

**Self-guided Approximate Linear Programs:
Randomized Multi-shot Approximation of Markov Decision Processes**

by

PARSHAN PAKIMAN
B.S., University of Tehran, 2016

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Business Administration
in the Graduate College of the
University of Illinois Chicago, 2023

Chicago, Illinois

Defense Committee:

Selvaprabu Nadarajah, Chair and Advisor
Boxiao Chen,
Negar Soheili,
Daniel Adelman, University of Chicago
Itai Gurvich, Northwestern University

Copyright by
PARSHAN PAKIMAN
2023

To my lovely wife and my wonderful parents.

ACKNOWLEDGMENT

The best way to begin my doctoral thesis is to express my deepest and most sincere gratitude to Selva Nadarajah, my advisor, mentor, co-author, and close friend. His generosity with his time and patience has been priceless to me, and his unwavering support and encouragement were indispensable, without which this research would not have been possible. I am profoundly grateful for the opportunity to be trained and mentored by Selva. Words cannot adequately convey my appreciation for all he has done for me. Additionally, I am grateful to Negar Soheili for her valuable guidance and insights that shaped my Ph.D. journey. Negar and Selva always cared about my academic growth and personal well-being, which has been remarkably meaningful to me. I am thankful to both of them for their support and mentorship.

During my doctoral studies, I had the privilege of collaborating with incredible scholars. Qihang Lin provided valuable guidance, particularly in strengthening proofs and streamlining their understanding. I am privileged to work with Beryl Chen and Stefanus Jasin. They encouraged me to work on a research project beyond my doctoral thesis, greatly enhancing my overall research experience. I was fortunate to also collaborate with Abhilash Chenreddy and Ranga Chandrasekaran on an industry-based research project resulting in my first conference paper. Last but certainly not least, I had the pleasure of collaborating with Yun Fong Lim, whose research insights and encouragement have been valuable in my academic growth.

ACKNOWLEDGMENT (Continued)

I express my deep gratitude to all my doctoral defense committee members, Dan Adelman, Beryl Chen, Itai Gurvich, Selva Nadarajah, and Negar Soheili, for their valuable feedback and insightful comments that significantly enhanced the quality of this thesis. I consider myself fortunate to have been guided by such a supportive and knowledgeable group of faculty members. Thank you for your time and thoughtful suggestions.

I am thankful to the Information and Decision Sciences family at the University of Illinois Chicago for their support and kindness. Your welcoming presence facilitated my academic growth. Thank you for creating such a warm and supportive research environment.

To Homai, my sweet wife, most supportive partner, and soulmate, I am at a loss for words to express my gratitude for everything you have done for me. I cannot imagine where I would be today without you by my side. Your patience and emotional support have been my rock throughout my Ph.D. journey. In addition, your insights and thought-provoking questions helped me immensely with refining this research. I am endlessly thankful for your presence in my life and for everything you have given me, my love.

I am deeply grateful to my parents (Ahmad Pakiman and Maryam Lak), who supported me from thousands of miles away. My brother (Koushan Pakiman) brought immense happiness, fun, and joy into my life, and I am truly grateful for his presence. My wonderful friends and family members, I thank you for your patience with me during times of heavy workload. I will forever hold the memories of those family members (Fariba, Mamani, and Baba-Bozorg) I lost during my Ph.D. and was unable to reunite with.

Parshan Pakiman

CONTRIBUTIONS OF AUTHORS

Chapter 1 outlines the overarching goal of my thesis and summarizes its contributions. Chapter 2 contains the content of a manuscript titled “Self-Guided Approximate Linear Programs: Randomized Multi-Shot Approximation of Discounted Cost Markov Decision Processes,” which is currently under minor review at Management Science. The co-authors of this work are Selva Nadarajah, Negar Soheili, and Qihang Lin, with me serving as the lead author who has done the majority of the work. My advisor, Dr. Nadarajah, defined the research question as part of my first-year summer research paper and subsequently helped with ideation, technical development, and writing. Drs. Soheili and Lin have helped refine the theory and provided feedback to improve the paper. Chapter 3 describes a working paper titled “Randomized Multi-Shot Approximation of Average Cost Markov Decision Processes.” Dr. Nadarajah and I co-authored this paper. I am the lead author and have done the majority of the work. Dr. Nadarajah helped with the ideation and technical development aspects and also provided valuable feedback on the exposition. Each chapter required the development of code to solve large-scale optimization models, which was done solely by me. All code has been open-sourced and can be accessed through the following repository: <https://github.com/Multi-Shot-Approximation-of-MDPs>.

TABLE OF CONTENTS

<u>CHAPTER</u>		<u>PAGE</u>
1	INTRODUCTION	1
2	SELF-GUIDED APPROXIMATE LINEAR PROGRAMS: RANDOMIZED MULTI-SHOT APPROXIMATION OF DISCOUNTED COST MARKOV DECISION PROCESSES	11
2.1	Introduction	12
2.2	Exact Mathematical Programs	19
2.2.1	Background	19
2.2.2	Feature-based Exact Program	22
2.3	Randomized One-Shot Approximation	25
2.3.1	Model and Theory	26
2.3.2	Implementation Guidelines	31
2.4	Randomized Multi-Shot Approximation	36
2.4.1	Model and Algorithm	37
2.4.2	Understanding the Self-guiding Mechanism	40
2.4.3	Theoretical Guarantees	44
2.4.4	Implementation Guidelines	48
2.5	Extensions	49
2.6	Perishable Inventory Control	52
2.6.1	MDP Formulation and Instances	52
2.6.2	Computational Setup	55
2.6.3	Results	56
2.7	Bermudan Options Pricing	61
2.7.1	MDP Formulation	62
2.7.2	Computational Setup and Benchmarks	63
2.7.3	Results	64
2.8	Conclusions	66
2.9	Proofs	69
2.9.1	Additional Details of Assumption 1	69
2.9.2	Proofs of Statements in §2.2	70
2.9.3	Proofs of Statements in §2.3	70
2.9.4	Proofs of Statements in §2.4	82
2.9.5	Proofs of Statements in §2.5	89
2.10	Relaxing Assumptions	91
2.10.1	Relaxing Assumption of $V^* \in \mathcal{R}$	91
2.10.2	Relaxing Assumption 3	92

TABLE OF CONTENTS (Continued)

