Randomized Policy Optimization for Optimal Stopping

Xinyi Guan
Department of Logistics and Maritime Studies, Hong Kong Polytechnic University, Hong Kong, China,
xinyi.guan@polyu.edu.hk

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 in two ways: first, we establish that under mild conditions, given a fixed basis function architecture, optimizing over randomized linear policies is equivalent to optimizing over deterministic linear policies; and second, we theoretically construct simple parameterized families of problem instances where the popular least squares Monte Carlo approach can be made to perform arbitrarily poorly, whereas the optimal deterministic policy is either exactly optimal or arbitrarily close to optimal. We formulate the problem of learning randomized linear policies from data as a smooth non-convex sample average approximation (SAA) problem. We show that the SAA problem is in general NP-Hard, and consequently develop a practical heuristic for solving it based on alternating maximization. We numerically demonstrate the value of our method through a rich set of experiments, involving a standard Bermudan max-call option pricing benchmark, a more complex problem involving fractional Brownian motion, and a stylized exit-timing problem calibrated using real cryptocurrency data.

Key words: optimal stopping, randomization, non-convex optimization, alternating maximization, option pricing, cryptocurrencies

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.

High-dimensional optimal stopping problems can in theory be solved exactly by dynamic programming. This approach involves obtaining 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 continuation value function. However, this approach is often untenable in practice due to the curse of dimensionality.

As a result, many approaches have been proposed based on approximate dynamic programming (ADP), wherein one considers a policy that is greedy with respect to an approximate 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 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 because the conditional expectation function $h(x) = \mathbb{E}[Y \mid X = x]$ minimizes squared error, i.e., it solves the optimization problem $\min_f \mathbb{E}[(Y - f(X))^2]$. In such an idealized situation, the LSM policy 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, in the LSM policy, 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 even 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 policy's expected reward. 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.

2. **Worst-case analysis of LSM**: We undertake a theoretical comparison of LSM and the deterministic linear policy optimization approach in an idealized setting, where one considers an infinite sample. Through two separate families of parameterized instances, we show that the optimality gap of LSM can be arbitrarily large when: (i) LSM is equipped with only the constant basis function; and (ii) LSM is equipped with the constant basis function and the current reward as its only two basis functions. In those same instances, we show that optimizing over deterministic linear policies either exactly recovers the optimal policy for (i), or yields a policy that is arbitrarily close to optimal for (ii). To the best of our knowledge, we believe these results to be the first to characterize the worst-case performance of LSM.
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 novel heuristic algorithm for solving the problem, based on applying alternating maximization to a biconjugate representation of the objective function that is marginally concave (but not jointly concave) in two different sets of decision variables. We show that under certain conditions, the iterates produced by this algorithm converge in objective value to a deterministic linear policy.
4. **Numerical experiments**: Using three different types of optimal stopping problems – a benchmark family of Bermudan max-call option pricing instances used in the recent literature, a more complex family of instances involving exponentially-transformed fractional Brownian motion and a family of stylized exit-timing problem instances calibrated using real cryptocurrency data – we show that our approach yields policies that in general are substantially better than policies produced by LSM and the pathwise optimization method (Desai et al. 2012), and are comparable or better than a recently proposed nonparametric method based on representing the stopping policy as a tree (Ciocan and Mišić 2022).

The rest of this paper is organized as follows. In Section 2, we review the relevant literature in optimal stopping and ADP, 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 full stochastic forms, define the randomized linear policy problem in its sample average and full stochastic forms, and prove that the randomized linear policy problem and deterministic linear problems are equivalent. In Section 4, we present our pathological instances that characterize the worst-case performance of LSM. In Section 5, we show that our randomized policy SAA problem is NP-Hard, and present our alternating maximization heuristic. 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

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. Carriere (1996) first introduced 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), Yang et al. (2024) for salient examples of this methodology, and to the review paper of Brown and Smith (2022) for a detailed overview of this technique as it applies to stochastic dynamic programming more broadly. Outside of martingale duality, Glanzer et al. (2025) consider an approach for obtaining probabilistically guaranteed upper (and lower) bounds by applying backward recursion with high-biased and low-biased continuation value functions, that result from applying uniform confidence bands to a kernel ridge regression estimate of the continuation value.

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 (2023) 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 expected reward estimate with respect to a sample of trajectories.

We note that the notion of a randomized linear policy has been previously proposed in the optimal stopping literature. In particular, Bayer et al. (2021) propose an approach where the stopping decision is randomized according to a probability that depends on the current state, and in their numerical experiments, specifically consider the logistic regression form. As another example, Becker et al. (2019) consider a deep learning approach for optimal stopping, where the stopping decision is randomized using a logistic probability, for which the logit is output by a deep neural network. Our work differs from this prior work in several regards. First, our work is the first to highlight that the randomized linear policy class is equivalent to the deterministic linear policy class (this is the focus of Section 3.5). Second, our work also highlights that optimizing over randomized policies (or equivalently, optimizing over deterministic policies) is fundamentally distinct from the regression/LSM approach. In particular, our work is the first to establish the existence of problem instances where, holding the basis function architecture fixed, optimization over deterministic policies either exactly or approximately recovers the optimal policy, while LSM can have an arbitrarily large suboptimality gap; this is the focus of Section 4. Lastly, our paper also develops a novel optimization approach for deriving randomized policies from a sample of trajectories. Our approach is based on a biconjugate representation of the randomized policy objective, and solving this biconjugate representation using alternating maximization. This approach is distinct from the backward recursion approaches used in Becker et al. (2019) and Bayer et al. (2021).

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 full 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 full 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 $\mathcal{A}=\{\mathbf{stop},\mathbf{continue}\}$.

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:

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

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 $t\in[T]$, $\mathbf{x}\in\mathcal{X}$, $0\le g(t,\mathbf{x})\le\bar{G}$.*

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 $\phi_{t,1},\ldots,\phi_{t,K}:\mathcal{X}\to\mathbb{R}$ to be a collection of basis functions for period t, which map a state to a real number; for convenience, we will use $\Phi_t(\mathbf{x}) = (\phi_{t,1}(\mathbf{x}), \ldots, \phi_{t,K}(\mathbf{x}))$ 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., $\mathbf{b} = (\mathbf{b}_1, \ldots, \mathbf{b}_T)$. For simplicity, we assume that the number of basis functions K does not vary over time, and is the same for each period. 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:

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

We let $\mathcal{B} \subseteq \mathbb{R}^{KT}$ be the set of feasible weight vectors, and let $\Pi_{\mathcal{B}} = \{\pi_{\mathbf{b}} \mid \mathbf{b} \in \mathcal{B}\}$ be the corresponding set of linear policies. The linear policy optimal stopping problem can then be written as:

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

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:

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

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 $\prod_{t'=1}^{t-1} \mathbb{I}\{\mathbf{b}_{t'} \bullet \Phi_{t'}(\mathbf{x}(t')) \leq 0\} \cdot \mathbb{I}\{\mathbf{b}_t \bullet \Phi_t(\mathbf{x}(t)) > 0\}$ is equal to 1 if and only if $\tau_{\pi_{\mathbf{b}}} = t$; 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:

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

Note that the term $\prod_{t'=1}^{t-1} \mathbb{I}\{\mathbf{b}_{t'} \bullet \Phi_{t'}(\mathbf{x}(\omega, t')) \leq 0\} \cdot \mathbb{I}\{\mathbf{b}_t \bullet \Phi_t(\mathbf{x}(\omega, t)) > 0\}$ 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) is always zero due to the presence of the indicator function, which 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

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

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

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

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

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

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 $\tilde{\Pi}_{\mathcal{B}} = \{\tilde{\pi}_{\mathbf{b}} \mid \mathbf{b} \in \mathcal{B}\}$ to be the set of feasible randomized policies. 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 then be written as

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

or equivalently, as

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

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

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

In other words, we seek to find the coefficients $\mathbf{b} = (\mathbf{b}_1, \dots, \mathbf{b}_T)$ 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 problem (10) 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, we can 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, we can additionally show that the optimal objective values of the deterministic and randomized policy full problems (problems (5) and (9) respectively) are 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 full problem (5), the deterministic policy SAA problem (7), the randomized policy full 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 sets of feasible weight vectors for the two 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. *For each $t \in [T]$, the first basis function $\phi_{t,1}(\cdot)$ is the constant basis function, i.e., $\phi_{t,1}(\mathbf{x}) = 1$ 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 $\mathbf{b}_t \bullet \Phi_t(\mathbf{x}) = 0$ 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 $\Phi_{t,2:K} : \mathcal{X} \to \mathbb{R}^{K-1}$ be defined as $\Phi_{t,2:K}(\mathbf{x}) = (\phi_{t,2}(\mathbf{x}), \dots, \phi_{t,K}(\mathbf{x}))$, which is just the vector-valued mapping of the state $\mathbf{x}$ to the basis function values $\phi_{t,2}(\mathbf{x})$ through $\phi_{t,K}(\mathbf{x})$ (in other words, it is just the mapping Φ_t, only with the first basis function $\phi_{t,1}(\cdot)$ 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* $\mathbf{c} \in \mathbb{R}^{K-1}$ *with* $\mathbf{c} \neq \mathbf{0}$*,* $d \in \mathbb{R}$*, and any* $t \in [T]$*,* $\mathbb{P}(\Phi_{t,2:K}(\mathbf{x}(t)) \in A) = 0$.

We can now state our counterpart of Theorem 1 for the true 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 $\Phi_t(\mathbf{x}(\omega, t))$ 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 $\mathbf{b}_t \bullet \Phi_t(\mathbf{x}(\omega, t)) = 0$ for any ω and any t). In the full stochastic optimization problem setting, this is no longer possible. For this reason, we introduce Assumption 4, which requires that $\Phi_{t,2:K}(\mathbf{x}(t))$ 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 $\mathbf{b}_t \bullet \Phi_t(\mathbf{x}(t))$ 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 $\phi_{t,2}(\mathbf{x}), \dots, \phi_{t,K}(\mathbf{x})$ are polynomials of $\mathbf{x} \in \mathcal{X}$ for each t. In this case, the set $\{\mathbf{x} \in \mathcal{X} \mid \mathbf{c} \bullet \Phi_{t,2:K}(\mathbf{x}) + d = 0\}$ 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 $\mathbb{P}(\Phi_{t,2:K}(\mathbf{x}(t)) \in A) = 0$ for any hyperplane $A \subseteq \mathbb{R}^{K-1}$.

Theorems 1 and 2 are significant because they assert 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 full 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 finite sample case, 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 smooth. 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. In fact, it turns out that $\hat{J}_R(\cdot)$ has a particularly nice structure that lends itself to an iterative algorithm based on alternating maximization, where one alternates between solving two concave maximization problems; we provide the details in Section 5.2.

4. Performance comparison of LSM policies and optimal deterministic linear policies

In this section, we seek to understand the suboptimality gap of LSM in an idealized setting. In Section 4.1, we review the LSM method and define an idealized version of LSM, called the population-level LSM algorithm, which can be thought of as LSM applied to an infinite sample of trajectories. In Section 4.2, we devise a simple pathological problem instance where we show that the LSM method with only the constant basis function can in general perform arbitrarily poorly relative to the true optimal policy, whereas the optimal deterministic linear policy (DLP) exactly coincides with the true optimal policy. Then, in Section 4.3, we devise a more complex pathological problem instance where we show that the LSM method with the constant basis function and the current reward can also achieve an arbitrarily large optimality gap, while the optimal deterministic linear policy simultaneously attains an arbitrarily small optimality gap.

4.1. Review of the LSM method

We briefly review both the exact dynamic programming approach to solving optimal stopping problems, as well as the LSM method of Longstaff and Schwartz (2001). Recall that when the stochastic process $\{\mathbf{x}(t)\}_{t=1}^{T}$ is a Markov process, then we can solve the optimal stopping problem (2) by dynamic programming. In particular, letting $J_t(\cdot)$ denote the optimal value function, then Algorithm 1 exactly represents the dynamic programming approach.

Algorithm 1 Dynamic programming approach for solving problem (2).

```
$J_T(\mathbf{x}) \leftarrow g(T, \mathbf{x})$ for all $\mathbf{x} \in \mathcal{X}$.
for $t = T-1, \ldots, 1$ do
    $C_t(\mathbf{x}) \leftarrow \mathbb{E}[J_{t+1}(\mathbf{x}(t+1)) \mid \mathbf{x}(t) = \mathbf{x}]$, for all $\mathbf{x} \in \mathcal{X}$.
    $J_t(\mathbf{x}) \leftarrow \max\{g(t, \mathbf{x}), C_t(\mathbf{x})\}$, for all $\mathbf{x} \in \mathcal{X}$.
end for
```

An equivalent restatement of dynamic programming can be obtained if one replaces the conditional expectation with a least squares problem, and replaces the maximization over $g(t,\cdot)$ and $C_t(\cdot)$ with a conditional assignment. In particular, let $\mathcal{F}$ denote the set of all functions $f:\mathcal{X}\rightarrow\mathbb{R}$. Then Algorithm 1 is equivalent to the following procedure, which we denote as Algorithm 2.

Algorithm 2 Equivalent dynamic programming approach using least squares for solving problem (2).

```
$J_T(\mathbf{x}) \leftarrow g(T,\mathbf{x})$ for all $\mathbf{x} \in \mathcal{X}$.
for $t = T-1,\ldots,1$ do
    $C_t(\cdot) \leftarrow \arg\min_{f\in\mathcal{F}} \mathbb{E}[(f(\mathbf{x}(t)) - J_{t+1}(\mathbf{x}(t+1)))^2]$
    $J_t(\mathbf{x}) \leftarrow \begin{cases} g(t,\mathbf{x}), & \text{if } g(t,\mathbf{x}) > C_t(\mathbf{x}), \\ \mathbb{E}[J_{t+1}(\mathbf{x}(t+1) \mid \mathbf{x}(t)=\mathbf{x}], & \text{if } g(t,\mathbf{x}) \leq C_t(\mathbf{x}). \end{cases}$
end for
```

The equivalence between Algorithms 1 and 2 arises because of the connection between conditional expectation and least squares: recall that given two random variables $Y\in\mathbb{R}$ and $X\in\mathbb{R}^n$, the function $h(x)=\mathbb{E}[Y \mid X=x]$ is the solution of the problem $\min_f \mathbb{E}[(Y-f(X))^2]$, where the minimization is carried out over all functions $f:\mathbb{R}^n\rightarrow\mathbb{R}$.

The LSM algorithm leverages this equivalence, and introduces two simplifications to transform Algorithm 2 into a practical algorithm. First, it is in general difficult to solve the least squares problem $\min_{f\in\mathcal{F}} \mathbb{E}[(f(\mathbf{x}(t))-J_{t+1}(\mathbf{x}(t+1)))^2$, as it will be an infinite dimensional problem whenever $\mathcal{X}$ is an infinite set. Thus, instead of solving this problem, we introduce a set of basis functions $\{\phi_{t,k}(\cdot)\}_{k=1}^K$ for each period t. For each t, we then consider the set of functions spanned by its set of basis functions, defined as

$$\tilde{\mathcal{F}}_t = \left\{ \sum_{k=1}^{K} b_{t,k}\phi_{t,k}(\cdot) \;\middle|\; b_{t,1},\ldots,b_{t,K}\in\mathbb{R} \right\}. \tag{11}$$

Now, rather than solving the least squares problem at each period over $\mathcal{F}$, we can instead solve it over $\tilde{\mathcal{F}}_t$, which leads to a finite dimensional optimization problem over the weights $b_{t,1},\ldots,b_{t,K}$. This leads to an approximate method that we call the *population-level LSM algorithm*, as all calculations are performed using the true expected values. We define this procedure below as Algorithm 3.

Finally, the population-level LSM algorithm generally cannot be applied, as it requires us to calculate true expected values. Instead of working with true expected values, we can consider simplifying Algorithm 3 so that we work with a sample of trajectories. As in Section 3.3 and 3.4, we introduce a sample of Ω trajectories, indexed by $\omega=1,\ldots,\Omega$. We carry out our updates with respect to the index of the random trajectory, so that $J_t(\omega)$ is the reward obtained along sample path ω, following the stopping rule defined by $\hat{C}_t(\cdot),\ldots,\hat{C}_{T-1}(\cdot)$. This leads to a new procedure,

Algorithm 3 Population-level LSM algorithm for solving problem (2).

```
$J_T(\mathbf{x}) \leftarrow g(T, \mathbf{x})$ for all $\mathbf{x} \in \mathcal{X}$.
for $t = T-1, \ldots, 1$ do
    $\hat{C}_t(\cdot) \leftarrow \arg\min_{f \in \tilde{\mathcal{F}}_t} \mathbb{E}[(f(\mathbf{x}(t)) - J_{t+1}(\mathbf{x}(t+1)))^2]$
    $J_t(\mathbf{x}) \leftarrow \begin{cases} g(t, \mathbf{x}), & \text{if } g(t, \mathbf{x}) > \hat{C}_t(\mathbf{x}), \\ \mathbb{E}[J_{t+1}(\mathbf{x}(t+1)) \mid \mathbf{x}(t) = \mathbf{x}], & \text{if } g(t, \mathbf{x}) \leq \hat{C}_t(\mathbf{x}). \end{cases}$
end for
```

Algorithm 4 (Sample-based) LSM algorithm of Longstaff and Schwartz (2001) for solving problem (2).

```
$J_T(\omega) \leftarrow g(T, \mathbf{x}(\omega, t))$ for all $\mathbf{x} \in \mathcal{X}$.
for $t = T-1, \ldots, 1$ do
    $\hat{C}_t(\cdot) \leftarrow \arg\min_{f \in \tilde{\mathcal{F}}_t} \sum_{\omega=1}^{\Omega} (f(\mathbf{x}(\omega, t)) - J_{t+1}(\omega))^2$
    $J_t(\omega) \leftarrow \begin{cases} g(t, \mathbf{x}(\omega, t)), & \text{if } g(t, \mathbf{x}(\omega, t)) > \hat{C}_t(\mathbf{x}(\omega, t)), \\ J_{t+1}(\omega), & \text{if } g(t, \mathbf{x}(\omega, t)) \leq \hat{C}_t(\mathbf{x}(\omega, t)). \end{cases}$
end for
```

Algorithm 4, which is exactly the (sample-based) LSM method as presented in Longstaff and Schwartz (2001).

We end this section with a couple of remarks about the algorithms we present here. First, as noted earlier, Algorithm 4 is what is usually known as the LSM algorithm. However, the population-level LSM algorithm 3 will be useful in the stylized analysis that we undertake in Sections 4.2 and 4.3, as it will allow us to compare LSM and deterministic linear policy optimization without worrying about the errors induced by finite samples. Second, from the step-by-step transformation we showed above, we hope that it becomes clear where the potential loss in LSM comes from. In particular, the idealized least squares problem $\min_{f \in \mathcal{F}} \mathbb{E}[(f(\mathbf{x}(t)) - J_{t+1}(\mathbf{x}(t+1)))^2]$ recovers the exact conditional expectation, but only when the minimization is carried out over $\mathcal{F}$; when we replace $\mathcal{F}$ with $\tilde{\mathcal{F}}_t$, this no longer needs to be the case. As we shall show in the next two sections, even in very simple settings, a poor choice of $\tilde{\mathcal{F}}_t$ can lead to arbitrarily poor performance for LSM.

4.2. Performance of LSM and DLP with constant-only basis function architecture

In this section, we develop a family of pathological instances where, for the same basis function architecture, LSM can perform arbitrarily poorly relative to the optimal policy, while DLP recovers the optimal policy. The specific basis function architecture that we consider here is one consisting only of the constant basis function, i.e., the function $\phi(\mathbf{x}) = 1$. For the remainder of this section and Section 4.3, we will use LSM to refer to the population-level LSM method (Algorithm 3).

Let $a \in (0, 1)$ and $\delta > 0$. Let $P^1_{a,\delta}$ denote an optimal stopping problem instance, defined as follows:

- $T = 2$;

- There is a single state variable, $x(t) \in \mathcal{X} = [0,1]$;
- $x(1)$ is uniformly distributed on $[0,1]$;
- $x(2) \mid x(1) = x(1)$, i.e., $x(2)$ is deterministically equal to $x(1)$ (there is no probabilistic transition from period 1 to period 2);
- For $t = 1$, the reward is
$$g(1,x) = \begin{cases} a + \delta, \text{ if } x \leq a, \\ a - \delta, \text{ if } x > a; \end{cases} \tag{12}$$
- For $t = 2$, the reward is
$$g(2,x) = \begin{cases} 1 \text{ if } x \leq a, \\ 0 \text{ if } x > a. \end{cases} \tag{13}$$
In other words, the reward in the second period is 1 or 0 depending on whether $x(2)$ is below or above a.

Let us define the basis functions $\phi_{1,1}, \phi_{1,2}, \phi_{2,1}, \phi_{2,2}$ as

$$\phi_{1,1}(x) = 1, \qquad \phi_{2,1}(x) = 1, \tag{14}$$

$$\phi_{1,2}(x) = g(1,x), \qquad \phi_{2,2}(x) = g(2,x), \tag{15}$$

i.e., $\phi_{1,1}$ and $\phi_{2,1}$ are the constant basis functions, and $\phi_{1,2}$ and $\phi_{2,2}$ are the reward functions for periods 1 and 2 respectively.

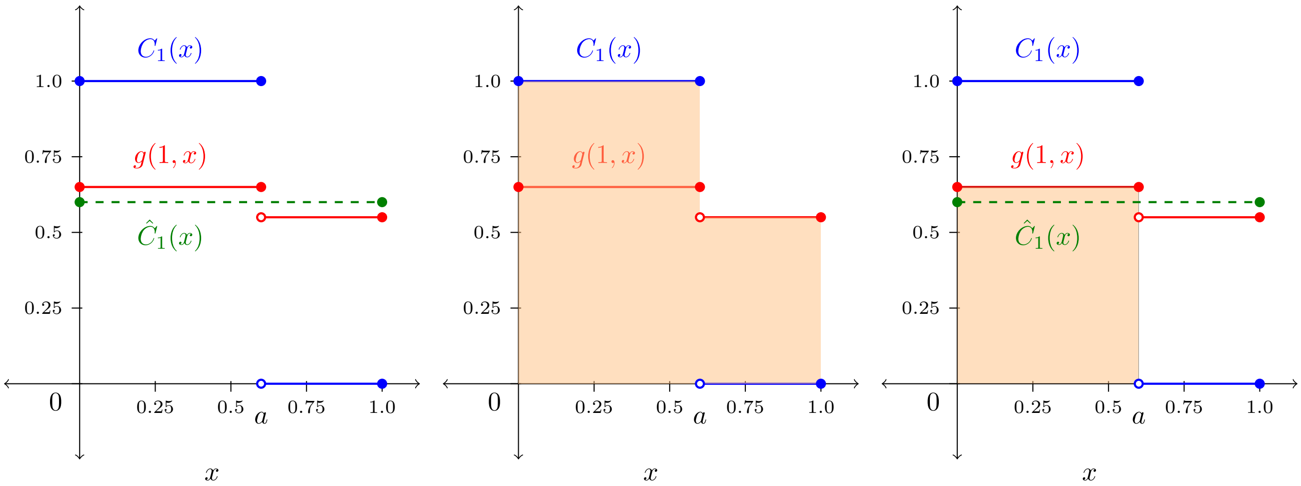
Let $V^*(P)$ denote the optimal expected reward for a problem instance P, where the expectation is taken over the initial state $x(1)$. Let $V^{\text{LSM}}(P)$ denote the expected reward of the LSM policy for an instance P where the LSM policy basis function architecture at $t = 1$ consists only of the constant basis function $\phi_{1,1}$. Let $V^{\text{DLP}}(P)$ denote the expected reward of the optimal deterministic linear policy for an instance P, where the deterministic linear policy approach is endowed with $\phi_{1,1}$ and $\phi_{1,2}$ at period 1, and $\phi_{2,1}$ and $\phi_{2,2}$ at period 2, i.e., additionally endowed with the reward function as another basis function. We remind readers here that LSM always stops in the second period, and hence we do not need to explicitly define basis functions for LSM in period 2. We additionally remind readers that the LSM policy incorporates the current reward automatically; thus, the DLP policy should be endowed with the current reward as a basis function to ensure the LSM and DLP policies are directly comparable. (The LSM policy involves stopping at period 1 if and only if $g(1,x) > b_{1,1}$ for some appropriately chosen weight $b_{1,1}$, which is exactly a deterministic linear policy in terms of $\phi_{1,1}(\cdot)$ and $\phi_{1,2}(\cdot)$.)

