Approximate Dynamic Programming for Large Scale Systems

Vijay V. Desai

Submitted in partial fulfillment of the

requirements for the degree

of Doctor of Philosophy

in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2012

ABSTRACT

Approximate Dynamic Programming for Large Scale Systems

Vijay V. Desai

Sequential decision making under uncertainty is at the heart of a wide variety of practical problems. These problems can be cast as dynamic programs and the optimal value function can be computed by solving Bellman's equation. However, this approach is limited in its applicability. As the number of state variables increases, the state space size grows exponentially, a phenomenon known as the curse of dimensionality, rendering the standard dynamic programming approach impractical.

An effective way of addressing curse of dimensionality is through parameterized value function approximation. Such an approximation is determined by relatively small number of parameters and serves as an estimate of the optimal value function. But in order for this approach to be effective, we need Approximate Dynamic Programming (ADP) algorithms that can deliver 'good' approximation to the optimal value function and such an approximation can then be used to derive policies for effective decision-making. From a practical standpoint, in order to assess the effectiveness of such an approximation, there is also a need for methods that give a sense for the suboptimality of a policy. This thesis is an attempt to address both these issues.

First, we introduce a new ADP algorithm based on linear programming, to compute value function approximations. LP approaches to approximate DP have typically relied on a natural 'projection' of a well studied linear program for exact dynamic programming. Such programs restrict attention to approximations that are lower bounds to the optimal cost-to-go function. Our program – the 'smoothed approximate linear program' – is distinct from such approaches and relaxes the restriction to lower bounding approximations in an appropriate fashion while remaining computationally tractable. The resulting program enjoys strong approximation guarantees and is

shown to perform well in numerical experiments with the game of Tetris and queueing network control problem.

Next, we consider optimal stopping problems with applications to pricing of high-dimensional American options. We introduce the pathwise optimization (PO) method: a new convex optimization procedure to produce upper and lower bounds on the optimal value (the 'price') of high-dimensional optimal stopping problems. The PO method builds on a dual characterization of optimal stopping problems as optimization problems over the space of martingales, which we dub the martingale duality approach. We demonstrate via numerical experiments that the PO method produces upper bounds and lower bounds (via suboptimal exercise policies) of a quality comparable with state-of-the-art approaches. Further, we develop an approximation theory relevant to martingale duality approaches in general and the PO method in particular.

Finally, we consider a broad class of MDPs and introduce a new tractable method for computing bounds by consider information relaxation and introducing penalty. The method delivers tight bounds by identifying the best penalty function among a parameterized class of penalty functions. We implement our method on a high-dimensional financial application, namely, optimal execution and demonstrate the practical value of the method vis-a-vis competing methods available in the literature. In addition, we provide theory to show that bounds generated by our method are provably tighter than some of the other available approaches.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

In everybody's life there are certain rare, definitive periods, which set the course for future life. The time I got to spend with my advisor, Ciamac Moallemi, will certainly be such a pivotal period for my career. During this time, I got a chance to work on some incredibly exciting projects, that not only introduced me to the area of *approximate dynamic programming*, but also provided me with a deeper understanding of operations research. I am very grateful to my advisor for providing me this opportunity.

I would like to thank Professors Mark Broadie, Martin Haugh, Garud Iyengar and Jay Sethuraman for agreeing to serve on my thesis committee. Their comments and discussion have improved the overall quality of the thesis.

Many thanks to Vivek Farias for collaborating on all the research projects. Working with him has been an immense learning opportunity for me. Both Vivek and Ciamac, together have been the perfect examples for how to conduct quality research. I would also like to thank Prof. Guillermo Gallego for collaborating with me on revenue management projects and giving me valuable research experience.

Doctoral studies requires considerable amount of solitary work and can get fairly stressful at times. A number of people have been instrumental in making my journey pleasant and smooth over the last six years. I would like to especially thank Freyr Hermannsson, Abhinav Verma, Anuj Kumar, Anuj Manuja, Shyam Sundar Chandramouli, Pannagadatta Shivaswamy, Vikram Deshpande, Srinivas Chivukula and Anurag Mathur with whom I have some fond memories of Columbia.

Probably my parents were happier than me on my successful defense. As always, their support has been critical for all my endeavors in life. I would like to thank my brother Ravi, for cultivating in me, early on, the joy of scientific pursuit. Ravi and my sister-in-law Smita's presence in New York, made sure home and good food were not more than an hour away. Finally, I would like to thank my wife Shailaja for creating a happy home during the last year of my studies.

1

INTRODUCTION

We are interested in complex systems that evolve with time. The evolution of the system can be influenced by the control applied, but is also subject to random disturbances. Associated with the system is a cost that depends on the 'state' and the control applied. The goal is to choose controls so that we minimize cost accumulated over the course of evolution of the system. Given the broad nature of the setting, a large number of problems from diverse areas like economics to business to engineering can be captured in this framework. However, for most of these problems, calculating optimal control is impossible in the face of current computing power limitations. The subject of this thesis is designing effective suboptimal control for such systems.

1.1. Markov Decision Problem

In order to bring out ideas and motivate discussion, we present one of the simplest Markov Decision Processes (MDPs). Consider a system characterized by finite state space $\mathcal{X}$ and finite action space $\mathcal{A}$ that evolves over a discrete time horizon $\mathcal{T} = \{0, 1, \ldots, T\}$. At time t, the controller observes the state of the system $x_t \in \mathcal{X}$ and takes action $a_t \in \mathcal{A}$ and the cost incurred is $g_t(x_t, a_t)$. We depict this pictorially in Figure 1.1. The system probabilistically transitions to the next state according to transition kernel P and the distribution over the next state is given by $P_{a_t}(x_t, \cdot)$.

Define *optimal value function* $J^*\colon \mathcal{X} \times \mathcal{T} \to \mathbb{R}$, where $J^*(x, t)$ captures the minimum cost incurred by starting from state $x \in \mathcal{X}$ at time $t \in \mathcal{T}$ and acting 'optimally' in the future.

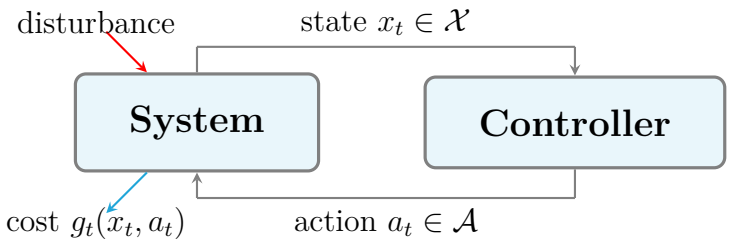


Figure 1.1 Markov Decision Problem

Then, J^* satisfies the following *Bellman's equation*:

$$J^*(x,t) = \begin{cases} \min_{a\in\mathcal{A}} \{g_t(x,a) + \sum_{x'\in\mathcal{X}} P_a(x,x')J^*(x',t+1)\} & \text{if } t < T \\ \min_{a\in\mathcal{A}} g_t(x,a) & \text{if } t = T, \end{cases} \tag{1.1}$$

for all $x \in \mathcal{X}$ and $t \in \mathcal{T}$. On the other hand, given optimal value function J^*, we can derive optimal policy. Given optimal policy $\mu^*\colon \mathcal{X} \times \mathcal{T} \to \mathcal{A}$, it satisfies

$$\mu^*(x,t) \in \operatorname*{argmin}_{a\in\mathcal{A}} \left\{ g_t(x,a) + \sum_{x'\in\mathcal{X}} P_a(x,x')J^*(x',t+1) \right\}, \tag{1.2}$$

for all $x \in \mathcal{X}$ and $t \in \mathcal{T}$.

For a finite state space problem as stated above, one can imagine creating a lookup table, which given a state $x \in \mathcal{X}$ and time $t \in \mathcal{T}$, returns the value $J^*(x,t)$[1]. This is referred to as the *lookup table representation* of the value function. However, as the number of state variables increases, the size of state space typically grows exponentially. This phenomenon is referred to as the *curse of dimensionality* and renders dynamic programming intractable in the face of problems of practical scale.

1.2. Motivating Applications

In order to gain appreciation for the kind of problems we have in mind, consider the following applications.

[1]Computing and storing a lookup table is not the only approach. In certain very special cases, for example, in the case of Linear Quadratic Control problem, one can guess the functional form of the optimal value function and then very efficiently determine the function through recursion.

1.2.1. Tetris

Tetris is a popular video game designed and developed by Alexey Pazhitnov in 1985. The Tetris board, illustrated in Figure 1.2, consists of a two-dimensional grid of 20 rows and 10 columns. The game starts with an empty grid and pieces fall randomly one after another. Each piece consists of four blocks and the player can impart any rotation and translation to the piece. The pieces come in seven different shapes and the next piece to fall is chosen from among these with equal probability. Whenever the pieces are placed such that there is an entire horizontal row or line of contiguous blocks formed, a point is earned and the line gets cleared. Once the board has enough blocks such that the incoming piece cannot be placed for all translations and rotations, the game terminates. Hence the goal of the player is to clear maximum number of lines before the board gets full.

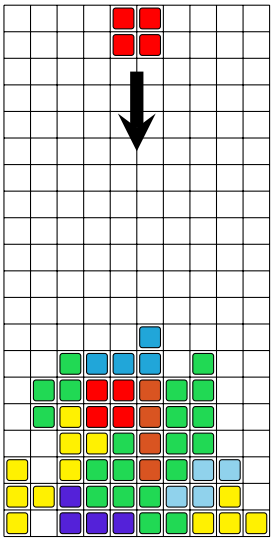


Figure 1.2 Example of a Tetris board configuration.

Tetris can be considered as a Markov decision problem (Farias and Van Roy, 2006) with the 'state' at a particular time encoding the current board configuration and the shape of the next falling piece, while the 'action' determines the placement of the falling piece. Thus, given a state and an action, the subsequent state is determined by the new configuration of the board following placement, and the shape of a new falling piece that is selected

uniformly at random. The objective of Tetris is to maximize reward, where, given a state and an action, the per stage reward is defined to be the number of rows that are cleared following the placement of the falling piece.

This MDP has the advantage of state space being finite, however, a crude estimate tell us that the state space size is approximately 2^{200}. Considering the staggering scale of the problem, solving it via standard approach of Bellman's equation, is a nonstarter.

1.2.2. Pricing of American Options

Financial derivatives are contracts whose payoff is contingent on the price of the underlying stock, bond or commodities. A simple example of a derivative is a European option whose payoff can be collected only on a predetermined exercise date. On the other hand, an American option provides more flexibility by allowing the exercise date to be any time during the lifetime of the contract.

These derivative securities have come to be fundamental financial products that are traded (and hence need to be valued) in a wide variety of markets including equity, commodity, foreign exchange, etc. Moreover, there are billions of dollars worth options that are held by various financial institutions who need to decide each day whether to exercise or continue to hold the option. Hence there is need for fast computational methods, whose solution time scales gracefully with the problem size.

Pricing of American options can be cast as a fundamental stochastic control problem, namely, optimal stopping. Solving it would give us a decision rule for when to exercise the option. Such a decision rule would have to be a function of the state i.e. all asset prices. Suppose the payoff depends on n stocks that evolve according to Geometric Brownian Motion (GBM), the decision rule is a function of $\mathbb{R}^n$. For 'small' n, approaches using finite difference and binomial method work well in practice, but when n is 'large', we have a high-dimensional space, which renders these approaches impractical.

1.3. Approximate Dynamic Programming

Intractable dynamic programs characterized by high-dimensional state space are a common phenomenon and some examples were presented in Section 1.2. The classical approach of solving Bellman's equation is a nonstarter for such problems. An effective approach, for a *minimization dynamic program*, would provide us with the following two things:

1. **Policies/Upper Bounds.** We are interested in policies that give us an easily implementable decision rule and in addition, by simulating such a policy, we can obtain an estimate of the upperbound on the optimal objective function value.

2. **Lower Bounds.** We would also like to complement these upper bounds by computing lower bounds to objective over all feasible policies. These lower bounds would give us a sense for the suboptimality of the computed policy.

Over the course of this section, we will see that approximations to value functions, will be key to computing policies/upper bounds and lower bounds on the objective value. We begin with a discussion of approximations to value function.

1.3.1. Approximations to Value Function

Value function approximations address the curse of dimensionality through the use of parameterized function approximations. In particular, it is common to focus on linear parameterizations. Consider a collection of *basis functions* $\{\phi_1, \ldots, \phi_K\}$ where each $\phi_i \colon \mathcal{X} \to \mathbb{R}$ is a real-valued function on the state space. ADP algorithms seek to find linear combinations of the basis functions that provide good approximations to the optimal value function function. In particular, we seek a vector of weights $r \in \mathcal{T} \times \mathbb{R}^K$ so that

$$J^r(x,t) \triangleq \sum_{i=1}^{K} \phi_i(x) r_{t,i} = \Phi r_t(x) \approx J^*(x,t).$$

Here, we define $\Phi \triangleq [\phi_1 \ \phi_2 \ \ldots \ \phi_K]$ to be a matrix with columns consisting of the basis functions. Given such an approximation, one can derive policies by using (1.2) and substituting

the value function approximation instead of optimal value function. Thus, the corresponding policy is given by

$$\mu^r(x,t) \in \underset{a\in\mathcal{A}}{\operatorname{argmin}} \left\{ g_t(x,a) + \sum_{x'\in\mathcal{X}} P_a(x,x')\Phi r_{t+1}(x') \right\}. \tag{1.3}$$

This approach is similar in spirit to statistical regression, where given a problem, the user selects a set of functions, whose linear combination is used as a predictor for the output. However, the key difference is, in the context of ADP, there is no training set with input-output pairs.

An alternate interpretation is basis functions can be viewed as 'features' that given a state, capture 'relevant' information for effective decision making. For example, in the case of Tetris introduced in Section 1.2.1, one set of basis functions that has been commonly employed was introduced by Bertsekas and Ioffe (1996). These basis functions essentially capture the height of the 'wall', 'jaggedness' of the top of the wall, and the number of holes in the wall; the features a human player might consider worth focussing on. Thus, we can intuitively think of basis functions, as a way of incorporating our knowledge or intuition about the 'crucial' features of the problem.

Another advantage of value function approximations is they are a *compact representation* and require very little storage. Typically, one would store the weights r and general structure of the basis functions and value function approximations $\Phi r_t(x)$ are generated only when needed. In contrast the optimal value function, generally requires a lookup table representation and requires storage space on the order of state space size.

Given a basis function architecture, there a number of algorithms for computing a value function approximation. We will introduce approximate linear programming method in the following section.

1.3.2. Approximate Linear Programming

A dynamic program solution is characterized by value function and is the main objective of computation. Although, value function is an arbitrary function defined on the state space, it

can be characterized as the solution to (exact) linear program and this approach is credited to Manne (1960). However, this linear program does not overcome the curse of dimensionality. In particular, it is a program with as many variables as the state space size and atleast as many constraints and hence could be a very large scale linear program.

Approximate Linear Program (ALP) reduces the number of variables in exact linear program by focussing on linearly parameterized value function approximations. This approach was introduced by Schweitzer and Seidmann (1985) and later analyzed by de Farias and Van Roy (2003, 2004). de Farias and Van Roy (2003, 2004) introduce the constraint sampling approach to obtain tractable dynamic programs and also establish approximation and performance guarantees to demonstrate the soundness of this approach. From a practical standpoint, ALP approach allows us to capitalize on linear programming, which is a mature technology and there are a number of reliable commercial solvers.

A testament to the success of the ALP approach is the number of applications it has seen in recent years in large scale dynamic optimization problems. Applications range from scheduling in queueing networks (Moallemi et al., 2008; Morrison and Kumar, 1999; Veatch, 2005), revenue management (Adelman, 2007; Farias and Van Roy, 2007; Zhang and Adelman, 2008), portfolio management (Han, 2005), inventory problems (Adelman, 2004; Adelman and Klabjan, 2009), and algorithms for solving stochastic games (Farias et al., 2011) among others. Remarkably, in applications such as network revenue management, control policies produced via the LP approach (namely, Adelman, 2007; Farias and Van Roy, 2007) are competitive with ADP approaches that carefully exploit problem structure, such as, for example, that of Topaloglu (2009).

ADP provides methods, for example ALP, to compute approximations to value functions. In the context of a minimization dynamic program, simulating a policy yields upper bounds. However, in order to assess the quality of these policies, we would require lower bounds on the optimal objective.

1.3.3. Dual Approach

A general approach to obtaining bounds is by considering relaxation of information process. By allowing oneself to look into future, one would expect only to do better and hence obtain bounds on the optimal value. Further, in the spirit of Lagrangian duality, we can also impose penalty for this relaxation. This approach of obtaining lower bounds by using information relaxation, while simultaneously introducing penalty for this relaxation, will be referred to as the *dual approach*.

These methods originated in the context of American option pricing literature and have become popular following the work of Rogers (2002), Haugh and Kogan (2004) and Andersen and Broadie (2004). Generalization of this approach, to control problems other than optimal stopping, have been studied by Rogers (2008) and Brown et al. (2010). Following their work, these methods have seen applications in areas like portfolio optimization (Brown and Smith, 2010), valuation of natural gas storage (Lai et al., 2010a,b), among others.

While these methods have been applied successfully to demonstrate near optimality of certain heuristics, one drawback is they require considerable amount of problem-specific work in identifying the 'right' penalty functions. We address this issue by providing a generic approach for computing the best penalty within a user specified parameterized family. We introduce this approach first in the context of optimal stopping problems in Chapter 3 and generalize this approach to MDPs in Chapter 4.

1.4. Organization of This Thesis

- **Chapter 2.** In this chapter, we present a novel linear program called the 'smoothed approximate linear program' for approximating the cost-to-go function in high-dimensional stochastic control problems. We demonstrate bounds on the quality of approximation to the optimal cost-to-go function afforded by our approach. These bounds are, in general, no worse than those available for extant LP approaches, and for specific problem instances can be shown to be arbitrarily stronger. Second, experiments with our

approach on a pair of challenging problems (the game of Tetris and a queueing network control problem) show that the approach outperforms the existing LP approach by a substantial margin.

- **Chapter 3.** In this chapter, we introduce the pathwise optimization (PO) method, a new convex optimization procedure to produce upper and lower bounds on the optimal value of a high-dimensional optimal stopping problem. The PO method builds on a dual characterization of optimal stopping problems as optimization problems over the space of martingales, which we dub the martingale duality approach. We demonstrate via numerical experiments that the PO method produces upper and lower bounds of a quality comparable with state-of-the-art approaches. Further, we develop an approximation theory relevant to martingale duality approaches in general and the PO method in particular. Finally, we compare the bounds produced by PO method with other linear programming based ADP methods and in doing so show that the PO method dominates those alternatives.

- **Chapter 4** In this chapter, we generalize the pathwise method to a broader class of MDPs. We propose a class of value function approximations, which can be used to generate penalties that result in a tractable formulation. Given a parameterization of this class, PO method provides a structured approach to determining the best penalty within this class by solving a convex optimization problem. As an application of this procedure, we consider the problem of optimal execution. In numerical experiments, we observe that PO provides stronger bounds relative to the Brown and Smith (2010) approach, with very little incremental computational burden. In theory, PO bounds can be shown to dominate bounds obtained via ALP approach and semidefinite programming based approach introduced by Wang and Boyd (2011).

- **Chapter 5** In this chapter, we offer some concluding remarks, and discuss directions for future work.

2

SMOOTHED APPROXIMATE LINEAR PROGRAM

2.1. Introduction

Many dynamic optimization problems can be cast as Markov decision problems (MDPs) and solved, in principle, via dynamic programming. Unfortunately, this approach is frequently untenable due to the 'curse of dimensionality'. Approximate dynamic programming (ADP) is an approach which attempts to address this difficulty. ADP algorithms seek to compute good approximations to the dynamic programming optimal cost-to-go function within the span of some pre-specified set of basis functions.

ADP algorithms are typically motivated by exact algorithms for dynamic programming. The approximate linear programming (ALP) method is one such approach, motivated by the LP used for the computation of the optimal cost-to-go function. Introduced by Schweitzer and Seidmann (1985) and analyzed and further developed by de Farias and Van Roy (2003, 2004), this approach is attractive for a number of reasons. First, the availability of efficient solvers for linear programming makes the ALP approach easy to implement. Second, the approach offers attractive theoretical guarantees. In particular, the quality of the approximation to the cost-to-go function produced by the ALP approach can be shown to compete, in an appropriate sense, with the quality of the best possible approximation afforded by the set of basis functions used. A testament to the success of the ALP approach is the number of applications it has seen in recent years in large scale dynamic optimization problems.

These applications range from the control of queueing networks to revenue management to the solution of large scale stochastic games.

The optimization program employed in the ALP approach is in some sense the most natural linear programming formulation for ADP. In particular, the ALP is identical to the linear program used for exact computation of the optimal cost-to-go function, with further constraints limiting solutions to the low-dimensional subspace spanned by the basis functions used. The resulting LP implicitly restricts attention to approximations that are lower bounds to the optimal cost-to-go function. The structure of this program appears crucial in establishing guarantees on the quality of approximations produced by the approach (de Farias and Van Roy, 2003, 2004); these approximation guarantees were remarkable and a first for any ADP method. That said, the restriction to lower bounds naturally leads one to ask whether the program employed by the ALP approach is the 'right' math programming formulation for ADP. In particular, it may be advantageous to consider a generalization of the ALP approach that relaxes the lower bound requirement so as to allow for a better approximation, and, ultimately, better policy performance. Is there an alternative formulation that permits better approximations to the cost-to-go function while remaining computationally tractable? Motivated by this question, the present chapter introduces a new linear program for ADP we call the 'smoothed' approximate linear program (or SALP). This program is a generalization of the ALP method. We believe that the SALP represents a useful new math programming formulation for ADP. In particular, we make the following contributions:

1. We are able to establish strong approximation and performance guarantees for approximations to the cost-to-go function produced by the SALP. Our analyses broadly follow the approach of de Farias and Van Roy (2003, 2004) for the ALP. The resultant guarantees are no worse than the corresponding guarantees for the ALP, and we demonstrate that they can be *substantially* stronger in certain cases.

2. The number of constraints and variables in the SALP scale with the size of the MDP state space. We nonetheless establish sample complexity bounds that demonstrate that an appropriate 'sampled' SALP provides a good approximation to the SALP solution

with a tractable number of sampled MDP states. Moreover, we identify structural properties of the sampled SALP that can be exploited for fast optimization. Our sample complexity results and these structural observations allow us to conclude that the SALP scales similarly in computational complexity as existing LP formulations for ADP.

3. We present computational studies demonstrating the efficacy of our approach in the setting of two different challenging control problems. In the first study, we consider the game of Tetris. Tetris is a notoriously difficult, 'unstructured' dynamic optimization problem and has been used as a convenient testbed problem for numerous ADP approaches. The ALP has been demonstrated to be competitive with other ADP approaches for Tetris, such as temporal difference learning or policy gradient methods (see Farias and Van Roy, 2006). In detailed comparisons with the ALP, we show that the SALP provides an *order of magnitude* improvement over controllers designed via that approach for the game of Tetris. In the second computational study, we consider the optimal control of a 'criss-cross' queueing network. this is a challenging network control problem and a difficult test problem as witnessed by antecedent literature. Under several distinct parameter regimes, we show here that the SALP adds substantial value over the ALP approach.

   In addition to these results, the SALP method has recently been considered in other applications with favorable results: this includes work on a high-dimensional production optimization problem in oil exploration (Wen et al., 2011), and work studying large scale dynamic oligopoly models (Farias et al., 2011).

The literature on ADP algorithms is vast and we make no attempt to survey it here. Van Roy (2002) or Bertsekas (2007, Chap. 6) provide good, brief overviews, while Bertsekas and Tsitsiklis (1996) and Powell (2007) are encyclopedic references on the topic. The exact LP for the solution of dynamic programs is attributed to Manne (1960). The ALP approach to ADP was introduced by Schweitzer and Seidmann (1985) and de Farias and Van Roy (2003,

2004). de Farias and Van Roy (2003) establish approximation guarantees for the ALP approach. These guarantees are especially strong if the basis spans suitable 'Lyapunov'-like functions. The approach we present yields strong bounds if any such Lyapunov function exists, whether or not it is spanned by the basis. de Farias and Van Roy (2006) introduce a program for average cost approximate dynamic programming that resembles the SALP; a critical difference is that their program requires the relative violation allowed across ALP constraints be specified as input. Contemporaneous with the present work, Petrik and Zilberstein (2009) propose a relaxed linear program for approximating the cost-to-go function of a dynamic program. This linear program is similar to the SALP program (2.14) herein. The crucial determinant of performance in either program is a certain choice of Lagrange multipliers. Our work explicitly identifies such a choice, and for this choice, develops concrete approximation guarantees that compare favorably with guarantees available for the ALP. In addition, the choice of Lagrange multipliers identified also proves to be practically valuable as is borne out by our experiments. In contrast, Petrik and Zilberstein (2009) stop short of identifying this crucial input and thus provide neither approximation guarantees nor a 'generic' choice of multipliers for practical applications.

The remainder of this chapter is organized as follows: In Section 2.2, we formulate the approximate dynamic programming setting and describe the ALP approach. The smoothed ALP is developed as a relaxation of the ALP in Section 2.3. Section 2.4 provides a theoretical analysis of the SALP, in terms of approximation and performance guarantees, as well as a sample complexity bound. In Section 2.5, we describe the practical implementation of the SALP method, illustrating how parameter choices can be made as well as how to efficiently solve the resulting optimization program. Section 2.6 contains the computational study of the game Tetris, while the computational study in Section 2.7 considers a queueing application.

2.2. Problem Formulation

Our setting is that of a discrete-time, discounted infinite-horizon, cost-minimizing MDP with a finite state space $\mathcal{X}$ and finite action space $\mathcal{A}$. At time t, given the current state x_t and

a choice of action a_t, a per-stage cost $g(x_t, a_t)$ is incurred. The subsequent state x_{t+1} is determined according to the transition probability kernel $P_{a_t}(x_t, \cdot)$.

A stationary policy $\mu\colon \mathcal{X} \to \mathcal{A}$ is a mapping that determines the choice of action at each time as a function of the state. Given each initial state $x_0 = x$, the expected discounted cost (cost-to-go function) of the policy μ is given by

$$J_\mu(x) \triangleq \mathsf{E}_\mu \left[\sum_{t=0}^{\infty} \alpha^t g(x_t, \mu(x_t)) \;\middle|\; x_0 = x \right].$$

Here, $\alpha \in (0, 1)$ is the discount factor. The expectation is taken under the assumption that actions are selected according to the policy μ. In other words, at each time t, $a_t \triangleq \mu(x_t)$.

Denote by $P_\mu \in \mathbb{R}^{\mathcal{X}\times\mathcal{X}}$ the transition probability matrix for the policy μ, whose (x, x')th entry is $P_{\mu(x)}(x, x')$. Denote by $g_\mu \in \mathbb{R}^{\mathcal{X}}$ the vector whose xth entry is $g(x, \mu(x))$. Then, the cost-to-go function J_μ can be written in vector form as

$$J_\mu = \sum_{t=0}^{\infty} \alpha^t P_\mu^t g_\mu.$$

Further, the cost-to-go function J_μ is the unique solution to the equation $T_\mu J = J$, where the operator T_μ is defined by $T_\mu J = g_\mu + \alpha P_\mu J$.

Our goal is to find an optimal stationary policy μ^*, that is, a policy that minimizes the expected discounted cost from every state x. In particular,

$$\mu^*(x) \in \operatorname*{argmin}_\mu J_\mu(x), \quad \forall\, x \in \mathcal{X}.$$

The Bellman operator T is defined component-wise according to

$$(TJ)(x) \triangleq \min_{a\in\mathcal{A}} \; g(x, a) + \alpha \sum_{x'\in\mathcal{X}} P_a(x, x') J(x'), \quad \forall\, x \in \mathcal{X}.$$

Bellman's equation is then the fixed point equation

$$TJ = J. \tag{2.1}$$

Standard results in dynamic programming establish that the optimal cost-to-go function J^* is the unique solution to Bellman's equation (see, for example, Bertsekas, 2007, Chap. 1). Further, if μ^* is a policy that is greedy with respect to J^* (i.e., μ^* satisfies $TJ^* = T_{\mu^*} J^*$), then μ^* is an optimal policy.

2.2.1. The Linear Programming Approach

A number of computational approaches are available for the solution of the Bellman equation. One approach involves solving the optimization program:

$$\begin{aligned} &\underset{J}{\text{maximize}} && \nu^\top J \\ &\text{subject to} && J \leq TJ. \end{aligned} \tag{2.2}$$

Here, $\nu \in \mathbb{R}^{\mathcal{X}}$ is a vector with positive components that are known as the *state-relevance weights*. The above program is indeed an LP since for each state x, the constraint $J(x) \leq (TJ)(x)$ is equivalent to the set of $|\mathcal{A}|$ linear constraints

$$J(x) \leq g(x,a) + \alpha \sum_{x' \in \mathcal{X}} P_a(x,x') J(x'), \quad \forall\ a \in \mathcal{A}.$$

We refer to (2.2), which is credited to Manne (1960), as the *exact LP*. A simple argument, included here for completeness, establishes that J^* is the unique optimal solution: suppose that a vector J is feasible for the exact LP (2.2). Since $J \leq TJ$, monotonicity of the Bellman operator implies that $J \leq T^k J$, for any integer $k \geq 1$. Since the Bellman operator T is a contraction, $T^k J$ must converge to the unique fixed point J^* as $k \to \infty$. Thus, we have that $J \leq J^*$. Then, it is clear that every feasible point for (2.2) is a component-wise lower bound to J^*. Since J^* itself is feasible for (2.2), it must be that J^* is the unique optimal solution to the exact LP.

2.2.2. The Approximate Linear Program

In many problems, the size of the state space is enormous due to the curse of dimensionality. In such cases, it may be prohibitive to store, much less compute, the optimal cost-to-go function J^*. In approximate dynamic programming (ADP), the goal is to find tractable approximations to the optimal cost-to-go function J^*, with the hope that they will lead to good policies.

Specifically, consider a collection of *basis functions* $\{\phi_1, \ldots, \phi_K\}$ where each $\phi_i \colon \mathcal{X} \to \mathbb{R}$ is a real-valued function on the state space. ADP algorithms seek to find linear combinations

of the basis functions that provide good approximations to the optimal cost-to-go function. In particular, we seek a vector of weights $r \in \mathbb{R}^K$ so that

$$J^*(x) \approx J_r(x) \triangleq \sum_{i=1}^{K} \phi_i(x) r_i = \Phi r(x).$$

Here, we define $\Phi \triangleq [\phi_1 \ \phi_2 \ \ldots \ \phi_K]$ to be a matrix with columns consisting of the basis functions. Given a vector of weights r and the corresponding value function approximation Φr, a policy μ_r is naturally defined as the 'greedy' policy with respect to Φr, i.e. as $T_{\mu_r}\Phi r = T\Phi r$.

One way to obtain a set of weights is to solve the exact LP (2.2), but restricting to the low-dimensional subspace of vectors spanned by the basis functions. This leads to the *approximate linear program* (ALP), introduced by Schweitzer and Seidmann (1985), which is defined by

$$\begin{array}{ll} \underset{r}{\text{maximize}} & \nu^\top \Phi r \\ \text{subject to} & \Phi r \leq T\Phi r. \end{array} \tag{2.3}$$

For the balance of the chapter, we will make the following assumption:

Assumption 1. *Assume the ν is a probability distribution ($\nu \geq \mathbf{0}$, $\mathbf{1}^\top \nu = 1$), and that the constant function $\mathbf{1}$ is in the span of the basis functions Φ.*

The geometric intuition behind the ALP is illustrated in Figure 2.1(a). Supposed that r_{ALP} is a vector that is optimal for the ALP. Then the approximate value function Φr_{ALP} will lie on the subspace spanned by the columns of Φ, as illustrated by the orange line. Φr_{ALP} will also satisfy the constraints of the exact LP, illustrated by the dark gray region. By the discussion in Section 2.2.1, this implies that $\Phi r_{\text{ALP}} \leq J^*$. In other words, the approximate cost-to-go function is necessarily a point-wise lower bound to the true cost-to-go function in the span of Φ.

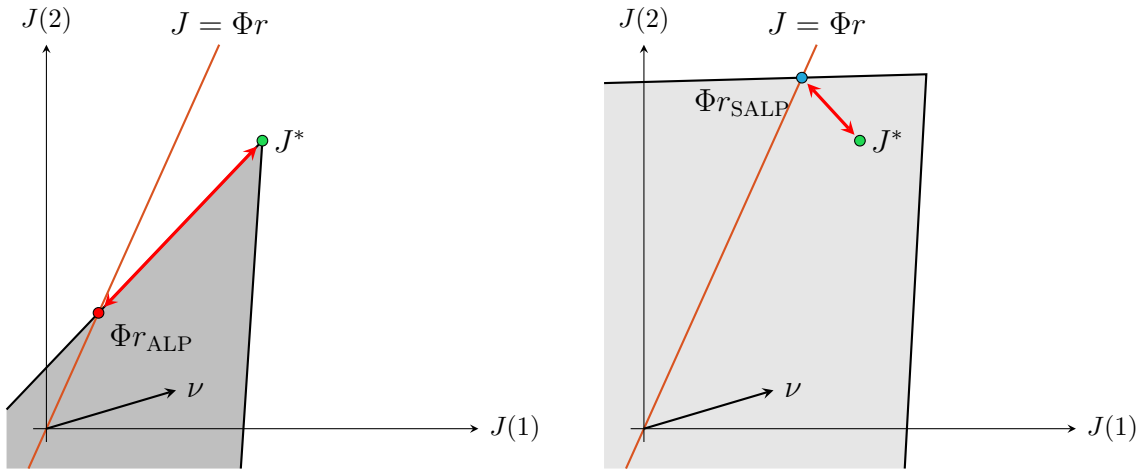
One can thus interpret the ALP solution r_{ALP} equivalently as the optimal solution to the program

$$\begin{array}{ll} \underset{r}{\text{minimize}} & \|J^* - \Phi r\|_{1,\nu} \\ \text{subject to} & \Phi r \leq T\Phi r. \end{array} \tag{2.4}$$

Here, the weighted 1-norm in the objective is defined by

$$\|J^* - \Phi r\|_{1,\nu} \triangleq \sum_{x \in \mathcal{X}} \nu(x)|J^*(x) - \Phi r(x)|.$$

This implies that the approximate LP will find the closest approximation (in the appropriate norm) to the optimal cost-to-go function, out of all approximations satisfying the constraints of the exact LP.



(a) ALP case. (b) SALP case.

Figure 2.1 A cartoon illustrating the feasible set and optimal solution for the ALP and SALP, in the case of a two-state MDP. The axes correspond to the components of the value function. A careful relaxation from the feasible set of the ALP to that of the SALP can yield an improved approximation. It is easy to construct a concrete two state example with the above features.

2.3. The Smoothed ALP

The $J \leq TJ$ constraints in the exact LP, which carry over to the ALP, impose a strong restriction on the cost-to-go function approximation: in particular they restrict us to approximations that are lower bounds to J^* at *every point in the state space*. In the case where the state space is very large, and the number of basis functions is (relatively) small, it may

be the case that constraints arising from rarely visited or pathological states are binding and influence the optimal solution.

In many cases, the ultimate goal is not to find a *lower bound* on the optimal cost-to-go function, but rather to find a *good approximation*. In these instances, it may be that relaxing the constraints in the ALP, so as not to require a uniform lower bound, may allow for better overall approximations to the optimal cost-to-go function. This is also illustrated in Figure 2.1. Relaxing the feasible region of the ALP in Figure 2.1(a) to the light gray region in Figure 2.1(b) would yield the point Φr_{SALP} as an optimal solution. The relaxation in this case is clearly beneficial; it allows us to compute a better approximation to J^* than the point Φr_{SALP}.

Can we construct a fruitful relaxation of this sort in general? The *smoothed approximate linear program* (SALP) is given by:

$$\begin{aligned} \underset{r,s}{\text{maximize}} \quad & \nu^\top \Phi r \\ \text{subject to} \quad & \Phi r \leq T\Phi r + s, \\ & \pi^\top s \leq \theta, \quad s \geq \mathbf{0}. \end{aligned} \tag{2.5}$$

Here, a vector $s \in \mathbb{R}^{\mathcal{X}}$ of additional decision variables has been introduced. For each state x, $s(x)$ is a non-negative decision variable (a slack) that allows for violation of the corresponding ALP constraint. The parameter $\theta \geq 0$ is a non-negative scalar. The parameter $\pi \in \mathbb{R}^{\mathcal{X}}$ is a probability distribution known as the *constraint violation distribution*. The parameter θ is thus a *violation budget*: the expected violation of the $\Phi r \leq T\Phi r$ constraint, under the distribution π, must be less than θ.

The SALP can be alternatively written as

$$\begin{aligned} \underset{r}{\text{maximize}} \quad & \nu^\top \Phi r \\ \text{subject to} \quad & \pi^\top (\Phi r - T\Phi r)^+ \leq \theta. \end{aligned} \tag{2.6}$$

Here, given a vector J, $J^+(x) \triangleq \max(J(x), 0)$ is defined to be the component-wise positive part. Note that, when $\theta = 0$, the SALP is equivalent to the ALP. When $\theta > 0$, the SALP replaces the 'hard' constraints of the ALP with 'soft' constraints in the form of a hinge-loss function.

The balance of the chapter is concerned with establishing that the SALP forms the basis of a useful approximate dynamic programming algorithm in large scale problems:

- We identify a concrete choice of violation budget θ and an idealized constraint violation distribution π for which the SALP provides a useful relaxation in that the optimal solution can be a better approximation to the optimal cost-to-go function. This brings the cartoon improvement in Figure 2.1 to fruition for general problems.
- We show that the SALP is tractable (i.e., it is well approximated by an appropriate 'sampled' version) and present computational experiments for a hard problem (Tetris) illustrating an order of magnitude improvement over the ALP.

2.4. Analysis

This section is dedicated to a theoretical analysis of the SALP. The overarching objective of this analysis is to provide some assurance of the soundness of the proposed approach. In some instances, the bounds we provide will be directly comparable to bounds that have been developed for the ALP method. As such, a relative consideration of the bounds in these two cases can provide a theoretical comparison between the ALP and SALP methods. In addition, our analysis will serve as a crucial guide to practical implementation of the SALP as will be described in Section 2.5. In particular, the theoretical analysis presented here provides intuition as to how to select parameters such as the state-relevance weights and the constraint violation distribution. We note that all of our bounds are relative to a measure of how well the approximation architecture employed is capable of approximating the optimal cost-to-go function; it is unreasonable to expect non-trivial bounds that are independent of the architecture used.

Our analysis will present three types of results:

- Approximation guarantees (Sections 2.4.2–2.4.4): We establish bounds on the distance between approximations computed by the SALP and the optimal value function J^*,

relative to the distance between the best possible approximation afforded by the chosen basis functions and J^*. These guarantees will indicate that the SALP computes approximations that are of comparable quality to the projection[1] of J^* on to the linear span of Φ. We explicitly demonstrate our approximation guarantees in the context of a simple, concrete queueing example, and show that they can be much stronger than corresponding guarantees for the ALP.

- Performance bounds (Section 2.4.5): While it is desirable to approximate J^* as closely as possible, an important concern is the quality of the policies generated by acting greedily according to such approximations, as measured by their performance. We present bounds on the performance loss incurred, relative to the optimal policy, in using an SALP approximation.

- Sample complexity results (Section 2.4.6): The SALP is a linear program with a large number of constraints as well as variables. In practical implementations, one may consider a 'sampled' version of this program that has a manageable number of variables and constraints. We present sample complexity guarantees that establish bounds on the number of samples required to produce a good approximation to the solution of the SALP. These bounds scale linearly with the number of basis function K and are independent of the size of the state space $\mathcal{X}$.

2.4.1. Idealized Assumptions

Our analysis of the SALP in this section is predicated on the knowledge of an idealized probability distribution over states. In particular, letting μ^* be an optimal policy and P_{μ^*} the associated transition matrix, we will require knowledge of the distribution $\pi_{\mu^*,\nu}$ given by

$$\pi_{\mu^*,\nu}^\top \triangleq (1-\alpha)\nu^\top (I - \alpha P_{\mu^*})^{-1} = (1-\alpha)\sum_{t=0}^{\infty} \alpha^t \nu^\top P_{\mu^*}^t. \tag{2.7}$$

[1] Note that it is intractable to directly compute the projection since J^* is unknown.