<u>CHAPTER</u>		<u>PAGE</u>
	2.11 Constraint Sampling Bound for Self-guided FALP	95
	2.12 Optimistic Bound Estimation	97
	2.12.1 Constraint Violation Learning	97
	2.12.2 Information Relaxation and Duality	101
	2.13 Addendum to Numerical Study	103
	2.13.1 Visualization of Self-guiding Mechanism	104
	2.13.2 Analyzing the Impact of Constraint Sampling on Policy-guided FALP	107
	2.13.3 Analyzing ReLU Basis Functions	109
	2.13.4 Upper and Lower Bound Values	113
3	RANDOMIZED MULTI-SHOT APPROXIMATION OF AVERAGE COST MARKOV DECISION PROCESSES	119
	3.1 Introduction	120
	3.1.1 Contributions	124
	3.1.2 Related work	126
	3.2 Markov Decision Processes	128
	3.3 Bound-Focused Programs	132
	3.3.1 Bound-Focused Exact Linear Program	133
	3.3.2 Bound-Focused Feature-based Exact Program	134
	3.3.3 Bound-Focused Approximate Linear Program	138
	3.4 Policy-Focused Programs	145
	3.4.1 Policy Performance Bound	145
	3.4.2 Discounted-cost Approach to Average-Cost MDPs	147
	3.4.3 Policy-Focused Exact Programs	151
	3.4.4 Policy-Focused Approximate Linear Program	155
	3.5 Algorithm	160
	3.6 Generalized Joint Replenishment	165
	3.6.1 Constraint Generation for Stump Basis Functions	168
	3.6.2 Instances and Computational Setup	171
	3.6.3 Results	172
	3.7 Perishable Inventory Control Problem	174
	3.7.1 Instances and Benchmarks	174
	3.7.2 Results	176
	3.8 Conclusion	178
	3.9 Addendum to Assumption 7	180
	3.10 Proofs	182
	CITED LITERATURE	196
	VITA	202

LIST OF TABLES

<u>TABLE</u>		<u>PAGE</u>
I	Examples of universal random basis functions.	24
II	Comparison of ALP^{LNS} and FALP on the three-dimensional perishable inventory control instances ($\sigma = 2$ and $c_1 = 100$).	57
III	Comparison of ALP^{LNS} , FALP, policy-guided FALP, and self-guided FALP on the five-dimensional perishable inventory control instances ($\gamma = 0.95$ and $c_1 = 1000$).	58
IV	Comparison of ALP^{LNS} , FALP, and self-guided FALP on the ten-dimensional perishable inventory control instances ($\gamma = 0.95$ and $c_1 = 1000$).	59
V	Comparison of optimality gaps on the Bermudan options pricing application.	65
VI	Comparison of the effect of different constraint sampling strategies on policy-guided FALP (extended version of Table III).	109
VII	Comparison of ReLU FALP and Fourier FALP on the three-dimensional perishable inventory control instances ($\sigma = 2$ and $c_1 = 100$).	110
VIII	Comparison of ReLU FALP and ReLU self-guided FALP with Fourier FALP and Fourier self-guided FALP on the Bermudan options pricing instances.	112
IX	Lower bound and upper bounds used to compute optimality and lower-bound gaps in Table II.	115
X	Lower bound and upper bounds used to compute optimality and lower-bound gaps in Table III.	115
XI	Lower bound and upper bounds used to compute optimality and lower-bound gaps in Table IV.	116
XII	Lower bound and upper bounds used to compute optimality and lower-bound gaps in Table V.	116
XIII	Lower bound and upper bounds used to compute optimality and lower-bound gaps in Table VI.	117
XIV	Lower bound and upper bounds used to compute optimality and lower-bound gaps in Table VII.	117
XV	Lower bound and upper bounds used to compute optimality and lower-bound gaps in Table VIII.	118
XVI	Comparison of BALP and AK-ALP lower and upper bounds in generalized joint replenishment problem instances.	173
XVII	Parameters of five-dimensional perishable inventory control instances. .	174
XVIII	Comparison of methods on perishable inventory control problem instances.	177

LIST OF FIGURES

<u>FIGURE</u>		<u>PAGE</u>
1	Illustration of self-guiding mechanism with \mathbf{v} equal to a uniform distribution.	40
2	$\text{FALP}_{600,7}^{\text{SG}}$ upper and lower bounds on two representative ten-dimensional perishable inventory control instances with (c_h, c_d, c_b, σ) equal to $(1, 8, 2, 5)$ and $(1, 8, 2, 2)$ in the left and right panels, respectively.	60
3	Comparison of FALP VFA $V(\beta_{50}^{\text{FA}})$ (left panel) and self-guided FALP VFA $V(\beta_{50}^{\text{SG}})$ (right panel) on a two-dimensional perishable inventory control instance.	105
4	Illustrating the impact of guiding constraints on greedy policy performance.	106
5	Self-guided FALP state-relevance distributions $\mathbf{v}'(\beta_{40}^{\text{SG}})$ (left panel), $\mathbf{v}'(\beta_{45}^{\text{SG}})$ (middle panel), and $\mathbf{v}'(\beta_{50}^{\text{SG}})$ (right panel) on a two-dimensional perishable inventory control instance.	108
6	Illustrating the connection between lower bound quality and BFA quality on a toy MDP.	141

LIST OF ABBREVIATIONS

AC	Average Cost
ALP	Approximate Linear Program
BALP	Bound-focused Approximate Linear Program
BELP	Bound-focused Exact Linear Program
BFEP	Bound-focused Feature-based Exact Program
BFA	Bias Function Approximation
BOP	Bermudan Options Pricing
DC	Discounted Cost
FALP	Feature-based Approximate Linear Program
FEP	Feature-based Exact Program
GJR	Generalized Joint Replenishment
IMP	Idealized Math Program
LB	Lower Bound
LSM	Least Squares Monte Carlo
MCMC	Markov Chain Monte Carlo
MDP	Markov Decision Process
PALP	Policy-focused Approximate Linear Program

LIST OF ABBREVIATIONS (Continued)

PELP	Policy-focused Exact Linear Program
PFEP	Policy-focused Feature-based Exact Program
PIC	Perishable Inventory Control
RL	Reinforcement Learning
SP	Separation Problem
Semi-MDP	Semi-Markov Decision Process
UB	Upper Bound
VFA	Value Function Approximation

SUMMARY

We revisit the well-established approximate linear programming approach to Markov decision processes (MDPs). This model-based reinforcement learning (RL) algorithm has strong theoretical properties and has been successfully applied to many Operations Management and Operations Research applications. However, guaranteeing this approach results in near-optimal control policies for new applications and problem instances poses known practical challenges. These challenges include (i) the design of approximation architectures and (ii) the formulation of approximate linear programs (ALPs) along with the fine-tuning of their parameters. Specifically, designing approximation architectures that ensure near-optimal MDP bias/value function approximations (B/VFAs) often involves tedious trial-and-error and exploiting problem structures. Additionally, formulating an ALP that ensures its B/VFA corresponds to high-quality control policies requires refining previously suggested ALP formulations and developing methods to tune ALP parameters. Although prior research has proposed solutions to these challenges for specific applications and problem instances, bridging the gap to design an application-agnostic ALP method that is both

This doctoral thesis research presents novel ALP methodologies for discounted-cost and average-cost MDPs that mitigate the aforementioned practical challenges. Our methods leverage random basis functions commonly used in Machine Learning and extend them to the ALP framework. Random basis functions allow us to replace the hand-engineering of B/VFAs with computationally low-cost sampling of random basis function parameters from known distribu-