We then have the following result.

Theorem 3. *For any $\epsilon > 0$, there exist $a \in (0,1)$ and $\delta > 0$ such that*

$$\frac{V^*(P^1_{a,\delta}) - V^{\text{LSM}}(P^1_{a,\delta})}{V^*(P^1_{a,\delta})} > 1 - \epsilon, \tag{16}$$

$$\frac{V^*(P^1_{a,\delta}) - V^{\text{DLP}}(P^1_{a,\delta})}{V^*(P^1_{a,\delta})} = 0. \tag{17}$$



(a) Reward function at $t = 1$, LSM approximate continuation value function $\hat{C}_1(\cdot)$ and true continuation value function $C_1(\cdot)$.

(b) Reward of optimal policy (shape shaded in orange; area of approximately $2a - a^2$).

(c) Reward of LSM policy (shape shaded in orange; area of approximately a^2).

Figure 1 **Visualization of pathological instance of Section 4.2.**

The proof of this result is provided in Section EC.1.3. The idea of the proof is that, by the construction of the $P^1_{a,\delta}$ instances, it is possible to trick the LSM policy into stopping exactly when the optimal policy chooses to continue, and to continue exactly when the optimal policy chooses to stop. This results in the optimal policy garnering a reward of approximately $2a - a^2$, while the LSM policy only garners approximately a^2 in reward; for sufficiently small a, the relative gap between the two policies can be made arbitrarily close to 1. On the other hand, the structure of the optimal policy is such that it can be represented exactly as a deterministic linear policy using the constant and reward basis functions, resulting in the gap between the optimal and DLP policies being exactly zero. Figure 1 provides a visualization of the reward function $g(1,\cdot)$, the optimal continuation value function $C_1(\cdot)$, the LSM approximate continuation value function $\hat{C}_1(\cdot)$, and the rewards garnered by the LSM and DLP/optimal policies.

This result is notable because it illustrates that, even in a very simple setting – only two periods, only the constant basis function and no issues arising from finite sample data – LSM can perform arbitrarily poorly relative to the optimal policy. In contrast, the DLP policy exactly returns the optimal policy.

A limitation of this result is that LSM is only endowed with the constant basis function, and a fair question is whether LSM would perform better if the current reward was included as an additional basis function. In this example, one can verify that LSM does recover the optimal policy if the current reward is included in its basis function architecture. However, the pathological performance of LSM that we describe here is not eliminated in other instances when LSM is provided with the current reward function, which we discuss in the next section.

4.3. Performance of LSM and DLP with constant-plus-reward basis function architecture

In this section, we continue our comparison of LSM and DLP where we construct a family of pathological instances for which LSM, when endowed with a constant and the current reward as basis functions, continues to perform arbitrarily poorly.

Let $k > 0$ and $\theta \in (0,1)$ denote parameters that we will use shortly to parametrize a particular family of instances. Define $S^r_{a,b}$ as

$$S^r_{a,b} = \frac{1}{r}(e^{rb} - e^{ra}) \tag{18}$$

Let α^*_k and β^*_k be constants defined as

$$\alpha^*_k = \theta - \frac{S^k_{0,\theta} - \theta S^k_{0,1}}{S^{2k}_{0,1} - (S^k_{0,1})^2} \cdot S^k_{0,1} \tag{19}$$

$$\beta^*_k = \frac{S^k_{0,\theta} - \theta S^k_{0,1}}{S^{2k}_{0,1} - (S^k_{0,1})^2}. \tag{20}$$

Let $P^2_{k,\theta}$ denote an optimal stopping problem instance, with the following properties:

- $T = 2$;
- There is a single state variable, $x(t) \in [0,1]$;
- $x(1)$ is uniformly distributed on $[0,1]$;
- $x(2) \mid x(1) = x(1)$, i.e., $x(2)$ is deterministically equal to $x(1)$ (there is no probabilistic transition from period 1 to period 2);
- For $t = 1$, the reward function is [1]

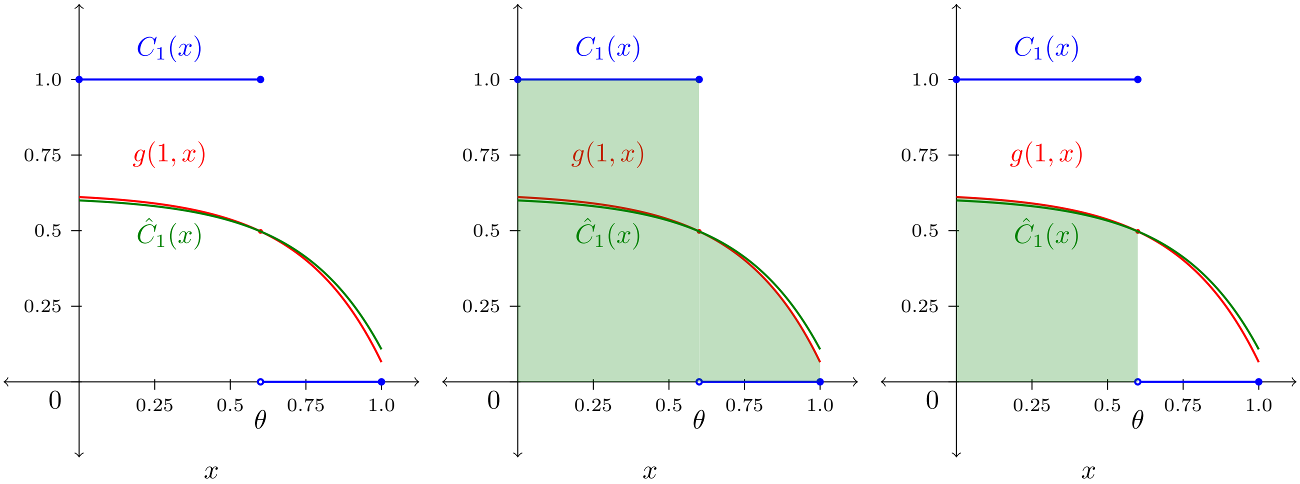
$$g(1,x) = (1 + e^{-k\theta})\beta^*_k e^{kx} + \alpha^*_k - \beta^*_k;$$

- For $t = 2$, the reward function is

$$g(2,x) = \mathbb{I}\{x \le \theta\} = \begin{cases} 1 \text{ if } x \le \theta, \\ 0 \text{ if } x > \theta. \end{cases}$$

We compare LSM and the deterministic linear policy approach, where both approaches use a constant and the current reward as basis functions, i.e., LSM uses $\phi_{1,1}$ and $\phi_{1,2}$ in period 1, while DLP uses $\phi_{1,1}, \phi_{1,2}$ in period 1, and $\phi_{2,1}, \phi_{2,2}$ in period 2. We then have the following result.

[1] We note that for sufficiently large k, $g(1,x)$ can be negative for some values of x. It turns out that this is not a major issue: by subtracting $g(1,1)$ from both $g(1,\cdot)$ and $g(2,\cdot)$, the new $g(1,\cdot)$ and $g(2,\cdot)$ will be nonnegative for all x since $g(1,1)$ is negative when k is sufficiently large, and $g(1,x)$ is decreasing in x. This has the effect of basically adding a constant $-g(1,1)$ to the reward of any policy that stops at period 1 or 2; moreover, this constant $-g(1,1)$ converges to θ as $k \to \infty$. All of the analysis for Theorem 4 goes through in this case, albeit with some additional tedious steps.



(a) Reward function at $t = 1$, LSM approximate continuation value function $\hat{C}_1(\cdot)$ and true continuation value function $C_1(\cdot)$.

(b) Reward of optimal policy (area of green shape; area approaches $2\theta - \theta^2$ as $k \to \infty$).

(c) Reward of LSM policy (area of green shape; area approaches θ^2 as $k \to \infty$).

Figure 2 **Visualization of pathological instance of Section 4.3.**

THEOREM 4. *For any $\epsilon > 0$, there exist $k > 0$ and $\theta \in (0,1)$ such that*

$$\frac{V^*(P^2_{k,\theta}) - V^{\text{DLP}}(P^2_{k,\theta})}{V^*(P^2_{k,\theta})} < \epsilon, \tag{21}$$

$$\frac{V^*(P^2_{k,\theta}) - V^{\text{LSM}}(P^2_{k,\theta})}{V^*(P^2_{k,\theta})} > 1 - \epsilon. \tag{22}$$

The proof of this result is presented in Section EC.1.4. This result is significant because it establishes that even when LSM is endowed with the current reward as a basis function, it can perform arbitrarily poorly, and DLP can perform arbitrarily well, compared to the optimal policy. The basic idea in these instances, as with those in Section 4.2, is to set up the second period reward so that the approximate continuation value function that LSM obtains leads to the wrong decision (i.e., $\hat{C}_1(x) > g(1,x)$ whenever the optimal policy says to stop, and $\hat{C}_1(x) < g(1,x)$ whenever the optimal policy says to continue). Figure 2 visualizes the period 1 reward function $g(1,\cdot)$, the LSM approximate continuation value function $\hat{C}_1(\cdot)$ and the true continuation value function $C_1(\cdot)$ (observe that for $x < \theta$, $C_1(x) > g(1,x) > \hat{C}_1(x)$, while for $x > \theta$, $C_1(x) < g(1,x) < \hat{C}_1(x)$, i.e., LSM is again tricked into making the wrong decision).

We offer two additional comments on Theorem 4. First, we note that these instances are not straightforward to devise. For the $P^1_{a,\delta}$ instances of the previous section, the task of devising the instance family described there made easier by the fact that the current reward is not a basis function; thus, after one constructs $g(2,\cdot)$, the approximate continuation value function is fixed and fully determined, and one is then free to design $g(1,\cdot)$ to lead to pathological behavior. This "linear" construction strategy does not work when the current reward is a basis function, because by changing $g(1,\cdot)$, the approximate continuation value function $\hat{C}_1(\cdot)$ also changes, which makes the construction of pathological instances in this setting that much more challenging.

Second, a key insight that emerges as one proceeds through the proof of Theorem 4 is how LSM and DLP respond to magnitudes of basis functions. For LSM, the magnitudes of the basis functions are extremely important. Observe that the reward function $g(1,x)$ is essentially a function of the form $c + de^{kx}$. When k is very large, the de^{kx} term explodes as x gets closer to 1, and in order to fit $J_2(x) = g(2,x) = \mathbb{I}\{x \leq \theta\}$, the coefficient of $g(1,x)$ in the regression model at $t = 1$ necessarily becomes very small. This results in $\hat{C}_1(x)$ being almost constant, and in particular it is possible to obtain regions in $[0,1]$ where $\hat{C}_1(x) < g(1,x) < C_1(x)$ (i.e., the LSM policy recommends to stop, but the optimal action is to continue) and where $\hat{C}_1(x) > g(1,x) > C_1(x)$ (i.e., the LSM policy recommends to stop, but the optimal action is to stop). On the other hand, for DLP, there exists a policy that continues when $x < \theta$ and stops when $x > \theta$, which agrees with the optimal policy for most x. What is crucial for this good performance is the fact that the function $g(1,x)$ is strictly monotonic, so the DLP policy is able to produce policies of the form $\pi(1,x) = \textbf{stop}$ if and only if $x < c$, or of the form $\pi(1,x) = \textbf{stop}$ if and only if $x \geq c$, for some threshold c. An important observation here is that such policies could be produced by any strictly monotonic $g(1,x)$: the fact that $g(1,x)$ is an exponential function plus a constant is not important to the DLP approach, and the DLP approach is unfazed by how quickly $g(1,x)$ grows. In the DLP approach, the basis functions are only important in the stopping regions that they induce.

5. Solution Methodology

Having shown that optimizing over randomized policies is equivalent to optimizing over deterministic policies (Section 3), and having shown that in theory there can be a significant performance difference between optimal deterministic/randomized policies and policies obtained using LSM (Section 4), 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 alternating maximization. Finally, in Section 5.3, we develop a convergence result for our alternating maximization 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 5. *The randomized policy SAA problem* (10) *is NP-Hard.*

We make a few remarks about this result. First, our proof of Theorem 5 (see Section EC.1.5 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. Alternating maximization 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 the *biconjugate* representation of problem (10) over the weight vectors of all periods using alternating maximization. In particular, recall that $\mathbf{b} = (\mathbf{b}_1, \ldots, \mathbf{b}_T)$, where each $\mathbf{b}_t$ is the weight vector for period t, and that $\hat{J}_R(\mathbf{b})$ is the SAA objective for a given $\mathbf{b}$. Assume that the set of feasible weight vectors can be written as $\mathcal{B} = \mathcal{B}_1 \times \cdots \times \mathcal{B}_T$, where $\mathcal{B}_1, \ldots, \mathcal{B}_T \subseteq \mathbb{R}^K$ is a collection of convex sets for the feasible weight vector at each period. Observe now that the objective function of the randomized policy SAA problem can be re-written as

$$\hat{J}_R(\mathbf{b}) = \frac{1}{\Omega} \sum_{\omega=1}^{\Omega} \sum_{t=1}^{T} g(t, \mathbf{x}(\omega, t)) \cdot \prod_{t'=1}^{t-1} (1 - \sigma(\mathbf{b}_{t'} \bullet \Phi_{t'}(\mathbf{x}(\omega, t'))))\sigma(\mathbf{b}_t \bullet \Phi_t(\mathbf{x}(\omega, t))) \tag{23}$$

$$= \sum_{\omega=1}^{\Omega} \sum_{t=1}^{T} \exp\left(\log\left(\frac{g(t, \mathbf{x}(\omega, t))}{\Omega} \right) + \sum_{t'=1}^{t-1} \log(1 - \sigma(\mathbf{b}_{t'} \bullet \Phi_{t'}(\mathbf{x}(\omega, t')))) + \log(\sigma(\mathbf{b}_t \bullet \Phi_t(\mathbf{x}(\omega, t)))) \right). \tag{24}$$

Let us use the function $\psi_{\omega,t} : \mathcal{B} \to \mathbb{R}$ to denote the function inside each $\exp(\cdot)$ for each trajectory ω and each period t, i.e., we define $\psi_{\omega,t}$ as

$$\psi_{\omega,t}(\mathbf{b}) = \log\left(\frac{g(t, \mathbf{x}(\omega, t))}{\Omega} \right) + \sum_{t'=1}^{t-1} \log(1 - \sigma(\mathbf{b}_{t'} \bullet \Phi_{t'}(\mathbf{x}(\omega, t')))) + \log(\sigma(\mathbf{b}_t \bullet \Phi_t(\mathbf{x}(\omega, t)))), \tag{25}$$

where we take $\psi_{\omega,t}(\mathbf{b}) = -\infty$ if $g(t, \mathbf{x}(\omega, t)) = 0$ (which would result in $\log(g(t, \mathbf{x}(\omega, t))/\Omega) = -\infty$). Observe that the function $\psi_{\omega,t}(\cdot)$ is concave in $\mathbf{b}$, as it is the sum of a linear function of $\mathbf{b}$ minus the sum of the softplus function, $u \mapsto \log(1 + e^u)$, which are convex in u. We can then write $\hat{J}_R(\cdot)$ as

$$\hat{J}_R(\mathbf{b}) = \sum_{\omega=1}^{\Omega} \sum_{t=1}^{T} e^{\phi_{\omega,t}(\mathbf{b})}. \tag{26}$$

Observe now that since $x \mapsto \log(x)$ is monotonic, the set of optimal solutions of the problem $\max_{\mathbf{b} \in \mathcal{B}} \hat{J}_R(\mathbf{b})$ remains unchanged if we consider the same problem with the log-transformed objective:

$$\log \hat{J}_R(\mathbf{b}) = \log \left[\sum_{\omega=1}^{\Omega} \sum_{t=1}^{T} e^{\psi_{\omega,t}(\mathbf{b})} \right]. \tag{27}$$

This new objective has a special structure; it is in the form of the log of sum of exponential (log-sum-exp) function, which is defined as $g(\mathbf{y}) = \log(\sum_{i=1}^{n} e^{y_i})$, for any $\mathbf{y} \in \mathbb{R}^n$. A standard result from convex analysis is that any proper, lower semi-continuous, convex function is equivalent to its biconjugate function, i.e., the convex conjugate of its convex conjugate (Rockafellar 1970). In particular, for the log-sum-exp function $g(\mathbf{y})$, it can be equivalently represented as

$$g(\mathbf{y}) = \max_{\boldsymbol{\mu} \in \Delta_{[n]}} \left\{ \sum_{i=1}^{n} \mu_i y_i - \sum_{i=1}^{n} \mu_i \log \mu_i \right\}, \tag{28}$$

where $\Delta_{[n]} = \{\boldsymbol{\mu} \in \mathbb{R}^n \mid \mathbf{1}^T \boldsymbol{\mu} = 1, \boldsymbol{\mu} \geq \mathbf{0}\}$ is the $(n-1)$-dimensional unit simplex. Using this result, we can re-write $\log \hat{J}_R(\cdot)$ as

$$\log \hat{J}_R(\mathbf{b}) = \max_{\boldsymbol{\mu} \in \Delta_{[\Omega] \times [T]}} \left\{ \sum_{\omega=1}^{\Omega} \sum_{t=1}^{T} \mu_{\omega,t} \cdot \psi_{\omega,t}(\mathbf{b}) - \sum_{\omega=1}^{\Omega} \sum_{t=1}^{T} \mu_{\omega,t} \log \mu_{\omega,t} \right\}. \tag{29}$$

Therefore, the randomized policy SAA problem becomes

$$\begin{aligned}
\underset{\mathbf{b}, \boldsymbol{\mu}}{\text{maximize}} \quad & \sum_{\omega=1}^{\Omega} \sum_{t=1}^{T} \mu_{\omega,t} \cdot \psi_{\omega,t}(\mathbf{b}) - \sum_{\omega=1}^{\Omega} \sum_{t=1}^{T} \mu_{\omega,t} \log \mu_{\omega,t} && \text{(30a)} \\
\text{subject to} \quad & \sum_{\omega=1}^{\Omega} \sum_{t=1}^{T} \mu_{\omega,t} = 1, && \text{(30b)} \\
& \mu_{\omega,t} \geq 0, \quad \forall\, \omega \in [\Omega],\ t \in [T], && \text{(30c)} \\
& \mathbf{b}_1 \in \mathcal{B}_1, \ldots, \mathbf{b}_T \in \mathcal{B}_T. && \text{(30d)}
\end{aligned}$$

Upon closer examination, we can see that this formulation is close to being a convex optimization problem. The feasible region is convex; for a fixed $\boldsymbol{\mu}$, the objective function is concave in $\mathbf{b}$, because each $\psi_{\omega,t}(\cdot)$ function is concave in $\mathbf{b}$ and for a fixed $\mathbf{b}$, the objective function is concave in $\boldsymbol{\mu}$, because it is the sum of a linear function of $\boldsymbol{\mu}$ minus the sum of negative entropies of the $\mu_{\omega,t}$'s, each of which are convex in $\mu_{\omega,t}$. The difficulty in solving this problem arises because the objective function, while marginally concave in $\mathbf{b}$ and $\boldsymbol{\mu}$ separately, is not jointly concave in $(\mathbf{b}, \boldsymbol{\mu})$, because of the product/bilinear terms $\mu_{\omega,t} \cdot \psi_{\omega,t}(\mathbf{b})$.

However, we observe that the constraints for $\boldsymbol{\mu}$ and $\mathbf{b}$ in this problem are separable, and thus we can at least solve the problem approximately using *alternating maximization*. The idea of the alternating maximization approach is that we start with an initial $\boldsymbol{\mu}$, and solve problem (30) for $\mathbf{b}$ while holding $\boldsymbol{\mu}$ constant. Then, we fix $\mathbf{b}$ to its current solution and solve problem (30) for $\boldsymbol{\mu}$. We repeat this process, alternating between optimizing over $\mathbf{b}$ and optimizing over $\boldsymbol{\mu}$, until the change in the objective function (30a) is within a pre-specified tolerance. We formally define this procedure as Algorithm 5 below.

Algorithm 5 Alternating maximization algorithm to approximately solve SAA problem (30).

```
Require: Initial choice of $\boldsymbol{\mu} \in \Delta_{[\Omega]\times[T]}$
b ← 0
$Z \leftarrow \sum_{\omega=1}^{\Omega} \sum_{t=1}^{T} \mu_{\omega,t} \cdot \psi_{\omega,t}(\mathbf{b}) - \sum_{\omega=1}^{\Omega} \sum_{t=1}^{T} \mu_{\omega,t} \log \mu_{\omega,t}$
repeat
    $Z_{previous} \leftarrow Z$
    $\mathbf{b} \leftarrow \arg\max_{\mathbf{b}' \in \mathcal{B}} \left\{ \sum_{\omega=1}^{\Omega} \sum_{t=1}^{T} \mu_{\omega,t} \cdot \psi_{\omega,t}(\mathbf{b}') - \sum_{\omega=1}^{\Omega} \sum_{t=1}^{T} \mu_{\omega,t} \log \mu_{\omega,t} \right\}$
    $\boldsymbol{\mu} \leftarrow \arg\max_{\boldsymbol{\mu}' \in \Delta_{[\Omega]\times[T]}} \left\{ \sum_{\omega=1}^{\Omega} \sum_{t=1}^{T} \mu'_{\omega,t} \cdot \psi_{\omega,t}(\mathbf{b}) - \sum_{\omega=1}^{\Omega} \sum_{t=1}^{T} \mu'_{\omega,t} \log \mu'_{\omega,t} \right\}$
    $Z \leftarrow \sum_{\omega=1}^{\Omega} \sum_{t=1}^{T} \mu_{\omega,t} \cdot \psi_{\omega,t}(\mathbf{b}) - \sum_{\omega=1}^{\Omega} \sum_{t=1}^{T} \mu_{\omega,t} \log \mu_{\omega,t}$
until $Z - Z_{previous} < \epsilon$
```

We make several important remarks about Algorithm 5. First, we note that with $\boldsymbol{\mu}$ fixed, solving problem (30) for $\mathbf{b}$ amounts to solving the concave maximization problem

$$\max_{\mathbf{b} \in \mathcal{B}} \sum_{\omega=1}^{\Omega} \sum_{t=1}^{T} \mu_{\omega,t} \psi_{\omega,t}(\mathbf{b}). \tag{31}$$

By leveraging the assumption that $\mathcal{B} = \mathcal{B}_1 \times \cdots \times \mathcal{B}_T$, it turns out that problem (31) decomposes by periods. In particular, letting $h_{\omega,t} = \log(g(t, \mathbf{x}(\omega, t))/\Omega)$, we have

$$\max_{\mathbf{b} \in \mathcal{B}} \sum_{\omega=1}^{\Omega} \sum_{t=1}^{T} \mu_{\omega,t} \psi_{\omega,t}(\mathbf{b}) \tag{32}$$

$$= \max_{\mathbf{b}\in\mathcal{B}} \sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} \mu_{\omega,t} \left(h_{\omega,t} - \sum_{t'=1}^{t-1} \log(1 + e^{\mathbf{b}_{t'} \bullet \Phi_{t'}(\mathbf{x}(\omega,t'))}) + \mathbf{b}_t \bullet \Phi_t(\mathbf{x}(\omega,t)) - \log(1 + e^{\mathbf{b}_t \bullet \Phi_t(\mathbf{x}(\omega,t))}) \right) \tag{33}$$

$$= \sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} \mu_{\omega,t} h_{\omega,t} + \max_{\mathbf{b}\in\mathcal{B}} \sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} \mu_{\omega,t} \left\{ - \sum_{t'=1}^{t-1} \log(1 + e^{b_{t'} \bullet \Phi_t(\mathbf{x}(\omega,t'))}) + \mathbf{b}_t \bullet \Phi_t(\mathbf{x}(\omega,t)) - \log(1 + e^{\mathbf{b}_t \bullet \Phi_t(\mathbf{x}(\omega,t))}) \right\} \tag{34}$$

$$= \sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} \mu_{\omega,t} h_{\omega,t} + \sum_{t=1}^{T} \max_{\mathbf{b}_t\in\mathcal{B}_t} \left\{ \sum_{\omega=1}^{\Omega} \left[- \left(\sum_{t'=t}^{T} \mu_{\omega,t'} \right) \cdot \log(1 + e^{\mathbf{b}_t \bullet \Phi_t(\mathbf{x}(\omega,t))}) + \mu_{\omega,t} (\mathbf{b}_t \bullet \Phi_t(\mathbf{x}(\omega,t))) \right] \Bigg] \right\} \tag{35}$$

From this, we can see that problem (31) can be solved by solving the period t problem

$$\max_{\mathbf{b}_t\in\mathcal{B}_t} \sum_{\omega=1}^{\Omega} \left[- \left(\sum_{t'=t}^{T} \mu_{\omega,t'} \right) \cdot \log(1 + e^{\mathbf{b}_t \bullet \Phi_t(\mathbf{x}(\omega,t))}) + \mu_{\omega,t} (\mathbf{b}_t \bullet \Phi_t(\mathbf{x}(\omega,t))) \right] \tag{36}$$

for each period t. Problem (36) has a nice structure: it is effectively a maximum likelihood estimation (MLE) problem corresponding to a binary logistic regression model. The two classes correspond to the two actions, **stop** and **continue**. The sample being used is a weighted sample with 2Ω weighted observations. Each trajectory ω corresponds to one observation where the class label is **stop** with weight $\mu_{\omega,t}$ and one observation where the class label is **continue** with weight $\sum_{t'=t}^{T} \mu_{\omega,t'} - \mu_{\omega,t} = \sum_{t'=t+1}^{T} \mu_{\omega,t'}$. When $\mathcal{B}_t = \mathbb{R}^K$, problem (36) becomes an unconstrained maximum likelihood estimation problem. This type of problem can be solved rapidly to global optimality using Newton's method; we take this approach in our numerical experiments in Section 6.