Here, ν is an initial distribution over states satisfying Assumption 1. The distribution $\pi_{\mu^*,\nu}$ may be interpreted as yielding the discounted expected frequency of visits to a given state when the initial state is distributed according to ν and the system runs under the policy μ^*. The distribution $\pi_{\mu^*,\nu}$ will be used as the SALP constraint violation distribution in order to develop approximation bounds (Theorems 1–2) and a performance bound (Theorem 3), and as a sampling distribution in our analysis of sample complexity (Theorem 4).

We note that assumptions such as knowledge of the idealized distribution described in the preceding paragraph are not unusual in the analysis of ADP algorithms. In the case of the ALP, one either assumes the the ability to solve a linear program with as many constraints as there are states, or absent that, the 'sampled' ALP introduced by de Farias and Van Roy (2004) requires access to states sampled according to precisely this distribution. Theoretical analyses of other approaches to approximate DP such as approximate value iteration and temporal difference learning similarly rely on the knowledge of specialized sampling distributions that cannot be obtained tractably (see de Farias and Van Roy, 2000).

2.4.2. A Simple Approximation Guarantee

This section presents a first, simple approximation guarantee for the following specialization of the SALP in (2.5),

$$\begin{array}{lll} \underset{r,s}{\text{maximize}} & \nu^\top \Phi r & \\ \text{subject to} & \Phi r \leq T\Phi r + s, & \\ & \pi_{\mu^*,\nu}^\top s \leq \theta, \quad s \geq \mathbf{0}. \end{array} \tag{2.8}$$

Here, the constraint violation distribution is set to be $\pi_{\mu^*,\nu}$.

Before we state our approximation guarantee, consider the following function:

$$\begin{array}{rlll} \ell(r,\theta) \triangleq & \underset{s,\gamma}{\text{minimize}} & \gamma/(1-\alpha) & \\ & \text{subject to} & \Phi r \leq T\Phi r + s + \gamma\mathbf{1}, & \\ & & \pi_{\mu^*,\nu}^\top s \leq \theta, \quad s \geq \mathbf{0}. \end{array} \tag{2.9}$$

We will denote by $s(r,\theta)$ the s component of the solution to (2.9). Armed with this definition, we are now in a position to state our first, crude approximation guarantee:

Theorem 1. *Suppose that r_{SALP} is an optimal solution to the SALP (2.8), and let r^* satisfy*

$$r^* \in \operatorname*{argmin}_r \ \|J^* - \Phi r\|_\infty.$$

Then,

$$\|J^* - \Phi r_{SALP}\|_{1,\nu} \leq \|J^* - \Phi r^*\|_\infty + \ell(r^*, \theta) + \frac{2\theta}{1-\alpha}. \tag{2.10}$$

As we will see shortly in the proof of Theorem 1, given a vector r of basis function weights and a violation budget θ, the quantity $\ell(r, \theta)$ obtained by solving (2.9) defines the minimal translation (in the direction of the constant vector $\mathbf{1}$) of r that yields a feasible solution for (2.8). The above theorem allows us to interpret $\ell(r^*, \theta) + 2\theta/(1-\alpha)$ as an upper bound to the approximation error (in the $\|\cdot\|_{1,\nu}$ norm) associated with the SALP solution r_{SALP}, relative to the error of the *best* approximation r^* (in the $\|\cdot\|_\infty$ norm). Note that this upper bound cannot be computed, in general, since r^* is unknown.

Theorem 1 provides justification for the intuition, described in Section 2.3, that a relaxation of the feasible region of the ALP will result in better value function approximations. To see this, first consider the following lemma (whose proof may be found in Appendix 2.8.1) that characterizes the function $\ell(r, \theta)$:

Lemma 1. *For any $r \in \mathbb{R}^K$ and $\theta \geq 0$:*

(i) $\ell(r, \theta)$ is a finite-valued, decreasing, piecewise linear, convex function of θ.

(ii)

$$\ell(r, \theta) \leq \frac{1+\alpha}{1-\alpha} \|J^* - \Phi r\|_\infty.$$

(iii) The right partial derivative of $\ell(r, \theta)$ with respect to θ satisfies

$$\frac{\partial^+}{\partial \theta^+} \ell(r, 0) = -\left((1-\alpha) \sum_{x \in \Omega(r)} \pi_{\mu^*, \nu}(x) \right)^{-1},$$

where

$$\Omega(r) \triangleq \operatorname*{argmax}_{\{x \in \mathcal{X} \,:\, \pi_{\mu^*,\nu}(x) > 0\}} \Phi r(x) - T\Phi r(x).$$

Then, we have the following corollary:

Corollary 1. *Define $U_{SALP}(\theta)$ to be the upper bound in* (2.10)*, i.e.,*

$$U_{SALP}(\theta) \triangleq \|J^* - \Phi r^*\|_\infty + \ell(r^*, \theta) + \frac{2\theta}{1-\alpha}.$$

Then:

(i)

$$U_{SALP}(0) \leq \frac{2}{1-\alpha}\|J^* - \Phi r^*\|_\infty.$$

(ii) The right partial derivative of $U_{SALP}(\theta)$ with respect to θ satisfies

$$\frac{d^+}{d\theta^+}U_{SALP}(0) = \frac{1}{1-\alpha}\left[2 - \left(\sum_{x\in\Omega(r^*)} \pi_{\mu^*,\nu}(x)\right)^{-1}\right].$$

Proof. The result follows immediately from Parts (ii) and (iii) of Lemma 1. ■

Suppose that $\theta = 0$, in which case the SALP (2.8) is identical to the ALP (2.3), thus, $r_{\text{SALP}} = r_{\text{ALP}}$. Applying Part (i) of Corollary 1, we have, for the ALP, the approximation error bound

$$\|J^* - \Phi r_{\text{ALP}}\|_{1,\nu} \leq \frac{2}{1-\alpha}\|J^* - \Phi r^*\|_\infty. \tag{2.11}$$

This is precisely Theorem 2 of de Farias and Van Roy (2003); we recover their approximation guarantee for the ALP.

Now observe that, from Part (ii) of Corollary 1, if the set $\Omega(r^*)$ is of very small probability according to the distribution $\pi_{\mu^*,\nu}$, we expect that the upper bound $U_{\text{SALP}}(\theta)$ may decrease rapidly as θ is increased from 0.[2] In other words, if the Bellman error $\Phi r^*(x) - T\Phi r^*(x)$ produced by r^* is maximized at states x that are collectively of very small probability, then we expect to have a choice of $\theta > 0$ for which $U_{\text{SALP}}(\theta) < U_{\text{SALP}}(0)$. In this case, the bound (2.10) on the SALP solution will be an improvement over the bound (2.11) on the ALP solution.

[2]Already if $\pi_{\mu^*,\nu}(\Omega(r^*)) < 1/2$, then $\frac{d^+}{d\theta^+}U_{\text{SALP}}(0) < 0$.

Before we present the proof of Theorem 1 we present an auxiliary claim that we will have several opportunities to use. The proof can be found in Appendix 2.8.1.

Lemma 2. *Suppose that the vectors $J \in \mathbb{R}^{\mathcal{X}}$ and $s \in \mathbb{R}^{\mathcal{X}}$ satisfy*

$$J \leq T_{\mu^*} J + s.$$

Then,

$$J \leq J^* + \Delta^* s,$$

where

$$\Delta^* \triangleq \sum_{k=0}^{\infty} (\alpha P_{\mu^*})^k = (I - \alpha P_{\mu^*})^{-1},$$

and P_{μ^*} is the transition probability matrix corresponding to an optimal policy.

A feasible solution to the ALP is necessarily a lower bound to the optimal cost-to-go function, J^*. This is no longer the case for the SALP; the above lemma characterizes the extent to which this restriction is relaxed. In particular, if (r, s) is feasible for the SALP (2.8), then,

$$\Phi r \leq J^* + \Delta^* s.$$

We now proceed with the proof of Theorem 1:

Proof of Theorem 1. Define the weight vector $\tilde{r} \in \mathbb{R}^m$ according to

$$\Phi \tilde{r} = \Phi r^* - \ell(r^*, \theta)\mathbf{1}.$$

Note that $\tilde{r}$ is well-defined since $\mathbf{1} \in \mathsf{span}(\Phi)$. Set $\tilde{s} = s(r^*, \theta)$, the s-component of a solution to the LP (2.9) with parameters r^* and θ. We will demonstrate that $(\tilde{r}, \tilde{s})$ is feasible for (2.8). Observe that, by the definition of the LP (2.9),

$$\Phi r^* \leq T\Phi r^* + \tilde{s} + (1 - \alpha)\ell(r^*, \theta)\mathbf{1}.$$

Then,

$$\begin{aligned} T\Phi\tilde{r} &= T\Phi r^* - \alpha\ell(r^*, \theta)\mathbf{1} \\ &\geq \Phi r^* - \tilde{s} - (1 - \alpha)\ell(r^*, \theta)\mathbf{1} - \alpha\ell(r^*, \theta)\mathbf{1} \\ &= \Phi\tilde{r} - \tilde{s}. \end{aligned}$$

Now, let $(r_{\text{SALP}}, \bar{s})$ be a solution to the SALP (2.8). By Lemma 2,

$$\begin{aligned}
\|J^* - \Phi r_{\text{SALP}}\|_{1,\nu} &\leq \|J^* - \Phi r_{\text{SALP}} + \Delta^* \bar{s}\|_{1,\nu} + \|\Delta^* \bar{s}\|_{1,\nu} \\
&= \nu^\top (J^* - \Phi r_{\text{SALP}} + \Delta^* \bar{s}) + \nu^\top \Delta^* \bar{s} \\
&= \nu^\top (J^* - \Phi r_{\text{SALP}}) + \frac{2\pi_{\mu^*,\nu}^\top \bar{s}}{1-\alpha} \\
&\leq \nu^\top (J^* - \Phi r_{\text{SALP}}) + \frac{2\theta}{1-\alpha}.
\end{aligned} \tag{2.12}$$

Since $(\tilde{r}, \tilde{s})$ is feasible for (2.8), we have that

$$\begin{aligned}
\|J^* - \Phi r_{\text{SALP}}\|_{1,\nu} &\leq \nu^\top (J^* - \Phi \tilde{r}) + \frac{2\theta}{1-\alpha} \\
&= \nu^\top (J^* - \Phi r^*) + \nu^\top (\Phi r^* - \Phi \tilde{r}) + \frac{2\theta}{1-\alpha} \\
&= \nu^\top (J^* - \Phi r^*) + \ell(r^*, \theta) + \frac{2\theta}{1-\alpha} \\
&\leq \|J^* - \Phi r^*\|_\infty + \ell(r^*, \theta) + \frac{2\theta}{1-\alpha},
\end{aligned} \tag{2.13}$$

as desired. ■

While Theorem 1 reinforces the intuition (shown via Figure 2.1) that the SALP will permit closer approximations to J^* than the ALP, the bound leaves room for improvement:

1. The right hand side of our bound measures projection error, $\|J^* - \Phi r^*\|_\infty$ in the $\|\cdot\|_\infty$ norm. Since it is unlikely that the basis functions Φ will provide a uniformly good approximation over the entire state space, the right hand side of our bound could be quite large.

2. As suggested by (2.4), the choice of state relevance weights can significantly influence the solution. In particular, it allows us to choose regions of the state space where we would like a better approximation of J^*. The right hand side of our bound, however, is independent of ν.

3. Our guarantee does not suggest a concrete choice of the violation budget parameter θ.

The next section will present a substantially refined approximation bound, that will address these issues.

2.4.3. A Stronger Approximation Guarantee

With the intent of deriving stronger approximation guarantees, we begin this section by introducing a 'nicer' measure of the quality of approximation afforded by Φ. In particular, instead of measuring the approximation error $J^* - \Phi r^*$ in the $\|\cdot\|_\infty$ norm as we did for our previous bounds, we will use a weighted max norm defined according to:

$$\|J\|_{\infty,1/\psi} \triangleq \max_{x\in\mathcal{X}} \frac{|J(x)|}{\psi(x)}.$$

Here, $\psi\colon \mathcal{X} \to [1,\infty)$ is a given 'weighting' function. The weighting function ψ allows us to weight approximation error in a non-uniform fashion across the state space and in this manner potentially ignore approximation quality in regions of the state space that are less relevant. We define Ψ to be the set of all weighting functions, i.e.,

$$\Psi \triangleq \left\{\psi \in \mathbb{R}^{\mathcal{X}} \ : \ \psi \geq \mathbf{1}\right\}.$$

Given a particular $\psi \in \Psi$, we define a scalar

$$\beta(\psi) \triangleq \max_{x,a} \frac{\sum_{x'} P_a(x,x')\psi(x')}{\psi(x)}.$$

Note that $\beta(\psi)$ is an upper bound on the one-step expected value of ψ relative to the current value when evaluated along a state trajectory under an arbitrary policy, i.e.,

$$\mathsf{E}\left[\psi(x_{t+1}) \mid x_t = x,\ a_t = a\right] \leq \beta(\psi)\psi(x), \quad \forall\ x \in \mathcal{X},\ a \in \mathcal{A}.$$

When $\beta(\psi)$ is small, then $\psi(x_{t+1})$ is expected to be small relative to $\psi(x_t)$, hence $\beta(\psi)$ can be interpreted as a measure of system 'stability'.

In addition to specifying the sampling distribution π, as we did in Section 2.4.2, we will specify (implicitly) a particular choice of the violation budget θ. In particular, we will consider solving the following SALP:

$$\begin{array}{ll} \underset{r,s}{\text{maximize}} & \nu^\top \Phi r - \dfrac{2\pi^\top_{\mu^*,\nu} s}{1-\alpha} \\ \text{subject to} & \Phi r \leq T\Phi r + s, \quad s \geq \mathbf{0}. \end{array} \tag{2.14}$$

Note that (2.14) is a Lagrangian relaxation of (2.8). It is clear that (2.14) and (2.8) are equivalent in the sense that there exists a specific choice of θ so any optimal solution to (2.14) is an optimal solution to (2.8) (for a formal statement and proof of this fact see Lemma 4 in Appendix 2.8.1). We then have:

Theorem 2. *If r_{SALP} is an optimal solution to* (2.14)*, then*

$$\|J^* - \Phi r_{SALP}\|_{1,\nu} \leq \inf_{r,\psi\in\Psi} \|J^* - \Phi r\|_{\infty,\mathbf{1}/\psi} \left(\nu^\top \psi + \frac{2(\pi_{\mu^*,\nu}^\top \psi)(\alpha\beta(\psi)+1)}{1-\alpha}\right).$$

Before presenting a proof for this approximation guarantee, it is worth placing the result in context to understand its implications. For this, we recall a closely related result shown by de Farias and Van Roy (2003) for the ALP. They demonstrate that a solution r_{ALP} to the ALP (2.3) satisfies

$$\|J^* - \Phi r_{\text{ALP}}\|_{1,\nu} \leq \inf_{r,\psi\in\bar{\Psi}} \|J^* - \Phi r\|_{\infty,1/\psi} \frac{2\nu^\top \psi}{1-\alpha\beta(\psi)}, \tag{2.15}$$

where

$$\bar{\Psi} \triangleq \{\psi \in \Psi \ : \ \psi \in \text{span}(\Phi),\ \alpha\beta(\psi) < 1\}.$$

Note that (2.15) provides a bound over a collection of weighting functions ψ that are within the span of the basis Φ and satisfy a 'Lyapunov' condition $\beta(\psi) < 1/\alpha$. Suppose that there is a particular Lyapunov function ψ such that under the $\|\cdot\|_{\infty,1/\psi}$ norm, J^* is well approximated by a function in the span of Ψ, i.e., $\inf_r \|J^* - \Phi r\|_{\infty,1/\psi}$ is small. In order for the left-hand side of (2.15) also to be small and hence guarantee a small approximation error for the ALP, it must be the case that ψ is contained in the basis. Hence, being able to select a basis that spans suitable Lyapunov functions is viewed to be an important task in ensuring good approximation guarantees for the ALP. This often requires a good deal of problem specific analysis; de Farias and Van Roy (2003) identify appropriate ψ for several queueing models. To contrast with the SALP, the guarantee we present holds over *all possible* ψ (including those ψ that do not satisfy the Lyapunov condition $\beta(\psi) < 1/\alpha$, and that are not necessarily in the span of Φ). As we will see in Section 2.4.4, this difference can be significant.

To provide another comparison, let us focus attention on a particular choice of ν, namely $\nu = \pi_{\mu^*} \triangleq \pi_*$, the stationary distribution induced under an optimal policy μ^*. In this case, restricting attention to the set of weighting functions $\bar{\Psi}$ so as to make the two bounds comparable, Theorem 2 guarantees that

$$\begin{aligned} \|J^* - \Phi r_{\text{SALP}}\|_{1,\nu} &\leq \inf_{r,\psi \in \bar{\Psi}} \|J^* - \Phi r\|_{\infty,\mathbf{1}/\psi} \left(\pi_*^\top \psi + \frac{2(\pi_*^\top \psi)(\alpha\beta(\psi) + 1)}{1-\alpha} \right) \\ &\leq \inf_{r,\psi \in \bar{\Psi}} \|J^* - \Phi r\|_{\infty,\mathbf{1}/\psi} \frac{5\pi_*^\top \psi}{1-\alpha}. \end{aligned} \tag{2.16}$$

On the other hand, observing that $\beta(\psi) \geq 1$ for all $\psi \in \Psi$, the right hand side for the ALP bound (2.15) is at least

$$\inf_{r,\psi \in \bar{\Psi}} \|J^* - \Phi r\|_{\infty,\mathbf{1}/\psi} \frac{2\pi_*^\top \psi}{1-\alpha}.$$

Thus, the approximation guarantee of Theorem 2 is at most a constant factor of 5/2 worse than the guarantee (2.15) for the ALP, and can be significantly better since it allows for the consideration of weighting functions outside the span of the basis.

Proof of Theorem 2. Let $r \in \mathbb{R}^m$ and $\psi \in \Psi$ be arbitrary. Define the vector $\tilde{s} \in \mathbb{R}^{\mathcal{X}}$ component-wise by

$$\tilde{s}(x) \triangleq \Big((\Phi r)(x) - (T\Phi r)(x) \Big)^+.$$

Observe that $(r, \tilde{s})$ is feasible for (2.14). Furthermore,

$$\pi_{\mu^*,\nu}^\top \tilde{s} \leq (\pi_{\mu^*,\nu}^\top \psi) \|\tilde{s}\|_{\infty,\mathbf{1}/\psi} \leq (\pi_{\mu^*,\nu}^\top \psi) \|T\Phi r - \Phi r\|_{\infty,\mathbf{1}/\psi}.$$

Finally, note that

$$\nu^\top (J^* - \Phi r) \leq (\nu^\top \psi) \|J^* - \Phi r\|_{\infty,\mathbf{1}/\psi}.$$

Now, suppose that $(r_{\text{SALP}}, \bar{s})$ is an optimal solution to the SALP (2.14). We have from the inequalities in (2.12) in the proof of Theorem 1 and the above observations,

$$\begin{aligned} \|J^* - \Phi r_{\text{SALP}}\|_{1,\nu} &\leq \nu^\top (J^* - \Phi r_{\text{SALP}}) + \frac{2\pi_{\mu^*,\nu}^\top \bar{s}}{1-\alpha} \\ &\leq \nu^\top (J^* - \Phi r) + \frac{2\pi_{\mu^*,\nu}^\top \tilde{s}}{1-\alpha} \\ &\leq (\nu^\top \psi) \|J^* - \Phi r\|_{\infty,\mathbf{1}/\psi} + \|T\Phi r - \Phi r\|_{\infty,1/\psi} \frac{2\pi_{\mu^*,\nu}^\top \psi}{1-\alpha}. \end{aligned} \tag{2.17}$$

Since our choice of r and ψ were arbitrary, we have:

$$\|J^* - \Phi r_{\text{SALP}}\|_{1,\nu} \leq \inf_{r,\psi \in \Psi} (\nu^\top \psi) \|J^* - \Phi r\|_{\infty,\mathbf{1}/\psi} + \|T\Phi r - \Phi r\|_{\infty,1/\psi} \frac{2\pi^\top_{\mu^*,\nu}\psi}{1-\alpha}. \tag{2.18}$$

We would like to relate the Bellman error term $T\Phi r - \Phi r$ on the right hand side of (2.18) to the approximation error $J^* - \Phi r$. In order to do so, first note that, for any vectors $F_1, F_2 \in \mathbb{R}^{\mathcal{A}}$ with $a_1 \in \text{argmin}_a \ F_1(a)$ and $a_2 \in \text{argmin}_a \ F_2(a)$,

$$\min_a \ F_1(a) - \min_a \ F_2(a) = F_1(a_1) - F_2(a_2) \leq F_1(a_2) - F_2(a_2) \leq \max_a \ |F_1(a) - F_2(a)| \,.$$

By swapping the roles of F_1 and F_2, it is easy to see that

$$\left| \min_a \ F_1(a) - \min_a \ F_2(a) \right| \leq \max_a \ |F_1(a) - F_2(a)| \,.$$

Examining the definition of the Bellman operator T, this implies that, for any vectors $J, \bar{J} \in \mathbb{R}^{\mathcal{X}}$ and any $x \in \mathcal{X}$,

$$|TJ(x) - T\bar{J}(x)| \leq \alpha \max_a \sum_{x' \in \mathcal{X}} P_a(x,x') |J(x') - \bar{J}(x')|.$$

Therefore,

$$\begin{aligned}
\|T\Phi r - J^*\|_{\infty,\mathbf{1}/\psi} &\leq \alpha \max_{x,a} \ \frac{\sum_{x'} P_a(x,x') |\Phi r(x') - J^*(x')|}{\psi(x)} \\
&\leq \alpha \max_{x,a} \ \frac{\sum_{x'} P_a(x,x') \psi(x') \frac{|\Phi r(x') - J^*(x')|}{\psi(x')}}{\psi(x)} \\
&\leq \alpha \beta(\psi) \|J^* - \Phi r\|_{\infty,1/\psi}.
\end{aligned}$$

Thus,

$$\begin{aligned}
\|T\Phi r - \Phi r\|_{\infty,\mathbf{1}/\psi} &\leq \|T\Phi r - J^*\|_{\infty,\mathbf{1}/\psi} + \|J^* - \Phi r\|_{\infty,\mathbf{1}/\psi} \\
&\leq \|J^* - \Phi r\|_{\infty,1/\psi} (1 + \alpha\beta(\psi)).
\end{aligned} \tag{2.19}$$

Combining (2.18) and (2.19), we get the desired result. ■

2.4.4. Approximation Guarantee: A Queueing Example

In this section, we will examine the strength of the approximation guarantee we have provided for the SALP (Theorem 2) in a simple, concrete model studied in the context of the ALP by de Farias and Van Roy (2003). In particular, we consider an autonomous queue whose queue-length dynamics evolve over the state space $\mathcal{X} \triangleq \{0, 1, \ldots, N-1\}$ according to

$$x_{t+1} = \begin{cases} \max(x_t - 1, 0) & \text{w.p. } 1-p, \\ \min(x_t + 1, N-1) & \text{w.p. } p. \end{cases}$$

Here, we assume that $p \in (0, 1/2)$ and $N \geq 1$ is the buffer size. For convenience so as to avoid integrality issues, we will assume that $N-1$ is a multiple of 4. For $0 < x < N-1$, define the cost function $g(x) \triangleq x^2$. As in de Farias and Van Roy (2003), we may and will select $g(0)$ and $g(N-1)$ so that $J^*(x) = \rho_2 x^2 + \rho_1 x + \rho_0$ for constants $\rho_2 > 0$, ρ_1, and $\rho_0 > 0$ that depend only on p and the discount factor α. We take ν to be the steady-state distribution over states of the resulting birth-death chain, i.e., for all $x \in \mathcal{X}$,

$$\nu(x) = \frac{1-q}{1-q^N} q^x, \quad \text{where} \quad q \triangleq \frac{p}{1-p}.$$

Note that since this system is uncontrolled, we have $\pi_{\mu^*,\nu} = \nu$.

Assume we have a constant basis function and a linear basis function, i.e., $\phi_1(x) \triangleq 1$ and $\phi_2(x) \triangleq x$, for $x \in \mathcal{X}$. Note that this is different than the example studied by de Farias and Van Roy (2003), which assumed basis functions $\phi_1(x) \triangleq 1$ and $\phi_2(x) \triangleq x^2$. Nonetheless, the best possible approximation to J^* within this architecture continues to have an approximation error that is uniformly bounded in N. In particular, we have that

$$\begin{aligned} \inf_r \ \|J^* - \Phi r\|_{1,\nu} \leq \|J^* - (\rho_0\phi_1 + \rho_1\phi_2)\|_{1,\nu} &= \rho_2 \frac{1-q}{1-q^N} \sum_{x=0}^{N-1} q^x x^2 \\ &\leq \rho_2 \sum_{x=0}^{\infty} q^x x^2 = \frac{\rho_2 q}{(1-q)^3}. \end{aligned}$$

We make two principal claims for this problem setting:

(a) Theorem 2 in fact shows that the SALP is guaranteed to find an approximation in the span of the basis functions with an approximation error that is also uniformly bounded in N.

(b) We will see that the corresponding guarantee, (2.15), for the ALP (de Farias and Van Roy, 2003, Theorem 4.2) can guarantee at best an approximation error that scales linearly in N.

The broad idea used in establishing the above claims is as follows: For (a), we utilize a (quadratic) Lyapunov function identified by de Farias and Van Roy (2003) for the very problem here to produce an upper bound on the approximation guarantee we have developed for the SALP; we are careful to exclude this Lyapunov function from our basis. We then consider the ALP with the same basis, and absent the ability to utilize the quadratic Lyapunov function alluded to, show that the bound in de Farias and Van Roy (2003) must scale at least linearly in N. We now present the details.

First, consider claim (a). To see the first claim, we consider the weighting function $\psi(x) \triangleq x^2 + 2/(1-\alpha)$, for $x \in \mathcal{X}$. Notice that this weighting function is *not* in the span of Φ but still permissible for the bound in Theorem 2. For this choice of ψ, we have

$$\inf_r \ \|J^* - \Phi r\|_{\infty,1/\psi} \leq \max_{0\leq x<N} \ \frac{\rho_2 x^2}{x^2 + 2/(1-\alpha)} \leq \rho_2. \tag{2.20}$$

Moreover, de Farias and Van Roy (2003) show that for this choice of ψ,

$$\beta(\psi) \leq \frac{1+\alpha}{2\alpha}, \qquad \nu^\top \psi \leq \frac{1-p}{1-2p}\left(\frac{2}{1-\alpha} + 2\frac{p^2}{(1-2p)^2} + \frac{p}{1-2p}\right). \tag{2.21}$$

Combining (2.20)–(2.21), Theorem 2, and, in particular, (2.16), yields the (uniform in N) upper bound

$$\|J^* - \Phi r_{\mathrm{SALP}}\|_{1,\nu} \leq \frac{5\rho_2(1-p)}{(1-\alpha)(1-2p)}\left(\frac{2}{1-\alpha} + 2\frac{p^2}{(1-2p)^2} + \frac{p}{1-2p}\right).$$

The analysis of de Farias and Van Roy (2003) applies identically to the more complex settings considered in that work (namely the controlled queue and queueing network considered there) to yield uniform approximation guarantees for SALP approximations.

The following lemma, whose proof may be found in Appendix 2.8.1, demonstrates that the right-hand side of (2.15) must increase linearly with N, establishing (b). The proof of the lemma reveals that this behavior is driven primarily by the fact that the basis does not span an appropriate weighting function ψ.

Lemma 3. *For the autonomous queue with basis functions* $\phi_1(x) \triangleq 1$ *and* $\phi_2(x) \triangleq x$*, if* N *is sufficiently large, then*

$$\inf_{r,\psi\in\bar{\Psi}} \frac{2\nu^\top\psi}{1-\alpha\beta(\psi)}\|J^* - \Phi r\|_{\infty,1/\psi} \geq \frac{3\rho_2 q}{32(1-q)}(N-1).$$

2.4.5. A Performance Bound

The analytical results provided in Sections 2.4.2 and 2.4.3 provide bounds on the quality of the approximation provided by the SALP solution to J^*. In this section, we derive performance bounds with the intent of understanding the increase in expected cost incurred in using a control policy that is greedy with respect to the SALP approximation in lieu of the optimal policy. In particular, we will momentarily present a result that will allow us to interpret the objective of the SALP (2.14) as an upper bound on the performance loss of a greedy policy with respect to the SALP solution.

To begin, we briefly introduce some relevant notation. For a given policy μ, we denote

$$\Delta_\mu \triangleq \sum_{k=0}^{\infty}(\alpha P_\mu)^k = (I - \alpha P_\mu)^{-1}.$$

Thus, $\Delta^* = \Delta_{\mu^*}$. Given a vector $J \in \mathbb{R}^{\mathcal{X}}$, let μ_J denote the greedy policy with respect to J. That is, μ_J satisfies $T_{\mu_J}J = TJ$. Recall that the policy of interest to us will be $\mu_{\Phi r_{\mathrm{SALP}}}$ for a solution r_{SALP} to the SALP. Finally, for an arbitrary starting distribution over states η, we define $\nu(\eta, J)$ to be the 'discounted' expected frequency of visits to each state under the policy μ_J, i.e.,

$$\nu(\eta, J)^\top \triangleq (1-\alpha)\eta^\top \sum_{k=0}^{\infty}(\alpha P_{\mu_J})^k = (1-\alpha)\eta^\top\Delta_{\mu_J}.$$

We have the following upper bound on the increase in cost incurred by using μ_J in place of μ^*:

Theorem 3.

$$\|J_{\mu_J} - J^*\|_{1,\eta} \leq \frac{1}{1-\alpha}\left(\nu(\eta, J)^\top(J^* - J) + \frac{2}{1-\alpha}\pi^\top_{\mu^*,\nu(\eta,J)}(J - TJ)^+\right).$$

Theorem 3 applies to general approximations J and is not specific to approximations produced by the SALP. Theorem 3 indicates that if J is close to J^*, so that $(J - TJ)^+$ is also small, then the expected cost incurred in using a control policy that is greedy with respect to J will be close to optimal. The bound indicates the impact of approximation errors over differing parts of the state space on performance loss.

Suppose that $(r_{\text{SALP}}, \bar{s})$ is an optimal solution to the SALP (2.14). Then, examining the proof of Theorem 2 and, in particular, (2.17), reveals that

$$\begin{aligned} &\nu^\top (J^* - \Phi r_{\text{SALP}}) + \frac{2}{1-\alpha} \pi_{\mu^*,\nu}^\top \bar{s} \\ &\leq \inf_{r,\psi \in \Psi} \ \|J^* - \Phi r\|_{\infty,\mathbf{1}/\psi} \left(\nu^\top \psi + \frac{2(\pi_{\mu^*,\nu}^\top \psi)(\alpha\beta(\psi) + 1)}{1-\alpha} \right). \end{aligned} \tag{2.22}$$

Assume that the state relevance weights ν in the SALP (2.14) satisfy

$$\nu = \nu(\eta, \Phi r_{\text{SALP}}). \tag{2.23}$$

Then, combining Theorem 3 and (2.22) yields

$$\|J_{\mu_{\Phi r_{\text{SALP}}}} - J^*\|_{1,\eta} \leq \frac{1}{1-\alpha} \left(\inf_{r,\psi \in \Psi} \ \|J^* - \Phi r\|_{\infty,\mathbf{1}/\psi} \left(\nu^\top \psi + \frac{2(\pi_{\mu^*,\nu}^\top \psi)(\alpha\beta(\psi) + 1)}{1-\alpha} \right) \right). \tag{2.24}$$

This bound *directly* relates the performance loss of the SALP policy to the ability of the basis function architecture Φ to approximate J^*. Moreover, assuming (2.23), we can interpret the SALP as minimizing the upper bound on performance loss given by Theorem 3.

Unfortunately, it is not clear how to make an a-priori choice of the state relevance weights ν to satisfy (2.23), since the choice of ν determines the solution to the SALP r_{SALP}; this is essentially the situation one faces in performance analyses for approximate dynamic programming algorithms such as approximate value iteration and temporal difference learning (de Farias and Van Roy, 2000). Indeed, it is not clear that there exists a ν that solves the fixed point equation (2.23). On the other hand, given a choice of ν so that $\nu \approx \nu(\eta, \Phi r_{\text{SALP}})$, in the sense of a bounded Radon-Nikodym derivative between the two distributions, then the performance bound (2.24) will hold, inflated by the quantity

$$\max_{x \in \mathcal{X}} \ \frac{\nu(x)}{\nu(\eta, \Phi r_{\mathsf{SALP}})(x)}.$$

As suggested by de Farias and Van Roy (2003) in the ALP case, one possibility for finding such a choice of state relevance weights is to iteratively re-solve the SALP, and at each time using the policy from the prior iteration to generate state relevance weights for the next iteration.

Proof of Theorem 3. Define $s \triangleq (J - TJ)^+$. From Lemma 2, we know that

$$J \leq J^* + \Delta^* s.$$

Using the fact that the operator T_{μ^*} is monotonic, we can apply T_{μ^*} to both sides to obtain

$$\begin{aligned} T_{\mu^*} J &\leq T_{\mu^*}(J^* + \Delta^* s) = g_{\mu^*} + \alpha P_{\mu^*}(J^* + \Delta^* s) = J^* + \alpha P_{\mu^*} \Delta^* s \\ &= J^* + \alpha P_{\mu^*}(I - \alpha P_{\mu^*})^{-1} s = J^* + \Delta^* s - s \leq J^* + \Delta^* s, \end{aligned}$$

so that

$$TJ \leq T_{\mu^*} J \leq J^* + \Delta^* s. \tag{2.25}$$

Then,

$$\begin{aligned} \eta^\top (J_{\mu_J} - J) &= \eta^\top \sum_{k=0}^{\infty} \alpha^k P_{\mu_J}^k (g_{\mu_J} + \alpha P_{\mu_J} J - J) \\ &= \eta^\top \Delta_{\mu_J} (TJ - J) \\ &\leq \eta^\top \Delta_{\mu_J} (J^* - J + \Delta^* s) \\ &= \frac{1}{1-\alpha} \nu(\eta, J)^\top (J^* - J + \Delta^* s). \end{aligned} \tag{2.26}$$

where the second equality is from the fact that $g_{\mu_J} + \alpha P_{\mu_J} J = T_{\mu_J} J = TJ$, and the inequality follows from (2.25).

Further,

$$\begin{aligned} \eta^\top (J - J^*) &\leq \eta^\top \Delta^* s \\ &\leq \eta^\top \Delta_{\mu_J} \Delta^* s \\ &= \frac{1}{1-\alpha} \nu(\eta, J)^\top \Delta^* s. \end{aligned} \tag{2.27}$$

where the second inequality follows from the fact that $\Delta^* s \geq \mathbf{0}$ and $\Delta_{\mu_J} = I + \sum_{k=1}^{\infty} \alpha^k P_{\mu_J}^k$.

It follows from (2.26) and (2.27) that

$$\begin{aligned}\eta^\top(J_{\mu_J} - J^*) &= \eta^\top(J_{\mu_J} - J) + \eta^\top(J - J^*) \\ &\leq \frac{1}{1-\alpha}\nu(\eta, J)^\top(J^* - J + 2\Delta^* s) \\ &= \frac{1}{1-\alpha}\left(\nu(\eta, J)^\top(J^* - J) + \frac{2}{1-\alpha}\pi^\top_{\mu^*,\nu(\eta,J)} s\right),\end{aligned}$$

which is the result. ■

2.4.6. Sample Complexity

Our analysis thus far has assumed we have the ability to solve the SALP. The number of constraints and variables in the SALP grows linearly with the size of the state space $\mathcal{X}$. Hence, this program will typically be intractable for problems of interest. One solution, which we describe here, is to consider a *sampled* variation of the SALP, where states and constraints are sampled rather than exhaustively considered. In this section, we will argue that the solution to the SALP is well approximated by the solution to a tractable, sampled variation.

In particular, let $\hat{\mathcal{X}}$ be a collection of S states drawn independently from the state space $\mathcal{X}$ according to the distribution $\pi_{\mu^*,\nu}$. Consider the following optimization program:

$$\begin{aligned}\underset{r,s}{\text{maximize}} \quad & \nu^\top \Phi r - \frac{2}{(1-\alpha)S}\sum_{x\in\hat{\mathcal{X}}} s(x) \\ \text{subject to} \quad & \Phi r(x) \leq T\Phi r(x) + s(x), \qquad \forall\, x \in \hat{\mathcal{X}}, \\ & s \geq \mathbf{0}, \quad r \in \mathcal{N}.\end{aligned} \tag{2.28}$$

Here, $\mathcal{N} \subset \mathbb{R}^K$ is a bounding set that restricts the magnitude of the sampled SALP solution, we will discuss the role of $\mathcal{N}$ shortly. Notice that (2.28) is a variation of (2.14), where only the decision variables and constraints corresponding to the sampled subset of states are retained. The resulting optimization program has $K+S$ decision variables and $S|\mathcal{A}|$ linear constraints. For a moderate number of samples S, this is easily solved. Even in scenarios where the size of the action space $\mathcal{A}$ is large, it is frequently possible to rewrite (2.28) as a compact linear program (Farias and Van Roy, 2007; Moallemi et al., 2008). The natural question, however,

is whether the solution to the sampled SALP (2.28) is a good approximation to the solution provided by the SALP (2.14), for a 'tractable' number of samples S.

Here, we answer this question in the affirmative. We will provide a sample complexity bound that indicates that for a number of samples S that scales linearly with the dimension of Φ, K, and that need not depend on the size of the state space, the solution to the sampled SALP nearly satisfies, with high probability, the approximation guarantee presented for the SALP solution in Theorem 2.

In order to establish a sample complexity result, we require control over the magnitude of optimal solutions to the SALP (2.14). This control is provided by the bounding set $\mathcal{N}$. In particular, we will assume that $\mathcal{N}$ is large enough so that it contains an optimal solution to the SALP (2.14), and we define the constant

$$B \triangleq \sup_{r \in \mathcal{N}} \|(\Phi r - T\Phi r)^+\|_\infty. \tag{2.29}$$

This quantity is closely related to the diameter of the region $\mathcal{N}$. Our main sample complexity result can then be stated as follows:

Theorem 4. *Under the conditions of Theorem 2, let r_{SALP} be an optimal solution to the SALP (2.14), and let $\hat{r}_{SALP}$ be an optimal solution to the sampled SALP (2.28). Assume that $r_{SALP} \in \mathcal{N}$. Further, given $\epsilon \in (0, B]$ and $\delta \in (0, 1/2]$, suppose that the number of sampled states S satisfies*

$$S \geq \frac{64B^2}{\epsilon^2}\left(2(K+2)\log\frac{16eB}{\epsilon} + \log\frac{8}{\delta}\right).$$

Then, with probability at least $1 - \delta - 2^{-383}\delta^{128}$,

$$\|J^* - \Phi\hat{r}_{SALP}\|_{1,\nu} \leq \inf_{\substack{r \in \mathcal{N} \\ \psi \in \Psi}} \|J^* - \Phi r\|_{\infty, \mathbf{1}/\psi}\left(\nu^\top\psi + \frac{2(\pi^\top_{\mu^*,\nu}\psi)(\alpha\beta(\psi)+1)}{1-\alpha}\right) + \frac{4\epsilon}{1-\alpha}.$$

The proof of Theorem 4 is based on bounding the pseudo-dimension of a certain class of functions, and is provided in Appendix 2.8.2.

Theorem 4 establishes that the sampled SALP (2.28) provides a close approximation to the solution of the SALP (2.14), in the sense that the approximation guarantee we established

for the SALP in Theorem 2 is approximately valid for the solution to the sampled SALP, with high probability. The theorem precisely specifies the number of samples required to accomplish this task. This number depends linearly on the number of basis functions and the diameter of the feasible region, but is otherwise independent of the size of the state space for the MDP under consideration.

It is worth juxtaposing our sample complexity result with that available for the ALP (2.3). Recall that the ALP has a large number of constraints but a *small* number of variables;[3] the SALP is thus, at least superficially, a significantly more complex program. Exploiting the fact that the ALP has a small number of variables, de Farias and Van Roy (2004) establish a sample complexity bound for a sampled version of the ALP analogous to the sampled SALP (2.28). The number of samples required for this sampled ALP to produce a good approximation to the ALP can be shown to depend on the same problem parameters we have identified here, viz.: the constant B and the number of basis functions K. The sample complexity in the ALP case is identical to the sample complexity bound established here, up to constants and a linear dependence on the ratio B/ϵ. This is as opposed to the quadratic dependence on B/ϵ of the sampled SALP. Although the two sample complexity bounds are within polynomial terms of each other, one may rightfully worry abut the practical implications of an additional factor of B/ϵ in the required number of samples. In the numerical study of Section 2.6, we will attempt to address this concern computationally.