SUMMARY (Continued)

tions. Our methods also involve generating multiple randomized approximations to the MDP bias/value function instead of constructing a single deterministic B/VFA, as predominantly done in the literature. Therefore, our randomized multiple-shot approximation approaches involve iteratively solving a sequence of ALPs, where two consecutive models in this sequence are connected using “guiding” constraints that utilize the ALP B/VFA obtained in the previous iteration to guide the computation of the B/VFAs in the current iteration. Thus, our methods iteratively refine their own ALP formulations and parameters, so we refer to them as “self-guided ALPs.” We establish several theoretical properties of our methods, including probabilistic convergence rates and policy performance bounds, that are new to the ALP and RL literature. We also apply our application-agnostic algorithms to challenging inventory control and options pricing problems. We show that they deliver excellent control policies and performance bounds and improve upon or compete with existing problem-specific benchmarks. More broadly, our research takes a meaningful step toward easy-to-implement model-based RL methods that are guaranteed to compute near-optimal policies and performance bounds in both discounted-cost and average-cost MDP settings.

Chapter 1 provides an introduction to this thesis and a summary of its contributions. In chapters 2 and 3, our randomized multiple-shot approximation approaches for discounted-cost and average-cost MDPs are outlined, respectively.

CHAPTER 1

INTRODUCTION

After a few months into my Ph.D., I asked the following question from my thesis advisor: *Despite the success of reinforcement learning algorithms for operations management and operations research applications, why has the implementation of such methods not gained much attention?* This question has influenced my doctoral thesis research.

Reinforcement learning (RL) is a subfield of Artificial Intelligence that focuses on solving large-scale Markov decision processes (MDPs). MDPs model a wide range of operations management and operations research problems for which model-based RL (i.e., approximate dynamic programming) provides tractable solutions for computing control policies. Several families of RL algorithms, such as Approximate Value Iteration, Approximate Policy Iteration, and Approximate Linear Programming, are developed for approximately solving these MDPs with high-dimensional state and/or action spaces. Nonetheless, deploying these algorithms relies on exploiting problem structure, engaging in tedious trial and error, and human intervention is necessary to ensure the quality of their solutions. While the literature documents how to tailor these RL methods for specific problems, it is unclear how to adapt them to new problem instances and applications. These deployment challenges are rooted in the formulations and the theory behind these algorithms, and they severely limit the potential of RL to be conveniently applied to business applications. These reasons answer the question above.

This thesis leverages the well-established approximate linear programming approach to MDPs because it has several strong theoretical properties and is shown to perform well across multiple application domains. This approach for discounted-cost (DC) and average-cost (AC) MDPs relies on solving a so-called approximate linear program (ALP). We present two novel approximate linear programming frameworks, one for DC MDPs and one for AC MDPs. Both frameworks (i) are application-agnostic such that they can be applied directly to different problem instances and applications, (ii) deliver theoretical support for the performance of models solved during their implementation, and (iii) exhibit “near-optimal” numerical performance.

DC ALP relies on approximating the MDP value function, and its formulation necessitates making two choices: basis functions and state-relevance distribution, which are both defined over the MDP state space. The linear combination of basis functions provides a value function approximation (VFA), and the state-relevance distribution assigns weights to different regions of the MDP state space. DC ALP minimizes a VFA error weighted by the state-relevance distribution, i.e., it reduces VFA error at states where state-relevance distribution assigns high weights. The ALP VFA enables computing a “greedy policy,” along with upper and lower bounds on the optimal policy cost. In particular, plugging this VFA into a so-called greedy optimization problem results in the greedy policy, whose cost is an upper bound. The VFA can also be plugged into other methods (e.g., information relaxation and duality, constraint violation learning) to obtain a lower bound. Combining these upper and lower bounds on the optimal policy cost, we can calculate an optimality gap that reflects how close the greedy policy cost is relative to the optimal policy cost without directly knowing the optimal policy.

AC ALP relies on approximating the MDP bias function. It leverages a linear combination of chosen basis functions to perform bias function approximation (BFA). AC ALP treats as a variable a lower bound on the optimal policy cost, and it maximizes this lower bound without having an explicit term in its objective to minimize a BFA error. AC ALP objective is thus different from the DC ALP objective that minimizes a VFA error weighted by a state-relevance distribution. This difference resulted in fundamentally different formulations, theoretical results, and algorithm development for AC ALP compared to DC ALP. The literature suggests using the AC ALP formulation for computing the lower bound. However, to compute BFAs, greedy policies, and upper bounds, it is standard to solve AC ALP modifications that include some notion of BFA error in their objective function.

Below we describe the main contributions of this research.

Quality of lower bound. The quality of lower bounds from DC ALP and AC ALP relies on the choice of basis functions used to define VFA and BFA, respectively. However, selecting appropriate basis functions to ensure these ALPs deliver near-optimal lower bounds is challenging without prior domain knowledge. Most existing theory for ALP assumes basis functions are provided as input. Only a few studies have investigated the selection of basis functions and developed innovative methods to generate basis functions. These studies either focus on a specific application or rely on the structure of MDP optimal policy, which is idealized information not available during the deployment of ALP.

This thesis is the first to extend random basis functions, also referred to as random features, widely used in machine learning tasks such as classification and regression, to RL approaches

based on mathematical programming, specifically DC ALP and AC ALP. We show that this extension addresses the issue with the suboptimal ALP lower bounds resulting from poorly selected basis functions. Consider two gaps: one between the DC ALP lower bound and the optimal policy DC, and the other between the AC ALP lower bound and the optimal policy AC. We develop probabilistic error bounds on these gaps and demonstrate that they converge to zero with a high probability and at a dimension-free rate of one divided by the square root of the number of random bases used to formulate VFA in DC ALP and BFA in AC ALP. These rates can be viewed as the convergence rate of a Markov chain Monte Carlo (MCMC) method extended to Hilbert spaces for function approximation instead of estimating a single value. Our rates are the first finite convergence rates in the ALP literature with respect to the number of basis functions. Although our error bounds converge to zero at the same rate for both DC and AC settings, the constants in these bounds differ. Specifically, in the case of AC ALP, we are able to show that approximating the MDP bias function at a possibly small region of the state space suffices to obtain tight lower bounds. Thus, the constant in our AC ALP error bound relies on how “hard” it is to approximate the MDP bias function in this region. For these results, which are our first main contribution, please see §2.3 and §3.3 that correspond to DC MDPs and AC MDPs, respectively.