More importantly, the decomposition over periods in problem (31) yields an important distinction between Algorithm 5 and backward recursion methods like LSM. Problem (31) can be solved by solving T instances of problem (36), each of which is independent from the other. Thus, in theory, one could leverage parallel computing to simultaneously solve the T instances of problem (36). In LSM, the weights must be optimized sequentially starting from the last period to the first, which rules out the use of parallelization. This also differs from prior work on randomized optimal stopping methods, which also apply a backward recursion approach.

The second important point to note about Algorithm 5 is in regard to the $\boldsymbol{\mu}$ optimization step. When $\mathbf{b}$ is held fixed, solving problem (30) for $\boldsymbol{\mu}$ becomes

$$\max_{\boldsymbol{\mu}\in\Delta_{[\Omega]\times[T]}} \left\{ \sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} \mu_{\omega,t} \psi_{\omega,t}(\mathbf{b}) - \sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} \mu_{\omega,t} \log \mu_{\omega,t} \right\}. \tag{37}$$

It turns out that this problem can be solved in closed form. The optimal value of $\mu_{\omega,t}$ for each ω and t is

$$\mu^*_{\omega,t} = \frac{e^{\psi_{\omega,t}(\mathbf{b})}}{\sum_{\omega'=1}^{\Omega} \sum_{t'=1}^{T} e^{\psi_{\omega',t'}(\mathbf{b})}}. \tag{38}$$

If one further uses the definition of $\psi_{\omega,t}$, we can see that $\mu^*_{\omega,t}$ is

$$\mu^*_{\omega,t} = \frac{(1/\Omega)\cdot g(t,\mathbf{x}(\omega,t))\cdot \prod_{t'=1}^{t-1}(1-\sigma(\mathbf{b}_{t'}\bullet\Phi_{t'}(\mathbf{x}(\omega,t'))))\sigma(\mathbf{b}_t\bullet\Phi_t(\mathbf{x}(\omega,t)))}{(1/\Omega)\sum_{\omega'=1}^{\Omega}\sum_{\tau=1}^{T} g(\tau,\mathbf{x}(\omega',\tau))\cdot\prod_{t'=1}^{\tau-1}(1-\sigma(\mathbf{b}_{t'}\bullet\Phi_{t'}(\mathbf{x}(\omega',t'))))\sigma(\mathbf{b}_\tau\bullet\Phi_\tau(\mathbf{x}(\omega',\tau)))}, \tag{39}$$

which can be interpreted as what fraction of the overall SAA objective can be attributed to our decision in period t of trajectory ω.

The third important point to note about Algorithm 5 is that this algorithm is relatively simple to deploy, as the only tunable parameter is the choice of the initial value of $\boldsymbol{\mu}$. Here, one can potentially leverage some prior knowledge to choose $\boldsymbol{\mu}$. For example, given a heuristic policy, one could devise a choice of $\boldsymbol{\mu}$ so that $\mu_{\omega,t}$ is large whenever the heuristic policy stops at period t of trajectory ω, and otherwise $\mu_{\omega,t}$ is small. In our preliminary experimentation, what we found seems to work best across all of the instances we tested is when $\boldsymbol{\mu}$ is initialized to $1/(\Omega T)$, i.e., all periods and trajectories are equally weighted in the first solve of problem (31). (Relatedly, we note that while Algorithm 5 requires the initial $\boldsymbol{\mu}$ as an input, the weight vector $\mathbf{b}$ is initialized to the vector of all zeros. This particular detail is not of major importance because the algorithm first maximizes over $\mathbf{b}$ followed by $\boldsymbol{\mu}$, and so this initial choice of $\mathbf{b}$ will be immediately overwritten in the first iteration of the algorithm.)

Lastly, we note that the reformulation technique applied here, of using the biconjugate form of the log-sum-exp function, has been applied previously, but in a different context. In particular, Guan and Mišić (2025) apply the same type of biconjugate representation to reformulate a multi-product pricing problem involving the semi-log and log-log demand models. The difference between the approach here and the approach in Guan and Mišić (2025) is that in the latter paper, the underlying optimization problem is discrete (one chooses a price for each product from a finite set of prices). As a result, by using standard techniques in integer programming for linearizing products of binary variables and continuous variables, it is possible to reformulate the problem exactly as a mixed-integer exponential cone program. In contrast, the same approach is not applicable for our problem (30), as the main decision variable $\mathbf{b}$ is continuous. Beyond this paper, the biconjugate form of the log-sum-exp function arises in choice modeling. In this domain, the log-sum-exp function is termed the choice welfare function, which represents the expected utility of an agent maximizing a random utility, and the right-hand side maximization in (28) is the representative agent model, which is an optimization problem where an agent chooses choice probabilities to maximize an expected utility with a concave regularization term that favors diversification. We refer the interested reader to Natarajan et al. (2009), Feng et al. (2017), Yan et al. (2022) and Ruan et al. (2022) for recent work in this space.

5.3. Convergence of alternating maximization procedure to deterministic policy

In this section, we develop a convergence guarantee for the alternating maximization (AM) algorithm presented in Section 5.2. In particular, we show that the objective value of the AM iterate sequence converges to the objective value of a deterministic policy.

To setup our result, we let the sequence of iterates be denoted by $(\mathbf{b}^1, \mathbf{b}^2, \dots)$, and denote the sequence of objective values by $(\hat{J}_R(\mathbf{b}^1), \hat{J}_R(\mathbf{b}^2), \hat{J}_R(\mathbf{b}^3), \dots)$. We make the following three assumptions.

Assumption 5. *For all* $\omega \in [\Omega]$, $t \in [T]$, $\Phi_t(\mathbf{x}(\omega, t)) \neq \mathbf{0}$.

Assumption 6. $\liminf_{s\to\infty} \frac{|\mathbf{b}_t^s \bullet \Phi_t(\mathbf{x}(\omega,t))|}{\|\mathbf{b}_t^s\|_2 \|\Phi_t(\mathbf{x}(\omega,t))\|_2} > 0$ *for all* $t \in [T]$, $\omega \in [\Omega]$.

Assumption 7. $\limsup_{s\to\infty} \min_{1\le t\le T} \|\mathbf{b}_t^s\|_2 = +\infty$.

With these assumptions, we can establish the following result.

Proposition 1. *Under Assumptions 5, 6 and 7,* $\hat{J}_R(\mathbf{b}^s) \to \hat{J}_D(\mathbf{v})$ *as* $s \to \infty$ *for some deterministic policy weight vector* $\mathbf{v} \in \mathbb{R}^{KT}$.

We prove this result in Section EC.1.6 of the ecompanion. The proof consists of showing that the objective values from AM converge to a finite limit, and then showing that within the sequence of iterates produced by AM, there exists a subsequence for which the magnitudes of the weight vectors explode to infinity, and within that subsequence, there is a sub-subsequence for which the normalized weight vectors converge to a point on the set $\mathcal{S}^T$, where $\mathcal{S} = \{\mathbf{b} \in \mathbb{R}^K \mid \|\mathbf{b}\|_2 = 1\}$ is the unit sphere in $\mathbb{R}^K$.

Assumption 7 plays an important role in the proof of the result because it ultimately allows one to extract a subsequence for which the magnitudes of the weight vectors go to infinity, while the weight vectors eventually converge to a single direction. While we do not have a proof, we conjecture that Assumption 7 in general holds. Intuitively, with each update of $\boldsymbol{\mu}$, the objective function weights $\boldsymbol{\mu}$ in problem (36) become more biased based on the stopping probabilities induced by the current weight vector, because with each update of $\boldsymbol{\mu}$, each $\mu_{\omega,t}$ is proportional to the probability of stopping based on the current weight vector $\mathbf{b}$. As this happens, problem (36) becomes closer and closer to a logistic regression MLE problem for a perfectly separable data set. This results in the magnitudes of the coefficient vectors $\mathbf{b}_1, \dots, \mathbf{b}_T$ becoming large, which causes the weights in problem (36) to become more biased, causing the coefficient vectors $\mathbf{b}_1, \dots, \mathbf{b}_T$ to become even larger in the next iteration. In our numerical experiments, Assumption 7 always holds: in fact, we generally see stronger behavior, in that the magnitudes of the weight vectors in each period are increasing with each iteration (i.e., for each t, $\|\mathbf{b}_t^1\|_2 \le \|\mathbf{b}_t^2\|_2 \le \|\mathbf{b}_t^3\|_2 \le \dots$, which implies $\lim_{s\to\infty} \min_{1\le t\le T} \|\mathbf{b}_t\|_2 = +\infty$).

With regard to the other assumptions, Assumption 5 is a reasonable assumption, as encountering a feature vector $\Phi_t(\mathbf{x}(\omega,t))$ which is identically zero would mean that any randomized policy randomizes 50-50 between stopping and continuing at that period in that trajectory, no matter what $\mathbf{b}_t$ is; it is also automatically satisfied if $\Phi_t(\cdot)$ includes the constant basis function. Assumption 6 is a more technical assumption, which we can interpret as requiring that the angle between $\mathbf{b}_t$ and $\Phi_t(\mathbf{x}(\omega,t))$ is bounded away in the limit from 90°; this is needed to rule out a case where the limiting deterministic policy weight vector $\mathbf{v}$ is one for which $\mathbf{v}_t \bullet \Phi_t(\mathbf{x}(\omega,t))$ is exactly zero. Although we do observe that Assumption 6 holds numerically in the instances we have tested, we do not have as strong an intuitive justification for this assumption holding as we do for Assumption 7.

In terms of the significance of Proposition 1, we regard this result as a positive result. In particular, suppose that the reward function g is such that the random variable $g(t,\mathbf{x}(t))$ has a bounded density. Then almost surely, any two deterministic linear policies $\pi_{\mathbf{b}}, \pi_{\mathbf{b}'} \in \Pi_{\mathcal{B}}$ which induce different stopping behavior – i.e., $\tau_{\pi_{\mathbf{b}},\omega} \neq \tau_{\pi_{\mathbf{b}'},\omega}$ for some $\omega \in [\Omega]$ – must have distinct rewards, i.e., $\hat{J}_D(\mathbf{b}) \neq \hat{J}_D(\mathbf{b}')$. On the other hand, recall that $\hat{J}_R(\tilde{\mathbf{b}})$ of a finite randomized linear policy weight vector $\tilde{\mathbf{b}}$ can be regarded as the expected reward of a randomly chosen deterministic linear policy (Theorem 1). Additionally, under the assumptions of Theorem 1, it is not difficult to see that any finite $\tilde{\mathbf{b}}$ will induce non-degenerate randomization over two or more deterministic linear policies with different stopping behavior. This implies that a randomized policy with a finite weight vector $\tilde{\mathbf{b}}$ cannot be optimal, i.e., cannot solve $\sup_{\tilde{\mathbf{b}} \in \tilde{\mathcal{B}}} \hat{J}_R(\tilde{\mathbf{b}})$. Stated differently, the optimal randomized policy SAA objective value $\sup_{\tilde{\mathbf{b}} \in \tilde{\mathcal{B}}} \hat{J}_R(\tilde{\mathbf{b}})$ is not attained by any finite $\tilde{\mathbf{b}}$, but is attained in the limit by considering weight vectors of increasing magnitude. Proposition 1 ensures that we will not converge to a truly random policy, which cannot be globally optimal, but instead to a "degenerate" randomized policy (in other words, a deterministic policy), which has the potential, though is not guaranteed, to be globally optimal.

6. Numerical experiments

In this section, we numerically test our randomized policy approach in two optimal stopping problems. The first that we consider is a standard option pricing problem, that of pricing a Bermudan max-call option, previously considered in a number of papers (e.g., Desai et al. 2012, Ciocan and Mišić 2022, Sturt 2023). We define this 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 in the case of 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. In Section 6.4, we consider a variation of the problem where the barrier price varies over time.

The second problem that we consider is that of optimally stopping an exponentially-transformed fractional Brownian motion process. We define this problem and present our results in Section 6.5.

The final problem that we will consider is that of optimally timing an exit from a cryptocurrency position. We consider two different types of instances calibrated using real data, based on taking a cross-sectional versus a longitudinal approach to a set of cryptocurrencies. We describe the calibration process for these two types of instances and present our results in Section 6.6.

We implement our methods in the Julia programming language, version 1.10.4 (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 10.01 (Gurobi Optimization, Inc. 2022). All our experiments are executed on Amazon Elastic Compute Cloud (EC2), on a single instance of type `m6a.48xlarge` (AMD EPYC 7R13 processor with 192 virtual CPUs and 768 GBs of memory).

6.1. Background on Bermudan max-call option with barrier

The first optimal stopping problem that we will focus on is that of 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) and Sturt (2023). We consider the same family of problem instances used in those papers; due to the somewhat involved mechanics of this problem instance, we use this section to briefly review the details of this problem family.

In this family of problem instances, 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:

$$y(t) = \mathbb{I}\left\{ \max_{1 \leq i \leq n, 1 \leq t' \leq t} p_i(t') < B \right\}. \tag{40}$$

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

$$g'(t) = y(t) \cdot \max\left\{ 0, \max_{1 \leq i \leq n} p_i(t) - K \right\}. \tag{41}$$

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 $t = 0$.

Our comparisons will primarily focus on 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). In addition to LSM and PO, we also test the tree method of Ciocan and Mišić (2022) in two of our sets of experiments (Sections 6.3 and 6.4). The inclusion of the tree method is intended as an additional benchmark and to provide a relative sense of the performance of the RPO method. However, we note here that the tree method of Ciocan and Mišić (2022) is fundamentally different from LSM, PO and RPO, in that it produces policies structured as trees with axis-aligned splits, which in general will be more expressive than a deterministic/randomized linear policy (mirroring how tree models exhibit lower bias than linear models in classification and regression in machine learning). As a result, we wish to emphasize that the primary goal of our experiments is to compare LSM, PO and RPO, which all operate within the same policy class, and to demonstrate that RPO is a more reliable method for obtaining best-in-class policies with a fixed basis function architecture. For this reason, we also do not consider more sophisticated optimal stopping methods, such as the deep neural network approach of Becker et al. (2019); again, our goal is to show that optimizing over randomized policies using our alternating maximization algorithm is the "right" way of finding good policies with respect to a fixed basis function architecture, rather than to argue the relative benefits of fixing a basis function architecture versus adopting a nonparametric approach (where the policy is represented as a tree or a neural network).

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 $i \in [n]$.
- 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 pathwise optimization method, we follow Desai et al. (2012) in generating 500 inner samples. In our implementation of the tree method, we use the construction heuristic of Ciocan and Mišić (2022) with the same relative improvement tolerance as that paper ($\gamma = 0.005$).

In our implementation of the RPO approach, we use our alternating maximization approach (Algorithm 5). We initialize the weight vector $\boldsymbol{\mu}$ so that $\mu_{\omega,t} = 1/(\Omega T)$, and set the termination parameter to $\epsilon = 10^{-3}$ (i.e., the additive improvement in the log-transformed average reward falls below 10^{-3}). We do not impose any constraints on the per-period weight vector, i.e., $\mathcal{B}_t = \mathbb{R}^K$. We solve the maximization step over $\mathbf{b}$ by solving T per-period maximization problems (problem (36)), where each instance of problem (36) is solved using Newton's method, implemented using the Optim package in Julia (Mogensen and Riseth 2018). We note that we also solve problem (36) over all t's serially, i.e., without parallel computing.

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

$$g'(t) = y(t) \cdot \max\{0, p_1(t) - K\}, \tag{42}$$

$$y(t) = \mathbb{I}\left\{ \max_{1 \leq t' \leq t} p_1(t') < B \right\}. \tag{43}$$

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 $\Omega = 20,000$ 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. 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.

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.

			Initial price	
Method	Basis functions	$\bar{p} = 90$	$\bar{p} = 100$	$\bar{p} = 110$
LSM	ONE	6.46 (0.006)	10.82 (0.011)	16.46 (0.018)
LSM	ONE, PAYOFF	11.36 (0.016)	16.61 (0.023)	22.00 (0.023)
PO	ONE	9.48 (0.012)	14.78 (0.011)	20.67 (0.016)
PO	ONE, PAYOFF	9.09 (0.060)	16.00 (0.036)	22.68 (0.016)
RPO	ONE, PAYOFF	**12.42 (0.014)**	**17.67 (0.016)**	**23.22 (0.013)**
PO-UB	ONE, PAYOFF	12.54 (0.011)	17.91 (0.019)	23.54 (0.017)
PO-UB	ONE	18.26 (0.025)	25.46 (0.031)	32.47 (0.031)

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

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 92% 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 9.3%. Comparing to PO with ONE and with ONE and PAYOFF, the randomized policy approach attains an improvement of up to 31% and 37%, 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 1.4% 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, LSM and PO can return policies that are substantially suboptimal.

It is also interesting to consider what the thresholds produced by the different methods look like. Figure 3 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 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, the threshold decreases at a very gentle rate until about $t = 48$, where the threshold begins to decrease much more quickly.

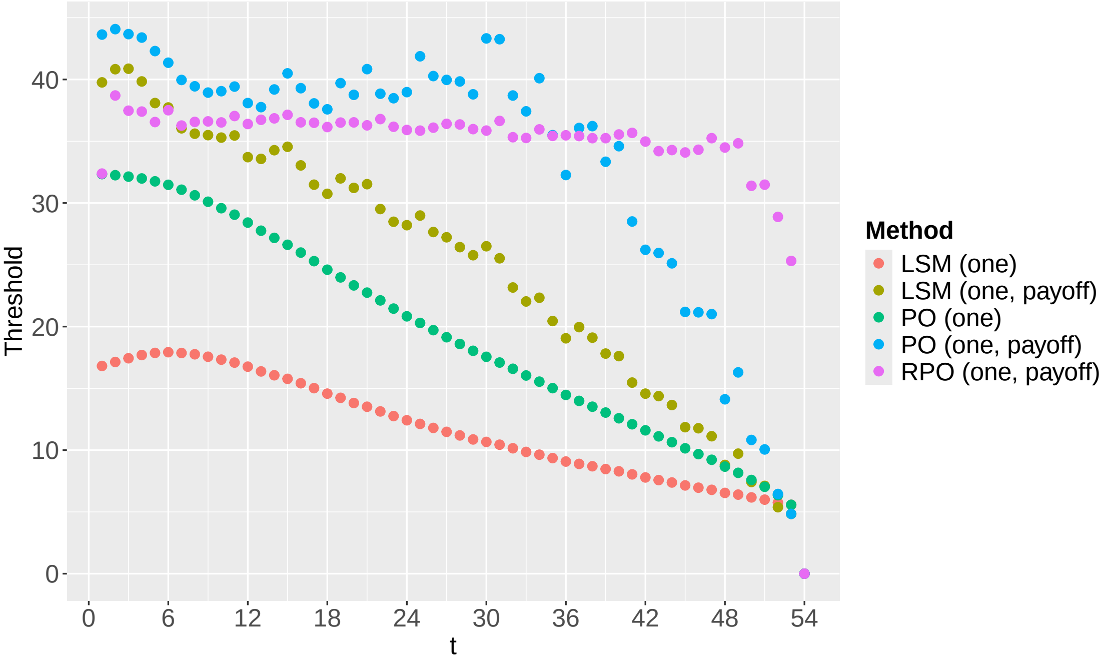


Figure 3 Plot of thresholds for policies in $n = 1$ experiment.

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.2.1 of the ecompanion.

We again test the LSM, PO and RPO methods with a variety of basis function architectures, as well as the tree method. 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 performance of the tree method and 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.) For each such group, we indicate the best out-of-sample reward in bold. The tree policies are presented as a separate group, as the class of tree policies does not coincide with the deterministic linear policy class for the basis function architecture.

From this table, we can see that within each policy class, the RPO method obtains policies that outperform those obtained by LSM and PO. 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 45.39, which is an improvement of 2.6%. Similarly, for that same policy class, the improvement of RPO over the best PO policy (reward of 44.82) is about 1.3%.

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 PAYOFF and either KOIND or ONE as the basis functions (45.39 for $\bar{p} = 90$, 51.31 for $\bar{p} = 100$, 54.46 for $\bar{p} = 110$). The best performance for the LSM method across any of the basis function architectures is substantially lower (44.26 for $\bar{p} = 90$, 50.09 for $\bar{p} = 100$, 53.15 for $\bar{p} = 110$). The best performance for the PO method is better, but still lower (44.82 for $\bar{p} = 90$, 50.90 for $\bar{p} = 100$, 54.35 for $\bar{p} = 110$). Outside of LSM and PO, we note that the best RPO policy (with PAYOFF and either KOIND or ONE) achieves comparable performance to the best tree policy (any of the policies that include PAYOFF and TIME).

Beside out-of-sample reward, it is also useful to compare the methods in terms of computation time. Due to space constraints, we present the average computation time of each method in Table EC.3 of Section EC.2.1 of the ecompanion. Each entry in the table shows the average computation time for each of the methods. For LSM, this is just the time to apply the LSM algorithm. For PO, this time includes the time to solve the PO linear program using Gurobi and the time to execute the regression, as well as the time to generate the inner paths and the time to formulate problem in JuMP. For RPO, this is the time required to run the AM algorithm. 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 100 seconds (approximately 1.5 minutes) on average. 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 100 seconds ($\bar{p} = 100$, PO with ONE; just under 2 minutes) to 1200 seconds ($\bar{p} = 100$, PO with PRICESKO, PRICES2KO, KOIND, PAYOFF; roughly 20 minutes). The majority of this time comes from the generation of the inner paths, which is in general a computationally intensive task.

The tree method generally requires less time than the RPO method. However, we note that the computation time for the tree method exhibits a more complex dependence on the set of available split variables, and is not monotone in the number of split variables. For example, with PRICES and TIME, the construction method requires more time than with PAYOFF and TIME because a single split on PAYOFF corresponds to n splits on PRICES; on the other hand, PRICES, TIME, PAYOFF is faster than PRICES, TIME because the construction method greedily chooses to split on PAYOFF than on each individual prices. On the other hand, the computation time of the RPO method generally grows with the number of basis functions.