Finally, note that the sampled SALP has $K + S$ variables and $S|\mathcal{A}|$ linear constraints whereas the sampled ALP has merely K variables and $S|\mathcal{A}|$ linear constraints. Nonetheless, we will show in the Section 2.5.1 that the special structure of the Hessian associated with the sampled SALP affords us a linear computational complexity dependence on S when applying interior point methods.

An alternative sample complexity bound of a similar flavor can be developed using results from the stochastic programming literature. The key idea is that the SALP (2.14) can be

[3]Since the ALP has a small number of variables, it may be possible to solve exactly the ALP without resorting to constraint sampling by using a cutting-plane method or by applying column generation to the dual problem. In general, this would require some form of problem-specific analysis. The SALP, on the other hand, has many variables and constraints, and thus some form of sampling seems necessary.

reformulated as the following convex stochastic programming problem:

$$\underset{r\in\mathcal{N}}{\text{maximize}}\ \mathsf{E}_{\nu,\pi_{\mu^*,\nu}}\left[\Phi r(x_0) - \frac{2}{1-\alpha}(\Phi r(x) - T\Phi r(x))^+\right], \tag{2.30}$$

where $x_0, x \in \mathcal{X}$ have distributions ν and $\pi_{\mu^*,\nu}$, respectively. Interpreting the sampled SALP (2.28) as a sample average approximation of (2.30), a sample complexity bound can be developed using the methodology of Shapiro et al. (2009, Chap. 5), for example. This proof is simpler than the one presented here, but yields a cruder estimate that is not as easily compared with those available for the ALP.

2.5. Practical Implementation

The analysis in Section 2.4 applies to certain 'idealized' SALP variants, as discussed in Section 2.4.1. In particular, our main approximation guarantees focused on the linear program (2.14), and the 'sampled' version on that program (2.28). (2.14) is equivalent to the SALP (2.5) for a specialized choice of the violation budget θ and an idealized choice of the distribution π, namely $\pi_{\mu^*,\nu}$. As such (2.14) is not implementable: $\pi_{\mu^*,\nu}$ is not available and the number of constraints and variables scales linearly with the size of $\mathcal{X}$ which will typically be prohibitively large for interesting problems. The sampled variant of that program, (2.28), requires access to the same idealized sampling distribution and the guarantees pertaining to that program require knowledge of a bounding set for the optimal solution to (2.14), $\mathcal{N}$. As such, this program is not directly implementable either. Finally, the specialized choice of θ implicit in both (2.14) and (2.28) may not yield the best policies.

Thus, the bounds in Section 2.4 do not apply directly in the practical settings we will consider. Nonetheless, they do provide some insights that allow us to codify a recipe for a practical and implementable variation that we discuss below.

Consider the following algorithm:

1. Sample S states independently from the state space $\mathcal{X}$ according to a sampling distribution ρ. Denote the set of sampled states by $\hat{\mathcal{X}}$.

2. Perform a line search over increasing choices of $\theta \geq 0$. For each choice of θ,

   (a) Solve the *sampled* SALP:

$$\begin{aligned} \underset{r,s}{\text{maximize}} \quad & \frac{1}{S}\sum_{x\in\hat{\mathcal{X}}}(\Phi r)(x) \\ \text{subject to} \quad & \Phi r(x) \leq T\Phi r(x) + s(x), \quad \forall\, x \in \hat{\mathcal{X}}, \\ & \frac{1}{S}\sum_{x\in\hat{\mathcal{X}}} s(x) \leq \theta, \\ & s \geq \mathbf{0}. \end{aligned} \tag{2.31}$$

   (b) Evaluate the performance of the policy resulting from (2.31) via Monte Carlo simulation.

3. Select the best of the evaluated policies over different choices of θ.

Note that our algorithm does not require the specific choice of the violation budget θ implicit in the program (2.14), since we optimize with a line search so as to guarantee the *best* possible choice of θ. Note that, in such a line search, the sampled SALP (2.31) can be efficiently re-solved for increasing values of θ via a 'warm-start' procedure. Here, the optimal solution of the sampled SALP given previous value of θ is used as a starting point for the solver in a subsequent round of optimization. Using this method we observe that, in practice, the total solution time for a series of sampled SALP instances that vary by their values of θ grows sub-linearly with the number of instances. However, the policy for each solution instance must be evaluated via Monte Carlo simulation, which may be a time-consuming task.

Barring a line search, however, note that a reasonable choice of θ is implicitly suggested by the SALP (2.14) considered in Section 2.4.3. Thus, alternatively, the line search in Steps 2 and 3 can be replaced with the solution of single LP as follows:

2'. Solve the *sampled* SALP:

$$
\begin{array}{ll}
\underset{r,s}{\text{maximize}} & \dfrac{1}{S}\displaystyle\sum_{x\in\hat{\mathcal{X}}}(\Phi r)(x) - \dfrac{2}{(1-\alpha)S}\sum_{x\in\hat{\mathcal{X}}} s(x) \\
\text{subject to} & \Phi r(x) \leq T\Phi r(x) + s(x), \qquad \forall\, x \in \hat{\mathcal{X}}, \\
& s \geq \mathbf{0}.
\end{array}
\tag{2.32}
$$

Note that the sampled SALP (2.32) is equivalent to (2.31) for a specific, implicitly determined choice of θ (cf. Lemma 4 in Appendix 2.8.1). The programs (2.31) and (2.32) do not employ a specialized choice of π, and the use of the bounding set $\mathcal{N}$ is omitted. In addition, (2.31) does not require the specific choice of violation budget θ implicit in (2.14) and (2.28). As such, our main approximation guarantees do not apply to these programs.

Our algorithm takes as inputs the following parameters:

- Φ, a collection of K basis functions.
- S, the number of states to sample. By sampling S states, we limit the number of variables and constraints in the sampled SALP (2.31). Thus, by keeping S small, the sampled SALP becomes tractable to solve numerically. On the other hand, the quality of the approximation provided by the sampled SALP may suffer if S is chosen to be too small. The sample complexity theory developed in Section 2.4.6 suggests that S can be chosen to grow linearly with K, the size of the basis set. In particular, a reasonable choice of S need not depend on the size of the underlying state space.

  In practice, we choose $S \gg K$ to be as large as possible subject to limits on the CPU time and memory required to solve (2.31). In Section 2.5.1, we will discuss how the program (2.31) can be solved efficiently via barrier methods for large choices of S. Finally, note that a larger sample size can be used in the evaluation of the objective of the sampled SALP (2.31) than in the construction of constraints. In other words, the objective in (2.31) can be constructed from a set of states distinct from $\hat{\mathcal{X}}$, since this does not increase the size of the LP.

- ρ, a sampling distribution on the state space $\mathcal{X}$. The distribution ρ is used, via Monte Carlo sampling, in place of both the distributions ν and π in the SALP (2.5).

  Recall that the bounds in Theorems 1 and 2 provide approximation guarantees in a ν-weighted 1-norm. This suggests that ν should be chosen to emphasize regions of the state space where the quality of approximation is most important. The important regions could be, for example, regions of the state space where the process spends the most time under a baseline policy, and they could emphasized by setting ν to be the stationary distribution induced by the baseline policy. Similarly, the theory in Section 2.4 also suggests that the distribution π should be chosen to be the discounted expected frequency of visits to each state given an initial distribution ν under the *optimal* policy. Such a choice of distribution is clearly impossible to compute. In its place, however, if ν is the stationary distribution under a baseline policy, it seems reasonable to use the same distribution for π.

  In practice, we choose ρ to be the stationary distribution under some baseline policy. States are then sampled from ρ via Monte Carlo simulation of the baseline policy. This baseline policy can correspond, for example, to a heuristic control policy for the system. More sophisticated procedures such as 'bootstrapping' can also be considered (Farias and Van Roy, 2006). Here, one starts with a heuristic policy to be used for sampling states. Given the sampled states, the application of our algorithm will result in a new control policy. The new control policy can then be used for state sampling in a subsequent round of optimization, and the process can be repeated.

2.5.1. Efficient Linear Programming Solution

In this section, we will discuss the efficient solution of the sampled SALP (2.31) via linear programming. Note that the discussion here applies to the variant (2.32) as well. To begin,

note that (2.31) can be written explicitly in the form of a linear program as

$$\begin{aligned} \underset{r,s}{\text{maximize}} \quad & c^\top r \\ \text{subject to} \quad & \begin{bmatrix} A_{11} & A_{12} \\ 0 & d^\top \end{bmatrix} \begin{bmatrix} r \\ s \end{bmatrix} \leq b, \\ & s \geq \mathbf{0}. \end{aligned} \tag{2.33}$$

Here, $b \in \mathbb{R}^{S|\mathcal{A}|+1}$, $c \in \mathbb{R}^K$, and $d \in \mathbb{R}^S$ are vectors, $A_{11} \in \mathbb{R}^{S|\mathcal{A}|\times K}$ is a dense matrix, and $A_{12} \in \mathbb{R}^{S|\mathcal{A}|\times S}$ is a sparse matrix. This LP has $K + S$ decision variables and $S|\mathcal{A}| + 1$ linear constraints.

Typically, the number of sampled states S will be quite large. For example, in Section 2.6, we will discuss an example where $K = 22$ and $S = 300{,}000$. The resulting LP has approximately 300,000 variables and 6,600,000 constraints. In such cases, with many variables *and* many constraints, one might expect the LP to be difficult to solve. However, the sparsity structure of the constraint matrix in (2.33) and, especially, that of the sub-matrix A_{12}, allows efficient optimization of this LP.

In particular, imagine solving the LP (2.33) with a barrier method. The computational bottleneck of such a method is the inner Newton step to compute a central point (see, for example, Boyd and Vandenberghe, 2004). This step involves the solution of a system of linear equations of the form

$$H \begin{bmatrix} \Delta r \\ \Delta s \end{bmatrix} = -g. \tag{2.34}$$

Here, $g \in \mathbb{R}^{K+S}$ is a vector and $H \in \mathbb{R}^{(K+S)\times(K+S)}$ is the Hessian matrix of the barrier function. Without exploiting the structure of the matrix H, this linear system can be solved with $O((K + S)^3)$ floating point operations. For large values of S, this may be prohibitive.

Fortunately, the Hessian matrix H can be decomposed according to the block structure

$$H \triangleq \begin{bmatrix} H_{11} & H_{12} \\ H_{12}^\top & H_{22} \end{bmatrix},$$

where $H_{11} \in \mathbb{R}^{K \times K}$, $H_{12} \in \mathbb{R}^{K \times S}$, and $H_{22} \in \mathbb{R}^{S \times S}$. In the case of the LP (2.33), it is not difficult to see that the sparsity structure of the sub-matrix A_{12} ensures that the sub-matrix H_{22} takes the form of a diagonal matrix plus a rank-one matrix. This allows the linear system (2.34) to be solved with $O(K^2S + K^3)$ floating point operations. This is linear in S, the number of sampled states. Note that this is the same computational complexity as that of an inner Newton step for the ALP, despite the fact that the SALP has more variables than the ALP. This is because the added slack variables in the SALP are 'local' and effectively do not contribute to the dimension of the problem.

2.6. Case Study: Tetris

Our interest in Tetris as a case study for the SALP algorithm is motivated by several facts. First, theoretical results suggest that design of an optimal Tetris player is a difficult problem. Brzustowski (1992) and Burgiel (1997) have shown that the game of Tetris has to end with probability one, under all policies. They demonstrate a sequence of pieces, which leads to termination state of game for all possible actions. Demaine et al. (2003) consider the offline version of Tetris and provide computational complexity results for 'optimally' playing Tetris. They show that when the sequence of pieces is known beforehand it is NP-complete to maximize the number of cleared lines, minimize the maximum height of an occupied square, or maximize the number of pieces placed before the game ends. This suggests that the online version should be computationally difficult.

Second, Tetris represents precisely the kind of large and unstructured MDP for which it is difficult to design heuristic controllers, and hence policies designed by ADP algorithms are particularly relevant. Moreover, Tetris has been employed by a number of researchers as a testbed problem. One of the important steps in applying these techniques is the choice of basis functions. Fortunately, there is a *fixed set of basis functions*, to be described shortly, which have been used by researchers while applying temporal-difference learning (Bertsekas and Ioffe, 1996; Bertsekas and Tsitsiklis, 1996), policy gradient methods (Kakade, 2002), and

approximate linear programming (Farias and Van Roy, 2006). Hence, application of SALP to Tetris allows us to make a clear comparison to other ADP methods.

The SALP methodology described in Section 2.5 was applied as follows:

- **MDP formulation.** We used the formulation of Tetris as a Markov decision problem of Farias and Van Roy (2006). Here, the 'state' at a particular time encodes the current board configuration and the shape of the next falling piece, while the 'action' determines the placement of the falling piece. Thus, given a state and an action, the subsequent state is determined by the new configuration of the board following placement, and the shape of a new falling piece that is selected uniformly at random.

- **Reward structure.** The objective of Tetris is to maximize reward, where, given a state and an action, the per stage reward is defined to be the number of rows that are cleared following the placement of the falling piece.

  Note that since every game of Tetris must ultimately end, Tetris is most naturally formulated with the objective of maximizing the expected total number of rows cleared, i.e., a maximum *total* reward formulation. Indeed, in the existing literature, performance is reported in terms of total reward. In order to accommodate the SALP setting, we will apply our methodology to a maximum *discounted* reward formulation with a discount factor[4] of $\alpha = 0.9$. When evaluating the performance of resulting policies, however, we will report both total reward (in order to allow comparison with the literature) and discounted reward (to be consistent with the SALP objective).

- **Basis functions.** We employed the 22 basis functions originally introduced by Bertsekas and Ioffe (1996). Each basis function takes a Tetris board configuration as its argument. The functions are as follows:

  - Ten basis functions, $\phi_0, \ldots, \phi_9$, mapping the state to the height h_k of each of the ten columns.

[4]The introduction of an artificial discount factor into an average cost problem is akin to analyzing a perturbed problem with a limited time horizon, a common feature in many ADP schemes (e.g., de Farias and Van Roy, 2006).

  - Nine basis functions, $\phi_{10}, \ldots, \phi_{18}$, each mapping the state to the absolute difference between heights of successive columns: $|h_{k+1} - h_k|$, $k = 1, \ldots, 9$.
  - One basis function, ϕ_{19}, that maps state to the maximum column height: $\max_k h_k$
  - One basis function, ϕ_{20}, that maps state to the number of 'holes' in the board.
  - One basis function, ϕ_{21}, that is equal to 1 in every state.

- **State sampling.** Given a sample size S, a collection $\hat{\mathcal{X}} \subset \mathcal{X}$ of S states was sampled. These samples were generated in an i.i.d. fashion from the stationary distribution of a (rather poor) baseline policy.[5] For each choice of sample size S, ten different collections of S samples were generated.

- **Optimization.** Given the collection $\hat{\mathcal{X}}$ of sampled states, an increasing sequence of choices of the violation budget $\theta \geq 0$ is considered. For each choice of θ, the optimization program (2.31) was solved. Separately, the optimization program (2.32), which implicitly defines a reasonable choice of θ, was also employed. The CPLEX 11.0.0 optimization package was used to solve the resulting linear programs.

- **Policy evaluation.** Given a vector of weights $\hat{r}$, the performance of the corresponding policy was evaluated using Monte Carlo simulation. We estimate the expected reward of the policy $\mu_{\hat{r}}$ over a series of 3,000 games. The sequence of pieces in each of the 3,000 games was fixed across the evaluation of different policies in order to reduce the Monte Carlo error in estimated performance differences.

  Performance is measured in two ways. The total reward is computed as the expected total number of lines eliminated in a single game. The discounted reward is computed as the expected discounted number of lines eliminated. The discounted reward depends on the initial state, and we sample the initial state from the sampling distribution, i.e., the stationary distribution of the baseline policy. This is consistent with the objective of the sampled SALPs (2.31) or (2.32), that seek to optimize the expectation of the value function under this same distribution.

[5] Our baseline policy had an expected total reward of 113 lines cleared.

For each pair (S, θ), the resulting *average* performance (averaged over each of the 10 policies arising from the different sets of sampled states) in terms of expected total lines cleared is shown in Figure 2.2. Note that the $\theta = 0$ curve in Figure 2.2 corresponds to the original ALP algorithm. Figure 2.2 provides experimental evidence for the intuition expressed in Section 2.3 and the analytic result of Theorem 1: Relaxing the constraints of the ALP even slightly, by allowing for a small slack budget, allows for better policy performance. As the slack budget θ is increased from 0, performance dramatically improves. At the peak value of $\theta = 0.0205$, the SALP generates policies with performance that is an order of magnitude better than ALP. Beyond this value, the performance of the SALP begins to degrade, as shown by the $\theta = 0.041$ curve. Hence, we did not explore larger values of θ.

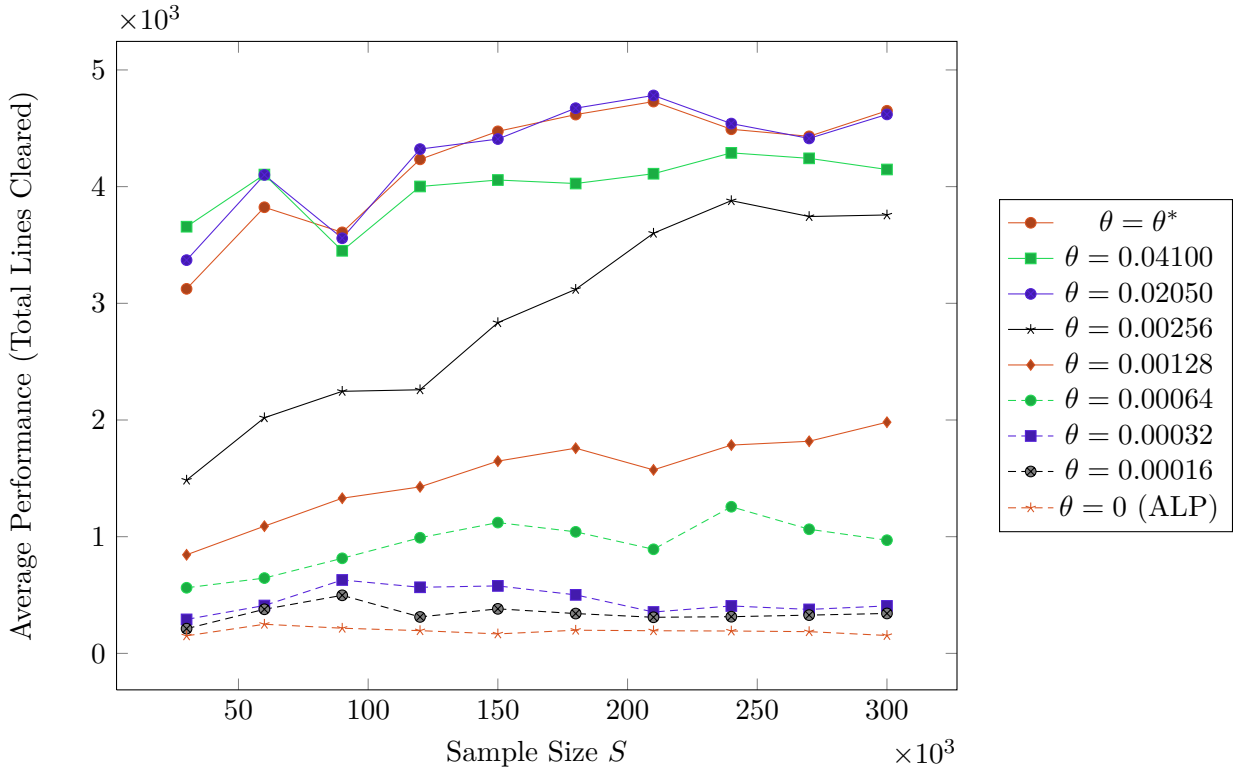


Figure 2.2 Expected total reward of the average SALP policy for different values of the number of sampled states S and the violation budget θ. Values for θ were chosen in an increasing fashion starting from 0, until the resulting average performance began to degrade. The $\theta = \theta^*$ curve corresponds to the implicit choice of θ made by solving (2.32).

As suggested in Section 2.5, instead of doing a line search over θ, one can consider solving the sampled SALP (2.32), which implicitly makes a choice of θ. We denote this implicit choice by $\theta = \theta^*$ in Figure 2.2. The results of solving (2.32) are given by the $\theta = \theta^*$ curve in Figure 2.2. We observe that, in our experiments, the results obtained by solving (2.32) are quite similar to the best results obtained by doing a line search over choices of θ. In fact, across these experiments, θ^* is observed to be roughly constant as a function of the sample size S, and approximately equal to 0.02. This is very close to the best values of θ found via line search.

In order to allow a comparison of our results with those reported elsewhere in the literature, Table 2.1 summarizes the expected total reward of the *best* policies obtained by various ADP algorithms. Note that all of these algorithms employ the same basis function architecture. The ALP and SALP results are from our experiments, while the other results are from the literature. Here, the reported ALP and SALP performance corresponds to that of the best performing policy among all of policies computed for Figure 2.2. Note that the best performance result of SALP is a factor of 2 better than the nearest competitors.

Algorithm	**Best Performance** (Total Lines Cleared)	**CPU Time**
ALP	698.4	hours
TD-Learning (Bertsekas and Ioffe, 1996)	3,183	minutes
ALP with bootstrapping (Farias and Van Roy, 2006)	4,274	hours
TD-Learning (Bertsekas and Tsitsiklis, 1996)	4,471	minutes
Policy gradient (Kakade, 2002)	5,500	days
SALP	11,574	hours

Table 2.1: Comparison of the performance of the best policy found with various ADP methods.

Note that significantly better policies are possible with this basis function architecture than *any* of the ADP algorithms in Table 2.1 discover. Using a heuristic global optimization method, Szita and Lőrincz (2006) report finding policies with a remarkable average performance of 350,000. Their method is very computationally intensive, however, requiring

one month of CPU time. In addition, the approach employs a number of rather arbitrary Tetris specific 'modifications' that are ultimately seen to be critical to performance — in the absence of these modifications, the method is unable to find a policy for Tetris that scores above a few hundred points. More generally, global optimization methods typically require significant trial and error and other problem specific experimentation in order to work well.

In Figure 2.3, we see the average performance for each (S, θ) pair measured as the expected *discounted* number of lines cleared, beginning from an initial configuration drawn according to the stationary distribution of the baseline policy. At a high level, these results are consistent with those reported in Figure 2.2. In particular, we see that according to this alternative metric, relaxing the ALP constraints also yields an improvement in performance. Note that the improvement under the discounted reward metric is less dramatic than under the total reward metric. This is to be expected: under the discounted metric we implicitly measure policy performance over a limited time horizon.

In Table 2.2, we see the effect of the choice of the discount factor α on the performance of the ALP and SALP methods. Here, we show both the expected discounted reward and the expected total reward, for different values of the discount factor α and the violation budget θ. Here, the policies were constructed using $S = 200{,}000$ sampled states. We find that:

1. For all discount factors, the SALP dominates the ALP. The performance improvement of the SALP relative to the ALP increases dramatically at high discount factors.

2. The absolute performance of both schemes degrades at high discount factors. This is consistent with our approximation guarantees, which degrade as $\alpha \to 1$, as well as prior theory that has been developed for the average cost ALP (de Farias and Van Roy, 2006). However, observe that the ALP degradation is drastic (scores in single digits) while the SALP degradation relatively mild (scores remain in the thousands).

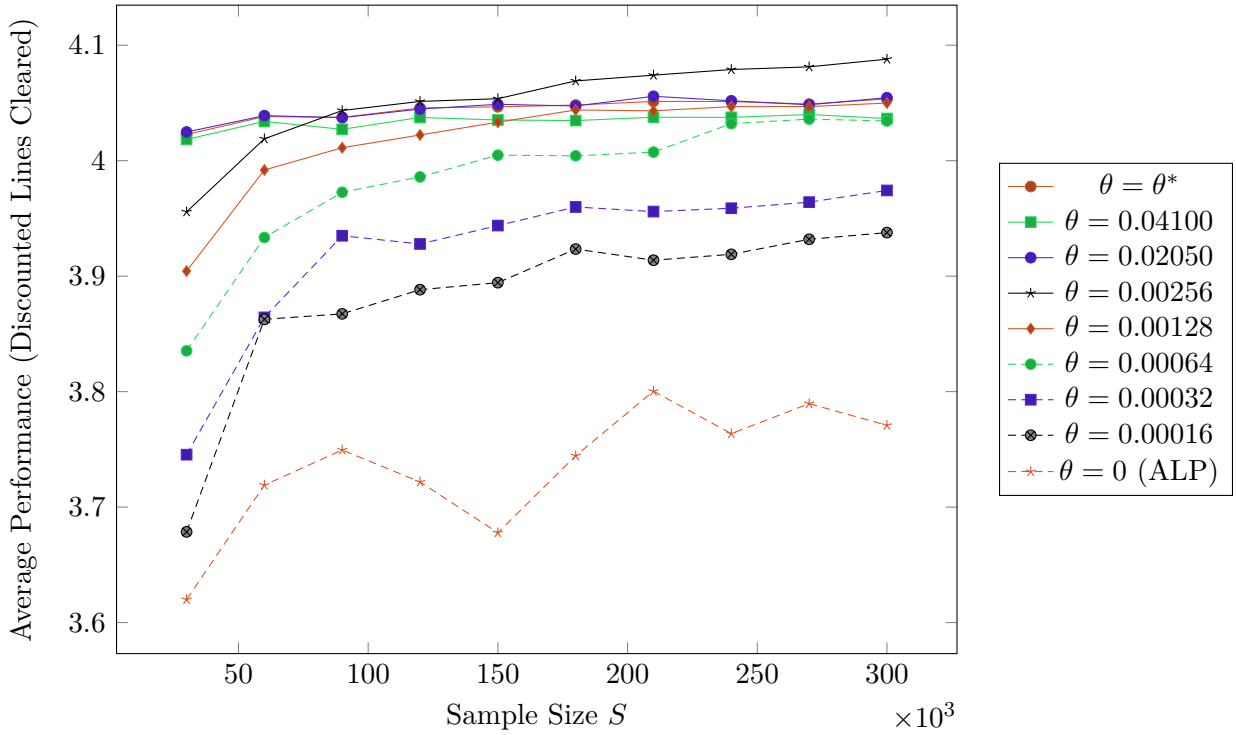


Figure 2.3 Expected discounted reward for the average SALP policy for different values of the number of sampled states S and the violation budget θ. The reward is measured starting from a random board configuration sampled from the stationary distribution of the baseline policy. The $\theta = \theta^*$ curve corresponds to the implicit choice of θ made by solving (2.32).

2.7. Case Study: A Queueing Network

In this section, we study the application of SALP and ALP to control of queueing networks. In particular, we consider a *criss-cross queueing network*, which has been considered extensively in the literature (e.g., Harrison and Wein, 1989; Kushner and Martins, 1996; Martins et al., 1996). Optimal control of a criss-cross network is a standard example of a challenging network control problem, and has eluded attempts to find an analytical solution (Kumar and Muthuraman, 2004).

The cross-cross queueing network consists of two servers and three queues connected as shown in Figure 2.4. There are two classes of jobs in this system. The first class of jobs takes a vertical path through the system, arriving to queue 1 according to a Poisson process of rate λ_1. The second class of jobs takes a horizontal path through the system,

Violation Budget	**Expected Total Reward** Discount Factor α				**Expected Discounted Reward** Discount Factor α			
θ	0.9	0.95	0.99	0.999	0.9	0.95	0.99	0.999
0 (ALP)	169.1	367.9	240.0	1.9	2.150	5.454	30.410	1.870
0.00002	201.7	844.6	295.9	44.1	2.111	5.767	34.063	39.317
0.00008	308.5	1091.7	355.7	93.9	2.249	5.943	34.603	79.086
0.00032	380.2	1460.2	792.1	137.4	2.261	6.011	35.969	108.554
0.00128	1587.4	2750.4	752.1	189.0	2.351	6.055	36.032	138.329
0.00512	5023.9	4069.9	612.5	355.1	2.356	6.116	35.954	202.640
0.01024	5149.7	4607.7	1198.6	1342.5	2.281	6.115	36.472	318.532
0.02048	4664.6	3662.3	1844.6	2227.4	2.216	6.081	36.552	340.718
0.04096	4089.9	2959.7	1523.3	694.5	2.200	6.044	36.324	262.462
0.08192	3085.9	2236.8	901.7	360.5	2.192	5.975	35.772	200.861
0.32768	1601.6	855.4	357.5	145.4	2.247	5.613	34.025	112.427
θ^*	4739.2	4473.7	663.5	138.7	2.213	6.114	35.827	109.341
Average θ^*	0.0204	0.0062	0.0008	0.0003	0.0204	0.0062	0.0008	0.0003

Table 2.2: Expected discounted reward and expected total reward for different values of the discount factor α and the violation budget θ. Here, the policies were constructed using $S = 200{,}000$ sampled states. The last row reports average value of the implicit violation budget θ^* for different values of the discount factor α.

arriving at queue 2 according to a Poisson process of rate λ_2. Server 1 can work on jobs in either queue 1 or queue 2, with service times distributed exponentially with rate $\mu_1 \triangleq 2$ and $\mu_2 \triangleq 2$ respectively. Vertical jobs exit the system after service, while horizontal jobs proceed to queue 3. There, they await service by server 2. The service times at server 2 are exponentially distributed with rate $\mu_3 \triangleq 1$. Given a common arrival rate $\lambda_1 \triangleq \lambda_2 \triangleq \lambda$, by analysis of the static planning LP (Harrison, 1988) associated with the network, it is straight forward to derive that the load of the network takes the form

$$\rho = \frac{\lambda_2}{\mu_2} + \max\left(\frac{\lambda_1}{\mu_1}, \frac{\lambda_2}{\mu_3}\right) = \tfrac{3}{2}\lambda.$$

The SALP and ALP methodologies were applied to this queueing network as follows:

- **MDP formulation.** The evolution of this queuing network is described by a continuous time Markov chain with the state $q \in \mathbb{Z}_+^3$ corresponding to the queue lengths. Via a

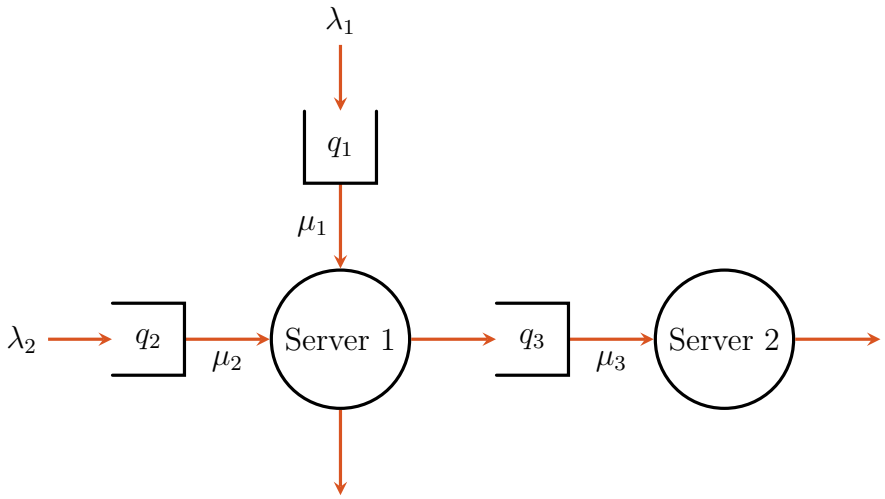


Figure 2.4 A criss-cross queueing network consisting of three queues and two servers. One class of jobs arrives to the system at queue 1, and departs after service by server 1. The second class of jobs arrives to the system at queue 2, and departs after sequential service from server 1 followed by server 2.

standard uniformization construction (see, e.g., Moallemi et al., 2008, for an explicit construction), we consider an equivalent discrete time formulation, where $q_t \in \mathbb{Z}_+^3$ is the vector of queue lengths after the tth event, for $t \in \{0, 1, \ldots\}$. At each time, the choice of action corresponds to an assignment of each server to an associated non-empty queue, and idling is allowed.

- **Reward structure.** We seek find a control policy that optimizes the discounted infinite horizon cost objective

$$\text{minimize } \mathsf{E}\left[\sum_{t=0}^{\infty} \alpha^t c^\top q_t\right].$$

  Here, the vector $c \in \mathbb{R}_+^3$ denotes the holding costs associated with each queue, and α is a discount factor.

- **Basis functions.** Four basis functions were used: the constant function, and, for each queue, a linear function in the queue length.

- **State sampling.** States were sampled from the stationary distribution of a policy which acts greedily according to the value function surrogate[6] $V(q) \triangleq \|q\|_2^2$. We use a collection $\hat{\mathcal{X}}$ of $S = 40{,}000$ sampled states as input to SALP. The results are averaged over 10 different collections of S samples.

- **Optimization.** The sampled states $\hat{\mathcal{X}}$ were used as input to optimization program (2.31). For increasing choices of violation budget $\theta \geq 0$, the linear program was solved to obtain policies. A policy corresponding to the implicit choice $\theta = \theta^*$ was obtained by separately solving linear program (2.32). Our implementation used CPLEX 11.0.0 to solve the resulting linear programs.

- **Policy evaluation.** Given a value function approximation, the expected discounted performance of the corresponding policy was evaluated by simulating 100 sample paths starting from an empty state ($q = 0$). Each sample path was evaluated over 50,000,000 time steps to compute the discounted cost.

We first consider the case where the holding costs are given by the vector $c \triangleq (1, 1, 3)$. This corresponds to Case IIB as considered by Martins et al. (1996), and the associated stochastic control problem is known to be challenging (Kumar and Muthuraman, 2004). These particular parameter settings are difficult because of the fact the holding costs for queue 3 are so much higher than those for queue 2. Hence, it may be optimal for server 1 to idle even if there are jobs in queue 2 so as to keep jobs in the cheaper buffer. On the other hand, too much idling at server 1 could lead to an empty queue 3, which would force idling at server 2. Hence, the policy decision for server 1 also depends on the downstream queue length.

In Table 2.3(a), we see the resulting performance of policies by solving SALP for various values of θ and for the ALP (i.e., $\theta = 0$). The results are shown for various levels of the load ρ. Overall, we observe a significant reduction in cost by policies generated via SALP in

[6]This corresponds approximately to a 'maximum pressure' policy (Dai and Lin, 2005; Tassiulas and Ephremides, 1992, 1993).

comparison to ALP. Using a line search to find the optimal choice of θ yields an SALP policy that results in a cost savings of 40% as compared to ALP. The policy corresponding to the implicit choice of $\theta = \theta^*$, obtained by solving LP (2.32), produces a comparable savings of 30%.

In Table 2.3(b), we consider the case when the holding costs are given by the vector $c \triangleq (1, 1, 1)$. This is a considerably easier control problem, since there is no need for server 1 to idle. In this case, the SALP is still a significant improvement over the ALP, however the magnitude of the improvement is smaller.

2.8. Proofs

In this section, we provide proofs for the results in the chapter.

2.8.1. Proofs for Sections 2.4.2–2.4.4

Lemma 1. *For any $r \in \mathbb{R}^K$ and $\theta \geq 0$:*

(i) $\ell(r, \theta)$ is a finite-valued, decreasing, piecewise linear, convex function of θ.

(ii)

$$\ell(r, \theta) \leq \frac{1+\alpha}{1-\alpha} \|J^* - \Phi r\|_\infty.$$

(iii) The right partial derivative of $\ell(r, \theta)$ with respect to θ satisfies

$$\frac{\partial^+}{\partial \theta^+} \ell(r, 0) = -\left((1-\alpha) \sum_{x \in \Omega(r)} \pi_{\mu^*, \nu}(x) \right)^{-1},$$

where

$$\Omega(r) \triangleq \operatorname*{argmax}_{\{x \in \mathcal{X} \,:\, \pi_{\mu^*,\nu}(x) > 0\}} \Phi r(x) - T\Phi r(x).$$

(a) Expected discounted cost for varying values of the load ρ, with holding costs $c = (1, 1, 3)$. The last row reports the average value of the implicit violation budget, for different values of the load ρ.

Violation Budget	**Expected Discounted Cost**					
	$\rho = 0.98$		$\rho = 0.95$		$\rho = 0.90$	
θ	Cost	Normalized	Cost	Normalized	Cost	Normalized
0 (ALP)	560.0	1.00	542.8	1.00	514.3	1.00
0.0001	560.0	1.00	542.8	1.00	514.4	1.00
0.0010	559.7	1.00	542.5	1.00	514.2	1.00
0.0100	588.7	1.05	570.7	1.05	541.2	1.05
0.1000	584.3	1.04	566.9	1.04	538.1	1.05
1.0000	502.8	0.90	486.1	0.90	459.0	0.89
θ^*	412.5	0.74	398.2	0.73	373.0	0.73
25.000	332.2	0.59	318.7	0.59	295.8	0.58
50.000	334.0	0.60	320.5	0.59	296.8	0.58
75.000	337.5	0.60	323.6	0.60	301.4	0.59
100.00	337.5	0.60	323.6	0.60	301.4	0.59
Average θ^*	17.79		17.73		17.67	

(b) Expected discounted cost for load $\rho = 0.98$, with holding costs $c = (1, 1, 1)$.

Violation Budget	**Expected Discounted Cost**	
	$\rho = 0.98$	
θ	Cost	Normalized
0 (ALP)	334.5	1.00
0.0001	334.5	1.00
0.0010	381.1	1.14
0.0100	284.4	0.85
0.1000	237.9	0.71
1.0000	246.7	0.74
θ^*	245.9	0.74
25.000	250.4	0.75
50.000	254.4	0.76
75.000	254.4	0.76
100.00	254.4	0.76
Average θ^*	11.81	

Table 2.3: Expected discounted cost for different values of the violation budget θ, load ρ, and holding costs c. The expected discounted cost is also reported after normalization by the performance of the corresponding ALP performance. Here, the expected discounted cost is measured starting from an empty state.

Proof. (i) Given any r, clearly $\gamma \triangleq \|\Phi r - T\Phi r\|_\infty$, $s \triangleq \mathbf{0}$ is a feasible point for (2.9), so $\ell(r,\theta)$ is feasible. To see that the LP is bounded, suppose (s,γ) is feasible. Then, for any $x \in \mathcal{X}$ with $\pi_{\mu^*,\nu}(x) > 0$,

$$\gamma \geq \Phi r(x) - T\Phi r(x) - s(x) \geq \Phi r(x) - T\Phi r(x) - \theta/\pi_{\mu^*,\nu}(x) > -\infty.$$

Letting (γ_1, s_1) and (γ_2, s_2) represent optimal solutions for the LP (2.9) with parameters (r,θ_1) and (r,θ_2) respectively, it is easy to see that $((\gamma_1+\gamma_2)/2, (s_1+s_2)/2)$ is feasible for the LP with parameters $(r,(\theta_1+\theta_2)/2)$. It follows that $\ell(r,(\theta_1+\theta_2)/2) \leq (\ell(r,\theta_1)+\ell(r,\theta_2))/2$. The remaining properties are simple to check.

(ii) Let $\epsilon \triangleq \|J^* - \Phi r\|_\infty$. Then, since T is an α-contraction under the $\|\cdot\|_\infty$ norm,

$$\|T\Phi r - \Phi r\|_\infty \leq \|J^* - T\Phi r\|_\infty + \|J^* - \Phi r\|_\infty \leq \alpha\|J^* - \Phi r\|_\infty + \epsilon = (1+\alpha)\epsilon.$$

Since $\gamma \triangleq \|T\Phi r - \Phi r\|_\infty$, $s \triangleq \mathbf{0}$ is feasible for (2.9), the result follows.