Different basis functions are used to define V/BFA and formulate ALP. We illustrate two choices below while a broader discussion on basis functions studied in the literature can be found in §2.1. [Trick and Zin \(1997\)](#) use spline basis functions to perform VFA in ALP. While this paper successfully applies such VFA to a low-dimensional MDP, using this VFA on large-scale prob-

lems has not been investigated. The challenge lies in generating splines with low computational cost while ensuring the resulting VFA maintains near-optimality. Alternatively, one can define a VFA using neural networks that are known to provide near-optimal approximations of MDP value function under mild conditions, but they depend non-linearly on their weights, necessitating solving non-linear programs. Random basis functions are generated by low-cost sampling from known distributions, and universal random basis functions, such as random Fourier bases, provide an arbitrarily close approximation of the MDP value function. In addition, a VFA constructed with random basis functions is linear in its weights, requiring the solution of a linear program. We leverage these low-cost sampling, universality, and linearity properties of VFAs based on random bases to develop our methods that are computationally appealing and provide convergence rates. Similar results may not carry over when using splines or neural networks in an ALP.

Quality of policy. In DC ALP, it is known that the state-relevance distribution used to define its objective function highly impacts the quality of greedy policy. Approaches for choosing this parameter are limited. For example, one can set the state-relevance distribution to the state-visit frequency of a baseline policy, but this approach lacks theoretical and computational justification (De Farias and Van Roy 2003, Farias and Van Roy 2006). In AC ALP, the state-relevance distribution does not appear in the ALP formulation. While it might be intriguing that there is no need to tune this ALP parameter in the AC setting, the absence of the state-relevance distribution can result in poor BFAs and greedy policies. Two lines of research studied this issue. The first line modifies the original ALP formulation by assuming a fixed set of basis

functions is given (De Farias and Van Roy 2002, 2006, Veatch 2013). Theoretical results for these modified models typically rely on idealized information, and there is no numerical evidence on how such ALP models perform. Klabjan and Adelman (2007) and Adelman and Klabjan (2012) proposed the second line that dynamically refines basis functions using dual ALP information and optimization. Their approach addresses this issue with AC ALP formulation via basis function generation. For a general MDP, their basis function generation requires solving a nonlinear math program, a practically challenging task. For a generalized joint replenishment (GJR) problem, they showed that this nonlinear program simplifies to a mixed-integer program. While extensions of their simplified model to other applications are, in principle, possible, no research has yet explored such extensions.

The second main contribution of this research involves taking multiple shots at randomly approximating the MDP value/bias function rather than performing a single-shot approximation, which is predominantly done in the literature, to mitigate the issues related to the state-relevance distribution in DC ALP and AC ALP. We propose two randomized multi-shot approximation mechanisms: one for DC MDPs and another for AC MDPs. Both these mechanisms involve an iterative process of sampling random basis functions in batches and constructing multiple ALP models with nested V/BFAs that are increasing in the number of random bases. Specifically, in the current iteration, we formulate an ALP using basis functions sampled thus far and include in its formulation additional “guiding constraints” that are defined based on V/BFA obtained from the previous iteration’s ALP with fewer random bases. We thus label an ALP constructed in this manner as “self-guided ALP” because it uses its own past V/BFA information to direct

the computation of the next V/BFA by incorporating guiding constraints. While the main idea behind both mechanisms is the same, they are fundamentally different, as explained next.

In the DC setting, our guiding constraints are added directly to the original DC ALP formulation to obtain self-guided ALP. We show that these constraints dynamically update the state-relevance distribution and thus avoid the need for hand-tuning this ALP parameter. Therefore, our self-guided ALP models do not rely on idealized information or problem structure to define state-relevance distribution. We develop an error rate for the quality of our self-guided ALP VFAs and demonstrate that a worst-case measure of their greedy policy performance is weakly improving as more random bases are sampled. To our knowledge, our approach is the only method in the literature that dynamically updates state-relevance distribution and has associated theoretical guarantees. These results, which serve as the third main contribution, are presented in §2.4.

In the AC setting, there is no hope of obtaining high-quality BFAs for a general MDP from the original AC ALP formulation with a fixed set of bases because it does not have any BFA error term in its objective function. Accordingly, there is no value in adding guiding constraints to this model. Therefore, we use an alternative ALP formulation proposed by [De Farias and Van Roy \(2002\)](#) that includes (an artificially added) state-relevance distribution to control BFA error in different regions of the MDP state space. Our self-guided ALP in the AC setting is thus based on the formulation in [De Farias and Van Roy \(2002\)](#), random basis functions, and guiding constraints. Note that the self-guided ALP model is solved only for computing good BFAs, not lower bounds, because we already discussed that the original AC ALP model with

universal random bases provides near-optimal lower bounds (please see our first contribution). We develop a weakly improving upper bound on the worst-case performance of policy obtained from our self-guided ALP for AC MDPs. The self-guided ALP formulation for AC MDPs and its theoretical properties are our fourth main contribution in this thesis. Please see §3.4 for details.

Numerical studies. We have numerically tested both DC and AC versions of our self-guided ALP method. In the DC setting, we applied it to high-dimensional instances of perishable inventory control (PIC) and Bermudan options pricing (BOP) problems. Self-guided ALP achieves excellent policies and bounds, leading to the best-known policies and lower bounds on the PIC instances and competing with a state-of-the-art benchmark for BOP. It is encouraging that our application-agnostic policies and lower bounds outperform benchmarks that use domain knowledge for basis function selection and/or heuristically updating state-relevance distribution. In the AC setting, we considered two applications: GJR, which gives rise to an AC semi-MDP, and an AC version of PIC. We benchmarked our method on GJR instances against the algorithm in [Adelman and Klabjan \(2012\)](#) that adaptively generates basis functions, as we do, but exploits the structure of GJR. On GJR instances without holding cost, the challenge is to find tight lower bounds, as shown in [Adelman and Klabjan \(2012\)](#). Therefore, we only perform the part of the self-guided ALP method that pertains to computing the lower bound (i.e., single shot). We observe that our application-agnostic lower bounds are comparable with the ones from this application-specific benchmark on these GJR instances, which is encouraging. To evaluate the effectiveness of our self-guiding mechanism in the AC setting, we applied our method to

PIC instances with AC criteria. We observed that our approach leads to near-optimal policies. It also outperforms several heuristics and benchmarks based on our DC self-guided ALP model. These extensive numerical studies serve as the fifth main contribution of this thesis. The results for DC and AC self-guided ALP models are reported in §2.6–§2.7 and §3.6–§3.7, respectively.

Solving ALPs. Our self-guided ALP models are semi-infinite linear programs if MDP state and/or state spaces are continuous. For PIC and BOP applications, we solve these programs via the widely-used constraint sampling approach (please see §2.6–§2.7). We observed that this method delivers high-quality approximations on the instances we considered. Nevertheless, for the GJR application, we use a more sophisticated approach called constraint generation (please see §3.6.1) for two reasons. First, the state and action spaces of GJR are both high-dimensional, so we expect constraint sampling to fail in providing a good approximation of semi-infinite linear programs. Second, solving the greedy policy optimization problem in GJR is challenging and cannot be done via discretization, given the high dimensionality of its MDP action space. Note that discretization of the action space is possible in PIC and BOP applications since they have one-dimensional action spaces. For GJR, we thus showcase how two families of random bases, namely random Stump and random ReLU, can be used to reformulate the constraint generation and greedy policy optimization problems as mixed-integer programs (please see §3.6). These solution approaches for (approximately) solving semi-infinite linear programs formulated with random bases are our last main contribution.