6.4. Experiment #3: multiple assets with time-varying barrier

In this experiment, we consider the following modification to the Bermudan max-call option: rather than fixing the knock-out barrier price to a constant B, we instead allow the barrier price to vary deterministically as the function $B(t) = B_0 e^{\delta t}$. This type of max-call option was previously considered in Sturt (2023). We consider the same parameter settings as in Section 6.3, except that we vary $n \in \{8, 16, 32\}$, and we set $B_0 = 150$, and set δ so that the annualized growth rate of the barrier price is 25% (for $T = 54$ periods over 3 years, this corresponds to $\delta = 0.25 \cdot (3/54) = 0.013888\ldots$). We note that these parameters exactly match those used in Sturt (2023) (see Section 5.2 of that paper). We compare the randomized policy approach against LSM, PO and the tree method. For each $\bar{p}$ and n, we generate 10 replications of 20,000 training trajectories and 100,000 testing trajectories. As in our experiments in Section 6.3, we also report the biased upper bound obtained as the objective value of the PO linear program. For simplicity, and due to the larger number of assets in these instances, we restrict our basis functions to ONE, PAYOFF, KOIND, PRICES, PRICESKO.

Table 3 shows the average out-of-sample reward, over the ten replications, for each of the methods with different choices of basis functions (for RPO, LSM and PO) and state variables (for the tree method), for $n = 16$. (The results for $n = 8$ and $n = 32$ are given in Section EC.2.2 of the ecompanion.) From this table, we can again see that holding the basis function architecture fixed, RPO always obtains the best policy out of the three linear policy approaches (RPO, LSM and PO). Additionally, whereas RPO and the tree approach were comparable when the barrier price was constant in our experiments in Section 6.3, the two methods now behave differently, and RPO now yields a significant improvement over the tree method in some cases (for example, for $\bar{p} = 100$, RPO with ONE, PAYOFF yields an average reward of 84.17, vs the tree method with PRICES, TIME, PAYOFF, KOIND as candidate split variables yields an average reward of 79.94).

6.5. Experiment #4: exponentially-transformed fractional Brownian motion

In this separate set of experiments, we depart from the Bermudan max-call option considered in Sections 6.1 - 6.4 and turn our attention to a more complex optimal stopping problem. In particular, we consider the stochastic process $\{y(t)\}_{t=1}^{T}$, where $y(t)$ is defined as $y(t) = e^{a+bB(\delta t)}$, $\delta > 0$ is a discretization parameter, and the process $\{B(t)\}_{0\leq t\leq 1}$ is a fractional Brownian motion on the interval $[0,1]$. A fractional Brownian motion process $\{B(t)\}_{0\leq t\leq 1}$ on the interval $[0,1]$ is a continuous time stochastic process which satisfies

$$\begin{aligned}
&B(0) = 0, \text{ a.s.},\\
&\mathbb{E}[B(t)] = 0, \quad \forall\, t \in [0,1],\\
&\mathbb{E}[B(t)B(s)] = (1/2)(|t|^{2H} + |s|^{2H} - |t-s|^{2H}), \quad \forall t, s \in [0,1],
\end{aligned}$$

where $H \in (0,1)$ is the Hurst parameter. Fractional Brownian motion can be viewed as a generalization of ordinary Brownian motion where the increments are allowed to be correlated: when $H < 1/2$, the increments are negatively correlated, whereas when $H > 1/2$, the increments are positively correlated. (When $H = 1/2$, $\{B(t)\}_{0\leq t\leq 1}$ coincides with ordinary Brownian motion.)

This optimal stopping problem is interesting for a couple of reasons. First, a fractional Brownian motion process is not a Markov process, and it is reasonable to expect that a good policy would need to use information about the process prior to period t. Second, the presence of the exponential function $s \mapsto e^{a+bs}$ in the definition of y induces the type of behavior seen in our pathological instances in Section 4, where LSM carried out with basis functions that explode in magnitude can lead to suboptimal behavior, leading to a challenging problem. We note that fractional Brownian motion has attracted interest in the recent optimal stopping research literature (see, for example, Section 4.3 of Becker et al. 2019). We also note that models similar to this one have been proposed as models of asset prices (see, for example, the fractional Samuelson/Black-Scholes model in Cheridito 2003).

We set $T = 50$, and $\delta = 1/50$. We set $a = 4$, $b = 3$. We focus on $H \in \{0.05, 0.1, 0.2, 0.3\}$, as we found that for H of 0.4 or higher, the best tree and RPO policies are comparable. We compare LSM, RPO and the tree method, with the latter two methods implemented with the same hyperparameters as in Section 6.1. We define the state variable $\mathbf{x}(t)$ as the T-dimensional vector

$$\mathbf{x}(t) = (y(t), y(t-1), y(t-2), \ldots, y(1), 0, 0, \ldots, 0), \tag{44}$$

and the state space as $\mathcal{X} = \mathbb{R}^T$. We define the reward function as $g(t,\mathbf{x}) = x_1$.

For LSM and RPO, we consider the constant basis function ONE as well as three different sets of basis functions. We use PH1:j to denote the set of j functions $\phi_{t,1}, \ldots \phi_{t,j}$ defined as

$$\phi_{t,i}(\mathbf{x}) = \begin{cases} y(t-i+1), & \text{if } t-i \geq 0,\\ 0, & \text{otherwise,}\end{cases} \tag{45}$$

for $i = 1, \ldots, j$. In other words, PH1:j gives the rewards of the process over the last j periods, including the current one. We vary $j \in \{2,3,4,5,10,15,20,50\}$, where we note that $j = 50$ implies that at each period, all of the prior rewards can be used by the policy. We similarly use PH2:j to denote the same set of functions with the first function $\phi_{t,1}$ omitted. Lastly, we also use LOGPH1:j to denote the set of functions in PH1:j upon logarithmic transformation, i.e., the collection of functions of the form $\mathbf{x} \mapsto \log y(t-i+1)$. For the tree model, we test it using combinations of TIME, together with either PH1:1 (current payoff) or PH1:50 (the entire payoff history). Given a fixed H, we generate a single replication by simulating $\Omega = 100,000$ trajectories of the system to serve as the training set, and a separate set of 100,000 trajectories to serve as the test set. For each H, we generate 10 replications.

Table 4 compares the average out-of-sample reward of LSM and RPO over the ten replications for different values of H. The table is organized so that the deterministic linear policies that use the same untransformed historical rewards are grouped together. For example, the three rows for LSM with ONE, PH1:3, LSM with ONE, PH2:3 and RPO with ONE, PH1:3 are grouped together, as all three policies involve stopping if $b_{t,0} + b_{t,1}y(t) + b_{t,2}y(t-1) + b_{t,3}y(t-2) > 0$ for appropriate choices of $b_{t,0}, \ldots, b_{t,3}$. Besides these groups of policies, we separately group the RPO policies that use the LOGPH1:j basis function set, and the tree policies.

What we find from this table is that as in our previous experiments, holding the policy class fixed, RPO outperforms LSM and often significantly so. For example, for $H = 0.2$ and using the payoff history up to 5 periods back (i.e., LSM/RPO with PH1:5 or PH2:5), the average out-of-sample reward of RPO is 8886, whereas the reward of LSM for either ONE, PH1:5 or ONE, PH2:5 is less than half of that.

Besides the comparison of LSM and RPO with a fixed policy class, it is also interesting to compare the RPO policies against the tree method. For all values of H, the best RPO policy is better than the best tree policy. Comparing the RPO policies with the LOGPH1:j basis function sets, the best RPO policy within that collection is even better than the best tree policy.

This last comparison brings us to another interesting aspect of the RPO approach: for the RPO policies with the basis functions ONE, LOGPH1:j, the policy stops if $b_{t,0} + \sum_{j'=1}^{j} b_{t,j'} \log y(t-j'+1) > 0$ and continues otherwise, for appropriately chosen $b_{t,0}, \ldots, b_{t,j}$. Note that these policies are distinct from those produced by LSM, as a deterministic linear policy produced by LSM necessarily must include the untransformed reward in the underlying linear function. This highlights another advantage of the RPO framework: it is capable of naturally producing policies that do not need to use the (untransformed) reward function $g(t, \mathbf{x})$ as a basis function.

6.6. Experiment #5: when to optimally exit a cryptocurrency position

In this section, we consider an experiment that uses real data. In particular, we will consider the problem of when to sell a position in cryptocurrency. A cryptocurrency is a form of digital currency for which transactions are stored using a digital ledger, and which can be purchased using fiat currency. Cryptocurrencies have exhibited astronomical price growth: for example, Dogecoin (symbol DOGE) was worth \$0.000299312 when it first launched on December 16, 2013, and is worth approximately \$0.2240 as of August 14, 2025, which corresponds to price growth by a factor of over 700. At the same time cryptocurrencies are also notorious for their volatility; for example, Bitcoin (symbol BTC) experienced five different days in 2022 on which its price dropped by over 10% over the course of a single day.

The data for these instances is obtained from CoinCodex (CoinCodex 2025) using their freely available API. We first queried the API to obtain the list of all cryptocurrencies that are registered with the platform as of August 14, 2025. From this set of cryptocurrencies, we filtered out those which are no longer trading. From the remaining set of cryptocurrencies, we extracted the 1000 largest by market capitalization as of August 14, 2025. For each of these cryptocurrencies, we queried daily price data from the first day that the currency began trading through to August 14, 2025. For each cryptocurrency c, we let s_c denote the first day of trading. We let $p_{c,d}$ denote the price of cryptocurrency c at the start of day d in US dollars. We let $d_{\max}$ denote the last day of available data (August 14, 2025).

Using this data, we considered two types of experiments:

Cross-sectional experiment: In this experiment, we first define two parameters: a burn-in period B and a duration T, both in days. We select a cryptocurrency c for which we have price data from day $s_c + B + 1$ through to $s_c + B + T$, i.e., we require $s_c + B + T \leq d_{\max}$. For this cryptocurrency, we define the sample path $x(c,t) = p_{c,s_c+B+t} / p_{c,s_c+B+1}$ where t ranges from 1 to T. Thus, each cryptocurrency c is associated with a sample path $x(c,1), x(c,2), \ldots, x(c,T)$, where $x(c,t)$ tells us what the price of the cryptocurrency is relative to its price on day $s_c + B + 1$. The reward function is defined as $g(t,x) = x$, i.e., the current relative price of the cryptocurrency. An instance is then defined by picking a fraction $q \in (0,1)$ of the sample paths (i.e., cryptocurrencies) randomly without replacement to serve as the training set, and the remainder to serve as the test set. We use the training set to obtain a policy, and evaluate its performance on the test set. Figure 4 visualizes the trajectories for ten randomly chosen cryptocurrencies.

The idea of this experiment is as follows. Many cryptocurrencies are volatile and can sometimes experience extreme price movements after they are initially launched. For example, the Akita Inu token (symbol AKITA) grew from 2.68×10^{-8} dollars a token when it was launched on February 27, 2021, reaching 2.805×10^{-5} dollars a token during the day of May 11, 2021, which corresponds to an over one thousand fold increase. Imagine now that a trader waits until a certain amount of time has passed after a cryptocurrency is launched, purchases 1 dollar worth of the cryptocurrency,

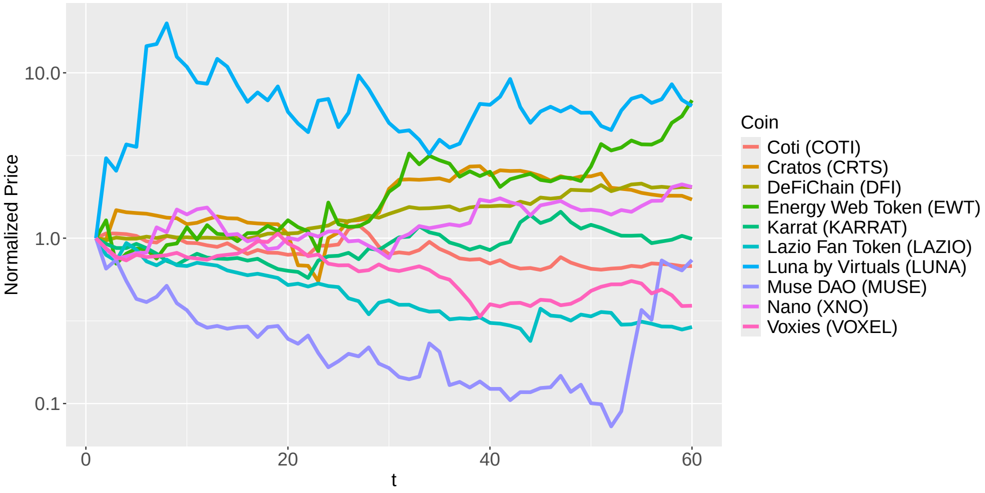


Figure 4 **Plot of $x(\omega, t)$ (shown on log scale) for ten randomly chosen cryptocurrencies ($B = 1$, $T = 60$).**

and then follows some kind of stopping policy to decide when to exit from (i.e., sell) their position within the ensuing T days. (Note that the trader does not intend to hold their position past T days from when they buy in, and their reward if they do not stop by day T is zero.) In this experiment, we are using the training set to learn a good policy for this trading setting, and then using the testing set to evaluate what our profit will be. The average reward on the test corresponds to the average value of our investment over all of the cryptocurrencies in our test set. A value of 1 means that in the aggregate, we broke even; a value greater than 1 means that we made a profit.

Longitudinal experiment: In this experiment, we create an instance as follows. For a given cryptocurrency c, we take all of the data from s_c to $d_{\max}$. We then divide this discrete set of days into consecutive intervals of T days. This results in $\bar{\Omega} = \lfloor (d_{\max} - s_c + 1)/T \rfloor$ intervals. We denote each interval by ω, and define the sample path as the sequence $x(\omega, 1), \ldots, x(\omega, T)$, where $x(\omega, t) = p_{c, s_c + (\omega-1)T+t} / p_{c, s_c + (\omega-1)T+1}$. Thus, each value $x(\omega, t)$ gives us the price of the cryptocurrency relative to its price at the beginning of interval ω. As in the cross-sectional experiment, the reward function is defined as $g(t, x) = x$, and the reward is assumed to be zero if we do not stop by day T.

For some fraction $q \in (0, 1)$, we take the first $\Omega = \lfloor q\bar{\Omega} \rfloor$ sample paths as the training data, and the remaining bar $\bar{\Omega} - \Omega$ sample paths as the test data. The training and test sets for a fixed cryptocurrency c define an instance. For each instance, we use the training set to construct a stopping policy and then evaluate its reward on the test set. We restrict our focus to cryptocurrencies for which we have at least $\bar{\Omega} \geq 20$ trajectories in total, resulting in a set of 612 cryptocurrencies.

The idea of this experiment is as follows. Suppose that we are interested in a particular cryptocurrency. We wish to regularly trade this cryptocurrency in the following way: at the start of an interval of T days, we purchase 1 dollar of the cryptocurrency, and then determine when to sell this position over the next T days. In this type of setup, an out-of-sample reward that is greater than 1 implies that on average we are able to make a profit when we apply the stopping policy repeatedly in the future.

For the cross-sectional experiment, we set $q = 0.7$ and independently at random generate 500 instances for each B. We set $T = 60$, and vary $B \in \{1, 15, 30, 60, 180\}$. For the longitudinal experiment, we set $q = 0.7$ to determine how many intervals are used for training versus testing. We compare the RPO, LSM and tree methods with a variety of basis function architectures using the basis function set PH used in our experiments in Section 6.5.

Table 5 shows the average out-of-sample reward for the cross-sectional experiment. What we find is that for many values of B, the best performing policy is obtained by RPO for some choice of basis functions. With regard to LSM, the LSM policy with ONE or ONE and PH1:1 tends to perform best across all of the values of B, while the reward for the other architecture tends to be significantly lower. As in our previous experiments, for a fixed policy class, the RPO policy with ONE, PH1:j always outperforms LSM with ONE, PH2:j and with ONE, PH1:j. With regard to the tree method, the tree method tends to perform well for high values of B (60, 180); in these settings, the tree policies tend to achieve the second highest reward compared to the best RPO policy. However, for lower values of B (1 and 15), the tree policies perform worse, and there is a larger separation in reward between the RPO policies and the tree policies.

To understand the difference between the tree policies and the RPO policies, it is worth comparing the tree policy with TIME, PH1:1 and the RPO policy with ONE,PH1:1 across the different values of B. The RPO policy here effectively involves setting a threshold for the reward for every period. The family of tree policies with TIME, PH1:1 can replicate such a policy, but to do so, the construction heuristic must make a large number of splits (to fully resolve the time period, it would need to make $T - 1$ splits on the TIME variable). Thus, in cases where good policies need to involve reward thresholds that vary a lot from period to period, we would expect RPO to perform better than the tree method, as the setup is more favorable to the RPO approach. On the other hand, when there exist good policies with thresholds that exhibit low variation, then one would expect the tree approach to either perform comparably or better than RPO. We note that for $B \in \{1, 15\}$, the sample paths exhibit a large amount of volatility (which is also reflected in the high average rewards), which would agree with the former case where a fully flexible policy is appropriate, whereas for $B \in \{60, 180\}$, the sample paths exhibit less volatility (reflected in the lower average rewards), which would agree with the latter case where one can do well with a simpler policy.

Table 6 presents the average out-of-sample reward of the different methods for the longitudinal experiment, where the average is taken over the 612 cryptocurrencies. In addition to the ordinary mean, we also compute the trimmed mean for $\alpha = 0.01$ and $\alpha = 0.025$, where we recall that a trimmed mean with α is the mean of a set of numbers with the smallest α and largest α fraction of the numbers removed. The reason for considering the trimmed mean is to reduce the influence of extreme instances (i.e., cryptocurrencies) for which the out-of-sample reward may be extremely large or extremely small. For the ordinary mean, the RPO policy with ONE, PH1:2 performs best (reward of 1.318); LSM with ONE gives the second highest reward (1.291), and the tree policy with TIME,PH1:10 and TIME,PH1:1 are slightly lower (1.274 and 1.255, respectively). Interestingly, for the two trimmed means, the RPO policies with ONE, PH1:j for $j = 1, 2, 3$ generally perform the best, while the performance of LSM with ONE and the tree policy with ONE, PH1:1 and ONE, PH1:10 both become considerably smaller. This highlights another strength of the RPO approach: RPO is able to consistently generate high quality policies across most cryptocurrencies, as opposed to generating policies for some cryptocurrencies that lead to outsized gains.

7. Conclusion

In this paper, we consider the problem of designing for high-dimensional optimal stopping problems. This problem was motivated by the observation that least squares Monte Carlo may not necessarily identify the best-in-class policy for a given basis function architecture. We established that optimizing over randomized policies is equivalent to optimizing over deterministic policies, and that the problem of optimizing over randomized policies can be solved at least approximately using an iterative algorithm based on alternating maximization, that allows the problem to be decomposed over periods. In numerical experiments, we showed that policies obtained by our approach have the same structure as those obtained by LSM and pathwise optimization, while yielding higher out-of-sample reward, and we also demonstrated settings where a parametric policy produced by our approach is preferable to a nonparametric tree policy. These results are complemented by our worst-case analysis of LSM, in which we show the existence of pathological instances where the best-in-class policy is either perfectly optimal or arbitrarily close to optimal, while LSM can perform arbitrarily poorly relative to the optimal policy.

An interesting question for future research is how the randomized policy framework can be extended beyond the optimal stopping problem here. Under the umbrella of this broad question, a promising direction is to extend the framework here to other types of stopping problems, such as search problems. Another direction is to study how the methodology here can be generalized to stochastic dynamic programming problems outside of optimal stopping. Yet another direction is to consider the application of this methodology to optimal stopping problems outside of financial applications; for example, to healthcare problems (Cheng et al. 2025), problems involving real options for energy production (Nadarajah et al. 2017, Yang et al. 2024) and marketing problems involving the timing of when new products or services are introduced (Kash et al. 2023).

References

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

C. Bayer, D. Belomestny, P. Hager, P. Pigato, and J. Schoenmakers. Randomized optimal stopping algorithms and their convergence analysis. *SIAM Journal on Financial Mathematics*, 12(3):1201–1225, 2021.

S. Becker, P. Cheridito, and A. Jentzen. Deep optimal stopping. *Journal of Machine Learning Research*, 20 (74):1–25, 2019.

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.

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.

G. Cheng, J. Xie, Z. Zheng, H. Luo, and O. C. Ooi. Extubation decisions with predictive information for mechanically ventilated patients in the icu. *Management Science*, 71(7):6069–6091, 2025.

P. Cheridito. Arbitrage in fractional brownian motion models. *Finance and stochastics*, 7(4):533–553, 2003.

D. F. Ciocan and V. V. Mišić. Interpretable optimal stopping. *Management Science*, 68(3):1616–1638, 2022.

CoinCodex. API Documentation — CoinCodex, 2025. URL `https://coincodex.com/page/api/`.

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.

G. Feng, X. Li, and Z. Wang. On the relation between several discrete choice models. *Operations research*, 65(6):1516–1525, 2017.

M. R. Garey and D. S. Johnson. *Computers and intractability*. W. H. Freeman New York, 1979.

M. Glanzer, S. Maier, and G. Ch. Pflug. Guaranteed bounds for optimal stopping problems using kernel-based non-asymptotic uniform confidence bands. *European Journal of Operational Research*, 2025.

X. Guan and V. V. Mišić. Randomized robust price optimization. *Management Science*, 2025. Forthcoming.

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.

I. A. Kash, P. B. Key, and S. I. Zoumpoulis. Optimal pricing and introduction timing of technology upgrades in subscription-based services. *Operations Research*, 71(2):665–687, 2023.

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.

P. K. Mogensen and A. N. Riseth. Optim: A mathematical optimization package for julia. *Journal of Open Source Software*, 3(24):615, 2018. doi: 10.21105/joss.00615. URL `https://doi.org/10.21105/joss.00615`.

S. Nadarajah, F. Margot, and N. Secomandi. Comparison of least squares monte carlo methods with applications to energy real options. *European Journal of Operational Research*, 256(1):196–204, 2017.

K. Natarajan, M. Song, and C.-P. Teo. Persistency model and its applications in choice modeling. *Management Science*, 55(3):453–469, 2009.

M. Okamoto. Distinctness of the eigenvalues of a quadratic form in a multivariate sample. *The Annals of Statistics*, pages 763–765, 1973.

R. T. Rockafellar. *Convex analysis*, volume 18. Princeton University Press, 1970.

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

Y. Ruan, X. Li, K. Murthy, and K. Natarajan. A nonparametric approach with marginals for modeling consumer choice. *arXiv preprint arXiv:2208.06115*, 2022.

B. Sturt. A nonparametric algorithm for optimal stopping based on robust optimization. *Operations Research*, 71(5):1530–1557, 2023.

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.

Z. Yan, K. Natarajan, C. P. Teo, and C. Cheng. A representative consumer model in data-driven multiproduct pricing optimization. *Management Science*, 68(8):5798–5827, 2022.

B. Yang, S. Nadarajah, and N. Secomandi. Least squares monte carlo and pathwise optimization for merchant energy production. *Operations Research*, 72(6):2758–2775, 2024.