(iii) Fix $r \in \mathbb{R}^K$, and define

$$\Delta \triangleq \max_{\{x \in \mathcal{X} \,:\, \pi_{\mu^*,\nu}(x) > 0\}} \Big(\Phi r(x) - T\Phi r(x)\Big) - \max_{\{x \in \mathcal{X}\setminus\Omega(r) \,:\, \pi_{\mu^*,\nu}(x) > 0\}} \Big(\Phi r(x) - T\Phi r(x)\Big) > 0.$$

Consider the program for $\ell(r,\delta)$. It is easy to verify that for $\delta \geq 0$ and sufficiently small, viz. $\delta \leq \Delta \sum_{x\in\Omega(r)} \pi_{\mu^*,\nu}(x)$, $(\bar{s}_\delta, \bar{\gamma}_\delta)$ is an optimal solution to the program, where

$$\bar{s}_\delta(x) \triangleq \begin{cases} \frac{\delta}{\sum_{x\in\Omega(r)} \pi_{\mu^*,\nu}(x)} & \text{if } x \in \Omega(r), \\ 0 & \text{otherwise,} \end{cases}$$

and

$$\bar{\gamma}_\delta \triangleq \gamma_0 - \frac{\delta}{\sum_{x\in\Omega(r)} \pi_{\mu^*,\nu}(x)},$$

so that

$$\ell(r,\delta) = \ell(r,0) - \frac{\delta}{(1-\alpha)\sum_{x\in\Omega(r)} \pi_{\mu^*,\nu}(x)}.$$

Thus,

$$\frac{\ell(r,\delta) - \ell(r,0)}{\delta} = -\left((1-\alpha)\sum_{x\in\Omega(r)} \pi_{\mu^*,\nu}(x)\right)^{-1}.$$

Taking a limit as $\delta \searrow 0$ yields the result. ■

Lemma 2. *Suppose that the vectors $J \in \mathbb{R}^{\mathcal{X}}$ and $s \in \mathbb{R}^{\mathcal{X}}$ satisfy*

$$J \leq T_{\mu^*} J + s.$$

Then,

$$J \leq J^* + \Delta^* s,$$

where

$$\Delta^* \triangleq \sum_{k=0}^{\infty} (\alpha P_{\mu^*})^k = (I - \alpha P_{\mu^*})^{-1},$$

and P_{μ^*} is the transition probability matrix corresponding to an optimal policy.

Proof. Note that the T_{μ^*}, the Bellman operator corresponding to the optimal policy μ^*, is monotonic and is a contraction. Then, repeatedly applying T_{μ^*} to the inequality $J \leq T_{\mu^*} J + s$ and using the fact that $T_{\mu^*}^k J \to J^*$, we obtain

$$J \leq J^* + \sum_{k=0}^{\infty} (\alpha P_{\mu^*})^k s = J^* + \Delta^* s.$$

■

Lemma 3. *For the autonomous queue with basis functions $\phi_1(x) \triangleq 1$ and $\phi_2(x) \triangleq x$, if N is sufficiently large, then*

$$\inf_{r, \psi \in \bar{\Psi}} \ \frac{2\nu^\top \psi}{1 - \alpha\beta(\psi)} \|J^* - \Phi r\|_{\infty, 1/\psi} \geq \frac{3\rho_2 q}{32(1-q)}(N-1).$$

Proof. We have:

$$\inf_{r, \psi \in \bar{\Psi}} \ \frac{2\nu^\top \psi}{1 - \alpha\beta(\psi)} \|J^* - \Phi r\|_{\infty, 1/\psi} \geq \inf_{\psi \in \bar{\Psi}} \ \frac{2\nu^\top \psi}{\|\psi\|_\infty} \ \inf_r \ \|J^* - \Phi r\|_\infty.$$

We will produce lower bounds on the two infima on the right-hand side above. Observe that

$$\begin{aligned}
\inf_r \ \|J^* - \Phi r\|_\infty &= \inf_r \ \max_x \ |\rho_2 x^2 + \rho_1 x + \rho_0 - r_1 x - r_0| \\
&\geq \inf_r \ \max \left(\max_x \ |\rho_2 x^2 + (\rho_1 - r_1) x| - |\rho_0 - r_0|, |\rho_0 - r_0| \right) \\
&= \inf_{r_0} \ \max \left(\inf_{r_1} \ \max_x \ |\rho_2 x^2 + (\rho_1 - r_1) x| - |\rho_0 - r_0|, |\rho_0 - r_0| \right),
\end{aligned}$$

which follows from the triangle inequality and the fact that

$$\max_x \ |\rho_2 x^2 + \rho_1 x + \rho_0 - r_1 x - r_0| \geq |\rho_0 - r_0|.$$

Routine algebra verifies that

$$\inf_{r_1} \max_x \ |\rho_2 x^2 + (\rho_1 - r_1)x| \geq \tfrac{3}{16}\rho_2 (N-1)^2. \tag{2.35}$$

It thus follows that

$$\inf_r \ \|J^* - \Phi r\|_\infty \geq \inf_{r_0} \ \max\left(\tfrac{3}{16}\rho_2(N-1)^2 - |\rho_0 - r_0|, |\rho_0 - r_0|\right) \geq \tfrac{3}{32}\rho_2(N-1)^2.$$

We next note that any $\psi \in \tilde{\Psi}$ must satisfy $\psi \in \text{span}(\Phi)$ and $\psi \geq \mathbf{1}$. Thus, $\psi \in \tilde{\Psi}$ must take the form $\psi(x) = \alpha_1 x + \alpha_0$ with $\alpha_0 \geq 1$ and $\alpha_1 \geq (1-\alpha_0)/(N-1)$. Thus, $\|\psi\|_\infty = \max(\alpha_1(N-1) + \alpha_0, \alpha_0)$. Define $\kappa(N)$ to be the expected queue length under the distribution ν, i.e.,

$$\kappa(N) \triangleq \sum_{x=0}^{N-1} \nu(x)x = \frac{1-q}{1-q^N} \sum_{x=0}^{N-1} xq^x = \frac{q}{1-q}\left[\frac{1 - Nq^{N-1}(1-q) - q^N}{1-q^N}\right],$$

so that $\nu^\top \psi = \alpha_1 \kappa(N) + \alpha_0$, Thus,

$$\inf_{\psi \in \tilde{\Psi}} \ \frac{2\nu^\top \psi}{\|\psi\|_\infty} \ \inf_r \ \|J^* - \Phi r\|_\infty \geq \tfrac{3}{16}\rho_2 \inf_{\substack{\alpha_0 \geq 1 \\ \alpha_1 \geq \frac{1-\alpha_0}{N-1}}} \ \frac{\alpha_1\kappa(N) + \alpha_0}{\max(\alpha_1(N-1) + \alpha_0, \alpha_0)}(N-1)^2$$

When $(1-\alpha_0)/(N-1) \leq \alpha_1 \leq 0$, we have

$$\begin{aligned}
\frac{\alpha_1\kappa(N) + \alpha_0}{\max(\alpha_1(N-1) + \alpha_0, \alpha_0)}(N-1)^2 &= \frac{\alpha_1\kappa(N) + \alpha_0}{\alpha_0}(N-1)^2 \\
&\geq \frac{(1-\alpha_0)\kappa(N)/(N-1) + \alpha_0}{\alpha_0}(N-1)^2 \\
&\geq \left(1 - \frac{\kappa(N)}{N-1}\right)(N-1)^2.
\end{aligned}$$

When $\alpha_1 > 0$, we have

$$\frac{\alpha_1\kappa(N) + \alpha_0}{\max(\alpha_1(N-1) + \alpha_0, \alpha_0)}(N-1)^2 = \frac{\alpha_1\kappa(N) + \alpha_0}{\alpha_1(N-1) + \alpha_0}(N-1)^2 \geq (N-1)\kappa(N),$$

where the inequality follows from the fact that $\kappa(N) \leq N-1$ and $\alpha_0 > 0$. It then follows that

$$\inf_{\psi \in \tilde{\Psi}} \frac{2\nu^\top \psi}{\|\psi\|_\infty} \inf_r \|J^* - \Phi r\|_\infty \geq \tfrac{3}{16}\rho_2 \min\left(\kappa(N)(N-1), \left(1 - \frac{\kappa(N)}{N-1}\right)(N-1)^2\right).$$

Now, observe that $\kappa(N)$ is increasing in N. Also, by assumption, $p < 1/2$, so $q < 1$ and thus $\kappa(N) \to q/(1-q)$ as $N \to \infty$. Then, for N sufficiently large, $\frac{1}{2}q/(1-q) \leq \kappa(N) \leq q/(1-q)$. Therefore, for N sufficiently large,

$$\inf_{\psi \in \tilde{\Psi}} \frac{2\nu^\top \psi}{\|\psi\|_\infty} \inf_r \|J^* - \Phi r\|_\infty \geq \frac{3\rho_2 q}{32(1-q)}(N-1),$$

as desired. ■

Lemma 4. *For every $\lambda \geq 0$, there exists a $\hat{\theta} \geq 0$ such that an optimal solution (r^*, s^*) to*

$$\begin{aligned} &\underset{r,s}{\text{maximize}} && \nu^\top \Phi r - \lambda \pi_{\mu^*,\nu}^\top s \\ &\textit{subject to} && \Phi r \leq T\Phi r + s, \quad s \geq \mathbf{0}. \end{aligned} \tag{2.36}$$

is also an optimal solution the SALP (2.8) with $\theta = \hat{\theta}$.

Proof. Let $\hat{\theta} \triangleq \pi_{\mu^*,\nu}^\top s^*$. It is then clear that (r^*, s^*) is a feasible solution to (2.8) with $\theta = \hat{\theta}$. We claim that it is also an optimal solution. To see this, assume to the contrary that it is not an optimal solution, and let $(\tilde{r}, \tilde{s})$ be an optimal solution to (2.8). It must then be that $\pi_{\mu^*,\nu}^\top \tilde{s} \leq \hat{\theta} = \pi_{\mu^*,\nu}^\top s^*$ and moreover, $\nu^\top \Phi \tilde{r} > \nu^\top \Phi r^*$ so that

$$\nu^\top \Phi r^* - \lambda \pi_{\mu^*,\nu}^\top s^* < \nu^\top \Phi \tilde{r} - \lambda \pi_{\mu^*,\nu}^\top \tilde{s}.$$

This, in turn, contradicts the optimality of (r^*, s^*) for (2.36) and yields the result. ■

2.8.2. Proof of Theorem 4

Our proof of Theorem 4 is based on uniformly bounding the rate of convergence of sample averages of a certain class of functions. We begin with some definitions: consider a family

$\mathcal{F}$ of functions from a set $\mathcal{S}$ to $\{0,1\}$. Define the *Vapnik-Chervonenkis (VC) dimension* $\dim_{\mathrm{VC}}(\mathcal{F})$ to be the cardinality d of the largest set $\{x_1, x_2, \ldots, x_d\} \subset \mathcal{S}$ satisfying:

$$\forall\ e \in \{0,1\}^d,\ \exists f \in \mathcal{F} \text{ such that } \forall\ i,\ f(x_i) = 1 \text{ iff } e_i = 1.$$

Now, let $\mathcal{F}$ be some set of *real*-valued functions mapping $\mathcal{S}$ to $[0, B]$. The *pseudo-dimension* $\dim_P(\mathcal{F})$ is the following generalization of VC dimension: for each function $f \in \mathcal{F}$ and scalar $c \in \mathbb{R}$, define a function $g\colon \mathcal{S} \times \mathbb{R} \to \{0,1\}$ according to:

$$g(x, c) \triangleq \mathbb{I}_{\{f(x)-c\geq 0\}}.$$

Let $\mathcal{G}$ denote the set of all such functions. Then, we define $\dim_P(\mathcal{F}) \triangleq \dim_{\mathrm{VC}}(\mathcal{G})$.

In order to prove Theorem 4, define the $\mathcal{F}$ to be the set of functions $f\colon \mathbb{R}^K \times \mathbb{R} \to [0, B]$, where, for all $x \in \mathbb{R}^K$ and $y \in \mathbb{R}$,

$$f(y, z) \triangleq \zeta\left(r^\top y + z\right).$$

Here, $\zeta(t) \triangleq \max\left(\min(t, B), 0\right)$, and $r \in \mathbb{R}^K$ is a vector that parameterizes f. We will show that $\dim_P(\mathcal{F}) \leq K + 2$. We will use the following standard result from convex geometry:

Lemma 5 (Radon's Lemma)**.** *A set $A \subset \mathbb{R}^m$ of $m + 2$ points can be partitioned into two disjoint sets A_1 and A_2, such that the convex hulls of A_1 and A_2 intersect.*

Lemma 6. $\dim_P(\mathcal{F}) \leq K + 2$

Proof. Assume, for the sake of contradiction, that $\dim_P(\mathcal{F}) > K + 2$. It must be that there exists a 'shattered' set

$$\left\{\left(y^{(1)}, z^{(1)}, c^{(1)}\right), \left(y^{(2)}, z^{(2)}, c^{(2)}\right), \ldots, \left(y^{(K+3)}, z^{(K+3)}, c^{(K+3)}\right)\right\} \subset \mathbb{R}^K \times \mathbb{R} \times \mathbb{R},$$

such that, for all $e \in \{0,1\}^{K+3}$, there exists a vector $r_e \in \mathbb{R}^K$ with

$$\zeta\left(r_e^\top y^{(i)} + z^{(i)}\right) \geq c^{(i)} \text{ iff } e_i = 1, \quad \forall\ 1 \leq i \leq K + 3.$$

Observe that we must have $c^{(i)} \in (0, B]$ for all i, since if $c^{(i)} \leq 0$ or $c^{(i)} > B$, then no such shattered set can be demonstrated. But if $c^{(i)} \in (0, B]$, for all $r \in \mathbb{R}^K$,

$$\zeta\left(r^\top y^{(i)} + z^{(i)}\right) \geq c^{(i)} \implies r_e^\top y^{(i)} \geq c^{(i)} - z^{(i)},$$

and

$$\zeta\left(r^\top y^{(i)} + z^{(i)}\right) < c^{(i)} \implies r_e^\top y^{(i)} < c^{(i)} - z^{(i)}.$$

For each $1 \leq i \leq K+3$, define $x^{(i)} \in \mathbb{R}^{K+1}$ component-wise according to

$$x_j^{(i)} \triangleq \begin{cases} y_j^{(i)} & \text{if } j < K+1, \\ c^{(i)} - z^{(i)} & \text{if } j = K+1. \end{cases}$$

Let $A = \{x^{(1)}, x^{(2)}, \ldots, x^{(K+3)}\} \subset \mathbb{R}^{K+1}$, and let A_1 and A_2 be subsets of A satisfying the conditions of Radon's lemma. Define a vector $\tilde{e} \in \{0,1\}^{K+3}$ component-wise according to

$$\tilde{e}_i \triangleq \mathbb{I}_{\left\{x^{(i)} \in A_1\right\}}.$$

Define the vector $\tilde{r} \triangleq r_{\tilde{e}}$. Then, we have

$$\sum_{j=1}^{K} \tilde{r}_j x_j \geq x_{K+1}, \quad \forall\, x \in A_1,$$

$$\sum_{j=1}^{K} \tilde{r}_j x_j < x_{K+1}, \quad \forall\, x \in A_2.$$

Now, let $\bar{x} \in \mathbb{R}^{K+1}$ be a point contained in both the convex hull of A_1 and the convex hull of A_2. Such a point must exist by Radon's lemma. By virtue of being contained in the convex hull of A_1, we must have

$$\sum_{j=1}^{K} \tilde{r}_j \bar{x}_j \geq \bar{x}_{K+1}.$$

Yet, by virtue of being contained in the convex hull of A_2, we must have

$$\sum_{j=1}^{K} \tilde{r}_j \bar{x}_j < \bar{x}_{K+1},$$

which is impossible. ■

With the above pseudo-dimension estimate, we can establish the following lemma, which provides a Chernoff bound for the *uniform* convergence of a certain class of functions:

Lemma 7. *Given a constant $B > 0$, define the function $\zeta\colon \mathbb{R} \to [0, B]$ by*

$$\zeta(t) \triangleq \max\left(\min(t, B), 0\right).$$

Consider a pair of random variables $(Y, Z) \in \mathbb{R}^K \times \mathbb{R}$. For each $i = 1, \ldots, n$, let the pair $\left(Y^{(i)}, Z^{(i)}\right)$ be an i.i.d. sample drawn according to the distribution of (Y, Z). Then, for all $\epsilon \in (0, B]$,

$$\mathsf{P}\left(\sup_{r \in \mathbb{R}^K} \left|\frac{1}{n}\sum_{i=1}^{n} \zeta\left(r^\top Y^{(i)} + Z^{(i)}\right) - \mathsf{E}\left[\zeta\left(r^\top Y + Z\right)\right]\right| > \epsilon\right) \leq 8\left(\frac{32eB}{\epsilon}\log\frac{32eB}{\epsilon}\right)^{K+2} \exp\left(-\frac{\epsilon^2 n}{64B^2}\right).$$

Moreover, given $\delta \in (0, 1)$, if

$$n \geq \frac{64B^2}{\epsilon^2}\left(2(K+2)\log\frac{16eB}{\epsilon} + \log\frac{8}{\delta}\right),$$

then this probability is at most δ.

Proof. Given Lemma 6, this follows immediately from Corollary 2 of of Haussler (1992, Section 4). ■

We are now ready to prove Theorem 4.

Theorem 4. *Under the conditions of Theorem 2, let r_{SALP} be an optimal solution to the SALP (2.14), and let $\hat{r}_{SALP}$ be an optimal solution to the sampled SALP (2.28). Assume that $r_{SALP} \in \mathcal{N}$. Further, given $\epsilon \in (0, B]$ and $\delta \in (0, 1/2]$, suppose that the number of sampled states S satisfies*

$$S \geq \frac{64B^2}{\epsilon^2}\left(2(K+2)\log\frac{16eB}{\epsilon} + \log\frac{8}{\delta}\right).$$

Then, with probability at least $1 - \delta - 2^{-383}\delta^{128}$,

$$\|J^* - \Phi\hat{r}_{SALP}\|_{1,\nu} \leq \inf_{\substack{r \in \mathcal{N} \\ \psi \in \Psi}} \|J^* - \Phi r\|_{\infty, \mathbf{1}/\psi}\left(\nu^\top\psi + \frac{2(\pi_{\mu^*,\nu}^\top\psi)(\alpha\beta(\psi)+1)}{1-\alpha}\right) + \frac{4\epsilon}{1-\alpha}.$$

Proof. Define the vectors

$$\hat{s}_{\mu^*} \triangleq (\Phi\hat{r}_{\text{SALP}} - T_{\mu^*}\Phi\hat{r}_{\text{SALP}})^+, \quad \text{and} \quad \hat{s} \triangleq (\Phi\hat{r}_{\text{SALP}} - T\Phi\hat{r}_{\text{SALP}})^+.$$

Note that $\hat{s}_{\mu^*} \leq \hat{s}$. One has, via Lemma 2, that

$$\Phi\hat{r}_{\text{SALP}} - J^* \leq \Delta^*\hat{s}_{\mu^*}$$

Thus, as in the last set of inequalities in the proof of Theorem 1, we have

$$\|J^* - \Phi\hat{r}_{\text{SALP}}\|_{1,\nu} \leq \nu^\top(J^* - \Phi\hat{r}_{\text{SALP}}) + \frac{2\pi_{\mu^*,\nu}^\top\hat{s}_{\mu^*}}{1-\alpha}. \tag{2.37}$$

Now, let $\hat{\pi}_{\mu^*,\nu}$ be the empirical measure induced by the collection of sampled states $\hat{\mathcal{X}}$. Given a state $x \in \mathcal{X}$, define a vector $Y(x) \in \mathbb{R}^K$ and a scalar $Z(x) \in \mathbb{R}$ according to

$$Y(x) \triangleq \Phi(x)^\top - \alpha P_{\mu^*}\Phi(x)^\top, \quad Z(x) \triangleq -g(x, \mu^*(x)),$$

so that, for any vector of weights $r \in \mathcal{N}$,

$$(\Phi r(x) - T_{\mu^*}\Phi r(x))^+ = \zeta\left(r^\top Y(x) + Z(x)\right).$$

Then,

$$\left|\hat{\pi}_{\mu^*,\nu}^\top\hat{s}_{\mu^*} - \pi_{\mu^*,\nu}^\top\hat{s}_{\mu^*}\right| \leq \sup_{r\in\mathcal{N}}\left|\frac{1}{S}\sum_{x\in\hat{\mathcal{X}}}\zeta\left(r^\top Y(x) + Z(x)\right) - \sum_{x\in\mathcal{X}}\pi_{\mu^*,\nu}(x)\zeta\left(r^\top Y(x) + Z(x)\right)\right|.$$

Applying Lemma 7, we have that

$$\mathsf{P}\left(\left|\hat{\pi}_{\mu^*,\nu}^\top\hat{s}_{\mu^*} - \pi_{\mu^*,\nu}^\top\hat{s}_{\mu^*}\right| > \epsilon\right) \leq \delta. \tag{2.38}$$

Next, suppose $(r_{\text{SALP}}, \bar{s})$ is an optimal solution to the SALP (2.14). Then, with probability at least $1-\delta$,

$$\begin{aligned}
\nu^\top(J^* - \Phi\hat{r}_{\text{SALP}}) + \frac{2\pi_{\mu^*,\nu}^\top\hat{s}_{\mu^*}}{1-\alpha} &\leq \nu^\top(J^* - \Phi\hat{r}_{\text{SALP}}) + \frac{2\hat{\pi}_{\mu^*,\nu}^\top\hat{s}_{\mu^*}}{1-\alpha} + \frac{2\epsilon}{1-\alpha} \\
&\leq \nu^\top(J^* - \Phi\hat{r}_{\text{SALP}}) + \frac{2\hat{\pi}_{\mu^*,\nu}^\top\hat{s}}{1-\alpha} + \frac{2\epsilon}{1-\alpha} \\
&\leq \nu^\top(J^* - \Phi r_{\text{SALP}}) + \frac{2\hat{\pi}_{\mu^*,\nu}^\top\bar{s}}{1-\alpha} + \frac{2\epsilon}{1-\alpha},
\end{aligned} \tag{2.39}$$

where the first inequality follows from (2.38), and the final inequality follows from the optimality of $(\hat{r}_{\text{SALP}}, \hat{s})$ for the sampled SALP (2.28).

Notice that, without loss of generality, we can assume that $\bar{s}(x) = (\Phi r_{\text{SALP}}(x) - T\Phi r_{\text{SALP}}(x))^+$, for each $x \in \mathcal{X}$. Thus, $0 \leq \bar{s}(x) \leq B$. Further,

$$\hat{\pi}_{\mu^*,\nu}^\top \bar{s} - \pi_{\mu^*,\nu}^\top \bar{s} = \frac{1}{S} \sum_{x \in \hat{\mathcal{X}}} \left(\bar{s}(x) - \pi_{\mu^*,\nu}^\top \bar{s} \right),$$

where the right-hand-side is of a sum of zero-mean bounded i.i.d. random variables. Applying Hoeffding's inequality,

$$\mathsf{P}\left(\left| \hat{\pi}_{\mu^*,\nu}^\top \bar{s} - \pi_{\mu^*,\nu}^\top \bar{s} \right| \geq \epsilon \right) \leq 2 \exp\left(-\frac{2S\epsilon^2}{B^2} \right) < 2^{-383}\delta^{128},$$

where final inequality follows from our choice of S. Combining this with (2.37) and (2.39), with probability at least $1 - \delta - 2^{-383}\delta^{128}$, we have

$$\begin{aligned} \|J^* - \Phi \hat{r}_{\text{SALP}}\|_{1,\nu} &\leq \nu^\top (J^* - \Phi r_{\text{SALP}}) + \frac{2\hat{\pi}_{\mu^*,\nu}^\top \bar{s}}{1-\alpha} + \frac{2\epsilon}{1-\alpha} \\ &\leq \nu^\top (J^* - \Phi r_{\text{SALP}}) + \frac{2\pi_{\mu^*,\nu}^\top \bar{s}}{1-\alpha} + \frac{4\epsilon}{1-\alpha}. \end{aligned}$$

The result then follows from (2.17)–(2.19) in the proof of Theorem 2. ■

3

PATHWISE METHOD FOR OPTIMAL STOPPING PROBLEMS

3.1. Introduction

Consider the following optimal control problem: a Markov process evolves in discrete time over the state space $\mathcal{X}$. Denote this process by $\{x_t,\ t \geq 0\}$. The process is associated with a state-dependent reward function $g\colon \mathcal{X} \to \mathbb{R}$. Our goal is to solve the optimization problem

$$\sup_{\tau}\ \mathsf{E}\Big[\alpha^{\tau} g(x_\tau)\ \big|\ x_0 = x\Big],$$

where the optimization is over stopping times τ adapted to the $\{x_t\}$ process, and $\alpha \in [0, 1)$ is a discount factor. In other words, we wish to pick a stopping time that maximizes the expected discounted reward. Such *optimal stopping* problems arise in a myriad of applications, most notably, in the pricing of financial derivatives.

In principle, the above stopping problem can be solved via the machinery of dynamic programming. However, the applicability of the dynamic programming approach is typically curtailed by the size of the state space $\mathcal{X}$. In particular, in many applications of interest, $\mathcal{X}$ is a high-dimensional space and thus intractably large.

Since high-dimensional stopping problems are important from a practical perspective, a number of alternative approaches that contend with the so-called 'curse of dimensionality' have emerged. There are two broad classes of methods by which one can develop bounds on the optimal value of a stopping problem, motivated essentially by distinct characterizations of the optimal solution to the stopping problem:

- **Lower Bounds / Approximate Dynamic Programming (ADP).** The optimal control is characterized by an optimal value function, which, in turn, is the unique solution to the so-called Bellman equation. A natural goal is to attempt to approximate this value function by finding 'approximate' solutions to the Bellman equation. This is the central goal of ADP algorithms such as *regression pricing* methods of the type pioneered by Carriere (1996), Longstaff and Schwartz (2001), and Tsitsiklis and Van Roy (2001). Such an approximate solution can then be used to both define a control policy and, via simulation of that (sub-optimal) policy, a lower bound on the optimal value function.

- **Upper Bounds / Martingale Duality.** At a high level, this approach may be thought of as relaxing the requirement of causality, while simultaneously introducing a penalty for this relaxation. The appropriate penalty function is itself a stochastic process (a martingale), and by selecting the 'optimal' martingale, one may in fact solve the original stopping problem. In the context of stopping problems, part of this characterization appears to date back at least to the work by Davis and Karatzas (1994), and this idea was subsequently fully developed by Rogers (2002) and Haugh and Kogan (2004).

  Not surprisingly, finding such an optimal martingale is no easier than solving the original stopping problem. As such, the martingale duality approach consists of heuristically selecting 'good' martingale penalty functions, using these to compute upper bounds on the price (i.e., the optimal value of the stopping problem). Here, two techniques are commonly employed. The first, which we will call a *dual value function* approach, derives a martingale penalty function from an approximation to the optimal value function. Such an approximation will typically be generated, for example, along the course of regression pricing procedures such as those described above. Alternatively, in what we will call a *dual policy* approach, a martingale penalty function can be derived from a heuristic control policy. This latter approach was proposed by Andersen and Broadie (2004). A good control policy will typically also be generated using a regression pricing procedure.

A combination of these methods have come to represent the state-of-the-art in financial applications (see, e.g., Glasserman, 2004). There, practitioners typically use regression pricing to derive optimal policies for the exercise of American and Bermudan options, and to derive lower bounds on prices. The martingale duality approach is then applied in a complementary fashion to generate upper bounds, using either the dual value function approach or the dual policy approach. Take together, these methods provide a 'confidence bound' on the true price. In this area, the development of such methodologies is thought to be worth considerable financial value, and thus may represent the greatest practical success of approximate dynamic programming.

In a nutshell, we introduce a new approach to solving high-dimensional stopping problems that draws on techniques from both of the methodologies above, and ultimately unifies our understanding of the two approaches. This new method is ultimately seen to be desirable from the practical perspective of rapidly pricing high-dimensional financial derivatives. In addition, we develop a theory that allows us to characterize the quality of the solutions produced by the approaches above.

In greater detail, we make the following contributions:

- **A New Algorithm.** ADP algorithms systematically explore approximations to the optimal value function within the span of some pre-defined set of basis functions. The duality approach, on the other hand, relies on an ad-hoc specification of an appropriate martingale penalty process. We introduce a new approach, which we call the *pathwise optimization* (PO) method. The PO method systematizes the search for a good martingale penalty process. In particular, given a set of basis functions whose linear span is expected to contain a good approximation to the optimal value function, we posit a family of martingales. As it turns out, finding a martingale within this family that produces the best possible upper bound to the value function is a convex optimization problem. The PO method seeks to solve this problem. We show that this method has several merits relative to extant schemes:

  1. The PO method is a specific instance of the dual value function approach. By

construction, however, the PO method produces an upper bound that is provably tighter than *any* other dual value function approach that employs a value function approximation contained in the span of the same basis function set. These latter approximations are analogous to what is typically found using regression methods of the type proposed by Longstaff and Schwartz (2001) and Tsitsiklis and Van Roy (2001). We demonstrate this fact in numerical experiments, where we will show that, given a fixed set of basis functions, the benefit of the PO method over the dual value function approach in concert with regression pricing can be substantial. We also see that the incremental computational overhead of the PO method over the latter method is manageable.

2. We compare the PO method to upper bounds generated using the dual policy approach in concert with policies derived from regression pricing. Given a fixed set of basis functions, we will see in numerical experiments that the PO method yields upper bounds that are comparable to but not as tight as those from the latter approach. However, the PO method does so in a substantially shorter amount of time, typically requiring a computational budget that is smaller by an order of magnitude.

3. The aforementioned regression techniques are the mainstay for producing control policies and lower bounds in financial applications. We illustrate that the PO method yields a continuation value approximation that can subsequently be used to derive control policies and lower bounds. In computational experiments, these control policies and lower bounds are substantially superior to those produced by regression methods.

In summary, the PO method is quite attractive from a practical perspective.

- **Approximation Theory.** We offer new guarantees on the quality of upper bounds of martingale penalty approaches in general, as well as specific guarantees for the PO method. We compare these guarantees favorably to guarantees developed for other

ADP methods. Our guarantees characterize the structural properties of an optimal stopping problem that are general determinants of performance for these techniques. Specifically:

1. In an infinite horizon setting, we show that the quality of the upper bound produced by the generic martingale duality approach depends on three parameters: the error in approximating the value function (measured in a root-mean-squared error sense), the square root of the effective time horizon (as also observed by Chen and Glasserman (2007)), and a certain measure of the 'predictability' of the underlying Markov process. We believe that this latter parameter provides valuable insight on aspects of the underlying Markov process that make a particular pricing problem easy or hard.

2. In an infinite horizon setting, we produce *relative* upper bound guarantees for the PO method. In particular, we produce guarantees on the upper bound that scale linearly with the approximation error corresponding to the *best possible* approximation to the value function within the span of the basis functions employed in the approach. Note that the latter approximation is typically not computable. This result makes precise the intuition that the PO method produces good price approximations if there exists *some* linear combination of the basis functions that is able to describe the value function well.

3. Upper bounds produced by the PO methods can be directly compared to upper bounds produced by linear programming-based ADP algorithms of the type introduced by Schweitzer and Seidmann (1985), de Farias and Van Roy (2003), and Desai et al. (2009). In particular, we demonstrate that the PO method produces provably tighter upper bounds than the latter methods. While these methods have achieved considerable success in a broad range of large scale dynamic optimization problems, they are dominated by the PO method for optimal stopping problems.

ADP algorithms are usually based on an approximate approach for solving Bellman's equation. In the context of optimal stopping, methods have been proposed that are variations of approximate value iteration (Tsitsiklis and Van Roy, 1999; Yu and Bertsekas, 2007), approximate policy iteration (Clément et al., 2002; Longstaff and Schwartz, 2001), and approximate linear programming (Borkar et al., 2009).

Martingale duality-based upper bounds for the pricing of American and Bermudan options, which rely on Doob's decomposition to generate the penalty process, were introduced by Rogers (2002) and Haugh and Kogan (2004). Rogers (2002) suggests the possibility of determining a good penalty process by optimizing of linear combinations of martingales; our method is a special case of this which uses a specific parametrization of candidate martingales in terms of basis functions. Andersen and Broadie (2004) show how to compute martingale penalties from rules and obtain upper bounds; practical improvements to these technique were studied by Broadie and Cao (2008). An alternative 'multiplicative' approach to duality was introduced by Jamshidian (2003). Its connections with martingale duality approach were explored in Chen and Glasserman (2007), who also develop approximation guarantees for martingale duality upper bounds. Belomestny et al. (2009) describe a variation of the martingale duality procedure that does not require inner simulation. Rogers (2010) describes a pure dual algorithm for pricing.

The remainder of the chapter is organized as follows: in Section 3.2, we formulate the optimal stopping problem and illustrate the general martingale penalty approach. In Section 3.3, we introduce our new algorithm, the PO method. Section 3.4 illustrates the benefits of the PO method in a numerical case study of pricing high-dimensional financial derivatives. In Section 3.5, we develop our theoretical results.

3.2. Formulation

Our framework will be that of an optimal stopping problem over a finite time horizon. Specifically, consider a discrete-time Markov chain with state $x_t \in \mathcal{X}$ at each time $t \in \mathcal{T} \triangleq \{0, 1, \ldots, d\}$. For simplicity, assume that the chain has a the state space $\mathcal{X}$ that is finite.

Denote by P the transition kernel of the chain. Without loss of generality, we will assume that P is time-invariant. Let $\mathcal{F} \triangleq \{\mathcal{F}_t\}$ be the natural filtration generated by the process $\{x_t\}$, i.e., for each time t, $\mathcal{F}_t \triangleq \sigma(x_0, x_1, \ldots, x_t)$.

Given a function $g\colon \mathcal{X} \to \mathbb{R}$, we define the payoff of stopping when the state is x_t as $g(x_t)$. A stationary exercise policy $\mu \triangleq \{\mu_t,\ t \in \mathcal{T}\}$ is a collection of functions where each $\mu_t\colon \mathcal{X} \to \{\mathsf{STOP}, \mathsf{CONTINUE}\}$ determines the choice of action at time t as a function of the state x_t. Without loss of generality, we will require that $\mu_d(x) = \mathsf{STOP}$ for all $x \in \mathcal{X}$, i.e., the process is always stopped at the final time d.

We are interested in finding a policy which maximizes the expected discounted payoff of stopping. The value of a policy μ assuming one starts at state x in period t is given by

$$J_t^\mu(x) \triangleq \mathsf{E}\left[\alpha^{\tau_\mu(t)-t} g(x_{\tau_\mu(t)}) \;\middle|\; x_t = x\right],$$

where $\tau_\mu(t)$ is the stopping time $\tau_\mu(t) \triangleq \min\{s \geq t \ :\ \mu(x_s) = \mathsf{STOP}\}$. Our goal is to find a policy μ that simultaneously maximizes the value function $J_t^\mu(x)$ for all t and x. The existence of such an optimal policy is a standard fact. We will denote such an *optimal policy* by μ^* and the corresponding *optimal value function* by J^*.

In principle, J^* may be computed via the following dynamic programming backward recursion, for all $x \in \mathcal{X}$ and $t \in \mathcal{T}$,

$$J_t^*(x) \triangleq \begin{cases} \max\left\{g(x),\ \alpha\mathsf{E}\left[J_{t+1}^*(x_{t+1}) \mid x_t = x\right]\right\} & \text{if } t < d. \\ g(x) & \text{if } t = d. \end{cases} \tag{3.1}$$

The corresponding optimal stopping policy μ^* is 'greedy' with respect to J^* and given by

$$\mu_t^*(x) \triangleq \begin{cases} \mathsf{CONTINUE} & \text{if } t < d \text{ and } g(x) < \alpha\mathsf{E}[J_{t+1}^*(x_{t+1}) \mid x_t = x], \\ \mathsf{STOP} & \text{otherwise.} \end{cases} \tag{3.2}$$

3.2.1. The Martingale Duality Approach

Let $\mathcal{S}$ be the space of real-valued functions on $\mathcal{X}$, i.e., functions of state, and let $\mathcal{P}$ be the space of real-valued functions on $\mathcal{X} \times \mathcal{T}$, i.e., time-dependent functions of state. We begin

by defining the *martingale difference operator* Δ. The operator Δ maps a function $V \in \mathcal{S}$ to the a function $\Delta V \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ according to

$$(\Delta V)(x_1, x_0) \triangleq V(x_1) - \mathsf{E}[V(x_1)|x_0].$$

Given an arbitrary function $J \in \mathcal{P}$, define the process

$$M_t \triangleq \sum_{s=1}^{t} \alpha^s (\Delta J_s)(x_s, x_{s-1}), \quad \forall\, t \in \mathcal{T}.$$

where $J_s \triangleq J(\cdot, s)$. Then, M is a martingale adapted to the filtration $\mathcal{F}$. Hence, we view Δ as a projection onto the space of martingale differences.

Next, we define for each $t \in \mathcal{T}$, the *martingale duality upper bound operator* $F_t \colon \mathcal{P} \to \mathcal{S}$ according to:

$$(F_t J)(x) \triangleq \mathsf{E}\left[\max_{t \le s \le d} \alpha^{s-t} g(x_s) - \sum_{p=t+1}^{s} \alpha^{p-t} \Delta J_p(x_p, x_{p-1}) \;\middle|\; x_t = x \right].$$

Finally, we define $J^* \in \mathcal{P}$ according to $J^*(x,t) \triangleq J_t^*(x)$. We are now ready to state the following key lemma, due to Rogers (2002) and Haugh and Kogan (2004). A proof is provided in Appendix 3.6 for completeness.

Lemma 8 (Martingale Duality).

(i) *(Weak Duality)* *For any $J \in \mathcal{P}$ and all $x \in \mathcal{X}$ and $t \in \mathcal{T}$, $J_t^*(x) \le F_t J(x)$.*

(ii) *(Strong Duality)* *For all $x \in \mathcal{X}$ and $t \in \mathcal{T}$, $J_t^*(x) = F_t J^*(x)$.*

The above result may be succinctly stated as follows: For any $t \in \mathcal{T}, x \in \mathcal{X}$,

$$J_t^*(x) = \inf_{J \in \mathcal{P}} F_t J(x). \tag{3.3}$$

This is an alternative (and somewhat convoluted) characterization of the optimal value function J^*. Its value, however, lies in the fact that *any* $J \in \mathcal{P}$ yields an upper bound, and evaluating this upper bound for a given J is for all practical purposes *not* impacted by the size of $\mathcal{X}$. Indeed, extant approaches to using the above characterization to produce upper

bounds on J^* use, as surrogates for J, an approximation of the optimal value function J^* (see, e.g., Glasserman, 2004). This approximation can be derived over the course of a regression pricing method of the type introduced by Longstaff and Schwartz (2001) or Tsitsiklis and Van Roy (2001). We call this the *dual value function* approach. Alternatively, an approximating value function corresponding to a sub-optimal policy (Andersen and Broadie, 2004) can be used, where the policy is typically produced by a regression pricing method. We call this the *dual policy* approach.

3.3. The Pathwise Optimization Method

Motivated by the (in general, intractable) optimization problem (3.3), we are led to consider the following: what if one chose to optimize over functions $J \in \hat{\mathcal{P}} \subset \mathcal{P}$, where $\hat{\mathcal{P}}$ is compactly parametrized and easy to optimize over? Motivated by ADP algorithms that seek approximations to the optimal value function that are linear combinations of some set of basis functions, we are led to the following parametrization: Assume we are given a collection of K *basis functions*

$$\Phi \triangleq \{\phi_1, \phi_2, \ldots, \phi_K\} \subset \mathcal{P}.$$

Ideally these basis functions capture features of the state space or optimal value function that are relevant for effective decision making, but frequently generic selections work well (e.g., all monomials up to a fixed degree). We may then consider restricting attention to functions that are linear combinations of elements of Φ, i.e., functions of the form

$$(\Phi r)_t(x) \triangleq \sum_{\ell=1}^{K} r_\ell \phi_\ell(x,t), \quad \forall\ x \in \mathcal{X},\ t \in \mathcal{T}.$$

Here, $r \in \mathbb{R}^K$ is known as a *weight vector*. Denote this sub-space of $\mathcal{P}$ by $\hat{\mathcal{P}}$ and note that $\hat{\mathcal{P}}$ is compactly parameterized by K parameters (as opposed to $\mathcal{P}$ which is specified by $|\mathcal{X} \times \mathcal{T}|$ parameters in general). Setting the starting epoch to $t = 0$ for convenience, we may rewrite the optimization problem (3.3) restricted to $\hat{\mathcal{P}}$ as:

$$\inf_r\ F_0 \Phi r(x). \tag{3.4}$$

We call this problem the *pathwise optimization* (PO) problem. The lemma below demonstrates that (3.4) is, in fact, a *convex* optimization problem.