Circling back to the question I posed at the beginning of this chapter, this dissertation takes a significant step toward application-agnostic policies and bounds for MDPs, making the

deployment of ALPs much easier. To make our approach accessible to a wider range of applications beyond those studied in this thesis, we have made publicly available Python codes that include the implementation of our methods, benchmarks, and the three applications considered (<https://github.com/Multi-Shot-Approximation-of-MDPs>).

More broadly, this thesis opened new research directions in RL to develop easy-to-implement methodologies that provide theoretical guarantees and are computationally efficient. While several model-based RL algorithms exist in the literature, such as approximate value iteration and least-square Monte Carlo, these methods, similar to ALP, often require significant hand-tuning to perform well on a specific problem. In addition, because they lack the same theoretical properties as the ALP approach when a fixed set of basis functions is used, it remains unclear what should be the correct notion of “self-guiding” or “randomized multi-shot approximation” in these methods. Therefore, it would be valuable to investigate model-based RL algorithms other than ALP within the multi-shot approximation framework of this thesis. Moreover, this research raises the broader question of extending randomized multi-shot approximation to model-free RL and offline RL.

CHAPTER 2

SELF-GUIDED APPROXIMATE LINEAR PROGRAMS: RANDOMIZED MULTI-SHOT APPROXIMATION OF DISCOUNTED COST MARKOV DECISION PROCESSES

(Co-authors: Parshan Pakiman, Selvaprabu Nadarajah, Negar Soheili, Qihang Lin)

Abstract

Approximate linear programs (ALPs) are well-known models based on value function approximations (VFAs) to obtain policies and lower bounds on the optimal policy cost of discounted-cost Markov decision processes (MDPs). Formulating an ALP requires (i) basis functions, the linear combination of which defines the VFA, and (ii) a state-relevance distribution, which determines the relative importance of different states in the ALP objective for the purpose of minimizing VFA error. Both these choices are typically heuristic: basis function selection relies on domain knowledge while the state-relevance distribution is specified using the frequency of states visited by a baseline policy. We propose a self-guided sequence of ALPs that embeds random basis functions obtained via inexpensive sampling and uses the known VFA from the previous iteration to guide VFA computation in the current iteration. In other words, this sequence takes multiple shots at randomly approximating the MDP value function with VFA-based guidance between consecutive approximation attempts. Self-guided ALPs mitigate domain knowledge during basis function selection and the impact of the state-relevance-distribution choice, thus

reducing the ALP implementation burden. We establish high probability error bounds on the VFAs from this sequence and show that a worst-case measure of policy performance is improved. We find that these favorable implementation and theoretical properties translate to encouraging numerical results on perishable inventory control and options pricing applications, where self-guided ALP policies improve upon policies from problem-specific methods. More broadly, our research takes a meaningful step toward application-agnostic policies and bounds for MDPs.

2.1 Introduction

Computing high-quality control policies in sequential decision making problems is an important task across several application domains. Markov decision processes (MDPs; [Puterman 1994](#)) provide a powerful framework to find optimal policies in such problems but are often intractable to solve exactly due to their large state and action spaces or the presence of high-dimensional expectations (see §1.2 and §4.1 of [Powell 2007](#)). Therefore, a class of approximate dynamic programming (ADP) approaches instead approximate the value functions of MDPs and use the resulting approximations to obtain control policies in simulations ([Bertsekas and Tsitsiklis 1996](#)). Approximate linear programming ([Schweitzer and Seidmann 1985](#), [De Farias and Van Roy 2003](#)) is a math-programming-based ADP approach for computing value function approximations (VFAs) that has been applied to a wide variety of domains, including operations research and artificial intelligence ([Adelman 2003](#), [Guestrin et al. 2003](#), [Forsell and Sabbadin 2006](#), [Desai et al. 2012a](#), [Adelman and Mersereau 2013](#), [Tong and Topaloglu 2013](#), [Nadarajah et al. 2015](#), [Mladenov et al. 2017](#), [Balseiro et al. 2019](#), [Blado and Toriello 2019](#)). It solves a so-called approximate linear program (ALP) to obtain a VFA, from which a control policy can be computed.

This VFA can also be used to obtain a lower bound on the optimal policy cost, which enables the computation of an optimality gap for the ALP policy as well as other heuristic policies.

Formulating an ALP requires (i) basis functions, the linear combination of which defines the VFA over the MDP state space, and (ii) a state-relevance distribution, which determines the relative importance of different states in the ALP objective for the purpose of minimizing VFA error. It is well known that the choices of basis functions and the state-relevance distribution are challenging to make and impact the VFA quality significantly. These choices are typically handled heuristically, the former using domain knowledge and the latter by considering the states visited by a baseline policy (see §5 in [Farias and Van Roy 2006](#) and §3.2.2 in [Sun et al. 2014](#)). The goal of this paper is to broaden the applicability of ALP by reducing the burden of making these choices.

Our contributions are the following.

1. Our starting point is to provide a new reformulation of a discounted-cost MDP as a large-scale mathematical program. This program has infinitely many variables corresponding to a weighted integral of a continuum of basis functions, referred to as random basis functions (or *random features* in machine learning), and a large number of constraints (possibly infinite), one for each MDP state and action pair. The class of random Fourier basis functions defined using cosines is a popular example ([Rahimi and Recht 2008](#)). A functional analogue of Monte Carlo sampling can be used to approximate the integral over random basis functions. The resulting model, dubbed feature-based approximate linear program (FALP), has variables corresponding to the VFA weights in a linear combination

of randomly sampled basis functions. This model can be viewed as a *randomized one-shot approximation* of the MDP value function. We establish high probability bounds on the worst-case error between the FALP VFA and the MDP value function. In particular, this error bound converges at the dimension-free rate of one divided by the square root of the number of sampled random basis functions, analogous to the convergence rate of standard Monte Carlo sampling with respect to the number of samples.

2. While FALP does not rely on defining basis functions using domain knowledge, its formulation still requires choosing a state-relevance distribution. Misspecifying this distribution can lead to poor ALP policies (De Farias and Van Roy 2003, Sun et al. 2014). To address this issue, we propose a *multi-shot randomized approximation* approach that leverages the ability to sample additional random basis functions inexpensively. This approach solves a sequence of FALP models with increasing numbers of random basis functions and guiding constraints that ensure successive VFAs weakly improve their distances to the MDP value function at each state. These constraints can be interpreted as adaptively updating the state-relevance distribution. We label our multi-shot approximation approach as self-guided FALP because the guiding constraints only require VFA information from a preceding approximation attempt. The sequence of self-guided FALP VFAs is guaranteed to provide monotonically increasing lower bounds and a monotonically non-increasing worst-case measure of policy performance. We establish an error bound for self-guided FALP that reflects the effect of the guiding constraints on this bound.