Method	Basis functions	$\bar{p} = 90$	$\bar{p} = 100$	$\bar{p} = 110$
LSM	KOind	39.43 (0.022)	48.09 (0.022)	53.18 (0.016)
LSM	KOind, payoff	44.26 (0.017)	50.09 (0.014)	53.15 (0.013)
PO	KOind	43.90 (0.016)	50.86 (0.012)	54.35 (0.014)
PO	KOind, payoff	44.82 (0.020)	50.90 (0.009)	53.90 (0.014)
RPO	KOind, payoff	**45.39 (0.015)**	**51.31 (0.012)**	**54.46 (0.012)**
PO-UB	KOind	49.30 (0.024)	53.47 (0.012)	55.69 (0.009)
PO-UB	KOind, payoff	46.20 (0.018)	52.04 (0.032)	55.05 (0.020)
LSM	one	33.81 (0.027)	38.68 (0.014)	43.15 (0.017)
LSM	one, payoff	41.14 (0.038)	43.22 (0.037)	45.01 (0.040)
PO	one	41.11 (0.025)	45.93 (0.018)	48.84 (0.026)
PO	one, payoff	22.27 (0.096)	16.24 (0.131)	10.94 (0.211)
RPO	one, payoff	**45.39 (0.015)**	**51.31 (0.012)**	**54.46 (0.012)**
PO-UB	one	52.19 (0.026)	57.44 (0.015)	60.33 (0.013)
PO-UB	one, payoff	46.42 (0.022)	52.65 (0.049)	56.02 (0.046)
LSM	prices	33.87 (0.030)	38.55 (0.015)	43.04 (0.020)
LSM	prices, payoff	39.53 (0.038)	41.75 (0.033)	44.15 (0.038)
PO	prices	40.96 (0.026)	44.84 (0.020)	47.48 (0.025)
PO	prices, payoff	22.14 (0.085)	16.08 (0.118)	10.74 (0.224)
RPO	prices, payoff	**44.57 (0.015)**	**50.14 (0.011)**	**53.28 (0.015)**
PO-UB	prices	51.40 (0.018)	57.21 (0.013)	60.31 (0.013)
PO-UB	prices, payoff	46.40 (0.021)	52.61 (0.050)	55.94 (0.045)
LSM	pricesKO, KOind	41.89 (0.025)	49.38 (0.013)	53.42 (0.011)
LSM	pricesKO, KOind, payoff	43.80 (0.019)	49.87 (0.014)	53.07 (0.009)
PO	pricesKO, KOind	44.05 (0.018)	50.94 (0.010)	54.26 (0.015)
PO	pricesKO, KOind, payoff	44.04 (0.019)	50.70 (0.010)	53.83 (0.014)
RPO	pricesKO, KOind, payoff	**45.39 (0.014)**	**51.29 (0.012)**	**54.42 (0.012)**
PO-UB	pricesKO, KOind	48.45 (0.020)	53.11 (0.011)	55.55 (0.008)
PO-UB	pricesKO, KOind, payoff	46.18 (0.020)	52.03 (0.032)	55.04 (0.020)
LSM	pricesKO	41.47 (0.029)	49.37 (0.012)	53.07 (0.013)
LSM	pricesKO, payoff	44.07 (0.014)	49.64 (0.011)	52.65 (0.011)
PO	pricesKO	44.34 (0.013)	49.85 (0.015)	52.81 (0.022)
PO	pricesKO, payoff	44.20 (0.018)	50.08 (0.011)	53.19 (0.015)
RPO	pricesKO, payoff	**44.57 (0.015)**	**50.15 (0.010)**	**53.28 (0.015)**
PO-UB	pricesKO	48.63 (0.019)	53.13 (0.010)	55.55 (0.007)
PO-UB	pricesKO, payoff	46.18 (0.019)	52.04 (0.032)	55.09 (0.021)
LSM	pricesKO, prices2KO, KOind	43.33 (0.017)	49.88 (0.018)	53.23 (0.009)
LSM	pricesKO, prices2KO, KOind, payoff	44.07 (0.020)	49.94 (0.016)	53.11 (0.008)
PO	pricesKO, prices2KO, KOind	44.34 (0.016)	50.79 (0.008)	53.93 (0.015)
PO	pricesKO, prices2KO, KOind, payoff	44.66 (0.017)	50.66 (0.009)	53.76 (0.015)
RPO	pricesKO, prices2KO, KOind, payoff	**45.18 (0.016)**	**51.07 (0.010)**	**54.21 (0.018)**
PO-UB	pricesKO, prices2KO, KOind	47.09 (0.017)	52.44 (0.008)	55.22 (0.006)
PO-UB	pricesKO, prices2KO, KOind, payoff	46.12 (0.018)	51.97 (0.030)	55.00 (0.019)
Tree	prices	35.49 (0.131)	43.49 (0.062)	47.29 (0.109)
Tree	prices, payoff	39.13 (0.019)	48.35 (0.020)	53.61 (0.015)
Tree	payoff, time	45.39 (0.020)	51.31 (0.019)	54.51 (0.014)
Tree	prices, time	38.43 (0.177)	40.55 (0.347)	42.94 (0.292)
Tree	prices, time, payoff	45.39 (0.019)	51.31 (0.019)	54.51 (0.014)
Tree	prices, time, payoff, KOind	45.39 (0.019)	51.31 (0.019)	54.51 (0.014)

Table 2 Average out-of-sample reward for different policies and different basis function architectures for constant barrier instances with $n = 8$ (Section 6.3).

Method	Basis functions	$\bar{p} = 90$	$\bar{p} = 100$	$\bar{p} = 110$
LSM	KOind	62.44 (0.019)	75.56 (0.036)	74.34 (0.044)
LSM	KOind, payoff	70.25 (0.040)	80.79 (0.063)	81.14 (0.083)
PO	KOind	68.02 (0.011)	79.16 (0.039)	74.02 (0.051)
PO	KOind, payoff	69.74 (0.030)	77.90 (0.047)	70.01 (0.073)
RPO	KOind, payoff	**71.68 (0.020)**	**84.17 (0.026)**	**84.62 (0.033)**
PO-UB	KOind	82.90 (0.045)	94.79 (0.064)	93.58 (0.047)
PO-UB	KOind, payoff	72.07 (0.042)	85.40 (0.074)	88.35 (0.056)
LSM	one	61.49 (0.021)	70.20 (0.053)	57.70 (0.094)
LSM	one, payoff	69.68 (0.035)	76.44 (0.055)	68.80 (0.057)
PO	one	67.65 (0.010)	77.37 (0.040)	69.38 (0.049)
PO	one, payoff	69.07 (0.019)	76.13 (0.056)	67.82 (0.055)
RPO	one, payoff	**71.68 (0.020)**	**84.17 (0.026)**	**84.62 (0.033)**
PO-UB	one	83.95 (0.049)	98.54 (0.076)	103.93 (0.068)
PO-UB	one, payoff	72.08 (0.043)	85.53 (0.077)	89.23 (0.066)
LSM	prices	63.25 (0.023)	70.85 (0.062)	55.60 (0.092)
LSM	prices, payoff	69.96 (0.037)	76.51 (0.046)	68.54 (0.051)
PO	prices	68.22 (0.010)	77.59 (0.040)	68.61 (0.050)
PO	prices, payoff	69.17 (0.021)	76.08 (0.056)	67.80 (0.047)
RPO	prices, payoff	**71.29 (0.020)**	**83.11 (0.033)**	**82.33 (0.025)**
PO-UB	prices	81.81 (0.031)	96.71 (0.060)	102.82 (0.058)
PO-UB	prices, payoff	72.07 (0.041)	85.51 (0.076)	89.17 (0.065)
LSM	pricesKO, KOind	65.24 (0.017)	77.30 (0.034)	74.75 (0.067)
LSM	pricesKO, KOind, payoff	70.58 (0.021)	81.22 (0.051)	81.16 (0.053)
PO	pricesKO, KOind	68.80 (0.016)	79.27 (0.034)	73.53 (0.046)
PO	pricesKO, KOind, payoff	70.24 (0.022)	79.17 (0.038)	73.13 (0.104)
RPO	pricesKO, KOind, payoff	**71.50 (0.022)**	**83.86 (0.036)**	**84.19 (0.030)**
PO-UB	pricesKO, KOind	80.76 (0.029)	92.96 (0.049)	92.42 (0.055)
PO-UB	pricesKO, KOind, payoff	72.05 (0.044)	85.38 (0.072)	88.30 (0.056)
LSM	pricesKO	64.39 (0.013)	77.09 (0.037)	74.52 (0.042)
LSM	pricesKO, payoff	70.59 (0.026)	81.41 (0.049)	80.84 (0.059)
PO	pricesKO	68.67 (0.014)	79.46 (0.038)	73.73 (0.041)
PO	pricesKO, payoff	70.20 (0.021)	79.06 (0.041)	73.49 (0.103)
RPO	pricesKO, payoff	**71.29 (0.020)**	**83.11 (0.034)**	**82.33 (0.025)**
PO-UB	pricesKO	81.26 (0.034)	93.53 (0.055)	92.74 (0.049)
PO-UB	pricesKO, payoff	72.05 (0.044)	85.39 (0.071)	88.31 (0.058)
Tree	payoff, time	70.47 (0.018)	79.94 (0.040)	74.88 (0.181)
Tree	prices	49.01 (0.042)	59.85 (0.111)	60.38 (0.161)
Tree	prices, payoff	52.97 (0.029)	65.40 (0.019)	66.49 (0.044)
Tree	prices, time	68.79 (0.024)	76.26 (0.076)	71.04 (0.136)
Tree	prices, time, payoff	70.47 (0.018)	79.94 (0.040)	74.88 (0.181)
Tree	prices, time, payoff, KOind	70.47 (0.018)	79.94 (0.040)	74.88 (0.181)

Table 3 Average out-of-sample reward for different policies and different basis function architectures for time-varying barrier instances with $n = 16$ (Section 6.4).

		Hurst parameter H			
Method	Basis functions	0.05	0.1	0.2	0.3
LSM	one, ph1:1	14288 (934)	8246 (239)	2397 (358)	1705 (214)
LSM	one	14583 (85)	7305 (33)	2408 (12)	1079 (7)
RPO	one, ph1:1	**26666 (850)**	**15215 (445)**	**7579 (192)**	**5433 (188)**
LSM	one, ph1:2	14447 (924)	8608 (278)	2591 (383)	1370 (214)
LSM	one, ph2:2	13267 (880)	7218 (124)	1904 (236)	758 (75)
RPO	one, ph1:2	**29024 (549)**	**17598 (465)**	**8503 (148)**	**5552 (118)**
LSM	one, ph1:3	14594 (1231)	9215 (292)	2378 (419)	1173 (209)
LSM	one, ph2:3	13488 (862)	7187 (157)	1786 (223)	584 (102)
RPO	one, ph1:3	**30788 (688)**	**18380 (333)**	**8746 (192)**	**5521 (129)**
LSM	one, ph1:4	13532 (1650)	9552 (573)	2328 (471)	978 (220)
LSM	one, ph2:4	13353 (926)	7204 (191)	1776 (228)	588 (91)
RPO	one, ph1:4	**31564 (745)**	**17740 (555)**	**8779 (194)**	**5551 (134)**
LSM	one, ph1:5	14345 (1730)	10315 (707)	2165 (467)	1027 (228)
LSM	one, ph2:5	12676 (1004)	7301 (154)	1665 (216)	509 (86)
RPO	one, ph1:5	**32322 (935)**	**18191 (595)**	**8886 (162)**	**5526 (124)**
LSM	one, ph1:10	16061 (1950)	9792 (1050)	2196 (609)	1048 (208)
LSM	one, ph2:10	12245 (1431)	6649 (576)	1257 (178)	428 (47)
RPO	one, ph1:10	**33196 (1510)**	**18792 (664)**	**8661 (143)**	**5420 (128)**
LSM	one, ph1:15	17135 (2109)	9440 (1111)	2469 (620)	1228 (211)
LSM	one, ph2:15	12582 (1381)	5918 (634)	1119 (173)	526 (39)
RPO	one, ph1:15	**34844 (1049)**	**19191 (587)**	**8724 (147)**	**5404 (103)**
LSM	one, ph1:20	18575 (2301)	9932 (1015)	2692 (614)	1359 (231)
LSM	one, ph2:20	13076 (1335)	5693 (522)	1235 (150)	554 (41)
RPO	one, ph1:20	**34128 (1476)**	**20057 (327)**	**8743 (140)**	**5272 (84)**
LSM	one, ph1:50	20974 (2153)	10501 (861)	2923 (595)	1492 (254)
LSM	one, ph2:50	12412 (1273)	5739 (232)	1348 (134)	644 (57)
RPO	one, ph1:50	**35311 (773)**	**19377 (329)**	**8516 (209)**	**5081 (111)**
RPO	one	4616 (398)	4574 (187)	4733 (333)	4705 (192)
RPO	one, logph1:1	31745 (508)	18274 (392)	8503 (140)	5409 (123)
RPO	one, logph1:2	34155 (488)	20427 (328)	9428 (146)	5678 (122)
RPO	one, logph1:3	35516 (660)	21644 (322)	9921 (131)	5821 (95)
RPO	one, logph1:4	36777 (651)	21946 (276)	9830 (138)	5846 (91)
RPO	one, logph1:5	37751 (776)	22566 (271)	9700 (129)	5723 (102)
RPO	one, logph1:10	38591 (587)	23070 (188)	9738 (135)	5506 (48)
RPO	one, logph1:15	39588 (616)	22784 (264)	9885 (109)	5468 (38)
RPO	one, logph1:20	39912 (861)	22933 (240)	9789 (122)	5433 (81)
RPO	one, logph1:50	39336 (641)	22677 (266)	9562 (149)	5137 (33)
Tree	time	4616 (398)	4710 (152)	4548 (372)	4705 (192)
Tree	time, ph1:1	29626 (490)	18246 (147)	8536 (189)	5112 (291)
Tree	time, ph1:50	29772 (454)	18075 (170)	8508 (131)	5084 (248)

Table 4 **Results for exponential-transformed fractional Brownian motion process instances (Section 6.5).**

		Burn-in period B				
Method	Basis functions	1	15	30	60	180
LSM	one	18.15 (0.741)	17.23 (0.733)	3.87 (0.108)	1.14 (0.004)	1.08 (0.003)
LSM	one, ph1:1	8.25 (0.433)	13.32 (0.729)	4.31 (0.124)	1.59 (0.035)	1.58 (0.020)
RPO	one, ph1:1	**19.03 (0.770)**	**18.75 (0.749)**	**5.34 (0.128)**	1.95 (0.048)	**1.80 (0.027)**
LSM	one, ph2:2	8.18 (0.485)	11.26 (0.654)	2.82 (0.094)	1.09 (0.003)	1.02 (0.003)
LSM	one, ph1:2	8.61 (0.473)	12.62 (0.737)	2.89 (0.103)	1.28 (0.031)	1.29 (0.006)
RPO	one, ph1:2	18.99 (0.786)	18.29 (0.757)	4.30 (0.109)	**2.33 (0.068)**	1.52 (0.012)
LSM	one, ph2:3	9.12 (0.508)	10.44 (0.573)	2.65 (0.095)	1.05 (0.002)	1.03 (0.003)
LSM	one, ph1:3	10.13 (0.548)	12.97 (0.692)	2.77 (0.100)	1.16 (0.020)	1.23 (0.006)
RPO	one, ph1:3	18.89 (0.773)	17.60 (0.742)	4.23 (0.107)	1.84 (0.050)	1.47 (0.009)
LSM	one, ph2:10	9.83 (0.527)	12.47 (0.636)	2.77 (0.097)	1.05 (0.002)	1.05 (0.003)
LSM	one, ph1:10	12.87 (0.651)	10.71 (0.566)	2.54 (0.079)	1.11 (0.004)	1.17 (0.005)
RPO	one, ph1:10	18.27 (0.758)	16.13 (0.694)	3.42 (0.103)	2.00 (0.062)	1.50 (0.019)
LSM	one, ph2:60	15.86 (0.721)	13.96 (0.664)	2.83 (0.097)	1.20 (0.017)	1.14 (0.005)
LSM	one, ph1:60	14.80 (0.671)	10.65 (0.540)	2.49 (0.073)	1.19 (0.005)	1.30 (0.010)
RPO	one, ph1:60	18.09 (0.767)	16.22 (0.704)	3.19 (0.101)	2.02 (0.063)	1.53 (0.022)
Tree	time, ph1:1	17.63 (0.763)	14.64 (0.559)	4.15 (0.112)	2.26 (0.083)	1.52 (0.009)
Tree	time, ph1:2	18.05 (0.803)	14.89 (0.627)	4.03 (0.109)	2.12 (0.078)	1.54 (0.013)
Tree	time, ph1:3	17.69 (0.768)	14.89 (0.628)	3.94 (0.108)	1.78 (0.064)	1.60 (0.016)
Tree	time, ph1:10	17.59 (0.764)	14.85 (0.629)	3.83 (0.101)	1.39 (0.039)	1.61 (0.016)
Tree	time, ph1:60	16.00 (0.653)	15.86 (0.672)	3.34 (0.098)	1.42 (0.027)	1.59 (0.017)

Table 5 Average out-of-sample reward for cross-sectional cryptocurrency experiment. Parenthesized values indicate standard errors. The best value for each burn-in period B is indicated in bold.

			Trimmed	Trimmed
Method	Basis functions	Mean	Mean (0.01)	Mean (0.025)
LSM	one, ph1:1	1.125	1.022	1.019
LSM	one, ph1:2	1.126	1.010	1.008
LSM	one, ph1:3	1.127	1.011	1.010
LSM	one, ph1:10	1.125	1.010	1.009
LSM	one, ph1:60	1.123	1.008	1.007
LSM	one	1.291	1.012	1.010
LSM	one, ph2:2	1.014	1.011	1.010
LSM	one, ph2:3	1.010	1.009	1.009
LSM	one, ph2:10	1.131	1.013	1.012
LSM	one, ph2:60	1.129	1.013	1.011
RPO	one, ph1:1	1.281	**1.037**	**1.033**
RPO	one, ph1:2	**1.318**	1.032	1.029
RPO	one, ph1:3	1.255	1.024	1.022
RPO	one, ph1:10	1.171	1.015	1.014
RPO	one, ph1:60	1.170	1.009	1.008
Tree	time, ph1:1	1.255	1.014	1.012
Tree	time, ph1:2	1.138	1.011	1.009
Tree	time, ph1:3	1.136	1.009	1.008
Tree	time, ph1:10	1.274	0.997	0.996
Tree	time, ph1:60	1.266	0.988	0.988

Table 6 Average out-of-sample reward for longitudinal cryptocurrency experiment. The best value for each type of mean is indicated in bold.

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 $\mathbf{b}_t \bullet \Phi_t(\mathbf{x}(\omega,t))$ satisfies either $\mathbf{b}_t \bullet \Phi_t(x(\omega,t)) > 0$ or $\mathbf{b}_t \bullet \Phi_t(\mathbf{x}(\omega,t)) < 0$ for each ω and t. (Stated differently, $\mathbf{b}_t \bullet \Phi_t(\mathbf{x}(\omega,t))$ 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 $\phi_{t,1}(\mathbf{x}) = 1$ 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 $\mathbf{b}_t \bullet \Phi_t(\mathbf{x}(\omega,t)) > 0$ or $\mathbf{b}_t \bullet \Phi_t(\mathbf{x}(\omega,t)) < 0$ for each ω and t, we have that

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

Consequently, we have that

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

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, $\xi_1,\dots,\xi_T$. Observe that we can write the reward of the randomized policy as

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

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 $\xi_1,\dots,\xi_T$ are independent; and the fifth by the linearity of expectation.

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

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

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

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

for each ω and t. We thus have that

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

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

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

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

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

where the third step follows because $\phi_{t,1}(\mathbf{x}) = 1$ for all $t \in [T]$ and $\mathbf{x} \in \mathcal{X}$ (this is Assumption 3).

Let E be the event defined as

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

We thus have that

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

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

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

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

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

Thus, we have that

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

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 prove this theorem, we first characterize the optimal expected reward, followed by the optimal reward of LSM and the optimal reward of DLP.

Optimal policy: The optimal value function at $t=2$ is clearly $J_2(x)=g(2,x)$, and so at $t=1$, the optimal continuation value function $C_1(x)$ is

$$C_1(x)=\mathbb{E}[J_2(x(2))\mid x(1)=x]=J_2(x)=g(2,x), \tag{EC.12}$$

since, by the definition of the $P^1_{a,\delta}$, the random variable $x(2)$ is always the same as $x(1)$. Therefore, the optimal value function is

$$\begin{aligned} J_1(x) &= \max\{g(1,x), C_1(x)\} && \text{(EC.13)}\\ &= \begin{cases} 1 & \text{if } x\le a,\\ a-\delta & \text{if } x>a, \end{cases} && \text{(EC.14)} \end{aligned}$$

and therefore the optimal expected reward, over the random initial state $x(1)$, is

$$\begin{aligned} V^*(P^1_{a,\delta}) = \mathbb{E}[J_1(x(1))] &= 1\cdot\mathbb{P}(x(1)\le a)+(a-\epsilon)\cdot\mathbb{P}(x(1)>a) && \text{(EC.15)}\\ &= a+(a-\delta)(1-a) && \text{(EC.16)}\\ &= 2a-\delta-a^2+\delta a. && \text{(EC.17)} \end{aligned}$$

The optimal policy in period 1 is consequently

$$\pi^*(1,x)=\begin{cases} \textbf{continue} & \text{if } x\le a,\\ \textbf{stop} & \text{if } x>a. \end{cases} \tag{EC.18}$$

LSM: Let us now consider the behavior of LSM where the basis function architecture consists of the constant basis function $\phi_1(\cdot)=1$. In period $t=2$, LSM will stop regardless of the state, just like the optimal policy. Therefore, the continuation value function of LSM at $t=1$ is the same as that of the optimal policy, which is $C_1(\cdot)$. Therefore, the regression step in LSM seeks to find the weight r of the basis function $\phi_{1,1}(\cdot)$ that minimizes expected squared error, where the expectation is taken over the random state $x(1)$ in the first period:

$$\begin{aligned} &\min_r \mathbb{E}[(r\cdot\phi_{1,1}(x(1))-C_1(x(1)))^2] && \text{(EC.19)}\\ &=\min_r \mathbb{E}[(r-C_1(x(1)))^2]. && \text{(EC.20)} \end{aligned}$$

Recall that $C_1(\cdot)$ is a piecewise constant function and that $x(1)$ is uniformly distributed on $[0,1]$; therefore, $C_1(x(1))$ is a random variable that is equal to 1 with probability a and 0 with probability $1-a$. Therefore, the solution r^* of the minimization above will be equal to

$$r^* = \mathbb{E}[C_1(x(1))] = 1 \cdot a + 0 \cdot (1-a) = a \tag{EC.21}$$

and thus, the approximate continuation value function $\hat{C}_1(\cdot)$ that is used by LSM is

$$\hat{C}_1(x) = r^* \cdot \phi_{1,1}(x) = r^* = a, \tag{EC.22}$$

i.e., it is simply the constant a. Observe now that the value function $\tilde{J}_1(\cdot)$ that is induced by this policy is

$$\tilde{J}_1(x) = \begin{cases} a+\delta & \text{if } x \leq a, \\ 0 & \text{if } x > a, \end{cases} \tag{EC.23}$$

because when $x \leq a$, $\hat{C}_1(x) < g(1,x)$ and LSM will choose to stop and earn the reward of $a+\delta$, whereas when $x > a$, $\hat{C}_1(x) > g(1,x)$ and LSM will choose to continue and earn the reward of 0. Taking expectations over the initial state $x(1)$, we have

$$V^{\text{LSM}}(P^1_{a,\delta}) = \mathbb{E}[\tilde{J}_1(x(1))] = a \cdot (a+\delta) + (1-a) \cdot 0 \tag{EC.24}$$

$$= a^2 + \delta a. \tag{EC.25}$$

Comparing the difference in the expected reward of LSM versus the optimal policy, we get

$$V^*(P^1_{a,\delta}) - V^{\text{LSM}}(P^1_{a,\delta}) = 2a - \delta - a^2 + \delta a - a^2 - \delta a \tag{EC.26}$$

$$= 2a - \delta - 2a^2 \tag{EC.27}$$

Relative to the optimal reward, this difference is

$$\frac{V^*(P^1_{a,\delta}) - V^{\text{LSM}}(P^1_{a,\delta})}{V^*(P^1_{a,\delta})} = \frac{2a - \delta - 2a^2}{2a - \delta - a^2 + \delta a} \tag{EC.28}$$

Observe that

$$\lim_{a \to 0} \frac{2a - 2a^2}{2a - a^2} = 1. \tag{EC.29}$$

Therefore, there exists an $\tilde{a} > 0$ such that

$$\frac{2\tilde{a} - 2\tilde{a}^2}{2\tilde{a} - \tilde{a}^2} > 1 - \epsilon. \tag{EC.30}$$

Additionally, observe that for any $a > 0$,

$$\lim_{\delta \to 0} \frac{2a - \delta - 2a^2}{2a - \delta - a^2 + \delta a} = \frac{2a - 2a^2}{2a - a^2}. \tag{EC.31}$$

Let $\epsilon' > 0$ be chosen so that

$$\epsilon' < \frac{2\tilde{a} - 2\tilde{a}^2}{2\tilde{a} - \tilde{a}^2} - 1 + \epsilon. \tag{EC.32}$$

Then, there exists a $\tilde{\delta} > 0$ such that

$$\frac{2a - \delta - 2a^2}{2a - \delta - a^2 + \delta a} > \frac{2\tilde{a} - 2\tilde{a}^2}{2\tilde{a} - \tilde{a}^2} - \epsilon' \tag{EC.33}$$

$$> 1 - \epsilon. \tag{EC.34}$$

Thus, the chosen $(\tilde{a}, \tilde{\delta})$ satisfy the requirement of the theorem, and establish that LSM can perform arbitrarily poorly compared to the optimal policy.