Lemma 9. *For every $t \in \mathcal{T}$ and $x \in \mathcal{X}$, the function $r \mapsto F_t \Phi r(x)$ is convex in r.*

Proof. Observe that, given a fixed (x, t) and as a function of r, $F_t \Phi r(x)$ is a non-negative linear combination of a set of pointwise suprema of affine functions of r, and hence must be convex as each of these operations preserves convexity. ■

Before devising a practical approach to solving (3.4), let us reflect on what solving this program accomplishes. We have devised a means to systematically and, anticipating the developments in the sequel, practically, find a martingale penalty process within a certain parametrized family of martingales. To appreciate the value of this approach, we note that it is guaranteed, by construction, to produce tighter upper bounds on price than *any* dual value function methods derived from value function approximations that are within the span of the same basis function set. These latter approximations are analogous to what is typically found using regression methods of the type proposed by Longstaff and Schwartz (2001) and Tsitsiklis and Van Roy (2001).[1]

Now, from a practical perspective, the optimization problem (3.4) is an unconstrained minimization of a convex function over a relatively low-dimensional space. Algorithmically, the main challenge is evaluating the objective, which is the expectation of a functional over paths in a high-dimensional space. We will demonstrate that this can be efficiently approximated via sampling.

3.3.1. Solution via Sampling

Consider sampling S independent *outer* sample paths of the underlying Markov process starting at some given state x_0; denote path i by $x^{(i)} \triangleq \left\{x_s^{(i)},\ s \in \mathcal{T}\right\}$ for $i = 1, 2, \ldots, S$. By

[1]Strictly speaking, the regression pricing approaches of Longstaff and Schwartz (2001) and Tsitsiklis and Van Roy (2001) seek linearly parameterized approximations to the optimal continuation value function, as is described in Section 3.4. However, the same ideas could easily be applied to find linearly parameterized approximations to the optimal value function.

the law of large numbers, we know that for a fixed r,

$$\frac{1}{S}\sum_{i=1}^{S}\max_{0\le s\le d}\left(\alpha^s g\Big(x_s^{(i)}\Big)-\sum_{p=1}^{s}\alpha^p\Delta(\Phi r)_p\Big(x_p^{(i)},x_{p-1}^{(i)}\Big)\right)\;\to\;F_0\Phi r(x_0),\quad\text{as } T\to\infty.$$

This suggests a useful proxy for the objective in the optimization problem (3.4).

Before writing down a final, implementable optimization program, however, consider the quantities that appear in the left-hand side of the expression above,

$$\Delta(\Phi r)_p\Big(x_p^{(i)},x_{p-1}^{(i)}\Big)=(\Phi r)_p(x_p^{(i)})-\mathsf{E}\left[(\Phi r)_p(x_p)\;\middle|\;x_{p-1}=x_{p-1}^{(i)}\right].$$

The expectation in the above expression may, in certain cases, be computed in closed form (see, e.g., Belomestny et al., 2009; Glasserman and Yu, 2002). Here, we choose to instead replace the expectation by its empirical counterpart. In particular, we generate I independent *inner* samples $\Big\{x_p^{(i,j)},\ j=1,\dots,I\Big\}$, conditional on $x_{p-1}=x_{p-1}^{(i)}$. In other words, these inner samples are generated according to the one-step transition distribution $P\Big(x_{p-1}^{(i)},\cdot\Big)$. Then, we employ the approximation

$$\hat{\mathsf{E}}\left[(\Phi r)_p(x_p)\,\middle|x_{p-1}^{(i)}\right]\triangleq\frac{1}{I}\sum_{j=1}^{I}(\Phi r)_p\Big(x_p^{(i,j)}\Big)\to\mathsf{E}\left[(\Phi r)_p(x_p)\mid x_{p-1}=x_{p-1}^{(i)}\right],\quad\text{as } I\to\infty.$$

Having thus replaced expectations by their empirical counterparts, we are ready to state a general, implementable, sampled variant of the optimization problem (3.4):

$$\begin{aligned}&\underset{r,u}{\text{minimize}}\quad\frac{1}{S}\sum_{i=1}^{S}u_i\\ &\text{subject to}\quad u_i+\sum_{p=1}^{s}\alpha^p\left\{(\Phi r)_p\Big(x_p^{(i)}\Big)-\hat{\mathsf{E}}\left[(\Phi r)_p(x_p)\,\middle|x_{p-1}^{(i)}\right]\right\}\ge\alpha^s g(x_s^{(i)}),\\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\forall\ 1\le i\le S,\ 0\le s\le d.\end{aligned}\tag{3.5}$$

Denoting a solution to (3.5) by $\hat{r}_{\mathsf{PO}}$, we propose, as an upper bound on $J_0^*(x_0)$, the quantity $F_0\Phi\hat{r}_{\mathsf{PO}}(x_0)$. This latter quantity may also be estimated via the same sampling procedure. However, in order to obtain an unbiased estimate of an upper bound, a second set of samples must be generated that is independent of those used in solving (3.5).

While we do not show it here, under suitable technical conditions one may establish that as S and I grow large, the value of the solution to the above optimization problem approaches the optimal value of the pathwise optimization problem (3.4).[2] The optimization problem we have proposed above is a linear program (LP) with $K+S$ variables and $S(d+1)$ constraints. Since no two variables $\{u_i, u_j\}$ with $i \neq j$ appear in the same constraint, it is easy to see that the Hessian corresponding to a logarithmic barrier function for the problem has block arrow structure. Inverting this matrix will require $O(K^2S)$ floating point operations (see, for example, Appendix C, page 675 Boyd and Vandenberghe, 2004). Consequently, one may argue that the complexity of solving this LP via an interior point method essentially scales linearly with S.

3.3.2. Lower Bounds and Policies

The PO method generates upper bounds on the performance of an optimal policy. We are also interested in generating good stopping policies, which, in turn, will yield lower bounds on optimal performance. Here, we describe a method that does so by computing a continuation value approximation.

In particular, for $0 \leq t < d$ and $x_t \in \mathcal{X}$, denote by $C_t^*(x_t)$ the optimal continuation value, or, the best value the can be achieved by any policy at time t and state x_t that does not immediately stop. Mathematically,

$$C_t^*(x_t) \triangleq \alpha \mathsf{E}\left[J_{t+1}^*(x_{t+1}) \middle| x_t \right].$$

Note that the optimal policy μ^* can be expressed succinctly in terms of C^* via

$$\mu_t^*(x) \triangleq \begin{cases} \mathsf{CONTINUE} & \text{if } t < d \text{ and } g(x) < C_t^*(x), \\ \mathsf{STOP} & \text{otherwise,} \end{cases} \tag{3.6}$$

for all $t \in \mathcal{T}$ and $x \in \mathcal{X}$. In other words, μ^* decides to stop or not by acting greedily using C^* to assess the value of not stopping. Inspired by this, given a good approximation $\tilde{C}$ to

[2]This may be established via an appeal to a uniform law of large numbers such as Theorem 7.48 of (Shapiro et al., 2009), under the assumption that the basis functions and reward function are bounded, and that r is restricted to a compact set.

the optimal continuation value, we can attempt to construct a good policy by replacing C^* with $\tilde{C}$ in (3.6).

Now, given a solution to (3.5), $\hat{r}_{\mathsf{PO}}$, we can generate generate upper bounds on continuation value and regress these against basis functions to generate a continuation value approximation. In particular, it follows from Lemma 8 that

$$C_t^*(x_t) \leq \mathsf{E}\left[\max_{t+1\leq s\leq d} \alpha^{s-t} g(x_s) - \sum_{p=t+2}^{s} \alpha^{p-t} \Delta(\Phi \hat{r}_{\mathsf{PO}})_p(x_p, x_{p-1}) \;\middle|\; x_t\right], \tag{3.7}$$

for all $0 \leq t < d$ and $x_t \in \mathcal{X}$. Thus, at time t along the ith sample path, a point estimate of an upper bound on $C_t^*\left(x_t^{(i)}\right)$ is given by

$$\bar{c}_t^{(i)} \triangleq \max_{t+1\leq s\leq d} \alpha^{s-t} g_s\left(x_s^{(i)}\right) - \sum_{p=t+2}^{s} \alpha^{p-t} \left\{(\Phi \hat{r}_{\mathsf{PO}})_p\left(x_p^{(i)}\right) - \hat{\mathsf{E}}\left[(\Phi \hat{r}_{\mathsf{PO}})_p(x_p) \,\middle|\, x_{p-1}^{(i)}\right]\right\}.$$

For each $0 \leq t < d-1$, the values $\left\{\bar{c}_t^{(i)},\ 1 \leq i \leq S\right\}$ can now be regressed against basis functions to obtain a continuation value approximation. In particular, defining a set of K basis functions of the state x_t,

$$\Psi_t \triangleq \{\psi_{1,t}, \psi_{2,t}, \ldots, \psi_{K,t}\} \subset \mathcal{S},$$

we can consider linear combinations of the form

$$(\Psi_t \kappa_t)(x) \triangleq \sum_{\ell=1}^{K} \kappa_{\ell,t} \psi_{\ell,t}(x), \quad \forall\ x \in \mathcal{X},$$

where $\kappa_t \in \mathbb{R}^K$ is a weight vector.[3] The weight vectors $\{\kappa_t,\ 0 \leq t < d\}$ can be computed efficiently in a recursive fashion as follows:

1. Iterate backward over times $t = d-1, d-2, \ldots, 0$.

2. For each sample path $1 \leq i \leq S$, we need to compute the continuation value estimate $\bar{c}_t^{(i)}$. If $t = d-1$, this is simply

$$\bar{c}_{d-1}^{(i)} = \alpha g_d\left(x_d^{(i)}\right).$$

[3]In our experimental work we used $\psi_{\ell,t}(\cdot) = \phi_i(\cdot, t)$. In other words, we used the same basis function architecture to approximate continuation values as were used for value functions.

If $t < d-1$, this can be computed recursively as

$$\bar{c}_t^{(i)} = \alpha \max \left\{ g_{t+1}\left(x_{t+1}^{(i)}\right),\ \bar{c}_{t+1}^{(i)} - \alpha \left((\Phi \hat{r}_{\mathsf{PO}})_{t+2}\left(x_{t+2}^{(i)}\right) - \hat{\mathsf{E}}\left[(\Phi \hat{r}_{\mathsf{PO}})_{t+2}(x_{t+2}) \,\middle|\, x_{t+1}^{(i)} \right] \right) \right\}.$$

3. Compute the weight vector κ_t via the regression

$$\kappa_t \in \operatorname*{argmin}_{\kappa}\ \frac{1}{S} \sum_{i=1}^{S} \left(\Psi_t \kappa\left(x_t^{(i)}\right) - \bar{c}_t^{(i)} \right)^2.$$

We may then use the sub-optimal policy that is greedy with respect to the continuation value approximation given by $\Psi_t \kappa_t$, for each $0 \leq t \leq d-1$.

Observe that, at a high-level, our algorithm is reminiscent of the regression pricing approach of Longstaff and Schwartz (2001). Both methods proceed backward in time over a collection of sample paths, regressing basis functions against point estimates of continuation values. Longstaff and Schwartz (2001) use point estimates of lower bounds derived from sub-optimal future policies. We, on the other had, use point estimates of upper bounds derived from the PO linear program (3.5). As we shall see in Section 3.4, despite the similarities, the PO-derived policy can offer significant improvements in practice.

3.4. Computational Results

In this section, we will illustrate the performance of the PO method versus a collection of competitive benchmark algorithms in numerical experiments. We begin by defining the benchmark algorithms in Section 3.4.1. In Section 3.4.2, we define the problem setting, which is that of pricing a high-dimensional Bermudan option. Implementation details such as the choice of basis functions and the state sampling parameters are given in Section 3.4.3. Finally, the results are presented in Section 3.4.4.

3.4.1. Benchmark Methods

The landscape of techniques available for pricing high-dimensional options is rich; a good overview of these is available from Glasserman (2004, Chapter 8). We consider the following

benchmarks, representative of mainstream methods, for purposes of comparison with the PO method:

- **Lower Bound Benchmark.** The line of work developed by Carriere (1996), Tsitsiklis and Van Roy (2001), and Longstaff and Schwartz (2001) seeks to produce approximations to the optimal continuation value function. These approximations are typically weighted combinations of pre-specified basis functions that are fit via a regression-based methodology. The greedy policies with respect to these approximations yield lower bounds on price.

  We generate a continuation value approximation $\hat{C}$ using the Longstaff and Schwartz (2001) (LS) method. Details are available from Glasserman (2004, Chapter 8, pg. 461). We simulate the greedy policy with respect to this approximation to generate lower bounds. We refer to this approach as LS-LB.

- **Upper Bound Benchmarks.** The martingale duality approach, originally proposed for this task by Rogers (2002) and Haugh and Kogan (2004) is widely used for upper bounds. Recall from Section 3.2.1 that a martingale for use in the duality approach is computed using the optimal value function, and extant heuristics use surrogates that approximate the optimal value function. We consider the following surrogates:

  1. DVF-UB: This is a dual value function approach that derives a value function approximation from the continuation value approximation of LS-LB regression pricing procedure. In particular, given the LS-LB continuation value approximation, $\hat{C}$, we generate a value function approximation $\hat{V}$ according to

  $$\hat{V}_t(x) \triangleq \max\{g(x), \hat{C}_t(x)\}, \quad \forall\, x \in \mathcal{X},\ t \in \mathcal{T}.$$

  This approach is described by Glasserman (2004, Section 8.7, pg. 473).

  2. DP-UB: This is a dual policy approach that derives a value function approximation from the policy suggested by the LS-LB regression pricing procedure. In

particular, let $\hat{\mu}$ denote the greedy policy derived from the LS-LB continuation value approximation $\hat{C}$, i.e., for all states x and times t,

$$\hat{\mu}_t(x) \triangleq \begin{cases} \mathsf{CONTINUE} & \text{if } t < d \text{ and } g(x) < \hat{C}_t(x), \\ \mathsf{STOP} & \text{otherwise.} \end{cases}$$

Define $V_t^{\hat{\mu}}(x)$ as the value of using the policy $\hat{\mu}$ starting at state x in time t. The quantity $V_t^{\hat{\mu}}(x)$ can be computed via an inner Monte Carlo simulation over paths that start at time t in state x. This can then be used as a value function surrogate to derive a martingale for the duality approach. This approach was introduced by Andersen and Broadie (2004) and a detailed description is available from Glasserman (2004, Section 8.7, pg. 474–475).

The LS-LB, DVF-UB, and DP-UB methods described above will be compared with upper bounds computed with the PO method (PO-UB) and their corresponding lower bounds (PO-LB), as described in Section 3.3. Further implementation details for each of these techniques will be provided in Section 3.4.3.

3.4.2. Problem Setting

Specifically, we consider a Bermudan option over a calendar time horizon T defined on multiple assets. The option has a total of d exercise opportunities at calendar times $\{\delta, 2\delta, \ldots, \delta d\}$, where $\delta \triangleq T/d$. The payoff of the option corresponds to that of a call option on the maximum of n assets with an *up-and-out* barrier. We assume a Black-Scholes framework, where risk-neutral asset price dynamics for each asset j are given by a geometric Brownian motion, i.e., the price process $\left\{P_s^j,\ s \in \mathbb{R}_+\right\}$ follows the stochastic differential equation

$$dP_s^j = (r - \zeta_j)P_s^j\, ds + \sigma_j P_s^j\, dW_s^j. \tag{3.8}$$

Here, r is the continuously compounded risk-free interest rate, ζ_j is the dividend rate of asset j, σ_j is the volatility of asset j, W_s^j is a standard Brownian motion, and the instantaneous correlation of each pair W_s^j and $W_s^{j'}$ is $\rho_{jj'}$. Let $\{p_t,\ 0 \le t \le d\}$ be the discrete time process

obtained by sampling P_s at intervals of length δ, i.e., $p_t^j \triangleq P_{\delta t}^j$ for each $0 \leq t \leq d$. On the discrete time scale indexed by t, the possible exercise times are given by $\mathcal{T} \triangleq \{1, 2, \ldots, d\}$, and the discount factor is given by $\alpha \triangleq e^{-r\delta}$.

The option is 'knocked out' (and worthless) at time t if, at any of the times preceding and including t, the maximum of the n asset prices exceeded the barrier B. We let $y_t \in \{0, 1\}$ serve as an indicator that the option is knocked out at time t. In particular, $y_t = 1$ if the option has been knocked out at time t or at some time prior, and $y_t = 0$ otherwise. The $\{y_t\}$ process evolves according to

$$y_t = \begin{cases} \mathbb{I}_{\left\{\max_{1\leq j\leq n}\ p_0^j \geq B\right\}} & \text{if } t = 0, \\ y_{t-1} \vee \mathbb{I}_{\left\{\max_{1\leq j\leq n}\ p_t^j \geq B\right\}} & \text{otherwise.} \end{cases}$$

A state in the associated stopping problem[4] is then given by the tuple $x \triangleq (p, y) \in \mathbb{R}^n \times \{0, 1\}$, and the payoff function is defined according to

$$g(x) \triangleq \Big(\max_j\ p_i(x) - K\Big)^+ \Big(1 - y(x)\Big).$$

where $y(x)$ and $p_j(x)$, respectively, are the knock-out indicator and the jth price coordinates of the composite state x.

3.4.3. Implementation Details

Basis Functions. We use the following set of $n + 2$ basis functions:

$$\begin{aligned} \phi_1(x, t) &= \Big(1 - y(x)\Big), \\ \phi_2(x, t) &= g(x), \\ \phi_{j+2}(x, t) &= \Big(1 - y(x)\Big)\, p_j(x), \quad \forall\ 1 \leq j \leq n. \end{aligned}$$

Described succinctly, our basis function architecture consists of a constant function, the payoff function, and linear functions of each asset price, where we have further ensured that

[4]Note that, as opposed to the setting of Section 3.2, the state space here is not finite. However, the results discussed earlier can easily be extended to the present setting.

each basis function takes the value zero in states where the option is knocked out. This is because zero is known to be the exact value of the option in such states. Note that many other basis functions are possible. For instance, the prices of barrier options on each of the individual stocks seems like a particularly appropriate choice. We have chosen a relatively generic basis architecture, however, in order to disentangle the study of the pricing methodology from the goodness of a particular tailor-made architecture.

State Sampling. Both the PO method as well as the benchmark methods require sampling states from the underlying Markov chain. In general, it may be possible to judiciously choose the sampling parameters so as to, for example, optimize the accuracy of a method given a fixed computational budget, and that such a good choice of parameters will likely vary from method to method. We have not attempted such an optimization. Instead, we have considered a setup with sampling parameters that generally follow those chosen by Andersen and Broadie (2004). Briefly, the sampling parameters chosen are as follows:

- LS-LB: This approach requires sample paths of the underlying Markov process to run the regression procedure. We used 200,000 sample paths for the regression. The greedy policy with respect to the regressed continuation values was evaluated over 2,000,000 sample paths.

- PO-UB: In the notation of Section 3.3.1, we solved the LP (3.5) using $S = 30{,}000$ outer sample paths, and $I = 500$ next state inner samples for one-step expectation computations. Given a solution, $\hat{r}_{\mathsf{PO}}$, we evaluated $F_0\Phi\hat{r}_{\mathsf{PO}}(x_0)$ using a distinct set of $S = 30{,}000$ outer sample paths, with $I = 500$ inner samples for one-step expectations.

- PO-LB: The policy here is constructed using computations entailed in the PO-UB method. We evaluate this policy to compute the lower bound using the same set of 2,000,000 sample paths used for the evaluation of LS-LB above.

- DVF-UB: As discussed earlier, a value function estimate $\hat{V}$ is obtained from the continuation value estimates of the regression procedure used for LS-LB above. We then

estimate the DVF-UB upper bound, $F_0\hat{V}(x_0)$, using the same set of 30,000 sample paths and one step samples in the evaluation of PO-UB above.

- DP-UB: As discussed earlier, this approach uses the value function approximation $V^{\hat{\mu}}$. We obtain continuation value estimates $\hat{C}$ via the regression computation for LS-LB. We estimate the upper bound $F_0V^{\hat{\mu}}(x_0)$ using 3,000 sample paths;[5] we evaluate $V^{\hat{\mu}}$ at each point along these sample paths using 10,000 inner sample paths.

3.4.4. Results

In the numerical results that follow, the following common problem settings were used:[6]

- strike price: $K = 100$
- knock-out barrier price: $B = 170$
- time horizon $T = 3$ years
- risk-free rate: $r = 5\%$ (annualized)
- dividend rate: $\zeta_j = 0$ (annualized)
- volatility: $\sigma_j = 20\%$ (annualized)

In Table 3.1, we see the upper and lower bounds produced by the PO approach and the benchmark schemes described above. Here, we vary the number of assets n and the initial price $p_0^j = \bar{p}_0$ common to all assets. Standard errors are in parentheses. Similarly, Tables 3.2 and 3.3 show the upper and lower bounds computed as, respectively, the number of exercise opportunities d and the common asset price correlation $\rho_{jj'} = \bar{\rho}$ is varied. We make the following broad conclusions from these experimental results:

[5]Andersen and Broadie (2004) used 1,500 sample paths. We chose the larger number to obtain standard errors comparable to the other approaches in the study.

[6]Note that while all the parameter choices here are symmetric across assets, and hence the assets are identical in the problems we consider. However, this symmetry was not exploited in our implementations.

- **Lower Bound Quality.** The PO-LB method provides substantially better exercise policies than does the LS-LB procedure and consequently tighter lower bounds. The exercise policies provide an improvement of over 100 basis points in most of the experiments; in some cases the gain was as much as 200 basis points.

- **Upper Bound Quality.** The DVF-UB upper bounds are the weakest while the DP-UB upper bounds are typically the strongest. The gap between these two bounds was typically on the order of 100 basis points. The upper bound produced via the PO-UB method was of intermediate quality, but typically recovered approximately 60% of the gap between the DVF-UB and DP-UB upper bounds.

Table 3.4 summarizes relative computational requirements of each method. Note that, for the dual upper bound methods, we report the time to compute both upper and lower bounds. This is for consistency, since for the DVF-UB and DP-UB methods, the LS-LB continuation value estimate is required and must be computed first. The running times are typically dominated by sampling requirements, and can be broken down as follows:

- LS-LB: The LS-LB method requires only the generation of outer sample paths and is thus the fastest.

- LS-LB + DVF-UB: Along each outer sample path, the DVF-UB method requires generation of inner samples for the next state.

- PO-LB + PO-UB: For the PO-UB method, the structure of the LP (3.5) permits extremely efficient solution via an interior point method as discussed in Section 3.3.1; the computation time is dominated by sampling rather than optimization. Qualitatively, the sampling requirements for the PO-UB method are the same as that of DVF-UB: next state inner samples are needed. However, in order to generate an unbiased estimate, the PO-UB method requires one set of sample paths for optimization, and a second set of sample paths for evaluation of the upper bound estimate. Hence, PO-UB takes about twice the computational time of DVF-UB.

(a) Upper and lower bounds, with standard errors.

$\bar{p}_0$	LS-LB	S.E.	PO-LB	S.E.	DP-UB	S.E.	PO-UB	S.E.	DVF-UB	S.E.
					$n = 4$ assets					
90	32.754	(0.005)	33.011	(0.011)	34.989	(0.014)	35.117	(0.026)	35.251	(0.013)
100	40.797	(0.003)	41.541	(0.009)	43.587	(0.016)	43.853	(0.027)	44.017	(0.011)
110	46.929	(0.003)	48.169	(0.004)	49.909	(0.016)	50.184	(0.017)	50.479	(0.008)
					$n = 8$ assets					
90	43.223	(0.005)	44.113	(0.009)	45.847	(0.016)	46.157	(0.037)	46.311	(0.015)
100	49.090	(0.004)	50.252	(0.006)	51.814	(0.023)	52.053	(0.027)	52.406	(0.014)
110	52.519	(0.005)	53.488	(0.007)	54.890	(0.020)	55.064	(0.019)	55.513	(0.005)
					$n = 16$ assets					
90	49.887	(0.003)	50.885	(0.006)	52.316	(0.020)	52.541	(0.010)	52.850	(0.011)
100	52.879	(0.001)	53.638	(0.004)	54.883	(0.020)	55.094	(0.016)	55.450	(0.013)
110	54.620	(0.002)	55.146	(0.003)	56.201	(0.009)	56.421	(0.016)	56.752	(0.007)

(b) Relative values of bounds.

$\bar{p}_0$	$\frac{\text{(PO-LB)} - \text{(LS-LB)}}{\text{LS-LB}}$	$\frac{\text{(PO-UB)} - \text{(DP-UB)}}{\text{LS-LB}}$	$\frac{\text{(DVF-UB)} - \text{(PO-UB)}}{\text{LS-LB}}$
	$n = 4$ assets		
90	0.78%	0.39%	0.41%
100	1.82%	0.65%	0.40%
110	2.64%	0.59%	0.63%
	$n = 8$ assets		
90	2.06%	0.72%	0.36%
100	2.37%	0.49%	0.72%
110	1.85%	0.33%	0.86%
	$n = 16$ assets		
90	2.00%	0.45%	0.62%
100	1.43%	0.40%	0.67%
110	0.96%	0.40%	0.61%

Table 3.1: A comparison of the lower and upper bound estimates of the PO and benchmarking methods, as a function of the common initial asset price $p_0^j = \bar{p}_0$ and the number of assets n. For each algorithm, the mean and standard error (over 10 independent trials) is reported. The number of exercise opportunities was $d = 54$ and the common correlation was $\rho_{jj'} = \bar{\rho} = 0$.

(a) Upper and lower bounds, with standard errors.

n	LS-LB	S.E.	PO-LB	S.E.	DP-UB	S.E.	PO-UB	S.E.	DVF-UB	S.E.
$d = 36$ exercise opportunities										
4	40.315	(0.004)	41.073	(0.008)	42.723	(0.016)	43.006	(0.021)	43.199	(0.009)
8	48.283	(0.004)	49.114	(0.006)	50.425	(0.019)	50.721	(0.027)	51.011	(0.008)
16	51.835	(0.003)	52.289	(0.004)	53.231	(0.009)	53.517	(0.020)	53.741	(0.006)
$d = 54$ exercise opportunities										
4	40.797	(0.003)	41.541	(0.009)	43.587	(0.016)	43.853	(0.027)	44.017	(0.011)
8	49.090	(0.004)	50.252	(0.006)	51.814	(0.023)	52.053	(0.027)	52.406	(0.014)
16	52.879	(0.001)	53.638	(0.004)	54.883	(0.020)	55.094	(0.016)	55.450	(0.013)
$d = 81$ exercise opportunities										
4	41.229	(0.004)	41.644	(0.017)	44.264	(0.023)	44.511	(0.030)	44.662	(0.006)
8	49.788	(0.003)	51.249	(0.004)	52.978	(0.018)	53.178	(0.027)	53.523	(0.013)
16	53.699	(0.003)	54.825	(0.005)	56.398	(0.024)	56.464	(0.007)	56.948	(0.008)

(b) Relative values of bounds.

n	$\frac{\text{(PO-LB)} - \text{(LS-LB)}}{\text{LS-LB}}$	$\frac{\text{(PO-UB)} - \text{(DP-UB)}}{\text{LS-LB}}$	$\frac{\text{(DVF-UB)} - \text{(PO-UB)}}{\text{LS-LB}}$
$d = 36$ exercise opportunities			
4	1.88%	0.70%	0.48%
8	1.72%	0.61%	0.60%
16	0.88%	0.55%	0.43%
$d = 54$ exercise opportunities			
4	1.82%	0.65%	0.40%
8	2.37%	0.49%	0.72%
16	1.43%	0.40%	0.67%
$d = 81$ exercise opportunities			
4	1.01%	0.60%	0.37%
8	2.93%	0.40%	0.69%
16	2.10%	0.12%	0.90%

Table 3.2: A comparison of the lower and upper bound estimates of the PO and benchmarking methods, as a function of the number of exercise opportunities d and the number of assets n. For each algorithm, the mean and standard error (over 10 independent trials) is reported. The common initial asset price was $p_0^j = \bar{p}_0 = 100$ and the common correlation was $\rho_{jj'} = \bar{\rho} = 0$.

(a) Upper and lower bounds, with standard errors.

n	LS-LB	S.E.	PO-LB	S.E.	DP-UB	S.E.	PO-UB	S.E.	DVF-UB	S.E.
					$\bar{\rho} = -0.05$ correlation					
4	41.649	(0.004)	42.443	(0.009)	44.402	(0.023)	44.644	(0.019)	44.846	(0.013)
8	50.077	(0.005)	51.136	(0.005)	52.581	(0.031)	52.799	(0.018)	53.163	(0.011)
16	53.478	(0.004)	54.076	(0.004)	55.146	(0.013)	55.360	(0.010)	55.708	(0.010)
					$\bar{\rho} = 0$ correlation					
4	40.797	(0.003)	41.541	(0.009)	43.587	(0.016)	43.853	(0.027)	44.017	(0.011)
8	49.090	(0.004)	50.252	(0.006)	51.814	(0.023)	52.053	(0.027)	52.406	(0.014)
16	52.879	(0.001)	53.638	(0.004)	54.883	(0.020)	55.094	(0.016)	55.450	(0.013)
					$\bar{\rho} = 0.1$ correlation					
4	39.180	(0.006)	39.859	(0.011)	42.001	(0.037)	42.187	(0.029)	42.425	(0.010)
8	47.117	(0.005)	48.371	(0.005)	50.139	(0.029)	50.362	(0.035)	50.700	(0.014)
16	51.414	(0.005)	52.498	(0.008)	54.141	(0.032)	54.217	(0.018)	54.654	(0.010)

(b) Relative values of bounds.

n	$\frac{\text{(PO-LB)} - \text{(LS-LB)}}{\text{LS-LB}}$	$\frac{\text{(PO-UB)} - \text{(DP-UB)}}{\text{LS-LB}}$	$\frac{\text{(DVF-UB)} - \text{(PO-UB)}}{\text{LS-LB}}$
	$\bar{\rho} = -0.05$ correlation		
4	1.91%	0.58%	0.49%
8	2.11%	0.44%	0.73%
16	1.12%	0.40%	0.65%
	$\bar{\rho} = 0$ correlation		
4	1.82%	0.65%	0.40%
8	2.37%	0.49%	0.72%
16	1.43%	0.40%	0.67%
	$\bar{\rho} = 0.1$ correlation		
4	1.73%	0.48%	0.61%
8	2.66%	0.47%	0.72%
16	2.11%	0.15%	0.85%

Table 3.3: A comparison of the lower and upper bound estimates of the PO and benchmarking methods, as a function of the common correlation $\rho_{jj'} = \bar{\rho}$ and the number of assets n. For each algorithm, the mean and standard error (over 10 independent trials) is reported. The common initial price was $p_0^j = \bar{p}_0 = 100$ and the number of exercise opportunities was $d = 54$.

method	time (normalized)
LS-LB (lower bound only)	1.0
LS-LB + DVF-UB (upper and lower bounds)	3.6
PO-LB + PO-UB (upper and lower bounds)	6.8
LS-LB + DP-UB (upper and lower bounds)	51.7

Table 3.4: Relative time values for different algorithms for the stopping problem setting of Table 3.1 with $n = 16$ assets. Here, all times are normalized relative to that required for the computation of the LS-LB lower-bound . All computations were single-threaded and performed on an Intel Xeon E5620 2.40 GHz CPU with 64 GB RAM. The PO-UB linear program was solved with IBM ILOG CPLEX 12.1.0 optimization software.

- LS-LB + DP-UB: The inner simulation requirements for DP-UB, on the other hand, result in that method requiring an order of magnitude more time than either of the other upper bound approaches. This is because along each outer sample path, inner samples not just for one time step, but for an entire trajectory until the option is knocked-out or exercised.

To summarize, these experiments demonstrate the two primary merits to using the PO method to produce upper and lower bounds:

1. The PO-UB method produces upper bounds that are superior to the DVF-UB method, and, in many cases, of comparable quality to the state-of-the-art DP-UB method. However, the PO-UB method requires an order of magnitude less computational effort than the DP-UB approach, and is highly practical.

2. The PO-LB method produces substantially superior exercise policies relative to the LS-LB method. These policies are effectively a by-product of the upper bound computation.

3.5. Theory

In this section, we will seek to provide theoretical guarantees for the martingale penalty approach in general as well as specific guarantees for the PO method.

Note that our setting here will be that of an optimal stopping problem that is discounted, stationary, and has an infinite horizon. This will yield us considerably simpler notation and easier statement of results, and is also consistent with other theoretical literature on ADP for optimal stopping problems (e.g., Tsitsiklis and Van Roy, 1999; Van Roy, 2010). Many of our results have finite horizon, non-stationary analogues, however, and we view intuition derived from the stationary setting as carrying over to the non-stationary setting. Our stationary setting is introduced in Section 3.5.1.

Our first class of theoretical results are *approximation guarantees*. These guarantee the quality of an upper bound derived from the martingale duality approach, relative to error in approximating the value function. A crucial parameter for our bounds measures the 'predictability' of a Markov chain; this is introduced in Section 3.5.2. In Section 3.5.3, we develop an approximation guarantee that applies generically to martingale duality upper bounds, and discuss the structural properties of optimal stopping problems that impact this bound. In Section 3.5.4, we develop a *relative* guarantee that is specific to the PO method; this guarantees the quality of the PO upper bound relative to the best approximation of the true value function within the span of the basis functions. In Section 3.5.5, we compare our guarantees to similar guarantees that have been developed for ADP lower bounds.

Our second class of theoretical results are *comparison bounds*, developed in Section 3.5.6. Here, we compare the upper bounds arising to the PO approach to other upper bounds which have been developed using ADP techniques based in linear programming. In this case, the upper bounds can be compared on a problem instance by problem instance basis, and we show that the PO method dominates the alternatives.

3.5.1. Preliminaries

Consider a discrete-time Markov chain with state $x_t \in \mathcal{X}$ at each time $t \in \{0, 1, \ldots\}$. Assume the chain has a the state space $\mathcal{X}$ that is finite. Denote by P the transition kernel of the chain. Assume that the chain is ergodic (i.e., aperiodic and irreducible), with stationary distribution π. Without loss of generality, assume that $\pi(x) > 0$ for every state x. Let

$\mathcal{F} \triangleq \{\mathcal{F}_t\}$ be the natural filtration generated by the process $\{x_t\}$, i.e., for each time t, $\mathcal{F}_t \triangleq \sigma(x_0, x_1, \ldots, x_t)$.

Given a function $g\colon \mathcal{X} \to \mathbb{R}$, we define the payoff of stopping when the state is x_t as $g(x_t)$. We are interested in maximizing the expected discounted payoff of stopping. In particular, given an initial state $x \in \mathcal{X}$, define the optimal value function

$$J^*(x) \triangleq \sup_{\tau} \ \mathsf{E}\Big[\alpha^\tau g(x_\tau) \mid x_0 = x\Big].$$

Here, the supremum is taken over all $\mathcal{F}$-adapted stopping times τ, and $\alpha \in [0, 1)$ is the discount factor.

We will define $\mathcal{P}$ to be the set of real-valued functions of the state space,[7] i.e., if $J \in \mathcal{P}$, then $J\colon \mathcal{X} \to \mathbb{R}$. We will abuse notation to also consider the transition kernel as a one-step expectation operator $P\colon \mathcal{P} \to \mathcal{P}$, defined by

$$(PJ)(x) \triangleq \mathsf{E}\Big[J(x_{t+1}) \mid x_t = x\Big], \quad \forall\ x \in \mathcal{X}.$$

Given a function $J \in \mathcal{P}$, define the *Bellman operator* $T\colon \mathcal{P} \to \mathcal{P}$ by

$$(TJ)(x) \triangleq \max\ \Big\{g(x), \alpha PJ(x)\Big\}, \quad \forall\ x \in \mathcal{X}.$$

Observe that the optimal value function is the unique fixed point $TJ^* = J^*$.

In order to define the pathwise optimization approach in this setting, we first define the *martingale difference operator* Δ. The operator Δ maps a function $J \in \mathcal{P}$ to a function $\Delta J\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, where

$$\Delta J(x_t, x_{t-1}) \triangleq J(x_t) - PJ(x_{t-1}), \quad \forall\ x_{t-1}, x_t \in \mathcal{X}.$$

Observe that, for any J, the process $\{\Delta J(x_t, x_{t-1}),\ t \geq 1\}$ is a martingale difference sequence. Now, for each J, the *martingale duality upper bound operator* $F\colon \mathcal{P} \to \mathcal{P}$ is given by

$$(FJ)(x) \triangleq \mathsf{E}\left[\sup_{s \geq 0}\ \alpha^s g(x_s) - \sum_{t=1}^{s} \alpha^t \Delta J(x_t, x_{t-1}) \ \middle|\ x_0 = x\right], \quad \forall\ x \in \mathcal{X}.$$

The following lemma establishes that the the F operator yields dual upper bounds to the original problem, the proof follows along the lines of Lemma 8 and is omitted:

[7] Note that earlier we defined $\mathcal{P}$ to be the set of real-valued functions of state and time. In the stationary infinite horizon setting, it suffices to consider only functions of state.

Lemma 10 (Infinite Horizon Martingale Duality).

(i) (Weak Duality) For any function $J \in \mathcal{P}$ and all $x \in \mathcal{X}$, $J^*(x) \leq FJ(x)$.

(ii) (Strong Duality) For all $x \in \mathcal{X}$, $J^*(x) = FJ^*(x)$.

In order to find a good upper bound, we begin with collection of K basis functions

$$\Phi \triangleq \{\phi_1, \phi_2, \ldots, \phi_K\} \subset \mathcal{P}.$$

Given a weight vector $r \in \mathbb{R}^K$, define the function $\Phi r \in \mathcal{P}$ as the linear combination

$$(\Phi r)(x) \triangleq \sum_{\ell=1}^{k} r_\ell \phi_\ell(x), \quad \forall\, x \in \mathcal{X}.$$

We will seek to find functions within the span of the basis Φ which yields the tightest *average* upper bound. In other words, we will see to solve the optimization problem

$$\underset{r}{\text{minimize}} \;\; \mathsf{E}_\pi\Big[F\Phi r(x_0)\Big]. \tag{3.9}$$

As before, this optimization problem is an unconstrained minimization of a convex function.

3.5.2. Predictability

Our approximation guarantees incorporate a notion of predictability of the underlying Markov chain, which we will define in this section. First, we begin with some notation. For functions $J, J' \in \mathcal{P}$, define the inner product

$$\langle J, J' \rangle_\pi \triangleq \mathsf{E}_\pi \left[J(x_0) J'(x_0) \right].$$

Here, E_π denotes that the expectation is taken with x_0 distributed according to the stationary distribution π. Similarly, define the norms

$$\|J\|_{p,\pi} \triangleq \left(\mathsf{E}_\pi \Big[|J(x_0)|^p \Big] \right)^{1/p}, \quad \forall\, p \in \{1, 2\}, \qquad \|J\|_\infty \triangleq \sup_{x \in \mathcal{X}} |J(x)|,$$

and define $\mathrm{Var}_\pi(J)$ to be the variance of $J(x)$ under the distribution π, i.e.,

$$\mathrm{Var}_\pi(J) \triangleq \mathsf{E}_\pi \left[\Big(J(x_0) - \mathsf{E}_\pi \left[J(x_0) \right] \Big)^2 \right].$$

Now, recall that P is the transition kernel of the Markov chain, that we also interpret as a one-step expectation operator. Define P^* to be the adjoint of P with respect to the inner product $\langle \cdot, \cdot \rangle_\pi$. P^* can be written explicitly according to

$$P^*(y,x) \triangleq \frac{\pi(x)P(x,y)}{\pi(y)}, \quad \forall\ x, y \in \mathcal{X}.$$

Note that P^* is the *time-reversal* of P; it corresponds to the transition kernel of the Markov chain running backwards in time.

The following quantity will be important for our analysis:

$$\lambda(P) \triangleq \sqrt{\rho(I - P^*P)}.$$

Here, $\rho(\cdot)$ is the spectral radius. We make the following elementary observations regarding $\lambda(P)$, the proof of which is deferred until Appendix 3.6:

Lemma 11.

(i) $0 \leq \lambda(P) \leq 1$.