3. We validate the performance of the proposed models on perishable inventory control and options pricing applications. We find that FALP outperforms ALP models with tailored application-specific basis functions and leads to near-optimal policies and bounds on low-dimensional instances for both applications, also closing the optimality gaps of prior ALP-based policies on known perishable inventory control instances. In other words, approximations based on one-shot randomization suffice in these low-dimensional instances. This is, however, not the case on higher dimensional instances, where FALP policies and/or bounds are suboptimal and randomized multi-shot approximations deliver value. Specifically, self-guided FALP provides excellent policies and bounds that significantly improve upon FALP as well as benchmarks. The benchmarks for the first and second applications are, respectively, FALP with state-relevance distribution updates based on the states visited by past FALP policies (see [De Farias and Van Roy 2003](#)) and the least-squares Monte Carlo algorithm, which is popular for options pricing ([Longstaff and Schwartz 2001](#)). Beyond these specific applications, the application-agnostic policies from self-guided FALP can serve as a useful benchmark to assess the value of procedures that exploit application structures. To facilitate such benchmarking, we have made Python code implementing the approaches developed in this paper publicly available at <https://github.com/multi-shot-approximation-of-mdps>.

Our contributions add to the research on ALPs, which predominantly assumes a fixed set of basis functions and a heuristic choice of the state-relevance distribution. Work relaxing these assumptions, as we do, is limited.

[Klabjan and Adelman \(2007\)](#) is a seminal paper that develops a convergent algorithm to generate basis functions for semi-Markov decision processes. It requires the solution of a challenging nonlinear program. Building on this work, [Adelman and Klabjan \(2012\)](#) considers an innovative algorithm for basis function generation in a generalized joint replenishment problem. Their algorithm leverages structure and numerical experience for this application. Our approach differs from this work because it uses low-cost sampling to generate basis functions, focuses on discounted-cost MDPs, and is application-agnostic.

[Bhat et al. \(2012\)](#) side-step basis function selection when computing VFAs by applying the kernel trick (see, e.g., chapter 5 of [Mohri et al. 2012](#)), which replaces the inner products of such functions in the dual of a regularized ALP relaxation by kernels. Guarantees on the approximation quality of their VFAs depend on the kernel and an idealized sampling distribution that assumes knowledge of an optimal policy. Our approach instead works directly on the primal ALP formulation and samples parameters that define a class of random basis functions as opposed to state-action pairs. Moreover, the sampling distribution is readily available in our framework and the error bounds that we develop are for models that do not rely on the knowledge of an optimal policy for their formulation and solution.

The papers above do not address the choice of the state-relevance distribution. Parametric forms for the state-relevance distribution that are close to the steady-state distribution of an optimal policy can be obtained for some queuing applications but not in general ([De Farias and Van Roy 2003](#)). [De Farias and Van Roy \(2003, page 854\)](#) and [Farias and Van Roy \(2006\)](#) propose dynamically updating this distribution using the state-visit frequency from simulating

a policy, versions of which are employed in [Sun et al. \(2014\)](#) and in conjunction with FALP in our numerical experiments. This strategy lacks theoretical backing and it can also be computationally expensive to simulate a policy each time an update of the state-relevance distribution is made. Self-guided FALP, while iterative, is fundamentally different as it leverages the ability to cheaply sample new random basis functions and uses only past VFA information available from solving an ALP model to guide the state-relevance distribution. Along with the theoretical guarantees mentioned earlier, one can view self-guided FALP as a conceptually sound mechanism for updating the state-relevance distribution.

To solve ALP models, which are large-scale, potentially semi-infinite, linear programs, we rely on the constraint sampling approach to obtain a linear program with a manageable number of variables and constraints that can be sent to a commercial solver such as Gurobi ([De Farias and Van Roy 2004](#), [Calafiore and Campi 2005](#)). To generate a lower bound on the optimal policy cost using a given VFA, we explore two approaches from the literature in our numerical study, neither of which deal with basis function selection or state-relevance distribution choice. For the perishable inventory control application, where the controllable part of state space is high dimensional, we embed VFAs within the primal-dual ALP approach from [Lin et al. \(2020\)](#), which is based on learning regions of high constraint violation and is thus referred to as constraint-violation learning. For the options pricing application, where there is essentially no controllable part of the state, we use VFAs within the information relaxation and duality approach studied by [Haugh and Kogan \(2004\)](#) and [Brown et al. \(2010\)](#), which solves penalized

hindsight optimization models and is known to be effective for this class of applications (see [Brown et al. 2022](#) for a tutorial).

Our work builds on the seminal research on random bases by [Rahimi and Recht \(2008\)](#), [Rahimi and Recht \(2008\)](#) and [Rahimi and Recht \(2009\)](#). There is extant literature applying this idea to data mining and machine learning applications ([Lu et al. 2013](#), [McWilliams et al. 2013](#), [Beevi et al. 2016](#), [Wu et al. 2018](#)) and to a value iteration algorithm by [Haskell et al. \(2020\)](#). These papers embed random bases in what amounts to a regression setting, whereas we show that such bases can be effectively used in ALPs that have complicated constraints. We also add to this literature in terms of theory. Our approximation guarantees for FALP adapt the arguments in [Rahimi and Recht \(2008\)](#) to an ALP setting and also strengthen the error bounds. A similar analysis of self-guided FALP, unfortunately, does not lead to insightful bounds. Therefore, we develop an error bound for self-guided FALP based on functional projections and a geometric notion of feasibility, which are new to this literature, and potentially of independent interest.

More broadly, our work adds to the rich literature on reinforcement learning that attempts to reduce the burden of feature engineering ([Mnih et al. 2015](#), [Silver et al. 2017](#)). Here, neural networks and deep learning have received significant research attention as they facilitate the approximation of complex functions with limited domain knowledge ([Fujimoto et al. 2018](#), [Osband et al. 2019](#), [Franke et al. 2021](#)). They give rise to VFAs that depend nonlinearly on the parameters but involve the solution of non-convex optimization problems ([Wang et al. 2020](#)). Our use of random basis functions in ALP mitigates domain knowledge while retaining linear programming structure; it can thus be viewed as a complementary strategy.

In §2.2, we present the standard linear programming formulation to solve MDPs and then introduce an alternative formulation in a randomized feature space. Randomized single-shot and multi-shot approximations of the MDP value function are discussed in §2.3 and §2.4, respectively. In §2.5, we present extensions to finite-state and finite-horizon MDPs. The numerical studies on perishable inventory control and options pricing are covered in §2.6 and §2.7, respectively. We conclude in §2.8. All proofs and supporting materials are available in §§2.9-2.13.