DLP: For DLP, observe that the optimal policy can be expressed as a deterministic linear policy in terms of $\phi_{1,1}, \phi_{1,2}, \phi_{2,1}, \phi_{2,2}$.

At $t = 2$, we stop if and only $b_{2,1}\phi_{2,1}(x) + b_{2,2}\phi_{2,2}(x) > 0$, where $b_{2,1}$ is any positive value and $b_{2,2} = 0$. Recall that $\phi_{2,1}(x) = 1$ and $\phi_{2,2}(x) = g(2,x)$; thus the inequality boils down to $b_{2,1} > 0$. In other words, we always stop, which is optimal since it is the last period and rewards are nonnegative.

At $t = 1$, observe that the optimal policy can also be written as

$$\pi^*(1,x) = \begin{cases} \textbf{continue} & \text{if } g(1,x) < a, \\ \textbf{stop} & \text{if } g(1,x) > a. \end{cases} \tag{EC.35}$$

This is equivalent to a deterministic linear policy that stops in period 1 if and only if $b_{1,1} \cdot \phi_{1,1}(x) + b_{1,2} \cdot \phi_{1,2}(x) > 0$, where $b_1 = -a$, $b_2 = +1$. (Recall that $\phi_{1,1}(\cdot) = 1$ is the constant basis function and $\phi_{1,2}(\cdot) = g(1,\cdot)$ is the current reward basis function.) Thus, we conclude that

$$V^*(P^1_{a,\delta}) = V^{\text{DLP}}(P^1_{a,\delta}), \tag{EC.36}$$

which immediately yields the result of the theorem. □

EC.1.4. Proof of Theorem 4

Recall the quantities $S^k_{0,1}$, $S^k_{0,\theta}$ and $S^{2k}_{0,1}$, which were defined as

$$S^k_{0,1} = \int_0^1 e^{kx}\,dx = \frac{1}{k}(e^k - 1), \tag{EC.37}$$

$$S^k_{0,\theta} = \int_0^\theta e^{kx}\,dx = \frac{1}{k}(e^{k\theta} - 1), \tag{EC.38}$$

$$S^{2k}_{0,1} = \int_0^1 e^{2kx}\,dx = \frac{1}{2k}(e^{2k} - 1). \tag{EC.39}$$

The optimal policy will choose to stop at $t = 2$, and hence the optimal value function at $t = 2$ is $J_2(x) = g(2,x)$. The optimal continuation value function at $t = 1$ is

$$C_1(x) = \mathbb{E}[J_2(x(2)) \mid x(1) = x] = \mathbb{E}[g(2,x(2)) \mid x(1) = x] = g(2,x) = \mathbb{I}\{x \le \theta\}, \tag{EC.40}$$

where we are using the fact that transitions are deterministic, i.e., the random variable $x(2)\,|\,x(1)=x$ is equal to x with probability 1.

To characterize the approximate continuation value function $\hat{C}_1(\cdot)$ used by LSM, we require several auxiliary results. Our first result characterizes the optimal solution of a generic least squares problem that will be central to LSM.

LEMMA EC.1. *The optimal solution of the problem*

$$\begin{aligned}
&\min_{\alpha,\beta}\mathbb{E}[(J_2(x(2))-(\alpha+\beta e^{kx(1)}))^2] && \text{(EC.41)}\\
&=\min_{\alpha,\beta}\int_0^1(\mathbb{I}\{x\leq\theta\}-\alpha-\beta e^{kx})^2dx && \text{(EC.42)}
\end{aligned}$$

is given by

$$\begin{aligned}
\beta_k^* &= \frac{S_{0,\theta}^k-\theta\cdot S_{0,1}^k}{S_{0,1}^{2k}-(S_{0,1}^k)^2}, && \text{(EC.43)}\\
\alpha_k^* &= \theta-\beta_k^*S_{0,1}^k && \text{(EC.44)}\\
&= \theta-\frac{S_{0,\theta}^k-\theta\cdot S_{0,1}^k}{S_{0,1}^{2k}-(S_{0,1}^k)^2}\cdot S_{0,1}^k. && \text{(EC.45)}
\end{aligned}$$

Proof of Lemma EC.1: Let $F(\alpha,\beta)$ denote the objective function,

$$F(\alpha,\beta)=\int_0^1(\mathbb{I}\{x\leq\theta\}-\alpha-\beta e^{kx})^2dx. \qquad \text{(EC.46)}$$

We can further simplify this objective function as

$$\begin{aligned}
&\int_0^1(\mathbb{I}\{x\leq\theta\}-\alpha-\beta e^{kx})^2dx && \text{(EC.47)}\\
&=\int_0^\theta(1-\alpha-\beta e^{kx})^2\,dx+\int_\theta^1(0-\alpha-\beta e^{kx})^2\,dx && \text{(EC.48)}\\
&=\int_0^\theta(\alpha-1+\beta e^{kx})^2\,dx+\int_\theta^1(\alpha+\beta e^{kx})^2\,dx && \text{(EC.49)}\\
&=\int_0^\theta(\alpha-1)^2+2(\alpha-1)\beta e^{kx}+\beta^2e^{2kx}\,dx+\int_\theta^1\alpha^2+2\alpha\beta e^{kx}+\beta^2e^{2kx}\,dx && \text{(EC.50)}\\
&=\int_0^\theta-2\alpha+1-2\beta e^{kx}\,dx+\int_0^1\alpha^2+2\alpha\beta e^{kx}+\beta^2e^{2kx}\,dx && \text{(EC.51)}\\
&=-2\alpha\theta+\theta-\left(\frac{2\beta e^{k\theta}}{k}-\frac{2\beta e^0}{k}\right)+\alpha^2+\frac{2\alpha\beta e^k}{k}-\frac{2\alpha\beta e^0}{k}+\frac{\beta^2e^{2k}}{2k}-\frac{\beta^2e^{2\cdot 0}}{2k} && \text{(EC.52)}\\
&=-2\alpha\theta+\theta-2\beta\cdot\frac{1}{k}(e^{k\theta}-1)+\alpha^2+2\alpha\beta\cdot\frac{1}{k}(e^k-1)+\beta^2\cdot\frac{1}{2k}(e^{2k}-1) && \text{(EC.53)}
\end{aligned}$$

Next, we have

$$\frac{\partial F}{\partial \alpha} = -2\theta + 2\alpha + 2\beta \cdot \frac{1}{k}(e^k - 1) \tag{EC.54}$$

$$= -2\theta + 2\alpha + 2\beta S^k_{0,1}, \tag{EC.55}$$

$$\frac{\partial F}{\partial \beta} = \frac{-2}{k}(e^{k\theta} - 1) + \frac{2\alpha}{k}(e^k - 1) + \frac{2\beta}{2k}(e^{2k} - 1) \tag{EC.56}$$

$$= -2S^k_{0,\theta} + 2\alpha S^k_{0,1} + 2\beta S^{2k}_{0,1}. \tag{EC.57}$$

Observe that $F(\alpha, \beta)$ is jointly convex in α and β, so the optimal solution can be found from the first order conditions. Using the first order condition for $\partial F/\partial \alpha$, we get:

$$\frac{\partial F}{\partial \alpha} = -2\theta + 2\alpha^*_k + 2\beta^*_k S^k_{0,1} = 0 \tag{EC.58}$$

$$\Rightarrow -\theta + \alpha^*_k + \beta^*_k S^k_{0,1} = 0 \tag{EC.59}$$

$$\Rightarrow \alpha^*_k = \theta - \beta^*_k S^k_{0,1}. \tag{EC.60}$$

Using the first order condition for $\partial F/\partial \beta$ and with the closed form solution of α^*_k, we get

$$\frac{\partial F}{\partial \beta} = -2S^k_{0,\theta} + 2\alpha^*_k S^k_{0,1} + 2\beta^*_k S^{2k}_{0,1} = 0 \tag{EC.61}$$

$$\Rightarrow -2S^k_{0,\theta} + 2(\theta - \beta^*_k S^k_{0,1})S^k_{0,1} + 2\beta^*_k S^{2k}_{0,1} = 0 \tag{EC.62}$$

$$\Rightarrow -S^k_{0,\theta} + (\theta - \beta^*_k S^k_{0,1})S^k_{0,1} + \beta^*_k S^{2k}_{0,1} = 0 \tag{EC.63}$$

$$\Rightarrow (S^{2k}_{0,1} - (S^k_{0,1})^2)\beta^*_k = S^k_{0,\theta} - \theta S^k_{0,1} \tag{EC.64}$$

$$\Rightarrow \beta^*_k = \frac{S^k_{0,\theta} - \theta S^k_{0,1}}{S^{2k}_{0,1} - (S^k_{0,1})^2}, \tag{EC.65}$$

as required. □

Our next lemma establishes the form of the optimal weights at $t = 1$ for the LSM policy. This result is just a generalization of a standard property in linear regression, which is that given a collection of features, the regression model one obtains is unchanged when each feature undergoes a linear transformation. The proof is straightforward and omitted.

LEMMA EC.2. *Suppose that $\phi_{1,1}(x) = 1$ and $\phi_{1,2}(x) = c + de^{kx}$, for some $d \neq 0$. Then the optimal solution of*

$$\min_{b_1, b_2} \mathbb{E}[(J_2(x(2)) - (b_1\phi_{1,1}(x(1)) + b_2\phi_{1,2}(x(1)))^2] \tag{EC.66}$$

is $b_1 = \alpha^*_k$, $b_2 = \beta^*_k$.

In particular, from the definition of $\phi_{1,1}(x) = 1$ and $\phi_{1,2}(x) = g(1, x)$, this result implies that the LSM continuation value function is exactly $\hat{C}_1(x) = \alpha^*_k + \beta^*_k e^{kx}$.

Next, we establish two universal properties of α^*_k and β^*_k.

Lemma EC.3. *The following properties of α_k^*, β_k^* hold, for all $k > 0$:*

$$\beta_k^* < 0, \tag{EC.67}$$

$$\alpha_k^* > \theta. \tag{EC.68}$$

Proof of Lemma EC.3: To verify the first claim, recall that

$$\beta_k^* = \frac{S_{0,\theta}^k - \theta S_{0,1}^k}{S_{0,1}^{2k} - (S_{0,1}^k)^2}. \tag{EC.69}$$

In this expression, observe that the denominator is positive, because it is equivalent to the variance of the (non-constant) random variable $e^{k \cdot x(1)}$. Thus, we just need to show that the numerator is negative. To see this, observe that

$$\begin{aligned} S_{0,\theta}^k - \theta S_{0,1}^k &= \frac{1}{k}(e^{k\theta} - 1) - \theta \cdot \frac{1}{k}(e^k - 1) && \text{(EC.70)} \\ &< \frac{1}{k}(\theta e^k + (1-\theta)e^0 - 1) - \frac{\theta}{k}(e^k - 1) && \text{(EC.71)} \\ &= \frac{1}{k}(\theta e^k - \theta) - \frac{\theta}{k}(e^k - 1) && \text{(EC.72)} \\ &= 0 && \text{(EC.73)} \end{aligned}$$

where the strict inequality follows by the strong convexity of $x \mapsto e^{kx}$ when $k > 0$.

The second claim, $\alpha_k^* \geq \theta$, follows straightforwardly from the first claim that $\beta^* < 0$, by using the definition of α_k^* as $\alpha_k^* = \theta - \beta_k^* S_{0,1}^k$. □

The next lemma catalogs a number of limiting properties of α_k^* and β_k^*. These properties are straightforward (albeit tedious) to verify, and the proof is therefore omitted.

Lemma EC.4. *The following limiting properties of α_k^*, β_k^* hold:*

$$\lim_{k\to\infty} \beta_k^* = 0, \tag{EC.74}$$

$$\lim_{k\to\infty} \alpha_k^* = \theta, \tag{EC.75}$$

$$\lim_{k\to\infty} \beta_k^* S_{0,\theta}^k = 0, \tag{EC.76}$$

$$\lim_{k\to\infty} \beta_k^* S_{0,1}^k = 0, \tag{EC.77}$$

$$\lim_{k\to\infty} \beta_k^* e^{k\theta} = 0, \tag{EC.78}$$

$$\lim_{k\to\infty} \beta_k^* e^k = -2\theta, \tag{EC.79}$$

$$\lim_{k\to\infty} \frac{1}{k} \log\left[\frac{-\alpha_k^* + \beta_k^*}{(1 + e^{-k\theta})\beta_k^*}\right] = 1, \tag{EC.80}$$

With all of these definitions, the first major result we will establish will be the characterization of the reward of LSM. Recall the definition of the function $g(1,\cdot)$, which is the immediate reward from stopping in $t=1$:

$$\begin{aligned} g(1,x) &= (1+e^{-k\theta})(\alpha_k^* + \beta_k^* e^{kx} - (\alpha_k^* + \beta_k^* e^{k\theta})) + (\alpha_k^* + \beta_k^* e^{k\theta}) \\ &= (1+e^{-k\theta})(\beta_k^* e^{kx} - \beta_k^* e^{k\theta}) + (\alpha_k^* + \beta_k^* e^{k\theta}) \\ &= (1+e^{-k\theta})\beta_k^* e^{kx} + \alpha_k^* - \beta_k^*. \end{aligned}$$

For the purpose of understanding how LSM will behave, there are two important insights to focus on. First, observe that $g(1,\cdot)$ is exactly of the form $g(1,x) = c + de^{kx}$. Thus, when one uses $g(1,\cdot)$ as a basis function within LSM, the approximate continuation value function $\hat{C}_1(x)$ will be exactly of the form $\hat{C}_1(x) = \alpha_k^* + \beta_k^* e^{kx}$; this is guaranteed by Lemma EC.2.

Second, observe that we have the following relationship between $g(1,\cdot)$ and $\hat{C}_1(\cdot)$:

- At $x=\theta$, $g(1,\theta)$ is exactly equal to $\hat{C}_1(\theta)$;
- For $x<\theta$, $g(1,x) > \hat{C}_1(x)$;
- For $x>\theta$, $g(1,x) < \hat{C}_1(x)$.

Thus, the LSM policy will choose to stop for $x<\theta$, and will choose to continue for $x>\theta$. This leads us to the following result, which characterizes the reward of the LSM policy.

PROPOSITION EC.1. *The expected reward of the LSM policy is*

$$V^{\mathrm{LSM}}(P^2_{k,\theta}) = \mathbb{E}[J^{LSM}(x(1))] = (1+e^{-k\theta})\beta_k^* S^k_{0,\theta} + \alpha_k^*\theta - \beta_k^*\theta. \tag{EC.81}$$

Additionally, for any fixed $\theta \in (0,1)$, we have

$$\lim_{k\to\infty} V^{\mathrm{LSM}}(P^2_{k,\theta}) = \theta^2 \tag{EC.82}$$

Proof of Proposition EC.1: Observe that $\hat{C}_1(x) = \alpha_k^* + \beta_k^* e^{kx}$, and that $g(1,x) > \hat{C}_1(x)$ for $x<\theta$, and $g(1,x) < \hat{C}_1(x)$ for $x>\theta$. Therefore, the optimal reward can be written as

$$\begin{aligned} \mathbb{E}[J^{LSM}(x(1))] &= \int_0^\theta g(1,x)\,dx + \int_\theta^1 C_1(x)\,dx && \text{(EC.83)} \\ &= \int_0^\theta (1+e^{-k\theta})\beta_k^* e^{kx} + \alpha_k^* - \beta_k^*\,dx + \int_\theta^1 0\,dx && \text{(EC.84)} \\ &= (1+e^{-k\theta})\beta_k^* S^k_{0,\theta} + \alpha_k^*\theta - \beta_k^*\theta, && \text{(EC.85)} \end{aligned}$$

which establishes the first part of the lemma. For the limiting statement, we have

$$\begin{aligned} &\lim_{k\to\infty} (1+e^{-k\theta})\beta_k^* S^k_{0,\theta} + \alpha_k^*\theta - \beta_k^*\theta && \text{(EC.86)} \\ &= \underbrace{\lim_{k\to\infty}(1+e^{-k\theta})}_{=1} \cdot \underbrace{\lim_{k\to\infty} \beta_k^* S^k_{0,\theta}}_{=0 \text{ by (EC.76)}} + \theta \cdot \underbrace{\lim_{k\to\infty} \alpha_k^*}_{=\theta \text{ by (EC.75)}} - \theta \cdot \underbrace{\lim_{k\to\infty} \beta_k^*}_{=0 \text{ by (EC.74)}} && \text{(EC.87)} \\ &= \theta^2, && \text{(EC.88)} \end{aligned}$$

as required. □

Having established the reward of the LSM policy, we now turn to the optimal policy. To understand the optimal policy, we need one additional definition. Let us define the quantity $\tilde{x}_k$ as the root of the equation $g(1,x)=0$; in closed form, it is

$$\tilde{x}_k = \frac{1}{k}\log\left[\frac{-\alpha_k^* + \beta_k^*}{(1+e^{-k\theta})\beta_k^*}\right] \tag{EC.89}$$

Lemma EC.5. *There exists k_0 such that for all $k > k_0$, $\tilde{x}_k \in (\theta, 1)$.*

Proof of Lemma EC.5: Note that when k is large enough, $g(1,0) > 0$, because

$$\begin{aligned} g(1,0) &= (1+e^{-k\theta})\beta_k^* e^{k\cdot 0} + \alpha_k^* - \beta_k^* & \text{(EC.90)}\\ &= e^{-k\theta}\beta_k^* + \alpha_k^* & \text{(EC.91)}\\ &\to \theta & \text{(EC.92)} \end{aligned}$$

as $k \to \infty$, which is guaranteed by limiting properties (EC.74) and (EC.75) of Lemma EC.4.

Additionally, when k is large enough, $g(1,1) < 0$, because

$$\begin{aligned} g(1,1) &= (1+e^{-k\theta})\beta_k^* e^{k\cdot 1} + \alpha_k^* - \beta_k^* & \text{(EC.93)}\\ &\to -\theta & \text{(EC.94)} \end{aligned}$$

as $k \to \infty$, which is guaranteed by limiting properties (EC.74) and (EC.79) of Lemma EC.4.

Together, these imply that when k is large enough, $g(1,0) > 0 > g(1,1)$ holds, and therefore the root $\tilde{x}_k$ must be in $(0,1)$. Additionally, limiting property (EC.80) of Lemma EC.4 directly implies that $\tilde{x}_k$ converges to 1 as $k \to \infty$. Therefore, for a sufficiently large k, we can assert that $\tilde{x}_k$ will be between θ and 1. □

With this lemma in hand, we can understand the behavior of the optimal policy for large k as follows. For large enough k, $g(1,x) < C_1(x)$ for $x < \theta$; $g(1,x) > C_1(x)$ for $x \in (\theta, \tilde{x}_k)$; and $g(1,x) < C_1(x)$ for $x > \tilde{x}_k$ (note that in this last case, this is true because $g(1,x)$ is negative for $x > \tilde{x}_k$, while $C_1(x) = 0$). The optimal policy at $t = 1$ therefore has the form

$$\pi^*(1,x) = \begin{cases} \textbf{continue} & \text{if } x < \theta, \\ \textbf{stop} & \text{if } \theta < x < \tilde{x}_k, \\ \textbf{continue} & \text{if } x > \tilde{x}_k. \end{cases} \tag{EC.95}$$

We can now use this to calculate the expected reward of the optimal policy.

Proposition EC.2. *There exists $k_0 > 0$ such that for all $k > k_0$, the expected reward of the optimal policy is*

$$V^*(P^2_{k,\theta}) = \mathbb{E}[J^*(x(1))] = \theta + \frac{1}{k}(\beta_k^* - \alpha_k^*) - \frac{1}{k}\beta_k^*(e^{k\theta}+1) + \alpha_k^*(\tilde{x}_k - \theta) - \beta_k^*(\tilde{x}_k - \theta) \tag{EC.96}$$

Additionally, for fixed $\theta \in (0,1)$, we have

$$\lim_{k\to\infty} V^*(P^2_{k,\theta}) = 2\theta - \theta^2. \tag{EC.97}$$

Proof: For the optimal policy, its expected reward is given by

$$\mathbb{E}[J^*(x(1))] = \int_0^1 \max\{g(1,x), C_1(x)\}\, dx \tag{EC.98}$$

To further analyze this quantity, observe that the quantity $\max\{g(1,x), C_1(x)\}$ has the following piecewise behavior:

$$\max\{g(1,x), C_1(x)\} = \begin{cases} C_1(x) & \text{if } x \in [0,\theta], \\ g(1,x) & \text{if } x \in (\theta, \tilde{x}_k), \\ 0 & \text{if } x \in [\tilde{x}_k, 1], \end{cases} \tag{EC.99}$$

Note that on the last piece, $(\tilde{x}_{k,\delta}, 1]$, the value of zero is exactly the value of the continuation value function $C_1(x)$.

Thus, the expected reward of the optimal policy is

$$\begin{aligned}
\mathbb{E}[J^*(x(1))] &= \int_0^1 \max\{g(1,x), C_1(x)\}\, dx \\
&= \int_0^\theta C_1(x)\, dx + \int_\theta^{\tilde{x}_k} g(1,x)\, dx + \int_{\tilde{x}_k}^1 0\, dx \\
&= \int_0^\theta 1\, dx + \int_\theta^{\tilde{x}_k} (1+e^{-k\theta})\beta_k^* e^{kx} + \alpha_k^* - \beta_k^*\, dx \\
&= \theta + (1+e^{-k\theta})\beta_k^* \cdot \frac{1}{k}(e^{k\tilde{x}_k} - e^{k\theta}) + \alpha_k^*(\tilde{x}_k - \theta) - \beta_k^*(\tilde{x}_k - \theta) \\
&= \theta + \frac{1}{k}(\beta_k^* - \alpha_k^*) - \frac{\beta_k^*}{k}(1+e^{-k\theta})e^{k\theta} + \alpha_k^*(\tilde{x}_k - \theta) - \beta_k^*(\tilde{x}_k - \theta) \\
&= \theta + \frac{1}{k}(\beta_k^* - \alpha_k^*) - \frac{\beta_k^*(e^{k\theta}+1)}{k} + \alpha_k^*(\tilde{x}_k - \theta) - \beta_k^*(\tilde{x}_k - \theta).
\end{aligned}$$

The second to last step in the above is justified due to $\tilde{x}_k$ being the root of $g(1,x) = 0$. In particular, the equation $g(1,\tilde{x}_k) = 0$ is

$$(1+e^{-k\theta})\beta_k^* e^{k\tilde{x}_k} + \alpha_k^* - \beta_k^* = 0, \tag{EC.100}$$

and by rearranging we obtain

$$(1+e^{-k\theta})\beta_k^* e^{k\tilde{x}_k} = \beta_k^* - \alpha_k^*, \tag{EC.101}$$

where we observe that the left hand side appears as one of the terms in the third to last equation.