(ii) If P is reversible, i.e., if $P = P^*$, then

$$\lambda(P) = \sqrt{\rho(I - P^2)} \leq \sqrt{2\rho(I - P)}.$$

In order to interpret $\lambda(P)$, first note that Part (i) of Lemma 11 guarantees that this quantity is bounded. Now, observe that the matrix P^*P, known as a *multiplicative reversiblization* (Fill, 1991), is also a stochastic matrix, corresponding to a transition one step backward in time in the original Markov chain, followed by an independent step forward in time. Suppose for the moment that the Markov chain is reversible, i.e., that $P = P^*$. Then, by Part (ii) of Lemma 11, $\lambda(P)$ will be small when $I \approx P$, or, the state x_{t+1} at time $t+1$ in the Markov chain is approximated well by the current state x_t. In other words, the Markov chain is closer to a deterministic process. Motivated by this intuition, we will call Markov chains where $\lambda(P) \approx 0$ *predictable*.[8]

[8]The spectral analysis of $I - P^*P$ is also important in the study of mixing times, or, the rate of convergence of a Markov chain to stationarity. In that context, one is typically concerned with the *smallest* non-zero eigenvalue (see, e.g., Montenegro and Tetali, 2006); informally, if this is large, the chain is said to be *fast mixing*. In the present context, we are interested in the *largest* eigenvalue, which is small in the case of a predictable chain. Thus, our predictable chains necessarily mix slowly.

Predictability is important because it provides a bound on the operator norm of the martingale difference operator Δ. When a Markov chain is predictable, it may be possible to approximate a particular martingale difference, say ΔJ^*, by some other martingale difference, say ΔJ, even if J^* is not particularly well approximated by J. This is captured in the following lemma:

Lemma 12. *Given a functions $J, J' \in \mathcal{P}$, define a distance between the martingale differences ΔJ, $\Delta J'$ by*

$$\|\Delta J - \Delta J'\|_{2,\pi} \triangleq \sqrt{\mathsf{E}_\pi\left[|\Delta J(x_1, x_0) - \Delta J'(x_1, x_0)|^2\right]}.$$

Then,

$$\|\Delta J - \Delta J'\|_{2,\pi} \leq \lambda(P)\ \sqrt{\mathrm{Var}_\pi(J - J')}.$$

Proof. Set $W \triangleq J - J'$, and observe that since π is an invariant distribution,

$$\begin{aligned}
\|\Delta W\|_{2,\pi}^2 &= \mathsf{E}_\pi\left[\Big(W(x_1) - \mathsf{E}[W(x_1)|x_0]\Big)^2\right] = \mathsf{E}_\pi\left[W(x_1)^2 - \Big(\mathsf{E}[W(x_1)|x_0]\Big)^2\right] \\
&= \mathsf{E}_\pi\left[W(x_0)^2 - \Big(\mathsf{E}[W(x_1)|x_0]\Big)^2\right] = \langle W, W\rangle_\pi - \langle PW, PW\rangle_\pi \\
&= \langle W, W\rangle_\pi - \langle W, P^*PW\rangle_\pi = \langle W, (I - P^*P)W\rangle_\pi \leq \rho(I - P^*P)\, \|W\|_{2,\pi}^2.
\end{aligned}$$

Now, note that the operator Δ is invariant to constant shifts, i.e., $\Delta(W + \gamma\mathbf{1}) = \Delta W$, where γ is a scalar and $\mathbf{1} \in \mathcal{P}$ is the constant function evaluating to 1 in every state. Then, define $\mu_W \triangleq \mathsf{E}_\pi[W(x_0)]$ to be the mean of W. We have that

$$\|\Delta W\|_{2,\pi}^2 = \Big\|\Delta\Big(W - \mu_W\mathbf{1}\Big)\Big\|_{2,\pi}^2 \leq \rho(I - P^*P)\ \|W - \mu_W\mathbf{1}\|_{2,\pi}^2 = \rho(I - P^*P)\ \mathrm{Var}_\pi(W).$$

The result follows. ■

One class of predictable Markov chains occurs when the calendar time scale between successive stopping opportunities is small:

Example 1 (Sampled State Dynamics). *Suppose that the Markov chain $\{x_t\}$ takes the form $x_t = z_{t\delta}$ for all integers $t \geq 0$, where $\delta > 0$ and $\{z_s \in \mathcal{X},\ s \in \mathbb{R}_+\}$ is a continuous time Markov chain with generator Q. In other words, $\{x_t\}$ are discrete time samples of an*

underlying continuous time chain over time scales of length δ. In this case, the transition probabilities take the form $P = e^{Q\delta}$ and $P^* = e^{Q^*\delta}$. As $\delta \to 0$,

$$\lambda(P) = \sqrt{\rho\left(I - e^{Q^*\delta}e^{Q\delta}\right)} = \sqrt{\delta\rho(Q^* + Q)} + o\left(\sqrt{\delta}\right) \to 0.$$

3.5.3. Upper Bound Guarantees

Lemma 10 establishes that, given a function $J \in \mathcal{P}$, FJ is an upper bound on J^*, and that if $J = J^*$, this upper bound is tight. Hence, it seems reasonable to pick J to be a good approximation of the optimal value function J^*. In this section, we seek to make this intuition precise. In particular, we will provide a guarantee on the quality of the upper bound, that is, a bound on the distance between FJ and J^*, as a function of the quality of the value function approximation J and other structural features of the optimal stopping problem.

The following lemma is provides the key result for our guarantee. It characterizes the difference between two upper bounds FJ and FJ' that arise from two different value function approximations $J, J' \in \mathcal{P}$. The proof is deferred until Appendix 3.6.

Lemma 13. *For any pair of functions $J, J' \in \mathcal{P}$,*

$$\|FJ - FJ'\|_{2,\pi} \leq \frac{R(\alpha)\alpha}{\sqrt{1-\alpha}}\,\lambda(P)\,\sqrt{\mathrm{Var}_\pi(J - J')},$$

where $R\colon [0,1) \to \left[1, \sqrt{5/2}\right]$ is a bounded function given by

$$R(\alpha) \triangleq \min\left\{\frac{1}{\sqrt{1-\alpha}}, \frac{2}{\sqrt{1+\alpha}}\right\}.$$

Taking $J' = J^*$ in Lemma 13, we immediately have the following:

Theorem 5. *For any function $J \in \mathcal{P}$,*

$$\|FJ - J^*\|_{2,\pi} \leq \frac{R(\alpha)\alpha}{\sqrt{1-\alpha}}\,\lambda(P)\,\sqrt{\mathrm{Var}_\pi(J - J^*)}.$$

Theorem 5 provides a guarantee on the upper bound FJ arising from an arbitrary function J. It is reminiscent of the upper bound guarantee of Chen and Glasserman (2007). In the present (discounted and infinite horizon) context, their upper bound guarantee can be stated as

$$\|FJ - J^*\|_\infty \leq \frac{4\alpha}{\sqrt{1-\alpha^2}} \|J - J^*\|_\infty. \tag{3.10}$$

It what follows, we will compare these two bounds, as well identify the structural features of the optimal stopping problem and the function J that lead to a tight upper bound FJ. In particular, notice that the right-hand side of the guarantee in Theorem 5 can be decomposed into three distinct components:

- **Value Function Approximation Quality.** Theorem 5 guarantees that the closer the value function approximation J is to J^*, the tighter the upper bound FJ will be. Importantly, the distance between J and J^* is measured in terms of the standard deviation of their difference. Under this metric, the relative importance of accurately approximating J^* in two different states is commensurate to their relative probabilities. On the other hand, the guarantee (3.10) requires a *uniformly* good approximation of J^*. In a large state space, this can be challenging.

- **Time Horizon.** Theorem 5 has dependence on the discount factor α. In typical examples, $\alpha \approx 1$, and hence we are most interested in this regime.

  One way to interpret α is as defining an effective time horizon. To be precise, consider an undiscounted stopping problem with the same state dynamics and reward function, but with a random finite horizon that is geometrically distributed with parameter α. We assume that the random time horizon is unknown to the decision maker, and that if the process is not stopped before the end of this time horizon, the reward is zero. This undiscounted, random but finite horizon formulation is mathematically equivalent to our discounted, infinite horizon problem. Hence, we define the *effective time horizon* $\mathsf{T}_{\mathsf{eff}}$ to be the expected length of the random finite time horizon, or

  $$\mathsf{T}_{\mathsf{eff}} \triangleq \frac{1}{1-\alpha}.$$

The guarantee of Theorem 5 is $O(\sqrt{\mathsf{T}_{\mathsf{eff}}})$, i.e., it grows as the square root of the effective time horizon. This matches (3.10), as well as the original finite horizon bound of Chen and Glasserman (2007).

- **Predictability.** Theorem 5 isolates the dynamics of the Markov chain through the $\lambda(P)$ term; if $\lambda(P)$ is small, then the upper bound FJ will be tight. In other words, all else being equal, chains that are more predictable yield better upper bounds. In some sense, optimal stopping problems on predictable Markov chains are closer to deterministic problems to begin with, hence less care is needed in relaxing non-anticipativity constraints.

  The dependence of Theorem 5 on predictability can be interpreted in the sampled state dynamics of Example 1. In this case, we assume that the transition probabilities of the Markov chain take the form $P = e^{Q\delta}$, where Q is the generator for a continuous time Markov chain and $\delta > 0$ is the calender time between successive stopping opportunities. In this setting, it is natural that the discount factor also scale as a function of the time interval δ, taking the form $\alpha = e^{-r\delta}$, where $r > 0$ is a continuously compounded interest rate. Then, as $\delta \to 0$,

$$\frac{R(\alpha)\alpha}{\sqrt{1-\alpha}}\,\lambda(P) = \sqrt{\frac{2\rho(Q^* + Q)}{r}} + o(1).$$

  In this way, the pre-multiplying constants on the right-hand side of Theorem 5 remain bounded as the number of stopping opportunities is increased. This is *not* the case for (3.10).

3.5.4. Pathwise Optimization Approximation Guarantee

The result of Section 3.5.3 provides a guarantee on the upper bounds produced by the martingale duality approach given an arbitrary value function approximation J as input. When the value function approximation J arises from the PO method, we have the following result:

Theorem 6. *Suppose that* r_{PO} *is an optimal solution for* (3.9)*. Then,*

$$\|F\Phi r_{\mathsf{PO}} - J^*\|_{1,\pi} \leq \frac{R(\alpha)\alpha}{\sqrt{1-\alpha}}\,\lambda(P)\,\min_r\,\sqrt{\mathrm{Var}_\pi(\Phi r - J^*)}.$$

Proof. Observe that, for any $r \in \mathbb{R}^K$, by the optimality of r_{PO} and Lemma 10,

$$\|F\Phi r_{\mathsf{PO}} - J^*\|_{1,\pi} = \mathsf{E}_\pi\left[F\Phi r_{\mathsf{PO}}(x_0) - J^*(x_0)\right] \leq \mathsf{E}_\pi\left[F\Phi r(x_0) - J^*(x_0)\right] = \|F\Phi r - J^*\|_{1,\pi}.$$

Since π is a probability distribution, $\|\cdot\|_{1,\pi} \leq \|\cdot\|_{2,\pi}$, thus, applying Theorem 5,

$$\|F\Phi r_{\mathsf{PO}} - J^*\|_{1,\pi} \leq \|F\Phi r - J^*\|_{2,\pi} \leq \frac{R(\alpha)\alpha}{\sqrt{1-\alpha}}\,\lambda(P)\,\sqrt{\mathrm{Var}_\pi(\Phi r - J^*)}.$$

The result follows after minimizing the right-hand side over r. ■

In order to compare Theorems 5 and 6, observe that Theorem 5 provides a guarantee that is a function of the distance between the value function approximation J and the optimal value function J^*. Theorem 6, on the other hand, provides a guarantee relative to the distance between the *best possible* approximation given the basis functions Φ and the optimal value function J^*. Note that it is not possible, in general, to directly compute this best approximation, which is the projection of J^* on to the subspace spanned by Φ, since J^* is unknown to begin with.

3.5.5. Comparison to Lower Bound Guarantees

It is instructive to compare the guarantees provided on upper bounds by Theorems 5 and 6 to guarantees that can be obtained on lower bounds derived from ADP methods. In general, the ADP approach to lower bounds involve identifying approximations to the optimal continuation value function C^*, which is related to the optimal value function J^* via

$$C^*(x) = \alpha\mathsf{E}[J^*(x_{t+1}) \mid x_t = x], \quad J^*(x) = \max\{g(x), C^*(x)\}, \quad \forall\, x \in \mathcal{X}.$$

Given the optimal continuation function C^*, an optimal policy is defined via

$$\mu^*(x) \triangleq \begin{cases} \mathsf{CONTINUE} & \text{if } g(x) < C^*(x), \\ \mathsf{STOP} & \text{otherwise.} \end{cases}$$

In other words, μ^* stops when $g(x) \geq C^*(x)$.

Similarly, given an approximate continuation value function C, we can define the policy

$$\mu(x) \triangleq \begin{cases} \mathsf{CONTINUE} & \text{if } g(x) < C(x), \\ \mathsf{STOP} & \text{otherwise.} \end{cases}$$

The value function J^μ for this policy can be estimated via Monte Carlo simulation. Since J^* is the optimal value function, we have that $J^\mu(x) \leq J^*(x)$ for every state x. In other words, J^μ is a lower bound to J^*.

Analogous to Theorem 5, Tsitsiklis and Van Roy (1999) establish that

$$\|J^* - J_\mu\|_{2,\pi} \leq \frac{1}{1-\alpha} \|C - C^*\|_{2,\pi}. \tag{3.11}$$

Given a set of basis functions Φ, there are a number of ways to select a weight vector r so that the linear function Φr can be used as an approximate continuation value function. Methods based on approximate value iteration are distinguished by the availability of theoretical guarantees. Indeed, Van Roy (2010) establishes a result analogous to Theorem 6 for approximate value iteration, that

$$\|J^* - J_\mu\|_{1,\pi} \leq \|J^* - J_\mu\|_{2,\pi} \leq \frac{L^*}{1-\alpha} \min_r \ \|\Phi r - C^*\|_{2,\pi}\,, \tag{3.12}$$

where $L^* \approx 2.17$.

Comparing (3.11)–(3.12) to Theorems 5 and 6, we see broad similarities: both sets of results provide guarantees on the quality of the lower (resp., upper) bounds produced, as a function of the quality of approximation of C^* (resp., J^*). There are key differences, however. Defining the effective time horizon $\mathsf{T}_{\mathsf{eff}} \triangleq (1-\alpha)^{-1}$ as in Section 3.5.3, the pre-multiplying constants in the lower bound guarantees are $O(\mathsf{T}_{\mathsf{eff}})$, while the corresponding terms in our upper bound guarantees are $O(\sqrt{\mathsf{T}_{\mathsf{eff}}})$. Further, Van Roy (2010) establishes that, for *any* ADP algorithm, a guarantee of the form (3.12) that applies over all problem instances must be linear in the effective time horizon. In this way, the upper bound guarantees of Theorems 5 and 6 have better dependence on the effective time horizon than is possible for lower bounds, independent of the choice of ADP algorithm. Further, the upper bound guarantees highlight

the importance of a structural property of the Markov chain, namely, predictability. There is no analogous term in the lower bound guarantees.

3.5.6. Comparison to Linear Programming Methods

We can compare upper bounds derived from the pathwise method directly to upper bounds derived from two other approximate dynamic programming techniques.

First, we consider the *approximate linear programming* (ALP) approach. In our discounted, infinite horizon optimal stopping setting, the ALP approach involves finding a value function approximation within the span of the basis by solving the optimization program

$$\begin{array}{lll} \underset{r}{\text{minimize}} & \mathsf{E}_c\Big[\Phi r(x_0)\Big] & \\ \text{subject to} & \Phi r(x) \geq g(x), & \forall\, x \in \mathcal{X}, \\ & \Phi r(x) \geq \alpha \mathsf{E}\left[\Phi r(x_{t+1}) \mid x_t = x\right], & \forall\, x \in \mathcal{X}. \end{array} \tag{3.13}$$

Here, c is a positive probability distribution over the state space know as the *state-relevance* distribution, it is natural (but not necessary) to take $c = \pi$. Note that (3.13) is a linear program, and that, for each state x, the pair of linear constraints in (3.13) are equivalent to the Bellman inequality $\Phi r(x) \geq T\Phi r(x)$. Denote the set of feasible r by $\mathcal{C}_{\mathsf{ALP}} \subset \mathbb{R}^K$.

As we shall see momentarily, if $r \in \mathcal{C}_{\mathsf{ALP}}$ is feasible for ALP (3.13), then Φr is a pointwise upper bound to the optimal value function J^*. The following theorem establishes that the martingale duality upper bound $F\Phi r$ is at least as good:

Theorem 7. *Suppose $r \in \mathcal{C}_{\mathsf{ALP}}$ is feasible for the ALP* (3.13)*. Then, for all $x \in \mathcal{X}$,*

$$J^*(x) \leq F\Phi r(x) \leq \Phi r(x).$$

Proof. Using Lemma 10 and the definition of the constraint set $\mathcal{C}_{\mathsf{ALP}}$,

$$\begin{aligned} J^*(x) \leq F\Phi r(x) &= \mathsf{E}\left[\sup_{s\geq 0}\ \alpha^s g(x_s) - \sum_{t=1}^{s} \alpha^t \Big(\Phi r(x_t) - \mathsf{E}[\Phi r(x_t) \mid x_{t-1}]\Big) \;\middle|\; x_0 = x\right] \\ &= \mathsf{E}\left[\sup_{s\geq 0}\ \alpha^s \Big(g(x_s) - \Phi r(x_s)\Big) + \Phi r(x_0) + \sum_{t=0}^{s-1} \alpha^t \Big(\alpha \mathsf{E}[\Phi r(x_{t+1}) \mid x_t] - \Phi r(x_t)\Big) \;\middle|\; x_0 = x\right] \\ &= \Phi r(x). \end{aligned}$$

■

We can interpret the ALP (3.13) as finding an upper bound in the set $\{\Phi r,\ r \in \mathcal{C}_{\mathsf{ALP}}\}$ that is smallest on average, as measured according to the state-relevance distribution c. Alternatively, consider solving the pathwise optimization problem

$$\underset{r}{\text{minimize}}\ \mathsf{E}_c\left[F\Phi r(x_0)\right]. \tag{3.14}$$

Theorem 7 implies that the resulting martingale duality upper bound will be, on average, at least as good. In this way, the PO method dominates ALP.

Similarly, the *smoothed approximate linear programming* (SALP) has been recently introduced by Desai et al. (2009). In our present context, this seeks to solve the linear program

$$\begin{array}{lll} \underset{r,s}{\text{minimize}} & \mathsf{E}_\pi\left[\Phi r(x_0) + \dfrac{1}{1-\alpha}s(x_0)\right] & \\ \text{subject to} & \Phi r(x) + s(x) \geq g(x), & \forall\ x \in \mathcal{X}, \\ & \Phi r(x) + s(x) \geq \alpha \mathsf{E}\left[\Phi r(x_{t+1}) \mid x_t = x\right], & \forall\ x \in \mathcal{X}, \\ & s(x) \geq 0, & \forall\ x \in \mathcal{X}. \end{array} \tag{3.15}$$

Observe that (3.15) is a relaxation of (3.13) when $c = \pi$, that is formed by introducing a vector of slack variables $s \in \mathbb{R}^{\mathcal{X}}$. Desai et al. (2009) argue that this relaxation yields a number of theoretical benefits relative to the ALP, and demonstrate superior practical performance in a computational study.

The following lemma allows us to interpret the SALP as an unconstrained convex minimization problem:

Lemma 14. *Given $J \in \mathcal{P}$, define the operator $F_{\mathsf{SALP}} \colon \mathcal{P} \to \mathcal{P}$ by*

$$(F_{\mathsf{SALP}}J)(x) \triangleq \mathsf{E}\left[J(x_0) + \sum_{t=0}^{\infty}\alpha^t\Big(TJ(x_t) - J(x_t)\Big)^+ \;\middle|\; x_0 = x\right], \quad \forall\ x \in \mathcal{X}.$$

Then, the SALP (3.15) *is equivalent to the convex optimization problem*

$$\underset{r}{\text{minimize}}\ \mathsf{E}_\pi\Big[F_{\mathsf{SALP}}\Phi r(x_0)\Big]. \tag{3.16}$$

Proof. Suppose (r, s) is feasible for the SALP (3.15). Then,

$$\begin{aligned}\mathsf{E}_\pi\left[\Phi r(x_0) + \frac{1}{1-\alpha}s(x_0)\right] &\geq \mathsf{E}_\pi\left[\Phi r(x_0) + \frac{1}{1-\alpha}\Big(T\Phi r(x_0) - \Phi r(x_0)\Big)^+\right] \\ &= \mathsf{E}_\pi\left[\Phi r(x_0) + \sum_{t=0}^{\infty}\alpha^t\Big(T\Phi r(x_t) - \Phi r(x_t)\Big)^+\right] \\ &= \mathsf{E}_\pi\Big[F_{\mathsf{SALP}}\Phi r(x_0)\Big],\end{aligned} \tag{3.17}$$

where we use the constraints of (3.15) and the fact that π is the stationary distribution. Hence, r achieves at least the same objective value in (3.16). Conversely, for any r, define $s \triangleq (T\Phi r - \Phi r)^+$ component-wise. Then, (r, s) is feasible for (3.15), and (3.17) holds with equality. Thus, (r, s) achieves same objective value in (3.15) as r in (3.16). ■

The following theorem shows that the F_{SALP} operator also yields dual upper bounds to the optimal value function, analogous to the F operator in the pathwise method. Critically, however, the upper bounds of the pathwise method pointwise dominate that of the SALP, which in turn pointwise dominate that of the ALP.

Theorem 8. *For an arbitrary weight vector $r \in \mathbb{R}^K$,*

$$J^*(x) \leq F\Phi r(x) \leq F_{\mathsf{SALP}}\Phi r(x), \quad \forall\, x \in \mathcal{X}.$$

In addition, if $r \in \mathcal{C}_{\mathsf{ALP}}$, i.e., r is feasible for the ALP (3.13), then

$$J^*(x) \leq F\Phi r(x) \leq F_{\mathsf{SALP}}\Phi r(x) \leq \Phi r(x), \quad \forall\, x \in \mathcal{X}.$$

Proof. Given a weight vector $r \in \mathbb{R}^K$, by Lemma 10,

$$\begin{aligned}J^*(x) \leq F\Phi r(x) &= \mathsf{E}\left[\sup_{s\geq 0}\ \alpha^s g(x_s) - \sum_{t=1}^{s}\alpha^t\Big(\Phi r(x_t) - \mathsf{E}[\Phi r(x_t) \mid x_{t-1}]\Big)\ \middle|\ x_0 = x\right] \\ &= \mathsf{E}\left[\sup_{s\geq 0}\ \alpha^s\Big(g(x_s) - \Phi r(x_s)\Big) + \Phi r(x_0) + \sum_{t=0}^{s-1}\alpha^t\Big(\alpha\mathsf{E}[\Phi r(x_{t+1}) \mid x_t] - \Phi r(x_t)\Big)\ \middle|\ x_0 = x\right] \\ &\leq \mathsf{E}\left[\sup_{s\geq 0}\ \alpha^s\Big(g(x_s) - \Phi r(x_s)\Big)^+ + \Phi r(x_0) + \sum_{t=0}^{s-1}\alpha^t\Big(\alpha\mathsf{E}[\Phi r(x_{t+1}) \mid x_t] - \Phi r(x_t)\Big)^+\ \middle|\ x_0 = x\right] \\ &\leq \mathsf{E}\left[\sup_{s\geq 0}\ \Phi r(x_0) + \sum_{t=0}^{s}\alpha^t\Big(T\Phi r(x_t) - \Phi r(x_t)\Big)^+\ \middle|\ x_0 = x\right] = F_{\mathsf{SALP}}\Phi r(x),\end{aligned}$$

which completes the first part of the result. If $r \in \mathcal{C}_{\mathsf{ALP}}$, it immediately follows that $F_{\mathsf{SALP}}\Phi r(x) \leq \Phi r(x)$. ■

In the context of the ALP and SALP optimization problems (3.13) and (3.15), Theorem 8 yields that that

$$\underset{r}{\text{minimize}}\ \mathsf{E}_\pi\Big[F\Phi r(x_0)\Big] \le \underset{r}{\text{minimize}}\ \mathsf{E}_\pi\Big[F_{\mathsf{SALP}}\Phi r(x_0)\Big] \le \underset{r\in\mathcal{C}_{\mathsf{ALP}}}{\text{minimize}}\ \mathsf{E}_\pi\Big[\Phi r(x_0)\Big].$$

In other words, given a fixed set of basis functions, the PO method yields an upper bound that is on average at least as tight as that of the SALP method, which in turn yields an upper bound that is on average at least as tight at that of the ALP method.

3.6. Proofs

Lemma 8 (Martingale Duality).

(i) (Weak Duality) For any $J \in \mathcal{P}$ and all $x \in \mathcal{X}$ and $t \in \mathcal{T}$, $J_t^*(x) \le F_t J(x)$.

(ii) (Strong Duality) For all $x \in \mathcal{X}$ and $t \in \mathcal{T}$, $J^*(x)_t = F_t J^*(x)$.

Proof. (i) Note that

$$J_t^*(x_t) = \sup_{\tau_t}\ \mathsf{E}\left[\alpha^{\tau_t - t} g(x_{\tau_t}) \;\middle|\; x_t\right] \tag{3.18}$$

$$= \sup_{\tau_t}\ \mathsf{E}\left[\alpha^{\tau_t - t} g(x_{\tau_t}) - \sum_{p=t+1}^{\tau_t} \alpha^{p-t}(\Delta J)(x_p, x_{p-1}) \;\middle|\; x_t\right] \tag{3.19}$$

$$\le \mathsf{E}\left[\max_{t\le s\le d}\ \alpha^{s-t} g(x_s) - \sum_{p=t+1}^{s} \alpha^{p-t}(\Delta J)(x_p, x_{p-1}) \;\middle|\; x_t\right]. \tag{3.20}$$

Here, in (3.18), τ_t is a stopping time that takes values in the set $\{t, t+1, \ldots, d\}$. (3.19) follows from the optimal sampling theorem for martingales. (3.20) follows from the fact that stopping times are non-anticipatory, and hence the objective value can only be increased by allowing policies with access to the entire sample path.

(ii) From (i) we know that $F_t J^*(x_t) \geq J_t^*(x_t)$. To see the opposite inequality,

$$\begin{aligned}
F_t J^*(x_t) &= \mathsf{E}\left[\max_{t\leq s\leq d}\ \alpha^{s-t}g(x_s) - \sum_{p=t+1}^{s}\alpha^{p-t}\left(\Delta J^*\right)(x_p, x_{p-1}) \;\middle|\; x_t\right] \\
&= \mathsf{E}\left[\max_{t\leq s\leq d}\ \alpha^{s-t}g(x_s) - \sum_{p=t+1}^{s}\alpha^{p-t}\left(J_p^*(x_p) - \mathsf{E}[J_p^*(x_p)|x_{p-1}]\right) \;\middle|\; x_t\right] \\
&= \mathsf{E}\left[\max_{t\leq s\leq d}\ \alpha^{s-t}g(x_s) - \alpha^{s-t}J_s^*(x_s) + J_t^*(x_t)\right. \\
&\qquad\qquad \left. + \sum_{p=t+1}^{s}\alpha^{p-t-1}\left(\alpha\mathsf{E}[J_p^*(x_p)|x_{p-1}] - J_{p-1}^*(x_{p-1})\right) \;\middle|\; x_t\right] \\
&\leq J_t^*(x_t)
\end{aligned}$$

The last inequality follows from the Bellman equation (3.1). ■

Lemma 11.

(i) $0 \leq \lambda(P) \leq 1$.

(ii) If P is reversible, i.e., if $P = P^*$, then

$$\lambda(P) = \sqrt{\rho(I - P^2)} \leq \sqrt{2\rho(I - P)}.$$

Proof. (i) Observe that $I - P^*P$ is self-adjoint, and hence must have real eigenvalues. Let $\sigma_{\min}$ and $\sigma_{\max}$ be the smallest and largest eigenvalues, respectively. By the Courant-Fischer variational characterization of eigenvalues,

$$\begin{aligned}
\sigma_{\max} &= \sup_{J\in\mathcal{P},\ \|J\|_{2,\pi}=1} \langle J, (I - P^*P)J\rangle_\pi = \sup_{J\in\mathcal{P},\ \|J\|_{2,\pi}=1} \langle J, J\rangle_\pi - \langle J, P^*PJ\rangle_\pi \\
&= 1 - \inf_{J\in\mathcal{P},\ \|J\|_{2,\pi}=1} \langle PJ, PJ\rangle_\pi = 1 - \inf_{J\in\mathcal{P},\ \|J\|_{2,\pi}=1} \|PJ\|_{2,\pi}^2 \leq 1.
\end{aligned} \tag{3.21}$$

Similarly,

$$\sigma_{\min} = \inf_{J\in\mathcal{P},\ \|J\|_{2,\pi}=1} \langle J, (I - P^*P)J\rangle_\pi = 1 - \sup_{J\in\mathcal{P},\ \|J\|_{2,\pi}=1} \|PJ\|_{2,\pi}. \tag{3.22}$$

Now, by Jensen's inequality and the fact that π is the stationary distribution of P,

$$\|PJ\|_{2,\pi}^2 = \mathsf{E}_\pi\left[\left(\mathsf{E}\left[J(x_1)|x_0\right]\right)^2\right] \leq \mathsf{E}_\pi\left[J(x_1)^2\right] = \|J\|_{2,\pi}^2.$$

That is, P is a non-expansive under the $\|\cdot\|_{2,\pi}$ norm. Combining this fact with (3.21)–(3.22), we have that $0 \leq \sigma_{\min} \leq \sigma_{\max} \leq 1$. Since $\rho(I - P^*P) = \max\left(|\sigma_{\min}|, |\sigma_{\max}|\right)$, the result follows.

(ii), suppose that $\zeta_1 \leq \zeta_2 \leq \cdots \leq \zeta_{|\mathcal{X}|}$ are the eigenvalues of the self-adjoint matrix P. By the same arguments as in (i), $0 \leq \zeta_i \leq 1$ for each i. Then,

$$\rho(I - P^2) = \max_i \ 1 - \zeta_i^2 = \max_i \ (1 - \zeta_i)(1 + \zeta_i) \leq \max_i \ 2(1 - \zeta_i) = 2\rho(I - P).$$

■

Lemma 13. *For any pair of functions $J, J' \in \mathcal{P}$,*

$$\|FJ - FJ'\|_{2,\pi} \leq \frac{R(\alpha)\alpha}{\sqrt{1-\alpha}}\,\lambda(P)\,\sqrt{\mathrm{Var}_\pi(J - J')},$$

where $R\colon [0,1) \to \left[1, \sqrt{5/2}\right]$ is a bounded function given by

$$R(\alpha) \triangleq \min\left\{\frac{1}{\sqrt{1-\alpha}}, \frac{2}{\sqrt{1+\alpha}}\right\}.$$

Proof. First, observe that if $y, y' \in \mathbb{R}^\infty$ are two infinite sequences of real numbers,

$$\sup_s y(s) - \sup_s y'(s) \leq \sup_s \ |y(s) - y'(s)|.$$

We can apply this fact to the pathwise maximization in the F operator to obtain that, for all $x_0 \in \mathcal{X}$,

$$FJ(x_0) - FJ'(x_0) \leq \mathsf{E}\left[\sup_{s\geq 0} \left|\sum_{t=1}^{s} \alpha^t \left(\Delta J(x_t, x_{t-1}) - \Delta J'(x_t, x_{t-1})\right)\right| \,\middle|\, x_0\right].$$

By symmetry,

$$|FJ(x_0) - FJ'(x_0)| \leq \mathsf{E}\left[\sup_{s\geq 0} \left|\sum_{t=1}^{s} \alpha^t \left(\Delta J(x_t, x_{t-1}) - \Delta J'(x_t, x_{t-1})\right)\right| \,\middle|\, x_0\right].$$

Using Jensen's inequality,

$$\begin{aligned} |FJ(x_0) - FJ'(x_0)|^2 &\leq \mathsf{E}\left[\sup_{s\geq 0} \left|\sum_{t=1}^{s} \alpha^t \left(\Delta J(x_t, x_{t-1}) - \Delta J'(x_t, x_{t-1})\right)\right|^2 \,\middle|\, x_0\right] \\ &\leq \mathsf{E}\left[\left(\sum_{t=1}^{\infty} \alpha^t \left|\Delta J(x_t, x_{t-1}) - \Delta J'(x_t, x_{t-1})\right|\right)^2 \,\middle|\, x_0\right]. \end{aligned} \tag{3.23}$$

Taking an expectation over x_0 and again applying Jensen's inequality,

$$\begin{aligned}
\|FJ - FJ'\|_{2,\pi}^2 &\leq \left(\frac{\alpha}{1-\alpha}\right)^2 \mathsf{E}_\pi \left[\left(\frac{1-\alpha}{\alpha}\sum_{t=1}^{\infty}\alpha^t \left|\Delta J(x_t, x_{t-1}) - \Delta J'(x_t, x_{t-1})\right|\right)^2\right] \\
&\leq \left(\frac{\alpha}{1-\alpha}\right)^2 \mathsf{E}_\pi \left[\frac{1-\alpha}{\alpha}\sum_{t=1}^{\infty}\alpha^t \left|\Delta J(x_t, x_{t-1}) - \Delta J'(x_t, x_{t-1})\right|^2\right] \\
&= \left(\frac{\alpha}{1-\alpha}\right)^2 \|\Delta J - \Delta J'\|_{2,\pi}^2 .
\end{aligned} \tag{3.24}$$

Here, the norm in the final equality is defined in Lemma 12, and we have used the fact that π is the stationary distribution.

On the other hand, following Chen and Glasserman (2007), Doob's maximal quadratic inequality and the orthogonality of martingale differences imply that, for every time $T \geq 1$,

$$\begin{aligned}
&\mathsf{E}_\pi \left[\sup_{0\leq s\leq T}\left|\sum_{t=1}^{s}\alpha^t \left(\Delta J(x_t, x_{t-1}) - \Delta J'(x_t, x_{t-1})\right)\right|^2\right] \\
&\leq 4\mathsf{E}_\pi \left[\left|\sum_{t=1}^{T}\alpha^t \left(\Delta J(x_t, x_{t-1}) - \Delta J'(x_t, x_{t-1})\right)\right|^2\right] \\
&\leq 4\mathsf{E}_\pi \left[\sum_{t=1}^{T}\alpha^{2t} \left|\Delta J(x_t, x_{t-1}) - \Delta J'(x_t, x_{t-1})\right|^2\right] \\
&= 4\alpha^2 \frac{1-\alpha^{2T-1}}{1-\alpha^2} \|\Delta J - \Delta J'\|_{2,\pi}^2 .
\end{aligned}$$

Using the monotone convergence theorem to take the limit as $T \to \infty$ and comparing with (3.23), we have that

$$\|FJ - FJ'\|_{2,\pi}^2 \leq \frac{4\alpha^2}{1-\alpha^2} \|\Delta J - \Delta J'\|_{2,\pi}^2 . \tag{3.25}$$

Combining the upper bounds of (3.24) and (3.25), we have that

$$\|FJ - FJ'\|_{2,\pi} \leq \frac{R(\alpha)\alpha}{\sqrt{1-\alpha}} \|\Delta J - \Delta J'\|_{2,\pi} . \tag{3.26}$$

Applying Lemma 12, the result follows. ■

4

PATHWISE METHOD FOR LINEAR CONVEX SYSTEMS

4.1. Introduction

Markov decision processes (MDPs) are a general framework for modeling sequential decision problems under uncertainity. A number of problems encountered in the areas from economics to business to engineering can be cast as a dynamic program. However, for many problems of interest, the size of state space is typically exponential in the dimension of state space. This phenomenon is referred to as the *curse of dimensionality* as it renders dynamic programming via standard approach intractable.

An effective way to address the curse of dimensionality is through the use of *value function approximations*. Consider a collection of real-valued functions of the state space, referred to as the *basis functions*. One can represent a value function approximation as a linear combination of these basis functions and such a parameterized form, involving relatively few parameters, provides a compact representation of the approximation. Using Approximate Dynamic Programming (ADP) techniques (see, for example, Bertsekas and Tsitsiklis, 1996; de Farias and Van Roy, 2003, 2004; Powell, 2007) one can tune the parameters to obtain 'good' approximation to the optimal value function. By standard dynamic programming results, we can use such an approximation to obtain policies, which are generally speaking suboptimal. We will couch our results in the context of a *minimization dynamic program* and in this setting, simulating a suboptimal policy provides us upper bounds on the optimal solution.

The upper bounds can be complemented by computing lower bounds, in order to get a sense for the suboptimality of the policy. A general approach to obtaining lower bounds is by considering relaxation of information process and allowing oneself to look into future. In a minimization dynamic program, one would expect only to do better by looking into the future and hence obtain lower bounds. Further, in the spirit of Lagrangian duality, we can also impose penalty for this relaxation. This approach of obtaining lower bounds by using information relaxation, while simultaneously introducing penalty for this relaxation, will be referred to as the *dual approach*. In this approach, the lower bounds are typically evaluated using Monte Carlo simulation and this involves generating sample paths of the underlying randomness and solving a deterministic optimization program, referred to as *inner problem*, for each sample path. The average of the optimal objective values of the inner problem, computed for each sample path, provides an estimate of the lower bound.

These methods originated in the context of American option pricing literature and have become popular following the work of Rogers (2002), Haugh and Kogan (2004) and Andersen and Broadie (2004). Generalization of this approach, to control problems other than optimal stopping, have been studied by Rogers (2008) and Brown et al. (2010). Following their work, these methods have seen applications in areas like portfolio optimization (Brown and Smith, 2010), valuation of natural gas storage (Lai et al., 2010a,b), among others.

A crucial input to duality based methods is the choice of the penalty function and intuitively speaking, the penalty nullifies the benefit that the policies may derive from prescience. Computing the optimal penalty, in general, may not be easier than solving the original MDP. However, this approach inspires heuristic selection of 'good' penalty functions. The choice of these penalties is guided by atleast two concerns: (a) the lower bounds evaluated via simulation should be tight (b) the inner problem obtained after information relaxation and introduction of penalty, should be tractable. Observe that these two concerns are at odds. On the one hand, we would prefer 'rich' penalty functions, in order to obtain tight lower bounds. But on the other, this needs to be balanced by the requirement to have a tractable inner problem. In certain cases, for example optimal stopping problem, the inner problem

is tractable on account of special structure available in the problem. However, in general, inner problem is a determinstic dynamic program subject to curse of dimensionality.

In application of dual methods, the penalty function is typically obtained from an approximation to value function, which in turn is obtained using other methods. In the context of American option pricing, the value function approximation is obtained by using ADP methods designed to obtain exercise policies. For more discussion see Section 8.7 of Glasserman (2004). In certain other cases, for example Brown and Smith (2010), application of these techniques can be quite specialized to the problem at hand. They consider dynamic portfolio optimization with transaction costs and demonstrate the effectiveness of certain heuristics by computing complementary bounds using dual method. The penalty function for the dual method is obtained from the value function associated with a simpler model, namely frictionless model. However, a direct application of the value function results in an intractable inner problem and hence the authors consider linearization of the value function along a heuristic strategy. Thus, application of these approaches have relied on other methods to obtain penalty function and might require considerable amount work to keep the overall approach tractable.

The present chapter, in summary, considers a broad class of MDPs and introduces a new tractable method for computing dual bounds. The method delivers tight bounds by identifying the best penalty function amongst a parameterized class of penalty functions. We implement our method on a high-dimensional financial application, namely, optimal execution and demonstrate the practical value of the method vis-a-vis competing methods available in the literature. In addition, we provide theory to show that the bounds generated by our method are provably tighter than some of the other available approaches, including *approximate linear programming* (ALP).

In greater detail, we make the following contributions:

- **New Methodology.** Computation of bounds via dual methods has been somewhat adhoc. We introduce a new method, which we call *pathwise optimization* (PO). We

propose a class of value function approximations that results in a tractable dual optimization problem. Given a parameterization of this class, PO method provides a structured approach to determining the best penalty within this class by solving a convex optimization problem.

- **Application: Optimal Execution.** We consider the application of PO to a high-dimensional problem that arises in the area of optimal trade execution. Brown and Smith (2010) considered the application of dual methods to portfolio optimization problem and their method can be applied in the context of our problem. Our method results in bounds that are provably stronger than the Brown and Smith (2010) bounds. In numerical experiments, we observe that PO provides much stronger bounds relative to the Brown and Smith (2010) approach with very little incremental computational burden.

- **Theory.** Lower bounds produced by the PO method can be directly compared to the bounds produced by linear programming based ADP algorithms of the type introduced by Schweitzer and Seidmann (1985) and de Farias and Van Roy (2003) and shown to result in provably tighter bounds. Wang and Boyd (2011) introduce a semidefinite programming based method to compute bounds and PO method dominates this alternative.