2.2 Exact Mathematical Programs

In §2.2.1, we provide background on infinite-horizon discounted-cost MDPs and their known linear programming MDP reformulation. In §2.2.2, we propose an alternative mathematical programming reformulation for MDPs based on randomized feature spaces, which plays a central role in the approximations we develop in later sections.

2.2.1 Background

Consider a decision maker controlling a system over an infinite horizon. A policy $\pi : \mathcal{S} \mapsto \mathcal{A}_s$ assigns an action $\mathbf{a} \in \mathcal{A}_s$ to each state $s \in \mathcal{S}$, where \mathcal{S} denotes the MDP state space and \mathcal{A}_s represents the feasible action space at state s . An action $\mathbf{a} \in \mathcal{A}_s$ taken at state $s \in \mathcal{S}$ results in an immediate cost of $c(s, \mathbf{a})$ and the transition of the system to the next state according to the probability distribution $P(\cdot | s, \mathbf{a})$.

The decision maker's objective is to find a stationary and deterministic optimal policy π that minimizes discounted expected costs. Starting from an initial state $s_0 = s \in \mathcal{S}$, the discounted expected cost of a policy π is

$$\text{PC}(s, \pi) := \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t c(s_t^\pi, \pi(s_t^\pi)) \mid s_0 = s \right],$$

where $\gamma \in [0, 1)$ denotes the discount factor, expectation \mathbb{E} is with respect to the state-action probability distribution induced by the transition probability distribution $P(\cdot | s, a)$ and the policy π , and s_t^π is the state reached at stage t when following this policy. The quality of a given policy is evaluated with respect to a distribution $\chi(s)$ for the initial state. Specifically, we define the cost of policy π as $\text{PC}(\pi) := \mathbb{E}_\chi[\text{PC}(s, \pi)]$.

The policy-cost minimization problem is

$$\inf_{\pi: \mathcal{S} \rightarrow \mathcal{A}_s} \text{PC}(\pi). \quad (2.1)$$

The MDP value function $V^* : \mathcal{S} \mapsto \mathbb{R}$ is defined as $V^*(s) = \inf_{\pi: \mathcal{S} \rightarrow \mathcal{A}_s} \text{PC}(s, \pi)$ for all $s \in \mathcal{S}$.

Assumption 1 *An optimal policy π^* that solves (2.1) exists and the MDP value function satisfies $V^*(s) = \text{PC}(s, \pi^*)$ for all $s \in \mathcal{S}$. The state space \mathcal{S} is a continuous, compact real-valued set and the action spaces \mathcal{A}_s for all $s \in \mathcal{S}$ either share this property or are finite. Moreover, the MDP value function $V^*(\cdot)$ is continuous.*

The existence of π^* in the literature is guaranteed under different requirements, mainly over the cost function $c(\cdot, \cdot)$ and state transition kernel $P(\cdot|s, a)$. Informally, one such set of conditions requires the lower semi-continuity of the immediate cost and the strong continuity of state transitions. We present them formally in §2.9.1 and refer to Theorem 4.2.3 in [Hernández-Lerma and Lasserre \(1996\)](#) for a more elaborate discussion. Continuous state spaces and value functions arise in applications such as lost-sales inventory control ([Zipkin 2008](#)), healthcare screening ([Steimle and Denton 2017](#)), dual sourcing ([Hua et al. 2015](#)), robotics ([Peters et al. 2003](#), [Haarnoja et al. 2019](#)), and flight simulators ([McGrew et al. 2010](#), [Yang et al. 2019](#)). Our models and analysis in the remainder of this section and §§2.3–2.4 focus on MDPs satisfying Assumption 1. We discuss in §2.5 how they apply to a broader class of MDPs, for instance, where the state space can have discrete components.

The computation of the MDP value function can be conceptually approached without knowing π^* via the exact linear program (ELP; see, e.g., pages 131-143 in [Hernández-Lerma and Lasserre 1996](#))

$$\begin{aligned} & \max_{V': \mathcal{S} \rightarrow \mathbb{R}} \quad \mathbb{E}_{\mathbf{v}}[V'(s)] \\ & \text{s.t.} \quad V'(s) - \gamma \mathbb{E}[V'(s') | s, a] \leq c(s, a), \quad \forall (s, a) \in \mathcal{S} \times \mathcal{A}_s, \end{aligned} \quad (2.2)$$

where \mathbf{v} is a state-relevance distribution that specifies the relative importance of each state in the state space. ELP is well defined because Assumption 1 ensures that the MDP value function V^* solves the optimality equations $V^*(s) = \min_{a \in \mathcal{A}_s} \{c(s, a) + \gamma \mathbb{E}[V^*(s') | s, a]\}$ for every $s \in \mathcal{S}$.

Thus, V^* is an optimal solution to ELP, which follows from its constraints holding as equalities at V^* . Since V^* is continuous over a compact domain (Assumption 1), it is bounded and the objective function of ELP, which is an expectation of V^* , is also bounded. However, ELP is intractable to solve since it is a doubly infinite linear program. It has continua of decision variables and constraints, one for each state and state-action pair, respectively.

2.2.2 Feature-based Exact Program

ELP directly computes the MDP value function. We present an alternative formulation that represents the MDP value function in a transformed feature space. This feature space is defined by a vector $\theta := (\theta_0, \theta_1, \dots, \theta_d) \in \Theta \subseteq \mathbb{R}^{d+1}$, a scalar mapping $\varphi(\cdot) : \mathbb{R} \mapsto \mathbb{R}$, and an associated sampling density $\rho(\theta)$, where integer d denotes the dimension of the state space \mathcal{S} . These elements can be used to represent a feature $\varphi(\theta^\top(1, s))$ using the inner product $\theta^\top(1, s) := \theta_0 + \sum_{i=1}^d \theta_i s_i$. In other words, for each θ sampled from ρ , we can define a “random” feature $\varphi(s; \theta) \equiv \varphi(\theta^\top(1, s))$. We define a representation of the MDP value function in this randomized feature space using the pair $\beta := (\beta_0, \mathbf{B})$ containing an intercept $\beta_0 \in \mathbb{R}$ and an integrable weighting function $\mathbf{B} : \Theta \mapsto \mathbb{R}$:

$$V(s; \beta) := \beta_0 + \int_{\Theta} \mathbf{B}(\theta) \varphi(s; \theta) d\theta. \quad (2.3)$$

The class of functions that can be covered by this construction is

$$\mathcal{R} := \left\{ V : \mathcal{S} \mapsto \mathbb{R} \mid \exists \beta = (\beta_0, \mathbf{B}) \text{ s.t. } V(s) = V(s; \beta), \forall s \in \mathcal{S}, \text{ and } \|\mathbf{B}/\rho\|_{2,\rho} < \infty \right\},$$

where the $(2, \rho)$ -norm of $\mathbf{B}(\cdot)/\rho(\cdot) : \Theta \mapsto \mathbb{R}$ is defined as

$$\|\mathbf{B}/\rho\|_{2,\rho} := \int_{\Theta} \left(\frac{\mathbf{B}(\theta)}{\rho(\theta)} \right)^2 \rho(d\theta) = \int_{\Theta} \frac{(\mathbf{B}(\theta))^2}{\rho(\theta)} d\theta.$$