To show the limiting statement, we make use of the several of the limiting properties in Lemma EC.4 to obtain

$$\lim_{k\to\infty} \theta + \frac{1}{k}(\beta_k^* - \alpha_k^*) - \frac{1}{k}\beta_k^*(e^{k\theta}+1) + \alpha_k^*(\tilde{x}_k - \theta) - \beta_k^*(\tilde{x}_k - \theta)$$

$$
\begin{aligned}
&= \lim_{k\to\infty} \theta - \frac{\alpha_k^*}{k} - \frac{1}{k}\beta_k^* e^{k\theta} + \alpha_k^*(\tilde{x}_k - \theta) - \beta_k^*(\tilde{x}_k - \theta) \\
&= \theta - \underbrace{\lim_{k\to\infty} \frac{\alpha_k^*}{k}}_{=0 \text{ by (EC.75)}} - \underbrace{\lim_{k\to\infty} \frac{1}{k}}_{=0} \cdot \underbrace{\lim_{k\to\infty} \beta_k^* e^{k\theta}}_{=0 \text{ by (EC.78)}} + \underbrace{\lim_{k\to\infty} \alpha_k^*}_{=\theta \text{ by (EC.75)}} \cdot \underbrace{\lim_{k\to\infty} (\tilde{x}_k - \theta)}_{=1-\theta \text{ by (EC.80)}} - \underbrace{\lim_{k\to\infty} \beta_k^*}_{=0 \text{ by (EC.74)}} \cdot \underbrace{\lim_{k\to\infty} (\tilde{x}_k - \theta)}_{=1-\theta \text{ by (EC.80)}} \\
&= \theta + \theta(1-\theta) \\
&= 2\theta - \theta^2,
\end{aligned}
$$

as required. □

We finally examine the deterministic linear policy. When we consider optimizing over all deterministic linear policies, observe that one possible policy is to set $b_{1,1}$ and $b_{1,2}$ as

$$b_{1,1} = \alpha_k^* + \beta_k^* e^{k\theta}, \tag{EC.102}$$

$$b_{1,2} = -1. \tag{EC.103}$$

The corresponding deterministic linear policy therefore stops when

$$
\begin{aligned}
&\alpha_k^* + \beta_k^* e^{k\theta} + (-1) \cdot \left[(1 + e^{-k\theta})(\alpha_k^* + \beta^* e^{kx} - (\alpha_k^* + \beta_k^* e^{k\theta})) + (\alpha_k^* + \beta_k^* e^{k\theta})\right] > 0 \\
\Leftrightarrow\ & -(1 + e^{-k\theta})(\alpha_k^* + \beta_k^* e^{kx} - (\alpha_k^* + \beta_k^* e^{k\theta})) > 0 \\
\Leftrightarrow\ & -\beta_k^* e^{kx} + \beta_k^* e^{k\theta} > 0 \\
\Leftrightarrow\ & -\beta_k^* e^{kx} > -\beta_k^* e^{k\theta} \\
\Leftrightarrow\ & e^{kx} > e^{k\theta} \quad \text{(follows by (EC.67) in Lemma EC.3)} \\
\Leftrightarrow\ & x > \theta,
\end{aligned}
$$

which is *almost* the same as the optimal policy. The discrepancy arises for $x > \tilde{x}_k$, where the optimal policy chooses to continue (because the current reward is negative), whereas the deterministic linear policy we constructed above would choose to stop.

This allows us to establish the next result, which gives a lower bound on the value of the optimal deterministic linear policy.

Proposition EC.3. *The expected reward of the optimal DLP policy satisfies*

$$V^{\text{DLP}}(P_{k,\theta}^2) = \mathbb{E}[J^{DLP}(x(1))] \geq \theta + (1 + e^{-k\theta})\beta_k^*(S_{0,1}^k - S_{0,\theta}^k) + \alpha_k^*(1-\theta) - \beta_k^*(1-\theta), \tag{EC.104}$$

Additionally, for any fixed $\theta \in (0,1)$, we have

$$\liminf_{k\to\infty} V^{\text{DLP}}(P_{k,\theta}^2) \geq 2\theta - \theta^2. \tag{EC.105}$$

Proof: For the DLP policy, observe that by our discussion above, a valid deterministic linear policy is to stop when $x > \theta$ and continue when $x < \theta$. The expected reward of such a policy is

$$\begin{aligned}
&\int_0^\theta C_1(x)\,dx + \int_\theta^1 g(1,x)\,dx && \text{(EC.106)}\\
&= \int_0^\theta C_1(x)\,dx + \int_\theta^1 (1+e^{-k\theta})\beta^* e^{kx} + \alpha_k^* - \beta_k^*\,dx && \text{(EC.107)}\\
&= \theta + (1+e^{-k\theta})\beta^*(S_{0,1}^k - S_{0,\theta}^k) + \alpha_k^*(1-\theta) - \beta_k^*(1-\theta). && \text{(EC.108)}
\end{aligned}$$

This expected reward thus provides a lower bound on $\mathbb{E}[J^{DLP}(x(1))]$. This establishes the first result.

For the limiting statement, observe that the limit of the right hand side of (EC.104) is

$$\begin{aligned}
&\lim_{k\to\infty} \theta + (1+e^{-k\theta})\beta_k^*(S_{0,1}^k - S_{0,\theta}^k) + \alpha_k^*(1-\theta) - \beta_k^*(1-\theta)\\
&= \theta + \underbrace{\lim_{k\to\infty}(1+e^{-k\theta})}_{=1}\cdot \underbrace{\lim_{k\to\infty}\beta_k^*(S_{0,1}^k - S_{0,\theta}^k)}_{=0 \text{ by (EC.76) and (EC.77)}} + (1-\theta)\cdot \underbrace{\lim_{k\to\infty}\alpha_k^*}_{=\theta \text{ by (EC.75)}} - (1-\theta)\cdot \underbrace{\lim_{k\to\infty}\beta_k^*}_{=0 \text{ by (EC.74)}}\\
&= \theta + \theta(1-\theta)\\
&= 2\theta - \theta^2
\end{aligned}$$

as required. □

We are now in a position to prove Theorem 4.

Proof of Theorem 4: We begin by first calculating the limits of the expected rewards of the three policies. First, observe that

$$\lim_{\theta\to 0}\frac{2\theta - 2\theta^2}{2\theta - \theta^2} = \lim_{\theta\to 0}\frac{2-4\theta}{2-2\theta} = 1. \tag{EC.109}$$

Therefore, there exists a $\tilde{\theta} \in (0,1)$ such that

$$\frac{2\tilde{\theta} - 2\tilde{\theta}^2}{2\tilde{\theta} - \tilde{\theta}^2} > 1 - \epsilon. \tag{EC.110}$$

Now, let $\delta > 0$ be such that

$$\begin{aligned}
&\frac{2\delta}{2\tilde{\theta} - \tilde{\theta}^2 - \delta} < \epsilon, && \text{(EC.111)}\\
&\frac{2\tilde{\theta} - 2\tilde{\theta}^2 - 2\delta}{2\tilde{\theta} - \tilde{\theta}^2 + \delta} > 1 - \epsilon. && \text{(EC.112)}
\end{aligned}$$

Such a δ is guaranteed to exist because the left-hand sides of inequalities (EC.111) and (EC.112) converge to 0 and $(2\tilde{\theta} - 2\tilde{\theta}^2)/(2\tilde{\theta} - \tilde{\theta}^2)$, respectively, as $\delta \to 0$.

Now, with such a δ in hand, and holding $\tilde{\theta}$ fixed, let us invoke the limiting properties (EC.82), (EC.97), and (EC.105) of Propositions EC.1, EC.2 and EC.3 respectively to assert the existence of a $\tilde{k}$ such that

$$V^{\mathrm{LSM}}(P^2_{\tilde{k},\tilde{\theta}}) < \tilde{\theta}^2 + \delta, \tag{EC.113}$$

$$2\tilde{\theta} - \tilde{\theta}^2 - \delta < V^*(P^2_{\tilde{k},\tilde{\theta}}) < 2\tilde{\theta} - \tilde{\theta}^2 + \delta, \tag{EC.114}$$

$$V^{\mathrm{DLP}}(P^2_{\tilde{k},\tilde{\theta}}) > 2\tilde{\theta} - \tilde{\theta}^2 - \delta. \tag{EC.115}$$

For this pair $(\tilde{k},\tilde{\theta})$, inequalities (EC.113) and (EC.114) imply that the gap of LSM is

$$\begin{aligned}\frac{V^*(P^2_{\tilde{k},\tilde{\theta}}) - V^{\mathrm{LSM}}(P^2_{\tilde{k},\tilde{\theta}})}{V^*(P^2_{\tilde{k},\tilde{\theta}})} &> \frac{(2\tilde{\theta} - \tilde{\theta}^2 - \delta) - (\tilde{\theta}^2 + \delta)}{2\tilde{\theta} - \tilde{\theta}^2 + \delta} \\ &= \frac{2\tilde{\theta} - 2\tilde{\theta}^2 - 2\delta}{2\tilde{\theta} - \tilde{\theta}^2 + \delta} \\ &> 1 - \epsilon,\end{aligned}$$

while inequalities (EC.114) and (EC.115) imply that the gap of DLP is

$$\begin{aligned}\frac{V^*(P^2_{\tilde{k},\tilde{\theta}}) - V^{\mathrm{DLP}}(P^2_{\tilde{k},\tilde{\theta}})}{V^*(P^2_{\tilde{k},\tilde{\theta}})} &< \frac{(2\tilde{\theta} - \tilde{\theta}^2 + \delta) - (2\tilde{\theta} - \tilde{\theta}^2 - \delta)}{2\tilde{\theta} - \tilde{\theta}^2 - \delta} \\ &= \frac{2\delta}{2\tilde{\theta} - \tilde{\theta}^2 - \delta} \\ &< \epsilon,\end{aligned}$$

which completes the proof. □

EC.1.5. Proof of Theorem 5

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

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

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_t at each t be just equal to the identity mapping, i.e., $\Phi_t(\mathbf{x})=\mathbf{x}$ for any $\mathbf{x}\in\mathcal{X}$. Thus, the dimension of the basis function vector K is equal to N.

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

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

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

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

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

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

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

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

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

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

$$\begin{aligned}\hat{J}_R(\mathbf{b}) &= \frac{1}{\Omega}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} g(t,\mathbf{x}(\omega,t)) \prod_{t'=1}^{t-1}(1-\sigma(\mathbf{b}_{t'} \bullet \Phi_{t'}(\mathbf{x}(\omega,t'))))\sigma(\mathbf{b}_t \bullet \Phi_t(\mathbf{x}(\omega,t)))\\ &= \frac{1}{\Omega}\sum_{\omega=1}^{\Omega} \Omega[\sigma(a_{\omega,1}b_{i_{\omega,1}}) + (1-\sigma(a_{\omega,1}b_{i_{\omega,1}}))\sigma(a_{\omega,2}b_{i_{\omega,2}}) + (1-\sigma(a_{\omega,1}b_{i_{\omega,1}}))(1-\sigma(a_{\omega,2}b_{i_{\omega,2}}))\sigma(a_{\omega,3}b_{i_{\omega,3}})]\\ &= \sum_{m=1}^{M}[\sigma(a_{m,1}b_{i_{m,1}}) + (1-\sigma(a_{m,1}b_{i_{m,1}}))\sigma(a_{m,2}b_{i_{m,2}}) + (1-\sigma(a_{m,1}b_{i_{m,1}}))(1-\sigma(a_{m,2}b_{i_{m,2}}))\sigma(a_{m,3}b_{i_{m,3}})].\end{aligned} \tag{EC.116}$$

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

$$\begin{aligned}& y_3 + (\neg y_3)\cdot(\neg y_4) + (\neg y_3)\cdot(\neg\neg y_4)\cdot y_7\\ &= y_3 + (1-y_3)(1-y_4) + (1-y_3)(y_4)y_7.\end{aligned} \tag{EC.117}$$

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.117) – are equivalent. The term in the square brackets in (EC.116) 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.116) 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:

$$b_i = \begin{cases} +\alpha \text{ if } y_i = 1, \\ -\alpha \text{ if } y_i = 0. \end{cases} \tag{EC.118}$$

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.117)), it follows that

$$\begin{aligned}
\lim_{\alpha \to +\infty} \hat{J}_R(\mathbf{b}) = \lim_{\alpha \to +\infty} \sum_{m=1}^{M} &[\sigma(a_{m,1} b_{i_{m,1}}) + (1 - \sigma(a_{m,1} b_{i_{m,1}}))\sigma(a_{m,2} b_{i_{m,2}}) \\
&+ (1 - \sigma(a_{m,1} b_{i_{m,1}}))(1 - \sigma(a_{m,2} b_{i_{m,2}}))\sigma(a_{m,3} b_{i_{m,3}})] \\
= \sum_{m=1}^{M} c_m,&
\end{aligned}$$

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

Randomized policy SAA answer is yes ⇒ MAX-3SAT answer is yes: To show the other direction of the equivalence, let us suppose we have a solution $\mathbf{b}$ for the randomized policy SAA problem with objective value $\hat{J}_R(\mathbf{b}) \geq W - 1/2$. We now need to construct a solution for the MAX-3SAT decision problem with objective value at least W.

Let us use $c_m(y_1, \ldots, y_N)$ to denote the value of clause m as a function of the binary variables $y_1, ..., y_N$. We claim that

$$\hat{J}_R(\mathbf{b}) = \mathbb{E}\left[\sum_{m=1}^{M} c_m(\mathbb{I}\{\xi_1 \leq b_1\}, \ldots, \mathbb{I}\{\xi_N \leq b_N\})\right], \tag{EC.119}$$

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.119), we argue that for any clause m,

$$\begin{aligned} &\mathbb{E}[c_m(\mathbb{I}\{\xi_1 \leq b_1\}, \ldots, \mathbb{I}\{\xi_N \leq b_N\})] \\ &= \sigma(a_{m,1} b_{i_{m,1}}) + (1 - \sigma(a_{m,1} b_{i_{m,1}}))\sigma(a_{m,2} b_{i_{m,2}}) + (1 - \sigma(a_{m,1} b_{i_{m,1}}))(1 - \sigma(a_{m,2} b_{i_{m,2}}))\sigma(a_{m,3} b_{i_{m,3}}). \end{aligned} \tag{EC.120}$$

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.120), 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

$$\begin{aligned} &\mathbb{P}(\xi_3 \leq b_3) + (1 - \mathbb{P}(\xi_3 \leq b_3))(1 - \mathbb{P}(\xi_4 \leq b_4)) + (1 - \mathbb{P}(\xi_3 \leq b_3)) \cdot \mathbb{P}(\xi_4 \leq b_4) \cdot \mathbb{P}(\xi_7 \leq b_7) && \text{(EC.121)} \\ &= \mathbb{E}[\mathbb{I}\{\xi_3 \leq b_3\}] + \mathbb{E}[1 - \mathbb{I}\{\xi_3 \leq b_3\}]\mathbb{E}[1 - \mathbb{I}\{\xi_4 \leq b_4\}] + \mathbb{E}[1 - \mathbb{I}\{\xi_3 \leq b_3\}]\mathbb{E}[\mathbb{I}\{\xi_4 \leq b_4\}]\mathbb{E}[\mathbb{I}\{\xi_7 \leq b_7\}] \\ &= \mathbb{E}[\mathbb{I}\{\xi_3 \leq b_3\} + (1 - \mathbb{I}\{\xi_3 \leq b_3\})(1 - \mathbb{I}\{\xi_4 \leq b_4\}) + (1 - \mathbb{I}\{\xi_3 \leq b_3\})\mathbb{I}\{\xi_4 \leq b_4\}\mathbb{I}\{\xi_7 \leq b_7\}], && \text{(EC.122)} \end{aligned}$$

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.122) can be written as

$$y_3 + (1 - y_3)(1 - y_4) + (1 - y_3)y_4 y_7$$

which is logically identical to $y_3 \vee \neg y_4 \vee y_7$. Thus, in this example, it follows that equation (EC.120) 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.120), can be used to turn the right-hand side of (EC.120) 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.120) holds, by linearity of expectation it must be the case that (EC.119) also holds. Consequently, there must exist values $\xi'_1, \ldots, \xi'_N$ of the random variables $\xi_1, \ldots, \xi_N$ which satisfy the following:

$$\begin{aligned} &\mathbb{E}[\sum_{m=1}^{M} c_m(\mathbb{I}\{\xi_1 \leq b_1\}, \ldots, \mathbb{I}\{\xi_N \leq b_N\})] \\ &\leq \sum_{m=1}^{M} c_m(\mathbb{I}\{\xi'_1 \leq b_1\}, \ldots, \mathbb{I}\{\xi'_N \leq b_N\}). \end{aligned} \tag{EC.123}$$

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.123) and (EC.119), 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.1.6. Proof of Proposition 1

First, observe that the sequence of objective values $(\hat{J}_R(\mathbf{b}^s))_{s=1}^{\infty}$ converges to a finite limit. To see this, observe that the sequence of objective values is non-decreasing, i.e.,

$$\hat{J}_R(\mathbf{b}^1) \leq \hat{J}_R(\mathbf{b}^2) \leq \hat{J}_R(\mathbf{b}^3) \leq \dots \tag{EC.124}$$

which is the case because the AM algorithm 5 in each iteration cannot cause the logarithm of the objective value (and hence the untransformed objective value) to decrease. Because each objective value $\hat{J}_R(\cdot)$ is upper bounded by $\bar{G}$, the sequence $(\hat{J}_R(\mathbf{b}^s))_{s=1}^{\infty}$ converges. Let us denote the finite limit of this sequence as $\bar{J}$.

Now, we will show that this $\bar{J}$ is attained by some deterministic policy weight vector. To see this, let us use Assumption 7, which asserts that there exists a subsequence $s_1, s_2, s_3, \dots$, for which

$$\lim_{i\to\infty} \min_{1\leq t\leq T} \|\mathbf{b}_t^{s_i}\|_2 = +\infty, \tag{EC.125}$$

and without loss of generality, we can assume that $\min_{1\leq t\leq T} \|\mathbf{b}_t^{s_i}\|_2 > 0$ for every i.

Now, for this sequence of weight vectors, let us define several new sequences. Let us define the sequence $(\mathbf{v}^i)_{i=1}^{\infty}$ in $\mathbb{R}^{KT}$ as

$$\mathbf{v}_t^i = \frac{\mathbf{b}_t^{s_i}}{\|\mathbf{b}_t^{s_i}\|_2}, \tag{EC.126}$$

where we note that this sequence is well-defined by our construction of the subsequence indices $s_1, s_2, \dots$ that guarantees that $\min_{1\leq t\leq T} \|\mathbf{b}_t^{s_i}\|_2 > 0$. Let us further define the sequence $(d_t^i)_{i=1}^{\infty}$ in $\mathbb{R}$, for each $t \in [T]$, as

$$d_t^i = \|\mathbf{b}_t^{s_i}\|_2. \tag{EC.127}$$

Observe that the sequence $(\mathbf{v}^i)_{i=1}^{\infty}$ is a sequence in $\mathcal{S}^T$, where $\mathcal{S} = \{\mathbf{y} \in \mathbb{R}^T \mid \|\mathbf{y}\|_2 = 1\}$ is the unit sphere. Since $\mathcal{S}$ is compact, $\mathcal{S}^T$ is compact, and hence there exists a subsequence, $i_1, i_2, i_3, \dots$ such that $\mathbf{v}^{i_j} \to \mathbf{v} \in \mathcal{S}^T \subseteq \mathbb{R}^{KT}$ as $j \to \infty$. Let $\mathbf{v}_t$ denote the weights for period t, so that $\mathbf{v} = (\mathbf{v}_1, \dots, \mathbf{v}_T)$.

With regard to this limiting vector $\mathbf{v}$, there are three mutually exclusive and collectively exhaustive cases to consider for the sign of the inner product $\mathbf{v}_t \bullet \Phi_t(\mathbf{x}(\omega,t))$: we either have $\mathbf{v}_t \bullet \Phi_t(\mathbf{x}(\omega,t)) > 0$, $\mathbf{v}_t \bullet \Phi_t(\mathbf{x}(\omega,t)) < 0$, or $\mathbf{v}_t \bullet \Phi_t(\mathbf{x}(\omega,t)) = 0$. Note that this last case cannot happen, for if it does, then by basic properties of limits and continuity of the map $z \mapsto |z|$, it is the case that $(|\mathbf{v}_t^{i_j} \bullet \Phi_t(\mathbf{x}(\omega,t))|)_{j=1}^{\infty}$ is convergent and

$$\lim_{j\to\infty} |\mathbf{v}_t^{i_j} \bullet \Phi_t(\mathbf{x}(\omega,t))| = |\mathbf{v}_t \bullet \Phi_t(\mathbf{x}(\omega,t))| = 0.$$

However, by Assumption 6, together with Assumption 5, there exists a $c>0$ such that

$$\begin{aligned}
&\lim_{j\to\infty} |\mathbf{v}_t^{i_j} \bullet \Phi_t(\mathbf{x}(\omega,t))| \\
&= \lim_{j\to\infty} \frac{|\mathbf{b}_t^{s_{i_j}} \bullet \Phi_t(\mathbf{x}(\omega,t))|}{\|\mathbf{b}_t^{s_{i_j}}\|_2} \\
&= \|\Phi_t(\mathbf{x}(\omega,t))\|_2 \cdot \lim_{j\to\infty} \frac{|\mathbf{b}_t^{s_{i_j}} \bullet \Phi_t(\mathbf{x}(\omega,t))|}{\|\mathbf{b}_t^{s_{i_j}}\|_2 \cdot \|\Phi_t(\mathbf{x}(\omega,t))\|_2} \\
&\geq c\|\Phi_t(\mathbf{x}(\omega,t))\|_2 \\
&> 0,
\end{aligned}$$

which leads to a contradiction. Hence, $\mathbf{v}_t \bullet \Phi_t(\mathbf{x}(\omega,t))$ cannot be equal to zero.

Therefore, for the subsequence $(\mathbf{v}^{i_j})_{j=1}^{\infty}$, as $j\to\infty$, observe that

$$\mathbf{b}_t^{s_{i_j}} \bullet \Phi_t(\mathbf{x}(\omega,t)) \tag{EC.128}$$
$$= d_t^{i_j} \cdot \mathbf{v}_t^{i_j} \bullet \Phi_t(\mathbf{x}(\omega,t)) \tag{EC.129}$$
$$\to \begin{cases} +\infty & \text{if } \mathbf{v}_t \bullet \Phi_t(\mathbf{x}(\omega,t)) > 0, \\ -\infty & \text{if } \mathbf{v}_t \bullet \Phi_t(\mathbf{x}(\omega,t)) < 0. \end{cases} \tag{EC.130}$$

Hence, as $j\to\infty$, we have

$$\sigma(\mathbf{b}_t^{s_{i_j}} \bullet \Phi_t(\mathbf{x}(\omega,t))) \tag{EC.131}$$
$$\to \begin{cases} 1 & \text{if } \mathbf{v}_t \bullet \Phi_t(\mathbf{x}(\omega,t)) > 0, \\ 0 & \text{if } \mathbf{v}_t \bullet \Phi_t(\mathbf{x}(\omega,t)) < 0 \end{cases} \tag{EC.132}$$
$$= \mathbb{I}\{\mathbf{v}_t \bullet \Phi_t(\mathbf{x}(\omega,t)) > 0\}. \tag{EC.133}$$

Thus, we have that

$$\hat{J}_R(\mathbf{b}^{s_{i_j}}) \tag{EC.134}$$
$$= \frac{1}{\Omega}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} g(t,\mathbf{x}(\omega,t)) \prod_{t'=1}^{t-1}(1-\sigma(\mathbf{b}_{t'}^{s_{i_j}} \bullet \Phi_{t'}(\mathbf{x}(\omega,t'))))\sigma(\mathbf{b}_t^{s_{i_j}} \bullet \Phi_t(\mathbf{x}(\omega,t))) \tag{EC.135}$$
$$\to \frac{1}{\Omega}\sum_{\omega=1}^{\Omega}\sum_{t=1}^{T} g(t,\mathbf{x}(\omega,t)) \prod_{t'=1}^{t-1} \mathbb{I}\{\mathbf{v}_{t'} \bullet \Phi_{t'}(\mathbf{x}(\omega,t')) \leq 0\}\mathbb{I}\{\mathbf{v}_t \bullet \Phi_t(\mathbf{x}(\omega,t)) > 0\} \tag{EC.136}$$
$$= \hat{J}_D(\mathbf{v}), \tag{EC.137}$$

as $j\to\infty$. Since $\hat{J}_R(\mathbf{b}^s) \to \bar{J}$ as $s\to\infty$, it follows that $\hat{J}_R(\mathbf{b}^{s_{i_j}}) \to \bar{J}$ as $j\to\infty$ and hence that $\bar{J} = \hat{J}_D(\mathbf{v})$. □

EC.2. Additional numerical results

EC.2.1. 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. Finally, Table EC.3 gives the average computation time of all of the methods for $n = 8$.