Duality based upper bounds for the pricing of American and Bermudan options, which rely on Doob's decomposition to generate the penalty process, were introduced by Rogers (2002) and Haugh and Kogan (2004). Andersen and Broadie (2004) show how to compute martingale penalties from rules and obtain upper bounds. Desai et al. (2010) provide a structured approach to identifying the best penalty within a parameterized family for optimal stopping problems. An alternative 'multiplicative' approach to duality was introduced by Jamshidian (2003) and its connections with 'additive' duality approach were explored in Chen and Glasserman (2007). Generalizations of the duality approach to control problems other than optimal stopping have been studied by Rogers (2008) and Brown et al. (2010).

Further, Brown et al. (2010) consider a broader class of information relaxations than causality. Applications of these methods were considered in portfolio optimal (Brown and Smith, 2010) and valuation of natural gas storage (Lai et al., 2010a,b), among others. Haugh and Lim (2011) provide a comparison of different penalties, including the ones introduced by Brown and Smith (2010), in the context of Linear Quadratic Controller problem.

The remainder of the chapter is organized as follows: in Section 4.2, we formulate the pathwise optimization problem and illustrate the general martingale penalty approach. In Section 4.3, we introduce our methodology, the PO method. Section 4.4 illustrates the benefits of the PO method in a numerical case study of optimal execution. In Section 4.5, we develop our theoretical results.

4.2. Formulation

Consider a finite horizon Markov decision process (MDP) with state $x_t \in \mathbb{R}^{m+\ell}$ at each time $t \in \mathcal{T} \triangleq \{0, 1, \ldots, T\}$. We assume that the state decomposes according to $x_t = (y_t, z_t)$, where $y_t \in \mathbb{R}^m$ and $z_t \in \mathbb{R}^\ell$. We assume that the process $\{y_t\}$ evolves according to the dynamics

$$y_{t+1} = A_t y_t + B_t z_t + C_t u_t + \epsilon_{t+1}, \tag{4.1}$$

for all $0 \leq t < T$. Here, $u_t \in \mathbb{R}^n$ is the control input applied at time t, and $\epsilon_t \in \mathbb{R}^m$ is a zero mean IID process with covariance matrix $W_t \triangleq \mathsf{E}[\epsilon_t \epsilon_t^\top]$. Thus, at each time t, the dynamics of y_t are *linear* and governed by the matrices $A_t \in \mathbb{R}^{m\times m}$, $B_t \in \mathbb{R}^{m\times \ell}$, and $C_t \in \mathbb{R}^{m\times n}$. We will refer to y_t as the *endogenous* state since it's dynamics are affected by the choice of control u_t. On the other hand, we call z_t that evolves independent of the control u_t applied, as an *exogenous* state and is given by dynamics of the form

$$z_{t+1} = f(z_t, \eta_{t+1}),\ \forall\ 0 \leq t < T, \tag{4.2}$$

where η_t are IID random variables. Thus, our system evolves according to:

Assumption 2. (Quasi-linear dynamics)

$$y_{t+1} = A_t y_t + B_t z_t + C_t u_t + \epsilon_{t+1},$$
$$z_{t+1} = f(z_t, \eta_{t+1}),\ \forall\ 0 \leq t < T,$$

where ϵ_t and η_t are sources of randomness.

Let $\mathbb{F} \triangleq \{\mathcal{F}_t\}$ be the natural filtration generated by the exogenous process z_t and noise term ϵ_t. Define $\mathbf{x}_t \triangleq (x_0, x_1, \ldots, x_t)$, $\mathbf{u}_t \triangleq (u_0, u_1, \ldots, u_t)$, $\mathbf{y}_t \triangleq (y_0, y_1, \ldots, y_t)$ and $\mathbf{z}_t \triangleq (z_0, z_1, \ldots, z_t)$. We define our cost function on domain $\mathcal{D} \triangleq \mathbb{R}^{(m+\ell)\times(T+1)} \times \mathbb{R}^{n\times(T+1)}$. Given a measurable function $g\colon \mathcal{D} \to \mathbb{R}$, we define the cost incurred over path $\mathbf{x}_T$ and decision variables $\mathbf{u}_T$ as $g(\mathbf{x}_T, \mathbf{u}_T)$. We make the following assumption:

Assumption 3. (Convex functionals of path)
Cost function $g\colon \mathcal{D} \to \mathbb{R}$ is measurable and jointly convex in $(\mathbf{y}_T, \mathbf{u}_T)$.

We allow for imposition of *convex constraints on the control*, i.e., given a convex set $\mathcal{K} \subseteq \mathbb{R}^{n\times(T+1)}$, we can impose the constraint $\mathbf{u}_T \in \mathcal{K}$. The control of the system is given by sequence of policy denoted by $\boldsymbol{\mu}_T = (\mu_0, \ldots, \mu_T)$. Define the set of feasible nonanticipative policies $\mathcal{A}^{\mathbb{F}} \triangleq \{\boldsymbol{\mu}_T : \boldsymbol{\mu}_T \in \mathcal{K}$ a.s. and is adapted to filtration $\mathbb{F}\}$. We are interested in the following optimization problem:

$$\inf_{\boldsymbol{\mu}_T \in \mathcal{A}^{\mathbb{F}}} \mathsf{E}\left[g(\mathbf{x}_T, \boldsymbol{\mu}_T)\right]. \tag{4.3}$$

Thus, our framework allows for very general dynamics, cost functions that are convex functionals of the path, and imposition of convex constraints on the controls. A large number of applications from the areas such as portfolio optimization, inventory control, revenue management, etc, can be readily addressed in this setup.

Pathwise method, to be described in Section 4.3, uses Assumptions 2 and 3 to ensure tractability of the approach. However, in the interest of ease of exposition and without loss of generality, we assume the cost function is separable across time for the rest of the chapter.

Assumption 4. (Separable cost and constraints)

1. *Cost function g is separable across time and is given by*

$$g(\mathbf{x}_T, \mathbf{u}_T) = \sum_{t=0}^{T} g_t(x_t, u_t),$$

where for all $t = 0, \ldots, T$, $g_t \colon \mathbb{R}^{m+\ell} \times \mathbb{R}^n \to \mathbb{R}$ is a measurable function of state x_t and decision variable u_t and is jointly convex in (y_t, u_t).

2. *Constraint on the control $\mathcal{K}$ is separable across time and takes the form $\mathcal{K} = \mathcal{U}_0 \times \mathcal{U}_1 \ldots \times \mathcal{U}_T$, where for all $t = 0, \ldots, T$, $\mathcal{U}_t \subseteq \mathbb{R}^n$ is a convex set constraining the action u_t.*

Thus, our optimization problem (4.3) can now be stated as

$$\inf_{\boldsymbol{\mu}_T \in \mathcal{A}^{\mathbb{F}}} \mathsf{E}\left[\sum_{t=0}^{T} g_t(x_t, \mu_t)\right]. \tag{4.4}$$

The policy μ_s at time s such that $\mu_s \in \mathcal{U}_s$ a.s. will be referred to as feasible policy. Let $\mathcal{A}_t^{\mathbb{F}} \triangleq \{(\mu_t, \ldots, \mu_T) \colon \mu_s \text{ is } \mathcal{F}_s - \text{measurable and feasible for all } s = t, \ldots, T\}$. We now define the optimal cost-to-go-functions or value functions, for all $x \in \mathbb{R}^{m+\ell}$ and $t \in \mathcal{T}$, as

$$J_t^*(x) = \inf_{(\mu_t, \ldots, \mu_T) \in \mathcal{A}_t^{\mathbb{F}}} \mathsf{E}\left[\sum_{s=t}^{T} g_s(x_s, \mu_s) | x_t = x\right]. \tag{4.5}$$

In order to ensure our problem is well defined, we assume following technical conditions:

Assumption 5. (Technical conditions)

1. $\mathsf{E}[|g_t(x_t, \mu_t)|] < \infty$ *for policies μ_s that are $\mathcal{F}_s-$measurable and feasible for all $s = 0, \ldots, t$*

2. $J_t^*(x) > -\infty$ *for all $x \in \mathbb{R}^{m+\ell}$ and $t \in \mathcal{T}$*

3. $J_t^*(\cdot)$ *is measurable for $t \in \mathcal{T}$*

4. $\mathsf{E}[|J_t^*(x_t)|] < \infty$ *for policies μ_s that are $\mathcal{F}_s-$measurable and feasible for all $s = 0, \ldots, t-1$*

Using Propostion 8.2 from Bertsekas and Shreve (1996), the optimal value functions will satisfy the Bellman's equation:

$$J_t^*(x) = \begin{cases} \inf_{u_t \in \mathcal{U}_t} \left\{ g_t(x, u_t) + \mathsf{E}\left[J_{t+1}^*(x_{t+1}) \middle| x_t = x,\ u_t \right] \right\} & \text{if } t < T, \\ \inf_{u_T \in \mathcal{U}_T} g_T(x, u_T) & \text{if } t = T. \end{cases} \tag{4.6}$$

In the following section, we introduce the dual formulation of the optimization problem (4.4) by using the so-called martingale duality approach. This approach relaxes the non-anticipativity constraint, while simultaneously introducing penalty for this relaxation and the dual problem is essentially choosing the 'optimal' penalty to solve the original problem (4.4).

4.2.1. The Martingale Duality Approach

Let $\mathcal{S}$ be the space of real-valued measurable functions defined on state space $\mathbb{R}^{m+l}$. Let $\mathcal{P}$ be the space of real-valued functions on $\mathbb{R}^{m+l} \times \mathcal{T}$, such that for $J \in \mathcal{P}$, for all $t \in \mathcal{T}$, $J_t \triangleq J(\cdot, t)$ is measurable and $\mathsf{E}[|J_t(x_t)|] < \infty$ for all measurable sequence of policies such that $\mu_s \in \mathcal{U}_s$ a.s. for $s = 0, \ldots, t-1$. Let us begin with defining the martingale difference operator Δ that maps $\mathcal{S}$ to the space of real valued functions on $\mathbb{R}^{m+l} \times \mathbb{R}^{m+l} \times \mathbb{R}^n$ according to

$$(\Delta J_t)(x_t, x_{t-1}, u_{t-1}) \triangleq J_t(x_t) - \mathsf{E}[J_t(x_t) | x_{t-1}, u_{t-1}].$$

We will abuse the notation slightly and write $(\Delta J_t)(x_t, x_{t-1}, u_{t-1})$ as $\Delta J_t(x_t, x_{t-1}, u_{t-1})$.

We are interested in computing lower bounds by considering a perfect information relaxation. Towards this end, define the set of all feasible controls $\mathcal{A}$ as a collection of measurable sequences $\mathbf{u}_T$ such that $\mathbf{u}_T \in \mathcal{K}$. Next, we define the *martingale duality operator* $F \colon \mathcal{P} \to \mathcal{S}$ according to:

$$(FJ)(x) \triangleq \mathsf{E}\left[\inf_{\mathbf{u}_T \in \mathcal{A}} g_0(x_0, u_0) + \sum_{t=1}^{T} g_t(x_t, u_t) - \Delta J_t(x_t, x_{t-1}, u_{t-1}) \middle| x_0 = x \right],$$

where $J_t \triangleq J(\cdot, t)$.

Given a sample path $\omega = (\{z_0\}, \{\epsilon_1, z_1\}, \ldots, \{\epsilon_{T-1}, z_{T-1}\}, \{\epsilon_T, z_T\})$, the deterministic optimization problem inside the expectation is a minimization over $u_t, t \in \mathcal{T}$, subject to dynamics of the system and will be referred to as the *inner optimization problem*. Or, viewing the dynamics as a set of constraints, the inner optimization problem can be stated as

$$\begin{array}{ll} \inf_{\mathbf{u}_T, \mathbf{y}_T} & g_0(x_0, u_0) + \sum_{t=1}^{T} g_t(x_t, u_t) - \Delta J_t(x_t, x_{t-1}, u_{t-1}) \\ \text{subject to} & y_{t+1} = A_t y_t + B_t z_t + C_t u_t + \epsilon_{t+1},\ 0 \le t \le T-1, \\ & u_t \in \mathcal{U}_t,\ 0 \le t \le T. \end{array} \tag{4.7}$$

We are now ready to state key proposition that shows martingale duality operator F can be used to generate lower bounds.

Proposition 1.

(i) (Weak Duality) For any $J \in \mathcal{P}$ and all $x \in \mathcal{X}$, $FJ(x) \le J_0^*(x)$.

(ii) (Strong Duality) For all $x \in \mathcal{X}$, $J_0^*(x) = FJ^*(x)$.

Proof. *(i)*

$$\begin{aligned} J_0^*(x) &= \inf_{\boldsymbol{\mu}_T \in \mathcal{A}^{\mathbb{F}}} \mathsf{E}\left[\sum_{t=0}^{T} g_t(x_t, \mu_t) \;\middle|\; x_0 = x \right], \\ &\overset{(a)}{=} \inf_{\boldsymbol{\mu}_T \in \mathcal{A}^{\mathbb{F}}} \mathsf{E}\left[\sum_{t=0}^{T} g_t(x_t, \mu_t) \; - \sum_{t=1}^{T} \Delta J_t(x_t, x_{t-1}, \mu_{t-1}) \middle| \; x_0 = x \right], \\ &= \inf_{\boldsymbol{\mu}_T \in \mathcal{A}^{\mathbb{F}}} \mathsf{E}\left[g_0(x_0, \mu_0) \; + \sum_{t=1}^{T} g_t(x_t, \mu_t) - \Delta J_t(x_t, x_{t-1}, \mu_{t-1}) \middle| \; x_0 = x \right], \\ &\overset{(b)}{\ge} \mathsf{E}\left[\inf_{\mathbf{u}_T \in \mathcal{A}} g_0(x_0, u_0) + \sum_{t=1}^{T} g_t(x_t, u_t) - \Delta J_t(x_t, x_{t-1}, u_{t-1}) \;\middle|\; x_0 = x \right], \\ &= FJ(x). \end{aligned}$$

Inequality (a) follows from the fact that martingale differences have zero mean and (b) holds because it allows for policies that look at entire sample path.

(ii) From weak duality we have $J_0^*(x) \geq FJ^*(x)$. We show that $J_0^*(x) \leq FJ^*(x)$ to establish the result.

$$\begin{aligned} FJ^*(x) &= \mathsf{E}\left[\inf_{\mathbf{u}_T \in \mathcal{A}} g_0(x_0, u_0) + \sum_{t=1}^{T} g_t(x_t, u_t) - J_t^*(x_t) + \mathsf{E}[J_t^*(x_t)|x_{t-1}, u_{t-1}] \;\middle|\; x_0 = x \right], \\ &= \mathsf{E}\Bigg[\inf_{\mathbf{u}_T \in \mathcal{A}} J_0^*(x_0) + \sum_{t=0}^{T-1} \Big(g_t(x_t, u_t) + \mathsf{E}[J_{t+1}^*(x_{t+1})|x_t, u_t] - J_t^*(x_t) \Big) \\ &\qquad\qquad + g_T(x_T, u_T) - J_T^*(x_T) \Bigg| \; x_0 = x \Bigg], \\ &\geq J_0^*(x) \end{aligned}$$

The last inequality follows from the fact that J^* satisfies (4.6). ■

This result allows us to define the *dual problem* as

$$\sup_{J \in \mathcal{P}} FJ(x). \tag{4.8}$$

Solving the optimization problem (4.8), in general, might be intractable because $\mathcal{P}$ is a very high-dimensional space and it is not clear how to optimize over it. In addition, observe that given a $J \in \mathcal{P}$, even evaluating $FJ(x)$ can be computationally challenging. A general approach to evaluate $FJ(x)$ is via Monte Carlo simulation, where we generate a number of sample paths, solve the inner optimization problem on each path and the sample average of optimal objective function values gives us an estimate of the upper bound. However, the inner optimization problem in general is a deterministic dynamic program subject to curse of dimensionality and can be a bottleneck of the procedure. In the following sections we address these challenges.

4.3. The Pathwise Optimization Method

Motivated by the challenges associated with solving program (4.8), we introduce a class of value function approximations that will not only result in a tractable inner optimization problem but will also enable us to identify the 'best' approximation that results in the tightest

possible lower bounds. Towards this end, consider following class of quadratic functions

$$\mathcal{C} \triangleq \{J \in \mathcal{P}\colon J_t(x_t) = y_t^\top \Gamma_t y_t + \beta_t(z_t)^\top y_t + \alpha_t(z_t), \text{ for some } \Gamma_t,\ \alpha_t, \text{ and } \beta_t,\ \forall\, t \in \mathcal{T}\}.$$

The following result guarantees the tractablility of inner optimization problem:

Theorem 9. *For all $J \in \mathcal{C}$, the inner problem is a convex optimization problem.*

Proof. Given a sample path $\omega = (\{z_0\}, \{\epsilon_1, z_1\}, \ldots, \{\epsilon_{T-1}, z_{T-1}\}, \{\epsilon_T, z_T\})$, the inner optimization problem is

$$\begin{aligned} \underset{\mathbf{u}_T, \mathbf{y}_T}{\text{minimize}} \quad & g_0(x_0, u_0) + \sum_{t=1}^{T} g_t(x_t, u_t) - \Delta J_t(x_t, x_{t-1}, u_{t-1}) \\ \text{subject to} \quad & y_{t+1} = A_t y_t + B_t z_t + C_t u_t + \epsilon_{t+1},\ 0 \le t \le T-1, \\ & u_t \in \mathcal{U}_t,\ 0 \le t \le T. \end{aligned} \tag{4.9}$$

For $J \in \mathcal{C}$, after some algebra, we can write the martingale difference as

$$\begin{aligned} \Delta J_t(x_t, x_{t-1}, u_{t-1}) = {} & 2\epsilon_t^\top \Gamma_t w_{t-1} + \beta_t(z_t)^\top y_t - \mathsf{E}\left[\beta_t(z_t)^\top | x_{t-1}\right] w_{t-1} \\ & + \epsilon_t^\top \Gamma_t \epsilon_t - \mathsf{E}\left[\epsilon_t^\top \Gamma_t \epsilon_t\right] + \alpha_t(z_t) - \mathsf{E}\left[\alpha_t(z_t) | x_{t-1}\right],\ \forall\ 1 \le t \le T, \end{aligned} \tag{4.10}$$

where $w_t = A_t y_t + B_t z_t + C_t u_t$. Notice that for fixed values of Γ_t, $\beta_t(z_t)$ and $\alpha_t(z_t)$, the expression is linear in w_{t-1} and y_t. Since both y_t and w_{t-1} are linear in u_{t-1} and $g_t(y_t, z_t, u_t)$ is jointly convex in (y_t, u_t), the inner problem is minimization of a convex function subject to linear dynamics constraints and can be solved efficiently. ■

Given a $J \in \mathcal{C}$, Theorem 9 guarantees that we can solve the inner problem efficiently and by repeatedly solving the inner problem over different sample paths, we can estimate the lower bound $FJ(x)$. However, for this procedure to be of practical value, we need a method for identifying $J \in \mathcal{C}$ that would yield tight bounds. Ideally, we would like to solve

$$\sup_{J \in \mathcal{C}} FJ(x),$$

but $\mathcal{C}$ involves arbitrary functions of exogenous variable z_t and it is unclear how to optimize over it. In the spirit of ADP algorithms, we introduce a basis function architecture with K *basis functions*:

$$\Phi \triangleq \{\phi_1, \phi_2, \ldots, \phi_K\},$$

where $\phi_i \colon \mathbb{R}^\ell \to \mathbb{R},\ 1 \leq i \leq K$ are functions of the exogenous state variable z_t. At each time $t \in \mathcal{T}$, a matrix $R_t \in \mathbb{R}^{K \times m}$, vector $r_t \in \mathbb{R}^K$ and symmetric matrix $\Gamma_t \in \mathbb{R}^{m \times m}$, determines the quadratic function $y_t^\top \Gamma_t y_t + \Phi(z_t) R_t y_t + \Phi(z_t) r_t$. Denote the collection of coefficients by $\kappa = \{\Gamma_t, R_t, r_t \colon t \in \mathcal{T}\}$ and the corresponding value function

$$J_t^\kappa(x_t) = y_t^\top \Gamma_t y_t + \Phi(z_t) R_t y_t + \Phi(z_t) r_t,\ \forall\ 1 \leq t \leq T.$$

Applying the martingale difference operator, we obtain

$$\begin{aligned}\Delta J_t^\kappa(x_t, x_{t-1}, u_{t-1}) = (2\epsilon_t^\top \Gamma_t + \Delta\Phi(z_t, z_{t-1}) R_t)(A_{t-1} y_{t-1} + B_{t-1} z_{t-1} + C_{t-1} u_{t-1}) \\ + \Delta\Phi(z_t, z_{t-1}) r_t + \Phi(z_t) R_t \epsilon_t + \epsilon_t^\top \Gamma_t \epsilon_t - \mathsf{E}\left[\epsilon_t^\top \Gamma_t \epsilon_t\right],\ \forall\ 1 \leq t \leq T,\end{aligned}$$

where $\Delta\Phi(z_t, z_{t-1}) = \Phi(z_t) - \mathsf{E}[\Phi(z_t)|z_{t-1}]$. Note, a number of terms in this expression are mean zero terms and do not interact with decision variables and can be safely ignored. Thus, given a collection of coefficients κ, the corresponding lower bound is given by

(4.11)

$$\begin{aligned}FJ^\kappa(x) = \mathsf{E}\Bigg[\inf_{\mathbf{u}_T \in \mathcal{A}} \sum_{t=0}^{T-1} \left(g_t(x_t, u_t) - \left(2\epsilon_{t+1}^\top \Gamma_{t+1} + \Delta\Phi(z_{t+1}, z_t) R_{t+1}\right)(A_t y_t + C_t u_t)\right) \\ + g_T(x_T, u_T) \Big| x_0 = x \Bigg].\end{aligned}$$

We propose to obtain the tightest possible lower bound, afforded by the chosen basis functions, by solving the problem:

$$\sup_\kappa FJ^\kappa(x). \tag{4.12}$$

Theorem 10 establishes that this is infact a concave optimization problem.

Theorem 10. *$FJ^\kappa(x)$ is concave in κ.*

Proof. Referring to equation (4.11), we observe that $FJ^\kappa(x)$ is expectation over an inner optimization problem. Since optimal objective of inner optimization problem is a concave function in κ, we conclude $FJ^\kappa(x)$ is concave in κ. ■

Optimization problem defined by (4.12) is a concave program in relatively small number of variables and the main challenge is objective function is expectation over an inner optimization problem. We provide two practical methods for solving this problem in the following section.

4.3.1. Computational Methods

We view the problem as a stochastic optimization problem and consider two sampling based approaches for solving it. The first method is based on the idea of stochastic gradient descent. Starting from an initial solution, one can use a step-size rule to move in the direction of the gradient. The main advantages of this method are that it is online and we can handle large problems with low a memory requirement.

The second approach uses a sample average to approximate the objective function. Using duality the inner problem is expressed as a maximization problem, thus enabling us to provide a convex optimization program to generate lower bounds. This approach allows us to use readily available convex optimization solvers to solve problems efficiently.

Stochastic Supergradient Method

Stochastic supergradient method is a simple approach that works with the first-order information, namely the supergradient. It works with unbiased estimates of the supergradient, which can be computed efficiently and under properly chosen step-size rule is guaranteed to converge. This approach can be used even for nondifferentiable functions and hence is broad in its applicability relative to second-order methods that require computation of the Hessian.

The choice of step-size is an important input to supergradient algorithms. Let the step-size at kth iterate be denoted by α_k. Typically one uses step-sizes that are square summable but not summable, i.e.

$$\alpha_k \geq 0, \quad \sum_{k=1}^{\infty} \alpha_k^2 < \infty, \quad \sum_{k=1}^{\infty} \alpha_k = \infty$$

(Bertsekas and Tsitsiklis, 1996). In our implementation, we used a step-size rule of the form

$$\alpha_k = \frac{A}{B+k}, \tag{4.13}$$

where the parameters A and B can be chosen to optimize practical performance.

The method starts with an initial guess for the solution. Given a current guess $\kappa = \{\Gamma_t, R_t, r_t, t \in \mathcal{T}\}$ and sample path $\omega = (\{z_0\}, \{\epsilon_1, z_1\}, \ldots, \{\epsilon_{T-1}, z_{T-1}\}, \{\epsilon_T, z_T\})$, we can define the optimal objective of inner optimization problem as

(4.14)

$$H(\kappa,\omega) \triangleq \begin{array}{ll} \min_{\mathbf{u}_T,\mathbf{y}_T} & \sum_{t=0}^{T-1} \left(g_t(x_t,u_t) - \left(2\epsilon_{t+1}^\top \Gamma_{t+1} + \Delta\Phi(z_{t+1},z_t)R_{t+1}\right)(A_t y_t + C_t u_t)\right) + g_T(x_T,u_T) \\ \text{s.t.} & y_{t+1} = A_t y_t + B_t z_t + C_t u_t + \epsilon_{t+1},\ 0 \le t \le T-1, \\ & u_t \in \mathcal{U}_t,\ 0 \le t \le T, \end{array}$$

and $y_t^*(\omega), u_t^*(\omega)$ be an optimal solution. Using this definition, our problem can be rewritten as

$$\sup_\kappa FJ^\kappa(x) = \sup_\kappa \mathsf{E}\left[H(\kappa,\omega)\ |x_0 = x\right].$$

In solving this problem, we are interested in the supergradient of $H(\kappa,\omega)$ and the following lemma helps us in that direction.

Lemma 15. *Assume $\mathcal{U}_t$ is a compact set for $t = 0,\ldots,T$. Supergradient of $H(\kappa,\omega)$ with respect to Γ_t and R_t is given by*

$$\begin{aligned} H(\kappa,\omega)_{\Gamma_t} &= -2\epsilon_t \left(A_{t-1}y_{t-1}^*(\omega) + C_{t-1}u_{t-1}^*(\omega)\right)^\top \\ H(\kappa,\omega)_{R_t} &= -\Delta\Phi(z_t,z_{t-1})^\top \left(A_{t-1}y_{t-1}^*(\omega) + C_{t-1}u_{t-1}^*(\omega)\right)^\top,\ \forall\ 1 \le t \le T. \end{aligned} \tag{4.15}$$

Proof. We first introduce some notation. Let the objective of inner optimization problem (4.14) be denoted by $Q(\kappa, \mathbf{u}_T, \mathbf{y}_T, \omega)$ and the feasible region by $Z(\omega)$. The collection of all optimal solutions to (4.14) be

$$\Omega(\kappa,\omega) \triangleq \operatorname*{argmin}_{(\mathbf{u}_T,\mathbf{y}_T)\in Z(\omega)} Q(\kappa,\mathbf{u}_T,\mathbf{y}_T,\omega).$$

For simplicity, assume κ' and κ'' be such that they differ only in Γ_t component. Then, we would like to show

$$H(\kappa', \omega) \leq H(\kappa'', \omega) + H(\kappa', \omega)_{\Gamma_t} \cdot (\Gamma'_t - \Gamma''_t),$$

where $H(\kappa', \omega)_{\Gamma_t} \cdot (\Gamma'_t - \Gamma''_t) = \sum_{i=1}^{m} \sum_{j=1}^{m} H(\kappa', \omega)_{\Gamma_t}(i, j)(\Gamma'_t - \Gamma''_t)(i, j)$.

By Danskin's theorem (see Proposition B.25 in Bertsekas, 1995), the directional derivative, denoted by $H'(\kappa, \omega; \kappa' - \kappa'')$, of $H(\kappa, \omega)$ in the direction $\kappa' - \kappa''$ is given by

$$\begin{aligned} H'(\kappa, \omega; \kappa' - \kappa'') &= \min_{(\mathbf{u}_T, \mathbf{y}_T) \in \Omega(\kappa, \omega)} Q'(\kappa, \mathbf{u}_T, \mathbf{y}_T, \omega; \kappa' - \kappa'') \\ &\leq Q'(\kappa, \mathbf{u}^*_T, \mathbf{y}^*_T, \omega; \kappa' - \kappa'') \\ &= H(\kappa', \omega)_{\Gamma_t} \cdot (\Gamma'_t - \Gamma''_t), \end{aligned}$$

where $\mathbf{u}^*_T, \mathbf{y}^*_T$ is an optimal solution to (4.14). By letting all the components Γ_t, R_t for all $t = 1, \ldots, T$ vary, we obtain the desired result. ■

We solve I independently sampled inner problems, compute the supergradient for each problem and use a sample average of the supergradients as an approximation to supergradient of $FJ^{\kappa}(x)$. We update our guess for the solution by taking a step in the direction of estimated gradient, according to the chosen step-size rule. Our approach is summarized in Algorithm 1.

The supergradient method is broad in its applicability and can be used to solve large convex optimization problems. Nonetheless, the performance of the algorithm is subject to choice of the initial guess for solution, parameters N and I and more importantly, step-size rule. In particular, if we pick a step-size that is small, then it will take a long time to get to the neighborhood of the optimal solution. On the other hand, if we picked a large step-size rule, then it would oscillate before converging. From a practical standpoint, effort is required to tune these parameters for efficient computation. The approach introduced in the next section addresses these concerns by introducing a convex optimization formulation.

Sample Average Approximation Method

The challenge in solving formulation (4.12) is that the objective function is an expectation of a random variable that is concave function of the decision variables. This suggests using

Step-size rule:

$$\alpha_k = \frac{A}{B+k}$$

where k is step number.
Set N to be iteration limit.
Set I to be number of samples used to estimate gradient.
For $1 \le t \le T$, set Γ_t, R_t to initial values.
for $k \leftarrow 1, N$ **do**
    for $i \leftarrow 1, I$ **do**
        Generate a sample path $\omega_i = (\{z_0^{(i)}\}, \{\epsilon_1^{(i)}, z_1^{(i)}\}, \ldots, \{\epsilon_T^{(i)}, z_T^{(i)}\})$.
        Solve the inner problem (4.14).
        Set $H(\kappa, \omega_i)_{\Gamma_t}$ and $H(\kappa, \omega_i)_{R_t}$ for $1 \le t \le T$ according to (4.15).
    end for
    Compute the sample average of the gradients

$$\bar{G}_{\Gamma_t} = \frac{1}{I}\sum_{i=1}^{I} H(\kappa, \omega_i)_{\Gamma_t}, \quad \bar{G}_{R_t} = \frac{1}{I}\sum_{i=1}^{I} H(\kappa, \omega_i)_{R_t}, \ \forall\ 1 \le t \le T.$$

    Update $R_t \leftarrow R_t + \alpha_k \bar{G}_{R_t}$ and $\Gamma_t \leftarrow \Gamma_t + \alpha_k \bar{G}_{\Gamma_t}$, for all $1 \le t \le T$.
end for

Algorithm 1: Supergradient algorithm

a sample average to approximate the objective function. Before we develop the method, we introduce Lemma 16 to obtain an alternate representation for FJ^κ. The idea is to convert the inner problem from a minimization problem to a maximization problem by using duality ideas and introduce the conjugate function $g_t^* \colon \mathbb{R}^m \times \mathbb{R}^l \times \mathbb{R}^n \to \mathbb{R}$ of the stage cost g_t defined as

$$g_t^*(\gamma_t, z_t, \mu_t) \triangleq \sup_{y_t, u_t} \left\{ \gamma_t^\top y_t + \mu_t^\top u_t - g_t(y_t, z_t, u_t) \right\}.$$

Lemma 16.

$$FJ^\kappa(x) = \mathsf{E}\left[\sup_{\lambda_0, \ldots, \lambda_{T-1}} \sum_{t=0}^{T-1} \lambda_t^\top \left(B_t z_t + \epsilon_{t+1}\right) - \sum_{t=0}^{T} g_t^*\left(\gamma_t, z_t, \mu_t\right) \right],$$

where

$$\mu_t^\top \triangleq \begin{cases} \left(2\epsilon_{t+1}^\top \Gamma_{t+1} + \Delta\Phi(z_{t+1}, z_t) R_{t+1} - \lambda_t^\top\right) C_t & 0 \le t \le T-1 \\ 0 & t = T, \end{cases}$$

$$\gamma_t^\top \triangleq \begin{cases} \left(2\epsilon_1^\top \Gamma_1 + \Delta\Phi(z_1, z_0) R_1 - \lambda_0^\top\right) A_0 & t = 0 \\ \left(2\epsilon_{t+1}^\top \Gamma_{t+1} + \Delta\Phi(z_{t+1}, z_t) R_{t+1} - \lambda_t^\top\right) A_t + \lambda_{t-1}^\top & 1 \le t \le T-1 \\ \lambda_{T-1}^\top & t = T, \end{cases}$$

and g_t^* is conjugate function of g_t.

Proof. Given a sample path $\omega = (\{z_0\}, \{\epsilon_1, z_1\}, \ldots, \{\epsilon_{T-1}, z_{T-1}\}, \{\epsilon_T, z_T\})$, the inner optimization problem is

$$\begin{aligned} \underset{u_t, y_t, t\in\mathcal{T}}{\text{minimize}} \quad & \sum_{t=0}^{T-1} \left(g_t(x_t, u_t) - \left(2\epsilon_{t+1}^\top \Gamma_{t+1} + \Delta\Phi(z_{t+1}, z_t) R_{t+1}\right)(A_t y_t + C_t u_t) \right) + g_T(x_T, u_T) \\ \text{subject to} \quad & y_{t+1} = A_t y_t + B_t z_t + C_t u_t + \epsilon_{t+1},\ 0 \le t \le T-1. \end{aligned}$$

Dualizing the constraints, the above problem is equivalent to

$$\sup_{\lambda_0,\ldots,\lambda_{T-1}} \inf_{u_t, y_t, t\in\mathcal{T}} \left\{ \sum_{t=0}^{T-1} \left(g_t(x_t, u_t) - \left(2\epsilon_{t+1}^\top \Gamma_{t+1} + \Delta\Phi(z_{t+1}, z_t) R_{t+1}\right)(A_t y_t + C_t u_t) \right) + g_T(x_T, u_T) \right.$$
$$\left. + \sum_{t=0}^{T-1} \lambda_t^\top \left(A_t y_t + B_t z_t + C_t u_t + \epsilon_{t+1} - y_{t+1}\right) \right\}.$$

After some algebra, we can rewrite it as

$$\sup_{\lambda_0,\ldots,\lambda_{T-1}} \left\{ \sum_{t=0}^{T-1} \lambda_t^\top \left(B_t z_t + \epsilon_{t+1}\right) + \inf_{u_t, y_t, t\in\mathcal{T}} \sum_{t=0}^{T} \left(g_t(y_t, z_t, u_t) - \gamma_t^\top y_t - \mu_t^\top u_t \right) \right\}, \tag{4.16}$$

where μ_t and γ_t are as defined in the statement of the Lemma. Substituting conjugate function of $g_t(y_t, z_t, u_t)$ in (4.16), we can write inner problem as

$$\sup_{\lambda_0,\ldots,\lambda_{T-1}} \left\{ \sum_{t=0}^{T-1} \lambda_t^\top \left(B_t z_t + \epsilon_{t+1}\right) - \sum_{t=0}^{T} g_t^*(\gamma_t, z_t, \mu_t) \right\}. \tag{4.17}$$

The desired result follows immediately from above expression. ■

Consider S independent *outer* sample paths $\omega_i = (\{z_0^{(i)}\}, \{\epsilon_1^{(i)}, z_1^{(i)}\}, \ldots, \{\epsilon_T^{(i)}, z_T^{(i)}\})$ for $i = 1, 2, \ldots, S$. Define $\hat{\mathsf{E}}$ to be empirical expectation with respect to the S sampled paths. Using sample average as an approximation to objective function in (4.12) and the representation of FJ^κ from Lemma 16, we are ready to state an implementable version of the optimization problem:

$$\max_{\kappa} \hat{\mathsf{E}} \left[\underset{\lambda_0,\ldots,\lambda_{T-1}}{\text{maximize}} \sum_{t=0}^{T-1} \lambda_t^\top \left(B_t z_t + \epsilon_{t+1}\right) - \sum_{t=0}^{T} g_t^* \left(\gamma_t, z_t, \mu_t\right) \right], \tag{4.18}$$

where

$$\mu_t^\top \triangleq \begin{cases} \left(2\epsilon_{t+1}^\top \Gamma_{t+1} + \Delta\Phi(z_{t+1}, z_t) R_{t+1} - \lambda_t^\top\right) C_t & 0 \le t \le T-1 \\ 0 & t = T, \end{cases}$$

$$\gamma_t^\top \triangleq \begin{cases} \left(2\epsilon_1^\top \Gamma_1 + \Delta\Phi(z_1, z_0) R_1 - \lambda_0^\top\right) A_0 & t = 0 \\ \left(2\epsilon_{t+1}^\top \Gamma_{t+1} + \Delta\Phi(z_{t+1}, z_t) R_{t+1} - \lambda_t^\top\right) A_t + \lambda_{t-1}^\top & 1 \le t \le T-1 \\ \lambda_{T-1}^\top & t = T, \end{cases}$$

and g_t^* is conjugate function of g_t. Observe that in order to compute μ_t and γ_t along a sample path, we need to be able to evaluate the martingale difference $\Delta\Phi(z_{t+1}, z_t)$. In general, these can be computed by one-step inner sampling, however, in many applications like the one illustrated in Section 4.4, these can be computed explicitly and hence only the outer sample paths need to be generated.

Computation of Unbiased Upper Bound

We have introduced two computational approaches for solving optimization problem (4.12). Using either of these methods we obtain an approximation $\hat{\kappa}$ to the true minimizer κ^*. We propose as an upper bound on $J_0^*(x_0)$, the quantity $FJ^{\hat{\kappa}}(x_0)$. The latter quantity may be estimated via a sampling procedure as follows.

1. Generate a second set of samples that is independent of those used in obtaining $\hat{\kappa}$.

2. An unbiased estimate of the upper bound $FJ^{\hat{\kappa}}(x_0)$ is given by

$$\hat{\mathsf{E}}\left[\underset{\lambda_0,\ldots,\lambda_{T-1}}{\text{maximize}} \sum_{t=0}^{T-1} \lambda_t^\top \left(B_t z_t + \epsilon_{t+1}\right) - \sum_{t=0}^{T} g_t^* \left(\gamma_t, z_t, \mu_t\right)\right],$$

where $\hat{\mathsf{E}}$ is the empirical expectation over second set of samples.

To summarize, we have presented two computational approaches to find an approximation to the optimum solution of (4.12). Given such a solution, we can compute unbiased upper bounds on optimal objective function value $J_0^*(x_0)$. We are ready to present a case study of these methods.

4.4. Case Study: Optimal Execution Problem

As an application of pathwise method we consider the problem of optimal execution. Consider a brokerage firm interested in liquidating a portfolio of securities on behalf of an investor. Due to the fiduciary role of the brokerage firm, only selling of the stock is allowed. In order to reduce transaction costs, the execution desk would like to time its orders to benefit from the short-term predictability in the stock prices. The execution literature is focused on this problem, see, for example, Heston et al. (2010).

We consider a model with short term return predictability presented in Moallemi and Sağlam (2011), describe the benchmark methods and then apply pathwise approach to the execution problem. We close the section with a numerical study, using the parameters described in Moallemi and Sağlam (2011) and compare pathwise bounds with benchmark methods including Brown and Smith (2010) bounds.

4.4.1. Problem Setting

We consider a setting where the economy consists of n different stocks. A trading desk is interested in liquidating a portfolio of assets $c \in \mathbb{R}^n_+$. The trading takes place over a discrete horizon $t = 1, \ldots, T$. The price of the assets at time t be $p_t \in \mathbb{R}^n$ and returns earned by holding a unit of each of the securities over time period $(t, t+1]$ be denoted by $r_{t+1} \triangleq p_{t+1} - p_t$.

The returns evolve according to a factor model $r_{t+1} = Bf_t + \epsilon_{t+1}$, where f_t denotes the factor values of L different factors at time t, matrix $B \in \mathbb{R}^{n\times L}$ is the factor loadings and $\epsilon_{t+1} \in \mathbb{R}^n$ is the zero mean noise with variance $\mathrm{Var}_t(\epsilon_{t+1}) = \Sigma$. The factors themselves evolve according to a mean reverting process given by $f_{t+1} = (I - \Theta)f_t + \varepsilon_{t+1}$, where $\Theta \in \mathbb{R}^{L\times L}$ is matrix of mean-reversion coefficients for the factors and ε_{t+1} is a mean zero noise with variance $\mathrm{Var}_t(\varepsilon_{t+1}) = \Xi$. This is a standard model, see, for example, Garleanu and Pedersen (2008). Let $\mathbb{F} \triangleq \{\mathcal{F}_t\}$ be the natural filtration generated by the exogenous process ϵ_{t+1} and noise term ε_{t+1}.

