side of (EC.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 (EC.38) holds, by linearity of expectation it must be the case that (EC.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:

$$\mathbb{E}[\sum_{m=1}^{M} c_m(\mathbb{I}\{\xi_1 \leq b_1\}, \ldots, \mathbb{I}\{\xi_N \leq b_N\})]$$

$$\leq \sum_{m=1}^{M} c_m(\mathbb{I}\{\xi_1' \leq b_1\}, \ldots, \mathbb{I}\{\xi_N' \leq b_N\}). \tag{EC.41}$$

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 (EC.41) and (EC.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. □

EC.3. Additional numerical results

EC.3.1. 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 (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 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 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 (20).

EC.3.2. Additional policy performance results for Section 6.3

Table EC.1 displays the results comparing LSM, PO and RPO for instances with $n = 4$ assets, while Table EC.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 EC.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 EC.2 Out-of-sample performance for different policies, for $n = 16$ assets.