EC.2.2. Additional policy performance results for Section 6.4

Tables EC.4 and EC.5 displays the results comparing LSM, PO, RPO and the tree method for the time-varying barrier instances (Section 6.4) with $n = 8$ and $n = 32$ assets, respectively.

			Initial price	
Method	Basis function architecture	$\bar{p} = 90$	$\bar{p} = 100$	$\bar{p} = 110$
LSM	KOind	26.20 (0.033)	35.64 (0.030)	44.07 (0.019)
LSM	KOind, payoff	33.38 (0.025)	41.92 (0.013)	48.06 (0.019)
PO	KOind	31.54 (0.023)	41.05 (0.014)	48.43 (0.017)
PO	KOind, payoff	32.21 (0.043)	42.28 (0.029)	48.97 (0.020)
RPO	KOind, payoff	**34.57 (0.022)**	**43.07 (0.014)**	**49.33 (0.019)**
PO-UB	KOind	41.50 (0.026)	48.37 (0.026)	52.83 (0.012)
PO-UB	KOind, payoff	35.08 (0.030)	43.90 (0.022)	50.18 (0.021)
LSM	one	24.69 (0.025)	31.78 (0.029)	37.47 (0.016)
LSM	one, payoff	32.85 (0.033)	40.01 (0.031)	43.11 (0.018)
PO	one	30.86 (0.025)	39.01 (0.021)	44.55 (0.015)
PO	one, payoff	22.73 (0.144)	20.47 (0.139)	16.59 (0.128)
RPO	one, payoff	**34.57 (0.022)**	**43.07 (0.014)**	**49.33 (0.019)**
PO-UB	one	43.26 (0.031)	51.09 (0.030)	56.47 (0.014)
PO-UB	one, payoff	35.12 (0.029)	44.05 (0.025)	50.56 (0.024)
LSM	prices	25.74 (0.026)	32.09 (0.027)	37.39 (0.021)
LSM	prices, payoff	32.35 (0.033)	38.18 (0.020)	40.72 (0.022)
PO	prices	31.41 (0.022)	38.96 (0.017)	43.39 (0.019)
PO	prices, payoff	23.01 (0.112)	20.04 (0.069)	15.57 (0.101)
RPO	prices, payoff	**34.00 (0.018)**	**42.17 (0.017)**	**48.14 (0.018)**
PO-UB	prices	40.55 (0.022)	49.26 (0.015)	55.61 (0.009)
PO-UB	prices, payoff	35.12 (0.029)	44.04 (0.026)	50.54 (0.027)
LSM	pricesKO, KOind	30.20 (0.025)	39.10 (0.015)	46.58 (0.018)
LSM	pricesKO, KOind, payoff	32.72 (0.021)	41.27 (0.021)	47.75 (0.026)
PO	pricesKO, KOind	31.88 (0.025)	40.67 (0.023)	48.41 (0.020)
PO	pricesKO, KOind, payoff	31.40 (0.029)	40.61 (0.036)	48.45 (0.017)
RPO	pricesKO, KOind, payoff	**34.59 (0.023)**	**43.11 (0.013)**	**49.35 (0.019)**
PO-UB	pricesKO, KOind	38.81 (0.019)	46.44 (0.014)	51.73 (0.007)
PO-UB	pricesKO, KOind, payoff	35.08 (0.028)	43.88 (0.022)	50.16 (0.021)
LSM	pricesKO	28.53 (0.033)	38.37 (0.024)	46.57 (0.024)
LSM	pricesKO, payoff	33.48 (0.023)	41.74 (0.015)	47.70 (0.018)
PO	pricesKO	32.70 (0.022)	41.87 (0.012)	47.75 (0.015)
PO	pricesKO, payoff	32.67 (0.026)	41.52 (0.026)	48.01 (0.019)
RPO	pricesKO, payoff	**34.00 (0.018)**	**42.17 (0.016)**	**48.15 (0.018)**
PO-UB	pricesKO	39.53 (0.015)	46.88 (0.019)	51.88 (0.007)
PO-UB	pricesKO, payoff	35.08 (0.027)	43.88 (0.021)	50.16 (0.020)
LSM	pricesKO, prices2KO, KOind	31.94 (0.025)	41.01 (0.014)	47.73 (0.016)
LSM	pricesKO, prices2KO, KOind, payoff	33.43 (0.022)	41.86 (0.020)	48.01 (0.019)
PO	pricesKO, prices2KO, KOind	32.18 (0.022)	41.90 (0.019)	48.70 (0.016)
PO	pricesKO, prices2KO, KOind, payoff	33.66 (0.019)	42.51 (0.017)	48.76 (0.016)
RPO	pricesKO, prices2KO, KOind, payoff	**34.51 (0.024)**	**43.07 (0.013)**	**49.30 (0.017)**
PO-UB	pricesKO, prices2KO, KOind	36.34 (0.018)	44.59 (0.013)	50.48 (0.011)
PO-UB	pricesKO, prices2KO, KOind, payoff	35.06 (0.028)	43.83 (0.021)	50.06 (0.019)
Tree	payoff, time	34.37 (0.061)	43.09 (0.020)	49.34 (0.020)
Tree	prices	27.12 (0.031)	36.87 (0.013)	45.13 (0.038)
Tree	prices, payoff	27.34 (0.025)	37.12 (0.021)	45.77 (0.022)
Tree	prices, time	33.97 (0.093)	39.18 (0.504)	41.49 (0.226)
Tree	prices, time, payoff	34.37 (0.061)	43.09 (0.020)	49.34 (0.020)
Tree	prices, time, payoff, KOind	34.37 (0.061)	43.09 (0.020)	49.34 (0.020)

Table EC.1 Average out-of-sample reward for different policies and different basis function architectures for constant barrier instances with $n = 4$ (Section 6.3).

			Initial price	
Method	Basis function architecture	$\bar{p} = 90$	$\bar{p} = 100$	$\bar{p} = 110$
LSM	KOind	49.84 (0.017)	53.79 (0.016)	55.13 (0.014)
LSM	KOind, payoff	50.66 (0.016)	53.35 (0.008)	54.82 (0.009)
PO	KOind	51.59 (0.012)	54.46 (0.008)	55.73 (0.010)
PO	KOind, payoff	51.39 (0.013)	53.95 (0.007)	55.30 (0.007)
RPO	KOind, payoff	**51.84 (0.013)**	**54.56 (0.008)**	**55.97 (0.009)**
PO-UB	KOind	53.45 (0.008)	55.50 (0.007)	56.76 (0.007)
PO-UB	KOind, payoff	52.51 (0.024)	55.06 (0.013)	56.40 (0.009)
LSM	one	39.13 (0.018)	43.17 (0.028)	47.15 (0.020)
LSM	one, payoff	43.26 (0.044)	45.15 (0.022)	47.47 (0.025)
PO	one	46.35 (0.022)	48.91 (0.013)	51.06 (0.016)
PO	one, payoff	17.96 (0.290)	15.86 (0.440)	34.74 (0.264)
RPO	one, payoff	**51.84 (0.013)**	**54.56 (0.008)**	**55.97 (0.009)**
PO-UB	one	57.55 (0.012)	60.29 (0.010)	61.88 (0.009)
PO-UB	one, payoff	53.25 (0.043)	56.10 (0.031)	57.39 (0.017)
LSM	prices	39.03 (0.021)	43.08 (0.029)	47.07 (0.019)
LSM	prices, payoff	42.30 (0.038)	44.55 (0.029)	47.13 (0.022)
PO	prices	45.61 (0.019)	48.04 (0.017)	50.36 (0.022)
PO	prices, payoff	18.09 (0.193)	16.31 (0.361)	34.74 (0.211)
RPO	prices, payoff	**50.92 (0.011)**	**53.59 (0.006)**	**55.21 (0.014)**
PO-UB	prices	57.45 (0.009)	60.27 (0.010)	61.85 (0.010)
PO-UB	prices, payoff	53.19 (0.039)	56.03 (0.032)	57.31 (0.017)
LSM	pricesKO, KOind	50.40 (0.012)	53.69 (0.009)	54.97 (0.010)
LSM	pricesKO, KOind, payoff	50.52 (0.011)	53.27 (0.009)	54.79 (0.007)
PO	pricesKO, KOind	51.60 (0.009)	54.33 (0.008)	55.54 (0.007)
PO	pricesKO, KOind, payoff	51.28 (0.011)	53.90 (0.006)	55.28 (0.007)
RPO	pricesKO, KOind, payoff	**51.77 (0.011)**	**54.47 (0.009)**	**55.88 (0.011)**
PO-UB	pricesKO, KOind	53.29 (0.007)	55.43 (0.008)	56.71 (0.010)
PO-UB	pricesKO, KOind, payoff	52.48 (0.024)	55.03 (0.015)	56.38 (0.011)
LSM	pricesKO	50.32 (0.013)	53.38 (0.006)	54.68 (0.010)
LSM	pricesKO, payoff	50.31 (0.012)	52.90 (0.008)	54.47 (0.007)
PO	pricesKO	50.87 (0.013)	53.45 (0.008)	55.02 (0.018)
PO	pricesKO, payoff	50.84 (0.013)	53.44 (0.007)	54.94 (0.011)
RPO	pricesKO, payoff	**50.93 (0.011)**	**53.60 (0.006)**	**55.21 (0.013)**
PO-UB	pricesKO	53.30 (0.007)	55.44 (0.008)	56.72 (0.011)
PO-UB	pricesKO, payoff	52.50 (0.028)	55.07 (0.015)	56.42 (0.011)
LSM	pricesKO, prices2KO, KOind	50.33 (0.009)	53.18 (0.010)	54.60 (0.012)
LSM	pricesKO, prices2KO, KOind, payoff	50.26 (0.009)	53.03 (0.010)	54.60 (0.010)
RPO	pricesKO, prices2KO, KOind, payoff	**51.00 (0.016)**	**53.67 (0.012)**	**55.17 (0.015)**
Tree	payoff, time	51.85 (0.016)	54.61 (0.008)	56.01 (0.009)
Tree	prices	39.48 (0.125)	42.33 (0.185)	43.86 (0.093)
Tree	prices, payoff	49.31 (0.017)	54.17 (0.009)	55.97 (0.008)
Tree	prices, time	39.47 (0.139)	43.11 (0.020)	46.14 (0.369)
Tree	prices, time, payoff	51.84 (0.015)	54.61 (0.011)	56.01 (0.009)
Tree	prices, time, payoff, KOind	51.84 (0.015)	54.61 (0.011)	56.01 (0.009)

Table EC.2 Average out-of-sample reward for different policies and different basis function architectures for constant barrier instances with $n = 16$ (Section 6.3).

Method	Basis functions	$\bar{p}=90$	$\bar{p}=100$	$\bar{p}=110$
LSM	KOind	1.52 (0.090)	1.41 (0.118)	1.51 (0.079)
LSM	KOind, payoff	0.12 (0.004)	0.11 (0.003)	0.41 (0.100)
PO	KOind	94.85 (1.185)	92.73 (0.635)	89.74 (0.880)
PO	KOind, payoff	102.90 (1.107)	100.44 (0.595)	101.17 (1.106)
RPO	KOind, payoff	11.85 (0.069)	11.66 (0.105)	11.46 (0.084)
LSM	one	0.24 (0.065)	0.17 (0.014)	0.08 (0.003)
LSM	one, payoff	0.15 (0.009)	0.22 (0.076)	0.16 (0.026)
PO	one	95.23 (0.826)	89.51 (0.511)	93.22 (0.816)
PO	one, payoff	100.99 (1.178)	94.56 (0.766)	92.58 (1.469)
RPO	one, payoff	17.34 (0.288)	16.28 (0.116)	16.52 (0.234)
LSM	prices	0.57 (0.045)	0.46 (0.026)	0.49 (0.050)
LSM	prices, payoff	0.43 (0.016)	0.43 (0.006)	0.44 (0.013)
PO	prices	113.49 (1.714)	110.38 (1.349)	109.68 (0.900)
PO	prices, payoff	130.91 (1.718)	128.34 (1.110)	129.85 (1.508)
RPO	prices, payoff	30.91 (1.254)	29.93 (0.531)	28.67 (0.121)
LSM	pricesKO, KOind	0.49 (0.019)	0.47 (0.020)	0.48 (0.014)
LSM	pricesKO, KOind, payoff	0.46 (0.014)	0.53 (0.023)	0.45 (0.014)
PO	pricesKO, KOind	149.32 (2.586)	150.80 (1.707)	147.96 (1.034)
PO	pricesKO, KOind, payoff	286.85 (4.488)	270.23 (3.728)	268.43 (2.355)
RPO	pricesKO, KOind, payoff	38.61 (1.230)	36.46 (0.430)	33.61 (0.375)
LSM	pricesKO	0.40 (0.014)	0.39 (0.010)	0.39 (0.009)
LSM	pricesKO, payoff	0.46 (0.017)	0.45 (0.013)	0.45 (0.007)
PO	pricesKO	275.34 (3.959)	255.65 (1.843)	251.46 (1.131)
PO	pricesKO, payoff	176.87 (2.920)	169.35 (1.181)	172.09 (1.170)
RPO	pricesKO, payoff	35.06 (0.600)	34.41 (0.262)	34.75 (0.385)
LSM	pricesKO, prices2KO, KOind	1.88 (0.018)	1.98 (0.038)	2.34 (0.182)
LSM	pricesKO, prices2KO, KOind, payoff	2.00 (0.028)	2.17 (0.040)	2.03 (0.034)
PO	pricesKO, prices2KO, KOind	1074.44 (17.873)	1177.70 (20.110)	1107.16 (22.704)
PO	pricesKO, prices2KO, KOind, payoff	1161.63 (28.464)	1371.78 (30.065)	1230.27 (34.084)
RPO	pricesKO, prices2KO, KOind, payoff	92.74 (1.973)	105.22 (3.007)	95.33 (2.065)
Tree	payoff, time	7.83 (0.197)	4.79 (0.197)	3.60 (0.142)
Tree	prices	49.22 (2.696)	68.31 (1.896)	67.10 (1.599)
Tree	prices, payoff	4.64 (0.072)	5.02 (0.164)	4.26 (0.205)
Tree	prices, time	81.74 (5.552)	56.28 (5.300)	44.03 (3.539)
Tree	prices, time, payoff	14.05 (0.231)	8.85 (0.442)	6.69 (0.234)
Tree	prices, time, payoff, KOind	14.23 (0.312)	8.55 (0.513)	6.86 (0.175)

Table EC.3 Average computation time for different policies and different basis function architectures for constant barrier instances with $n=8$ (Section 6.3).

Method	Basis functions	$\bar{p} = 90$	$\bar{p} = 100$	$\bar{p} = 110$
LSM	KOind	45.25 (0.046)	58.94 (0.034)	68.00 (0.031)
LSM	KOind, payoff	54.51 (0.039)	66.44 (0.035)	72.90 (0.067)
PO	KOind	51.51 (0.025)	64.97 (0.030)	70.61 (0.034)
PO	KOind, payoff	54.27 (0.047)	65.24 (0.047)	68.03 (0.053)
RPO	KOind, payoff	**55.34 (0.034)**	**68.98 (0.028)**	**76.99 (0.028)**
PO-UB	KOind	67.65 (0.062)	81.53 (0.053)	88.32 (0.065)
PO-UB	KOind, payoff	55.52 (0.027)	69.60 (0.066)	78.90 (0.060)
LSM	one	44.96 (0.051)	57.12 (0.041)	60.38 (0.066)
LSM	one, payoff	54.38 (0.044)	65.22 (0.043)	67.47 (0.036)
PO	one	51.40 (0.025)	64.27 (0.033)	68.35 (0.051)
PO	one, payoff	54.13 (0.034)	64.60 (0.034)	66.85 (0.049)
RPO	one, payoff	**55.34 (0.034)**	**68.98 (0.028)**	**76.99 (0.028)**
PO-UB	one	68.15 (0.064)	83.37 (0.063)	93.81 (0.071)
PO-UB	one, payoff	55.52 (0.029)	69.64 (0.069)	79.14 (0.061)
LSM	prices	47.39 (0.044)	59.41 (0.048)	59.69 (0.072)
LSM	prices, payoff	54.67 (0.035)	65.63 (0.041)	67.37 (0.038)
PO	prices	52.22 (0.023)	64.86 (0.035)	68.23 (0.056)
PO	prices, payoff	54.38 (0.038)	64.57 (0.031)	66.79 (0.039)
RPO	prices, payoff	**55.12 (0.029)**	**68.23 (0.037)**	**75.32 (0.024)**
PO-UB	prices	64.62 (0.034)	79.73 (0.069)	90.93 (0.042)
PO-UB	prices, payoff	55.52 (0.028)	69.63 (0.065)	79.13 (0.065)
LSM	pricesKO, KOind	49.40 (0.026)	62.84 (0.039)	69.72 (0.038)
LSM	pricesKO, KOind, payoff	54.71 (0.030)	67.08 (0.027)	73.47 (0.047)
PO	pricesKO, KOind	52.88 (0.023)	65.58 (0.036)	70.36 (0.038)
PO	pricesKO, KOind, payoff	54.62 (0.031)	66.26 (0.034)	70.15 (0.038)
RPO	pricesKO, KOind, payoff	**55.26 (0.032)**	**68.82 (0.031)**	**76.78 (0.028)**
PO-UB	pricesKO, KOind	64.13 (0.033)	77.93 (0.060)	85.46 (0.051)
PO-UB	pricesKO, KOind, payoff	55.52 (0.031)	69.59 (0.064)	78.88 (0.062)
LSM	pricesKO	47.83 (0.042)	61.87 (0.040)	69.59 (0.038)
LSM	pricesKO, payoff	54.80 (0.034)	67.13 (0.031)	73.71 (0.046)
PO	pricesKO	52.39 (0.021)	65.72 (0.032)	70.78 (0.033)
PO	pricesKO, payoff	54.60 (0.032)	66.10 (0.040)	69.82 (0.040)
RPO	pricesKO, payoff	**55.12 (0.029)**	**68.24 (0.036)**	**75.34 (0.025)**
PO-UB	pricesKO	64.49 (0.035)	78.77 (0.047)	86.31 (0.055)
PO-UB	pricesKO, payoff	55.51 (0.029)	69.60 (0.063)	78.89 (0.057)
Tree	payoff, time	54.78 (0.030)	66.74 (0.021)	71.27 (0.128)
Tree	prices	37.66 (0.032)	49.16 (0.039)	56.89 (0.080)
Tree	prices, payoff	39.38 (0.022)	51.42 (0.033)	59.89 (0.033)
Tree	prices, time	53.91 (0.030)	65.37 (0.085)	70.62 (0.048)
Tree	prices, time, payoff	54.78 (0.030)	66.74 (0.021)	71.27 (0.128)
Tree	prices, time, payoff, KOind	54.78 (0.030)	66.74 (0.021)	71.27 (0.128)

Table EC.4 Average out-of-sample reward for different policies and different basis function architectures for time-varying barrier instances with $n = 8$ (Section 6.4).

Method	Basis functions	$\bar{p} = 90$	$\bar{p} = 100$	$\bar{p} = 110$
LSM	KOind	78.77 (0.035)	85.79 (0.026)	65.16 (0.062)
LSM	KOind, payoff	84.62 (0.040)	90.72 (0.082)	79.76 (0.064)
PO	KOind	82.99 (0.030)	86.76 (0.040)	62.53 (0.053)
PO	KOind, payoff	83.33 (0.030)	84.10 (0.056)	70.30 (0.179)
RPO	KOind, payoff	**86.88 (0.020)**	**94.56 (0.027)**	**81.14 (0.030)**
PO-UB	KOind	96.91 (0.055)	103.22 (0.027)	88.01 (0.090)
PO-UB	KOind, payoff	87.67 (0.078)	97.17 (0.058)	86.20 (0.083)
LSM	one	76.19 (0.056)	72.11 (0.066)	52.23 (0.034)
LSM	one, payoff	82.17 (0.035)	81.00 (0.097)	59.34 (0.093)
PO	one	82.01 (0.035)	82.88 (0.032)	57.56 (0.043)
PO	one, payoff	81.59 (0.045)	80.70 (0.083)	55.34 (0.062)
RPO	one, payoff	**86.88 (0.020)**	**94.56 (0.027)**	**81.14 (0.030)**
PO-UB	one	99.05 (0.055)	110.71 (0.030)	105.18 (0.075)
PO-UB	one, payoff	87.72 (0.079)	97.68 (0.068)	88.43 (0.077)
LSM	prices	77.14 (0.053)	70.73 (0.079)	51.73 (0.034)
LSM	prices, payoff	82.31 (0.039)	80.90 (0.073)	58.60 (0.095)
PO	prices	82.31 (0.032)	82.66 (0.032)	56.88 (0.039)
PO	prices, payoff	81.56 (0.034)	80.56 (0.061)	55.23 (0.055)
RPO	prices, payoff	**86.16 (0.013)**	**92.84 (0.046)**	**78.21 (0.038)**
PO-UB	prices	97.77 (0.043)	109.82 (0.034)	104.88 (0.072)
PO-UB	prices, payoff	87.66 (0.082)	97.62 (0.071)	88.32 (0.075)
LSM	pricesKO, KOind	80.32 (0.033)	86.18 (0.030)	68.41 (0.133)
LSM	pricesKO, KOind, payoff	84.88 (0.032)	90.69 (0.066)	79.44 (0.059)
PO	pricesKO, KOind	83.30 (0.024)	86.50 (0.045)	62.10 (0.055)
PO	pricesKO, KOind, payoff	83.92 (0.025)	85.83 (0.051)	71.61 (0.135)
RPO	pricesKO, KOind, payoff	**86.45 (0.019)**	**93.82 (0.034)**	**80.19 (0.040)**
PO-UB	pricesKO, KOind	95.63 (0.039)	102.26 (0.040)	87.62 (0.084)
PO-UB	pricesKO, KOind, payoff	87.62 (0.077)	97.09 (0.057)	86.11 (0.075)
LSM	pricesKO	79.96 (0.033)	86.22 (0.030)	65.53 (0.091)
LSM	pricesKO, payoff	84.89 (0.026)	90.83 (0.074)	**78.24 (0.046)**
PO	pricesKO	83.34 (0.029)	86.63 (0.045)	62.09 (0.051)
PO	pricesKO, payoff	83.88 (0.026)	85.83 (0.051)	70.60 (0.120)
RPO	pricesKO, payoff	**86.17 (0.013)**	**92.85 (0.045)**	78.20 (0.038)
PO-UB	pricesKO	96.04 (0.046)	102.57 (0.021)	87.68 (0.087)
PO-UB	pricesKO, payoff	87.62 (0.078)	97.10 (0.056)	86.11 (0.077)
Tree	payoff, time	84.34 (0.033)	86.79 (0.088)	61.21 (0.054)
Tree	prices	55.70 (0.240)	62.45 (0.328)	52.13 (0.165)
Tree	prices, payoff	67.10 (0.021)	75.68 (0.026)	60.98 (0.043)
Tree	prices, time	81.40 (0.030)	80.67 (0.060)	57.39 (0.092)
Tree	prices, time, payoff	84.34 (0.033)	86.79 (0.088)	61.21 (0.057)
Tree	prices, time, payoff, KOind	84.34 (0.033)	86.79 (0.088)	61.21 (0.057)

Table EC.5 Average out-of-sample reward for different policies and different basis function architectures for time-varying barrier instances with $n = 32$ (Section 6.4).