The control problem associated with liquidating a portfolio is to decide how to sell the stock in order to benefit from the short term predictability of the returns. At time t, we observe the amount of unsold stock y_{t-1}, the factors f_t, and decide the amount of stock to sell u_t. The holdings at time t is $y_t = y_{t-1} - u_t$. We define our state at time t to be $x_t \triangleq (y_{t-1}, f_t)$. The proceeds from the liquidation of the stock, wealth w_T, is then given by $w_T = \sum_{t=1}^{T} p_t^\top u_t$. In terms of returns and portfolio holdings we can rewrite terminal wealth as $w_T = p_1^\top y_0 + \sum_{t=2}^{T} r_t^\top y_{t-1}$, where y_0, p_1 are constants unaffected by our decision. Using the return evolution equation, $r_{t+1} = Bf_t + \epsilon_{t+1}$, we can rewrite terminal wealth as

$$w_T = p_1^\top y_0 + + \sum_{t=1}^{T-1} y_t^\top (Bf_t + \epsilon_{t+1}).$$

Observe that ϵ_t's are mean zero and do not matter in expectation. We assume that the costs associated with trading are quadratic and are of the form $\frac{1}{2}u_t^\top \Lambda u_t$. Let the trading policy be given by $\boldsymbol{\mu}_T = (\mu_1, \ldots, \mu_T)$. In order to maximize the net expected wealth after paying for the transaction costs, the decision maker solves the following optimization problem[1]:

$$\begin{aligned} \underset{\boldsymbol{\mu}_T \in \mathcal{A}^{\mathbb{F}}}{\text{maximize}} \quad & \mathsf{E}\left[\sum_{t=1}^{T}\left(y_t^\top B f_t - \frac{1}{2}\mu_t^\top \Lambda \mu_t\right)\right] \\ \text{subject to} \quad & y_t = y_{t-1} - \mu_t, \ \forall\, t = 1, \ldots, T \\ & \mu_t \geq 0, \ y_t \geq 0, \ \forall\, t = 1, \ldots, T \\ & y_0 = c, \ y_T = 0, \end{aligned} \tag{4.19}$$

[1] Note that constraints on the controls are not strictly separable as was assumed in our formulation. However, this assumption was introduced for the sake of ease of exposition and can be generalized to handle joint constraints.

where $\mathcal{A}^{\mathbb{F}}$ is collection of processes adapted to filtration $\mathbb{F}$. Observe that the nonnegativity constraints makes sure that the stock is sold via a pure sell algorithm.

4.4.2. Benchmark Methods

We consider the following benchmarks, representative of mainstream methods, for purposes of comparison with pathwise methods.

- **Perfect Hindsight Bounds.** Given the realization of factors $f_1, \ldots, f_T$ over the entire horizon, we can compute the perfect hindsight bound, referred to as PH-UB, by solving the perfect information relaxation of the problem (4.19), which is equivalent to solving a deterministic quadratic program. By repeating this procedure over a number of sample paths, we obtain our estimate of PH-UB.

- **Linear Quadratic Control Bounds.** Our optimal execution problem (4.19) can be interpreted as a constrained linear quadratic control problem (LQC). In particular, if we relax the nonnegativity constraints: $u_t \geq 0,\ y_t \geq 0,\ \forall\ t = 1, \ldots, T$, then our problem is equivalent to LQC. For more discussion refer to Garleanu and Pedersen (2008).

  The value function of LQC are known in closed form, they can be used to obtain upper bounds on the execution problem and will be referred as LQC-UB. The value function in the LQC problem, by the standard results in the literature, is of the form

$$J_t^{\mathsf{LQC}}(y_{t-1}, f_t) = -\frac{1}{2} y_{t-1}^\top A_{yy}^t y_{t-1} + y_{t-1}^\top A_{yf}^t f_t + \frac{1}{2} f_t^\top A_{ff}^t f_t + \frac{1}{2} m_t, \tag{4.20}$$

  and coefficients satisfy the following recursion:

$$\begin{aligned}
A_{ff}^t &= (B + A_{yf}^{t+1}(I-\Theta))^\top \left(\Lambda + A_{yy}^{t+1}\right)^{-1} (B + A_{yf}^{t+1}(I-\Theta)) + (I-\Theta) A_{ff}^{t+1}(I-\Theta) \\
A_{yf}^t &= \Lambda(\Lambda + A_{yy}^{t+1})^{-1} \left(B + A_{yf}^{t+1}(I-\Theta)\right) \\
A_{yy}^t &= -\Lambda \left(\Lambda + A_{yy}^{t+1}\right)^{-1} \Lambda + \Lambda \\
m_t &= \mathsf{tr}(A_{ff}^{t+1}\Xi) + m_{t+1}.
\end{aligned}$$

Using the boundary condition,

$$A_{yy}^T = \Lambda, \quad A_{yf}^T = \mathsf{Zeros}(n, K), \quad A_{ff}^T = \mathsf{Zeros}(K, K), \quad m_T = 0,$$

we can determine the value function at all times. The optimal objective function of starting in state (c, f_1) is given by $J_1^{\mathsf{LQC}}(c, f_1) = -\frac{1}{2}c^\top A_{yy}^1 c + c^\top A_{yf}^1 f_1 + \frac{1}{2}f_1^\top A_{ff}^1 f_1 + \frac{1}{2}m_1$. Assuming initial factor $f_1 \sim \mathsf{N}(0, \Xi_0)$,

$$\mathsf{E}[J_1^{\mathsf{LQC}}(y_0, f_1)] = -\frac{1}{2}c^\top A_{yy}^1 c + \frac{1}{2}\mathsf{tr}(A_{ff}^1 \Xi_0) + \frac{1}{2}m_1.$$

Bounds computed using the above expression will be referred to as LQC-UB.

- **Dual Value Function Upper Bounds.**

  Brown and Smith (2010) study dynamic portfolio optimization with transaction costs and consider information relaxations to obtain bounds. They introduce value function approximation based penalties. In our problem setting, it is natural to use LQC value function to generate penalties. More discussion about this approach is available in Haugh and Lim (2011). Using the functional form of LQC value function (4.20) and applying the martingale difference operator, we obtain

$$\begin{aligned}\Delta J_t^{\mathsf{LQC}}(x_t, x_{t-1}) &\triangleq J_t^{\mathsf{LQC}}(y_{t-1}, f_t) - \mathsf{E}\left[J_t^{\mathsf{LQC}}(y_{t-1}, f_t) \middle| y_{t-2}, f_{t-1} \right] \\ &= 2\varepsilon_t^\top A_{ff}^t (I - \Phi) f_{t-1} + \varepsilon_t^\top A_{ff}^t \varepsilon_t - \mathsf{E}[\varepsilon_t^\top A_{ff}^t \varepsilon_t] + y_{t-1}^\top A_{yf}^t \varepsilon_t\end{aligned}$$

  Note that the first three terms in the expression above are mean zero and do not interact with decision variables; therefore they can be dropped. Thus, we obtain the following penalty

$$\pi^{\mathsf{DVF\text{-}UB}} = \sum_{t=2}^{T} y_{t-1}^\top A_{yf}^t \varepsilon_t,$$

  where ε_t is the factor noise. Observe that this penalty is linear in the factors and in our framework, as will become clear in the Section 4.4.3, one can think of this penalty as resulting from a basis function architecture that consists of terms linear in factors. We will refer to the bounds computed using the above defined penalty as DVF-UB.

4.4.3. Implementation Details

Basis Functions. We consider a basis function architecture that is linear in the factors i.e.

$$\Phi(f_t) = [f_{t,1}, \ldots, f_{t,L}]$$

where L is the number of factors in our model. Note that many other basis functions are possible, for instance, we can consider polynomials of the factors. Our choice was motivated by the fact that this allows us to directly compare our bounds with DVF-UB. Given Φ, our value function surrogate is

$$J_t^\kappa(y_{t-1}, f_t) = y_{t-1}^\top \Gamma_t y_{t-1} + \Phi(f_t) R_t y_{t-1} \tag{4.21}$$

Inner Problem. Applying martingale difference operator to the value function surrogate (4.21), after some algebra, we obtain

$$\Delta J_t^\kappa(x_t, x_{t-1}) = y_{t-1}^\top R_t \varepsilon_t,$$

where ε_t is the noise in the factor process. Hence the martingale penalty for our problem is $\sum_{t=2}^T \Delta J_t^\kappa(x_t, x_{t-1}) = \sum_{t=2}^T y_{t-1}^\top R_t \varepsilon_t$. Given a sample path $\omega = (\{f_1\}, \{\epsilon_2\}, \ldots, \{\epsilon_T\})$, we would like to solve the following deterministic inner optimization problem

$$\begin{aligned} \underset{y_t, u_t}{\text{maximize}} \quad & \sum_{t=1}^{T-1} y_t^\top (B f_t - R_{t+1}\varepsilon_{t+1}) - \frac{1}{2}\sum_{t=1}^{T} u_t^\top \Lambda u_t \\ \text{subject to} \quad & y_t = y_{t-1} - u_t, \ \forall\, t = 1, \ldots, T \\ & u_t \geq 0, \ y_t \geq 0, \ \forall\, t = 1, \ldots, T \\ & y_0 = c, \ y_T = 0. \end{aligned} \tag{4.22}$$

We used CPLEX 12.1.0 to solve the quadratic program.

Stochastic Subgradient[2] Method. Given a sample path $\omega = (\{f_1\}, \{\epsilon_2\}, \ldots, \{\epsilon_T\})$, we can solve formulation (4.22) to compute an optimal solution $y_t^*(\omega), u_t^*(\omega)$ and the unbiased estimate of the subgradient is given by

$$G_{R_t}(\omega) = -y_{t-1}^*(\omega)\varepsilon_t^\top(\omega), \ \forall\ 2 \leq t \leq T.$$

[2] Note that our outer problem is minimization of a convex function and hence we work with the subgradient instead of supergradient.

Sample Average Approximation Method. The inner optimization problem (4.22) can be written as a minimization program using duality.

Lemma 17. *Inner optimization problem 4.22 is equivalent to*

(4.23)
$$\min_{\nu,\eta_t\geq 0} \quad \nu^\top c + \frac{1}{2}\sum_{t=1}^{T}\left(\sum_{s=1}^{t-1}(Bf_s - R_{s+1}\varepsilon_{s+1})^\top + \eta_t^\top - \nu^\top\right)\Lambda^{-1}\left(\sum_{s=1}^{t-1}(Bf_s - R_{s+1}\varepsilon_{s+1}) + \eta_t - \nu\right)$$

Proof. Consider the formulation (4.22). Substitute $y_t = \sum_{s=t+1}^{T} u_s$ for all $t = 1, \ldots, T$. Dualize the constraint $c = \sum_{t=1}^{T} u_t$, using dual variable ν, and the constraints $u_t \geq 0$, using dual variable η_t, we obtain:

$$\underset{\nu,\eta_t\geq 0}{\text{minimize}} \ \max_{u_t} \quad \nu^\top c + \sum_{t=1}^{T} u_t^\top\left(\sum_{s=1}^{t-1}(Bf_s - R_{s+1}\varepsilon_{s+1}) + \eta_t - \nu\right) - \frac{1}{2}\sum_{t=1}^{T} u_t^\top \Lambda u_t.$$

Maximization with respect to u_t, yields $u_t^* = \Lambda^{-1}\left(\sum_{s=1}^{t-1}(Bf_s - R_{s+1}\varepsilon_{s+1}) + \eta_t - \nu\right)$ for $1 \leq t \leq T$. Substituting u_t^* we obtain desired conclusion. ■

Thus, our convex optimization problem can be written as,

$$\min_{R_t} \ \hat{\mathsf{E}}\left[\min_{\nu,\eta_t\geq 0} \ \nu^\top c + \frac{1}{2}\sum_{t=1}^{T}\left(\sum_{s=1}^{t-1}(Bf_s - R_{s+1}\varepsilon_{s+1})^\top + \eta_t^\top - \nu^\top\right)\Lambda^{-1}\left(\sum_{s=1}^{t-1}(Bf_s - R_{s+1}\varepsilon_{s+1}) + \eta_t - \nu\right)\right]$$

where $\hat{\mathsf{E}}$ is expectation with respect to empirical distribution over S independently generated sample paths $\omega_i = (\{f_1^{(i)}\}, \{\epsilon_2^{(i)}\}, \ldots, \{\epsilon_T^{(i)}\})$ for $i = 1, \ldots, S$.

4.4.4. Results

We make use the experiment setup described in Moallemi and Sağlam (2011). Consider a block of $c = 100$ stocks of AAPL that need to be liquidated over a short trading horizon. There is an opportunity to trade every 5 minutes. Using 'momentum' and 'value' type signal, they construct a factor model with $L = 2$ return predicting factors. Using the stock data of AAPL from January 4, 2010 and January 5, 2010, the following model was estimated via pooled regression.

- Factor loadings: $B = [0.3375 \quad -0.0720]$.
- Variance of noise in returns: $\Sigma = 0.0428$.
- Returns: $r_{t+1} = 0.0726 + 0.3375 f_{1,t} - 0.0720 f_{2,t} + \epsilon_{t+1}$, where $\epsilon_t \sim \mathsf{N}(0, \Sigma)$.
- Transaction cost matrix assumed proportional to variance of return noise Σ, i.e. $\Lambda = \lambda\Sigma$, where $\lambda = 0.5$.
- Matrix of mean reversion coefficients:

$$\Theta = \begin{bmatrix} 0.0353 & 0 \\ 0 & 0.7146 \end{bmatrix}.$$

- Factor model:

$$\Delta f_{1,t+1} = -0.0353 f_{1,t} + \varepsilon_{1,t}$$
$$\Delta f_{2,t+1} = -0.7146 f_{2,t} + \varepsilon_{2,t},$$

  where $\varepsilon_t \sim \mathsf{N}(\mathbf{0}, \Xi)$ and

$$\Xi = \begin{bmatrix} 0.0378 & 0 \\ 0 & 0.0947 \end{bmatrix}$$

- Initial factor f_1 sampled from the stationary distribution. Thus, $f_1 \sim \mathsf{N}(\mathbf{0}, \Xi_0)$, where

$$\Xi_0 = \sum_{t=0}^{\infty} (I - \Theta)^t \Xi (I - \Theta)^t = \begin{bmatrix} 0.0412 & 0 \\ 0 & 1.3655 \end{bmatrix}$$

We solve the pathwise problem (4.12) using the stochastic subgradient approach described in Section 4.3.1. In our implementation we made the following parameter choices:

- Step-size rule: $A = 1$ and $B = 0$.
- Outer iteration limit $N = 1,000$.
- Number of samples used to estimate gradient $I = 100$.

We used $1,000,000$ sample paths to estimate unbiased pathwise upper bound and benchmark bounds with reasonably small standard errors.

Table 4.1 reports upper bounds on the optimal value generated by pathwise method and the benchmark algorithms. We compare the algorithms under three different parameter settings: (a), referred to as base case, uses parameters described above, (b) uses the base case parameters with variance of factor noise scaled up by a factor of 5 and transaction costs scaled down by a factor of 10 and the last case (c) uses base case parameters with variance of factor noise scaled up by a factor of 10 and transaction cost scaled down by a factor of 100. For each case we consider the values of $T = 12, 30, 60, 120, 240$. It is clear that among the benchmark methods, DVF-UB is the tightest. Observe, that pathwise upper bound (will be referred to as PATHWISE-UB) is consistently better than DVF-UB.

Table 4.1 (d) compares PATHWISE-UB with DVF-UB. We observe that PATHWISE-UB shows greater improvement over DVF-UB, when the number of time periods, T, increases. Further, as the variance of factor noise increases and transaction costs are reduced, we observe PATHWISE-UB offers more benefit over DVF-UB. Typically, PATHWISE-UB offers an *improvement of about* 5% over DVF-UB. In terms of computation time, PATHWISE-UB takes slightly longer time than DVF-UB. However, it is more by only about 10%, a very manageable increase in computational burden.

Moallemi and Sağlam (2011) obtain a lower bound, by simulating a policy, for the base case with $T = 12$. Their lower bound is 6.12 with a standard error of 0.224. Our upper bound for this case, shown in Table 4.1 (a), is 6.46 with a standard error of 0.04. This suggests that the upper bounds obtained by PATHWISE-UB can be fairly tight.

4.5. Theory

We compare bounds obtained using pathwise method to upper bounds derived from other approximate dynamic programming based methods.

(a) Performance and computation times of algorithms for *base case* parameters itemized at the beginning of Section 4.4.4

		Pathwise UB			**DVF-UB**			**PH-UB**			**LQC-UB**
T	$T_{\mathsf{comp.}}$	Mean	S.E.	$T_{\mathsf{sim.}}$	Mean	S.E	$T_{\mathsf{sim.}}$	Mean	S.E	$T_{\mathsf{sim.}}$	Value
12	21	6.46	0.04	195	6.48	0.04	187	8.48	0.05	199	12.59
30	27	56.32	0.10	259	57.52	0.09	254	65.48	0.12	251	236.5545
60	39	121.18	0.16	356	126.05	0.14	350	147.45	0.21	358	1.25E+03
120	64	212.38	0.21	573	222.29	0.18	580	281.94	0.35	575	5.14E+03
240	119	322.53	0.26	1066	333.57	0.21	1044	490.04	0.54	1065	1.64E+04

(b) Performance and computation times of algorithms for base case parameters with variance of factor noise scaled up by a factor of 5 and transaction cost scaled down by factor of 10.

		Pathwise UB			**DVF-UB**			**PH-UB**			**LQC-UB**
T	$T_{\mathsf{comp.}}$	Mean	S.E.	$T_{\mathsf{sim.}}$	Mean	S.E	$T_{\mathsf{sim.}}$	Mean	S.E	$T_{\mathsf{sim.}}$	Value
12	20	75.49	0.11	193	76.96	0.10	190	80.24	0.12	197	1.07E+03
30	29	186.01	0.23	258	194.51	0.21	257	204.46	0.27	268	1.20E+04
60	40	333.32	0.37	375	354.51	0.30	357	384.69	0.48	381	6.26E+04
120	65	543.56	0.50	588	578.64	0.40	596	681.74	0.79	581	2.57E+05
240	120	798.68	0.60	1082	835.87	0.48	1073	1143.55	1.21	1107	8.21E+05

(c) Performance and computation times of algorithms for base case parameters with variance of factor noise scaled up by a factor of 10 and transaction cost scaled down by factor of 100.

		Pathwise UB			**DVF-UB**			**PH-UB**			**LQC-UB**
T	$T_{\mathsf{comp.}}$	Mean	S.E.	$T_{\mathsf{sim.}}$	Mean	S.E	$T_{\mathsf{sim.}}$	Mean	S.E	$T_{\mathsf{sim.}}$	Value
12	20	118.57	0.15	206	120.73	0.14	191	125.01	0.17	194	2.15E+04
30	27	275.00	0.33	275	287.23	0.29	264	300.39	0.39	249	2.40E+05
60	44	483.70	0.52	484	513.82	0.43	347	554.98	0.68	352	1.25E+06
120	69	781.54	0.71	605	831.10	0.56	606	974.75	1.11	569	5.14E+06
240	128	1143.26	0.85	1053	1195.10	0.68	1067	1627.53	1.71	1120	1.64E+07

(d) Comparison of PATHWISE-UB with DVF-UB across above three cases.

T	Case (a)	Case (b)	Case (c)
12	0.40%	1.94%	1.83%
30	2.13%	4.57%	4.45%
60	4.01%	6.36%	6.23%
120	4.67%	6.45%	6.34%
240	3.42%	4.66%	4.53%

Table 4.1: Comparison of bounds by Pathwise method with benchmark algorithms and the computation times. T refers to the number of trading opportunities. $T_{\mathsf{comp.}}$ and $T_{\mathsf{sim.}}$ refer to computation and simulation time, respectively.

First, we consider the *approximate linear programming* (ALP) approach. For our optimization problem (4.4), the Bellman operator $T\colon \mathcal{P} \to \mathcal{P}$ can be defined as

$$(TJ)_t(x) = \begin{cases} \min_u \{g_t(x,u) + \mathsf{E}[J_{t+1}(x_{t+1})|x_t = x, u]\} & \text{if } t < T, \\ \min_u g_T(x,u) & \text{if } t = T. \end{cases} \tag{4.24}$$

The optimal solution J^* to the problem (4.4) can be characterized as a fixed point to the Bellman's equation: $TJ = J$. Consider the basis function architecture $\Psi = \{\psi_1, \ldots, \psi_K\}$, where $\psi_i \in \mathcal{P},\ \forall\ 1 \le i \le K$. We are interested in computing $r \in \mathbb{R}^K$ so that

$$J_t^*(x) \approx J_t^r(x) \triangleq \sum_{i=1}^{K} \psi_i(x,t) r_i = \Psi r_t(x)$$

We are ready to state the ALP for our problem:

$$\begin{aligned} \underset{r}{\text{maximize}} \quad & \Psi r_0(x) \\ \text{subject to} \quad & \Psi r(x) \le T\Psi r(x)\ \forall\ x \in \mathbb{R}^{m+\ell} \end{aligned} \tag{4.25}$$

Denote the set of feasible r in (4.25) by $\mathcal{C}_{\mathsf{ALP}}$. Notice that the number of constraints is infinite and it is not clear if we can solve this program in general. However, based on this idea, tractable programs have been introduced for certain special cases (see, Wang and Boyd, 2011). The following result shows that the bounds generated by PO based approach are provably tighter than the ALP based bounds.

Theorem 11. *Suppose $r \in \mathcal{C}_{ALP}$ is feasible for the ALP* (4.25)*. Then, for all $x \in \mathbb{R}^{m+\ell}$,*

$$\Psi r_0(x) \le F\Psi r(x) \le J_0^*(x).$$

Proof. By weak duality, we have

$$\begin{aligned} J_0^*(x) &\ge F\Psi r(x) \\ &= \mathsf{E}\left[\inf_{\mathbf{u}_T \in \mathcal{A}} g_0(x_0,u_0) + \sum_{t=1}^{T} \Big(g_t(x_t,u_t) - (\Psi r_t(x_t) - \mathsf{E}[\Psi r_t(x_t)|x_{t-1},u_{t-1}]) \Big) \,\middle|\, x_0 = x \right] \\ &\ge \mathsf{E}\Bigg[\inf_{\mathbf{u}_T \in \mathcal{A}} \Psi r_0(x_0) + \sum_{t=1}^{T} \Big(g_t(x_t,u_t) + \mathsf{E}[\Psi r_{t+1}(x_{t+1})|x_t,u_t] - \Psi r_t(x_t) \Big) \\ &\qquad\qquad + g_T(x_T,u_T) - \Psi r_T(x_T) \,\Bigg|\, x_0 = x \Bigg] \\ &\ge \Psi r_0(x). \end{aligned}$$

The last inequality follows (a) $\Psi r_t(x) \leq \min_{u_t} \{g_t(x, u_t) + \mathsf{E}[\Psi r_{t+1}(x_{t+1})|x_t = x, u_t]\}$ for $0 \leq t \leq T-1$, and (b) $\Psi r_T(x_T) \leq \min_{u_T} g_T(x_T, u_T)$. ■

Thus, we can interpret the ALP as finding the tightest possible lower bound $\Psi r_0(x)$ for $r \in \mathcal{C}_{\mathsf{ALP}}$ and by Theorem 11 the PO bounds are pointwise tighter than ALP.

Next, we compare PO based bounds with the bounds introduced by Wang and Boyd (2011). Their dynamics in our framework can be written as $y_{t+1} = A_t y_t + C_t u_t + \epsilon_{t+1},\ 0 \leq t \leq T-1$. Notice that they do not allow for exogenous z_t. They consider a cost function that is separable across time and the stage cost at time t is $g_t \colon \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is a function of the current state y_t and the control u_t and is not required to be convex. Given these dynamics and cost function, the optimization problem can be written as

$$J_0^*(y) = \underset{u_0,\ldots,u_T}{\text{minimize}}\ \mathsf{E}\left[\sum_{t=0}^{T} g_t(y_t, u_t) \middle| y_0 = y\right], \tag{4.26}$$

where the decisions $u_0, \ldots, u_T$ are nonanticipative. Their method relies on following observations:

1. Suppose stage cost $g_t(y_t, u_t)$ is quadratic (but not necessarily convex), then lower bounds on the optimal objective value can be constructed by solving ALP (4.25). In particular, for a quadratic value function approximation, the Bellman inequality can be written as an LMI and hence can be solved efficiently.

2. For a general stage cost (i.e., not necessarily convex), if we are able to find a quadratic stage cost function that are lower bounds to the actual stage cost, then using ALP approach lower bounds on the optimal objective value can be generated.

For certain types of stage cost and constraints on controls, the above two steps can be automated so that tightest possible bounds afforded by the above outlined approach can be computed. We can establish the following result about pathwise method.

Theorem 12. *The lower bounds generated by the pathwise method are tighter than the bounds generated by Wang and Boyd (2011) approach.*

Proof. Let the quadratic functions identified by Wang and Boyd (2011) procedure be $\ell_t(y_t, u_t)$ for all $t \in \mathcal{T}$ and the value function approximation be $J_t^\ell(y) = y^\top \tilde{P}_t y + 2\tilde{p}_t^\top y + \tilde{c}_t$ for all $t \in \mathcal{T}$. Then, $\ell_t(y_t, u_t) \leq g_t(y_t, u_t)$ for all $y_t \in \mathbb{R}^m$, $u_t \in \mathbb{R}^n$ and $0 \leq t \leq T$.

$$
\begin{aligned}
J_0^\ell(y) &\overset{(a)}{\leq} FJ^\ell(y) \\
&= \mathsf{E}\left[\inf_{u_0,\ldots,u_T} \ell_0(y_0, u_0) + \sum_{t=1}^{T} \ell_t(y_t, u_t) - \Delta J_t^\ell(y_t, y_{t-1}, u_{t-1}) \;\middle|\; y_0 = y \right] \\
&\overset{(b)}{\leq} \mathsf{E}\left[\inf_{u_0,\ldots,u_T} g_0(y_0, u_0) + \sum_{t=1}^{T} g_t(y_t, u_t) - \Delta J_t^\ell(y_t, y_{t-1}, u_{t-1}) \;\middle|\; y_0 = y \right] \\
&\overset{(c)}{\leq} \sup_{\kappa} FJ^\kappa(y),
\end{aligned}
$$

(a) follows by Theorem 11 (b) is true because $(\ell_0, \ldots, \ell_T)$ are lower bounds on the stage cost and (c) holds because J^ℓ is in the span of the parametric value functions determined by $\kappa = \{P_t, p_t, c_t, \forall t \in \mathcal{T}\}$. ■

5

CONCLUSION

Approximate Dynamic Programming is a growing field that holds a lot of potential in addressing large scale stochastic dynamic programs. On the one hand, it provides techniques for generating suboptimal policies for complex problems in sequential decision making under uncertainty. On the other, in the recent years, dual approaches provide a framework that can be used to quantify the suboptimality. In this thesis, we have sought to further our understanding in both these aspects.

The ALP approach to approximate DP is interesting at the outset for two reasons. First, it gives us the ability to leverage commercial linear programming software to solve large ADP problems, and second, the ability to prove rigorous approximation guarantees and performance bounds. We asked whether the formulation considered in the ALP approach was the ideal formulation. In particular, we asked whether certain restrictions imposed on approximations produced by the approach can be relaxed in a tractable fashion and whether such a relaxation has a beneficial impact on the quality of the approximation produced. We have answered both of these questions in the affirmative.

Further, we focused on a general method for computing bounds, namely, martingale duality approach. This approach, referred to as pathwise method, was developed in Chapter 3 in the context of optimal stopping problems and later generalized to MDPs in Chapter 4. Pathwise methods provides a structured procedure for obtaining tightest possible bounds afforded by the chosen basis functions. These methods, together with the methods for generating suboptimal policy, like the SALP, can provide 'confidence' bounds for the true optimal solution.

There are a number of directions that merit further investigation. We highlight some below:

- **Online Method for SALP.** SALP may be written as an unconstrained stochastic optimization problem given by (2.30). Such problems suggest natural *online* update rules for the weights r, based on stochastic gradient methods, yielding 'data-driven' ADP methods. The menagerie of online ADP algorithms available at present are effectively iterative methods for solving a projected version of Bellman's equation. TD-learning is a good representative of this type of approach and, as can be seen from Table 2.1, is not among the highest performing algorithms in our computational study. An online update rule that effectively solves the SALP promises policies that will perform on par with the SALP solution, while at the same time retaining the benefits of an online ADP algorithm.

- **Bootstrapping.** In our implementation of the SALP procedure, we start with a baseline policy. The sampled SALP is obtained by sampling states from the stationary distribution of the baseline policy. Suppose that the policies resulting from solution to SALP are superior to the baseline policy. Then, it is natural to consider sampling a new set of states from the improved policy. This procedure can repeated iteratively to obtain better policies over the course of iterations. Understanding the dynamics of such a 'bootstrapping' procedure can lead to iterative refinement of the policies.

- **Policy Generation.** The dual approach to solving stochastic dynamic programs, as discussed in Chapters 3 and 4, allows us to generate bounds. However, in practice, along with bounds we are interested in policies. In the case of optimal stopping problem, policy was obtained by regressing upper bounds on continuation value. It is natural to ask whether a more direct method is possible — for instance, the greedy policy with respect to the value function surrogate. This appears to be a non-trivial question. In particular, it is not hard to see that if the constant function were a basis function, then the pathwise method cannot identify a unique optimal coefficient for this basis

function. On the other hand, if one chose to use a policy that were greedy with respect to value function surrogate, it is clear that the coefficient corresponding to this basis function can dramatically alter the nature of the policy.

BIBLIOGRAPHY

D. Adelman. A price-directed approach to stochastic inventory/routing. *Operations Research*, 52(4):499–514, 2004.

D. Adelman. Dynamic bid prices in revenue management. *Operations Research*, 55(4): 647–661, 2007.

D. Adelman and D. Klabjan. Computing near optimal policies in generalized joint replenishment. Working paper, January 2009.

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

D. Belomestny, C. Bender, and J. Schoenmakers. True upper bounds for Bermudan products via non-nested Monte Carlo. *Mathematical Finance*, 19(1):53–71, 2009.

D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 1995.

D. P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 2. Athena Scientific, Belmont, MA, 3rd edition, 2007.

D. P. Bertsekas and S. Ioffe. Temporal differences–based policy iteration and applications in neuro–dynamic programming. Technical Report LIDS–P–2349, MIT Laboratory for Information and Decision Systems, 1996.

D. P. Bertsekas and S. E. Shreve. *Stochastic Optimal Control: The Discrete Time Case*. Athena Scientific, Belmont, MA, 1996.

D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA, 1996.

V. S. Borkar, J. Pinto, and T. Prabhu. A new learning algorithm for optimal stopping. *Discrete Event Dynamic Systems*, 19(1):91–113, 2009.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.

M. Broadie and M. Cao. Improved lower and upper bound algorithms for pricing American options by simulation. *Quantitative Finance*, 8(8):845–861, 2008.

D. B. Brown and J. E. Smith. Dynamic portfolio optimization with transaction costs: Heuristics and dual bounds. *Management Science*, Forthcoming, 2010.

D. B. Brown, J. E. Smith, , and P. Sun. Information relaxations and duality in stochastic dynamic programs. *Operations Research*, 58(4):785–801, July-August 2010.

J. Brzustowski. Can you win at Tetris? Master's thesis, University of British Columbia, 1992.

H. Burgiel. How to lose in Tetris. *Mathematical Gazette*, page 194, 1997.

J. F. Carriere. Valuation of the early-exercise price for derivative securities using simulations and splines. *Insurance: Math. Econom.*, 19:19–30, 1996.

N. Chen and P. Glasserman. Additive and multiplicative duals for American option pricing. *Finance and Stochastics*, 11(2):153–179, 2007.

E. Clément, D. Lamberton, and P. Protter. An analysis of a least squares regression method for American option pricing. *Finance and Stochastics*, 6(4):449–471, 2002.

J. G. Dai and W. Lin. Maximum pressure policies in stochastic processing networks. *Operations Research*, 53:197–218, 2005.

M. Davis and I. Karatzas. A deterministic approach to optimal stopping. In F. P. Kelly, editor, *Probability, Statistics and Optimization: A Tribute to Peter Whittle*, pages 455–466. J. Wiley and Sons, 1994.

D. P. de Farias and B. Van Roy. On the existence of fixed points for approximate value iteration and temporal-difference learning. *Journal of Optimization Theory and Applications*, 105(3), 2000.

D. P. de Farias and B. Van Roy. The linear programming approach to approximate dynamic programming. *Operations Research*, 51(6):850–865, 2003.

D. P. de Farias and B. Van Roy. On constraint sampling in the linear programming approach to approximate dynamic programming. *Mathematics of Operations Research*, 293(3):462–478, 2004.

D. P. de Farias and B. Van Roy. A cost-shaping linear program for average-cost approximate dynamic programming with performance guarantees. *Mathematics of Operations Research*, 31(3):597–620, 2006.

E. D. Demaine, S. Hohenberger, and D. Liben-Nowell. Tetris is hard, even to approximate. In *Proceedings of the 9th International Computing and Combinatorics Conference*, 2003.

V. V. Desai, V. F. Farias, and C. C. Moallemi. Approximate dynamic programming via a smoothed linear program. *Submitted*, 2009. Working paper.

V. V. Desai, V. F. Farias, and C. C. Moallemi. Pathwise optimization for optimal stopping problems. *Submitted*, 2010.

V. F. Farias and B. Van Roy. Tetris: A study of randomized constraint sampling. In *Probabilistic and Randomized Methods for Design Under Uncertainty*. Springer-Verlag, 2006.

V. F. Farias and B. Van Roy. An approximate dynamic programming approach to network revenue management. Working paper, 2007.

V. F. Farias, D. Saure, and G. Y. Weintraub. An approximate dynamic programming approach to solving dynamic oligopoly models. Working paper, 2011.

J. A. Fill. Eigenvalue bounds on convergence to stationarity for nonreversible Markov chains, with an application to the exclusion process. *The Annals of Applied Probability*, 1(1):62–87, February 1991.

Nicolae Garleanu and Lasse Heje Pedersen. Dynamic trading with predictable returns and transaction costs. *Submitted*, 2008.

P. Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer-Verlag, 2004.

P. Glasserman and B. Yu. Simulation for American options: Regression now or regression later? In H. Niederreiter, editor, *Monte Carlo and Quasi-Monte Carlo Methods*, pages 213–226. Springer Verlag, 2002.

J. Han. *Dynamic Portfolio Management - An Approximate Linear Programming Approach*. PhD thesis, Stanford University, 2005.

J. M. Harrison. Brownian models of queueing networks with heterogeneous customer populations. In *Stochastic Differential Systems, Stochastic Control Theory and Applications (Minneapolis, Minn., 1986)*, volume 10 of *IMA Vol. Math. Appl.*, pages 147–186. Springer, New York, 1988.

J. M. Harrison and L. M. Wein. Scheduling network of queues: Heavy traffic analysis of a simple open network. *Queueing Systems*, 5:265–280, 1989.

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

M. B. Haugh and Andrew Lim. Linear-quadratic control and information relaxations. *Working Paper*, 2011.

D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100:78–150, 1992.

Steven L. Heston, Robert A. Korajczyk, Ronnie Sadka, and Lewis D. Thorson. Are you trading predictably? *SSRN eLibrary*, 2010.

F. Jamshidian. Minimax optimality of Bermudan and American claims and their Monte-Carlo upper bound approximation. Technical report, NIB Capitial, The Hague, 2003.

S. Kakade. A natural policy gradient. In *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.

S. Kumar and K. Muthuraman. A numerical method for solving singular stochastic control problems. *Operations Research*, 52(4):563–582, 2004.

H. J. Kushner and L. F. Martins. Heavy traffic analysis of a controlled multiclass queueing network via weak convergence methods. *SIAM J. Control Optim.*, 34(5):1781–1797, 1996.

G. Lai, F. Margot, and N. Secomandi. An approximate dynamic programming approach to benchmark practice-based heuristics for natural gas storage valuation. *Operations research*, 58(3):564–582, 2010a.

Guoming Lai, Mulan X. Wang, Sunder Kekre, Alan Scheller-Wolf, and Nicola Secomandi. Valuation of storage at a liquefied natural gas terminal. *Operations Research*, Forthcoming, 2010b.

F. A. Longstaff and E. S. Schwartz. Valuing american options by simulation: A simple least-sqaures approach. *The Review of Financial Studies*, 14(1):113–147, 2001.

A. S. Manne. Linear programming and sequential decisions. *Management Science*, 60(3): 259–267, 1960.

L. F. Martins, S. E. Shreve, and H. M. Soner. Heavy traffic convergence of a controlled multiclass queueing network. *SIAM J. Control Optim.*, 34(6):2133–2171, 1996.

C. C. Moallemi and Mehmet Sağlam. Dynamic portfolio choice with transaction costs and return predictability: Linear rebalancing rules. *Working Paper*, 2011.

C. C. Moallemi, S. Kumar, and B. Van Roy. Approximate and data-driven dynamic programming for queueing networks. Working paper, 2008.

R. Montenegro and P. Tetali. *Mathematical Aspects of Mixing Times in Markov Chains*. Foundations and Trends in Theoretical Computer Science. NOW Publishers, Boston-Delft, June 2006.

J. R. Morrison and P. R. Kumar. New linear program performance bounds for queueing networks. *Journal of Optimization Theory and Applications*, 100(3):575–597, 1999.

M. Petrik and S. Zilberstein. Constraint relaxation in approximate linear programs. In *Proceedings of the 26th International Conference on Machine Learning*, Montreal, Canada, 2009.

W. B. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. John Wiley and Sons, 2007.

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

L. C. G. Rogers. Pathwise stochastic optimal control. *SIAM Journal on Control and Optimization*, 46(3):1116–1132, 2008.

L. C. G. Rogers. Dual valuation and hedging of Bermudan options. *SIAM Journal on Financial Mathematics*, 1:604–608, 2010.

P. Schweitzer and A. Seidmann. Generalized polynomial approximations in Markovian decision processes. *Journal of Mathematical Analysis and Applications*, 110:568–582, 1985.

A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on Stochastic Programming: Modeling and Theory*. SIAM, Philadelphia, PA, 2009.

I. Szita and A. Lőrincz. Learning Tetris using the noisy cross-entropy method. *Neural Computation*, 18:2936–2941, 2006.

L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 37(12):1936–1948, December 1992.

L. Tassiulas and A. Ephremides. Dynamic server allocation to parallel queues with randomly varying connectivity. *IEEE Transactions on Information Theory*, 39(2):466–478, March 1993.

H. Topaloglu. Using Lagrangian relaxation to compute capacity-dependent bid prices in network revenue management. *Operations Research*, 2009. To appear.

J. N. Tsitsiklis and B. Van Roy. Optimal Stopping of Markov Processes: Hilbert Space Theory, Approximation Algorithms, and an Application to Pricing High-Dimensional Financial Derivatives. *IEEE Transactions on Automatic Control*, 44(10):1840–1851, 1999.

J. N. Tsitsiklis and B Van Roy. Regression methods for pricing complex american-style options. *IEEE Transactions on Neural Networks*, 12(4):694–703, July 2001.

B. Van Roy. Neuro-dynamic programming: Overview and recent trends. In A. Shwartz E. Feinberg, editor, *Handbook of Markov Decision Processes*. Kluwer, Boston, 2002.

B. Van Roy. On Regression-Based Stopping Times. *Discrete Event Dynamic Systems*, 20 (3):307–324, 2010.

M. H. Veatch. Approximate dynamic programming for networks: Fluid models and constraint reduction. Working paper, 2005.

Y. Wang and S. Boyd. Performance bounds and suboptimal policies for linear stochastic control via lmis. *International Journal of Robust and Nonlinear Control*, 2011.

Z. Wen, L. J. Durlofsky, B. Van Roy, and K. Aziz. Use of approximate dynamic programming for production optimization. To appear in *Society of Petroleum Engineerins Proceedings*, 2011.

H. Yu and D. P. Bertsekas. A least squares q-learning algorithm for optimal stopping problems. Technical Report LIDS REPORT 2731, Laboratory for Information and Decision Systems, M.I.T., June 2007.

D. Zhang and D. Adelman. An approximate dynamic programming approach to network revenue management with customer choice. Working paper, 2008.