Replacing the MDP value function with the integral form (2.3) and requiring the weighting function to have a finite $(2, \rho)$ -norm as in the definition of \mathcal{R} gives the feature-based exact program (FEP):

$$\begin{aligned} & \sup_{\beta_0, \mathbf{B}} \quad \beta_0 + \int_{\Theta} \mathbf{B}(\theta) \mathbb{E}_{\nu}[\varphi(s; \theta)] d\theta \\ & \text{s.t.} \quad (1 - \gamma)\beta_0 + \int_{\Theta} \mathbf{B}(\theta) (\varphi(s) - \gamma \mathbb{E}[\varphi(s') \mid s, a]) d\theta \leq c(s, a), \quad \forall (s, a) \in \mathcal{S} \times \mathcal{A}_s \\ & \quad \|\mathbf{B}/\rho\|_{2,\rho} < \infty. \end{aligned}$$

Unlike ELP, which directly optimizes a value function, the above program optimizes the weights associated with the *feature-based* representation of the value function in the set \mathcal{R} . It is a reformulation of ELP when $V^* \in \mathcal{R}$ as shown in Proposition 1.

Proposition 1 *If $V^* \in \mathcal{R}$, there is an optimal FEP solution $\beta^* = (\beta_0^*, \mathbf{B}^*)$ such that $V^*(s) = \beta_0^* + \int_{\Theta} \mathbf{B}^*(\theta) \varphi(s; \theta) d\theta$ for all $s \in \mathcal{S}$.*

The assumption V^* belonging to \mathcal{R} can be restrictive if \mathcal{R} is not rich enough to contain an arbitrarily close approximation of V^* . When random features satisfy a “universality” property, this assumption is mild because \mathcal{R} contains a function that is arbitrarily close to any continuous function, and in particular V^* , as we formally show in §2.10.1. Thus, FEP should not result in any

Table I: Examples of universal random basis functions.

	$\varphi(\cdot)$	$\rho(\theta)$	Parameter
Fourier	$\cos(\cdot)$	$\theta_0 \sim \text{uniform}([- \pi, \pi]); \theta_i \sim \text{normal}(0, c_\rho), \text{ for } i \geq 1$	c_ρ
ReLU	$\max(\cdot, 0)$	$\theta \sim \text{uniform}(\text{d-dimensional unit sphere})$	None
Stump	$\text{sign}(\cdot)$	$\theta_0 \sim \text{uniform}([-c_\rho, c_\rho]); (\theta_1, \dots, \theta_d) \sim \text{uniform}(\{\mathbf{e}^1, \dots, \mathbf{e}^d\})$	c_ρ

significant error when using universal random features defined below. For function $V : \mathcal{S} \mapsto \mathbb{R}$, define the ∞ -norm as $\|V\|_\infty := \max_{s \in \mathcal{S}} |V(s)|$, and consider shorthand $V(\boldsymbol{\beta}) \equiv V(\cdot; \boldsymbol{\beta})$ for each function $V \in \mathcal{R}$.

Definition 1 *A class of random features φ with sampling density ρ is called universal if for any continuous function $V : \mathcal{S} \mapsto \mathbb{R}$ and $\varepsilon > 0$, there exists $\boldsymbol{\beta}_\varepsilon := (\beta_{0,\varepsilon}, \mathbf{B}_\varepsilon)$ such that $V(\boldsymbol{\beta}_\varepsilon) \in \mathcal{R}$ and $\|V - V(\boldsymbol{\beta}_\varepsilon)\|_\infty < \varepsilon$.*

Each random feature $\varphi(s; \theta)$ in the definition of FEP is a mapping from the state space to the real line. As a result, we refer to it as a random basis function because this terminology is more common in the ALP literature. Table I lists the components of three universal random basis functions that satisfy Definition 1: the mapping $\varphi(\cdot)$, the sampling density $\rho(\cdot)$ for the vector θ , and the parameters defining this density. Fourier basis functions are defined using a cosine mapping with θ_0 sampled from a uniform distribution with support involving the Archimedes constant π and the remaining elements of θ sampled from a normal distribution with mean zero and standard deviation c_ρ , which is a tunable scalar parameter. ReLU basis functions employ

a mapping that is a maximum with respect to zero. It samples θ from a uniform distribution over a unit sphere with no tunable parameters. Stump basis functions use a signum mapping that evaluates to a -1 , 0 , or 1 , depending on whether the input is negative, zero, or positive, respectively. The element θ_0 is sampled from a uniform distribution with support over an interval that depends on a tunable scalar parameter c_ρ . The remaining elements of θ are sampled from a uniform distribution defined on the discrete set $\{e^1, \dots, e^d\}$, where e^i , $i \in \{1, 2, \dots, d\}$ is a d -dimensional unit vector with 1 in the i -th coordinate and zero elsewhere.

Assumption 2, which holds for the rest of this chapter, includes $V^* \in \mathcal{R}$ and additional conditions needed for our theoretical analysis, all of which are standard in the random basis functions literature (see, e.g., [Rahimi and Recht 2008](#), Theorem 3.2).

Assumption 2 *The MDP value function V^* belongs to \mathcal{R} . Random basis function φ is universal, and its sampling distribution ρ has a finite second moment. Moreover, φ has a Lipschitz constant $L > 0$ and satisfies $\|\varphi\|_\infty \leq 1$ and $\varphi(0) = 0$.*

This assumption is satisfied by Fourier and ReLU basis functions in Table I but not by Stump basis functions as they are not continuous. While Assumption 2 is needed for analysis, the algorithms we present in §§2.3–2.4 can be applied even when this assumption fails to hold.

2.3 Randomized One-Shot Approximation

In §2.3.1, we introduce and analyze FALP, which uses a single set of sampled random basis functions to approximate FEP. That is, FALP is a randomized single-shot approximation of FEP. In §2.3.2, we provide implementation guidelines for FALP.

2.3.1 Model and Theory

In the literature, an ALP is derived from ELP by substituting its decision variable $V'(s)$ with a linear combination of pre-specified basis functions. Our starting point is instead FEP. We replace the integral form (2.3) with a sampled VFA

$$V(s; \beta) := \beta_0 + \sum_{i=1}^N \beta_i \varphi(s; \theta^i),$$

where $\theta^1, \theta^2, \dots, \theta^N$ are iid samples of the basis function vector from ρ and β is the finite weight vector $(\beta_0, \beta_1, \dots, \beta_N) \in \mathbb{R}^{N+1}$. The weight β_0 represents an intercept as in FEP and the remaining elements of β are weights associated with the random basis functions. In other words, $\beta_1, \beta_2, \dots, \beta_N$ is the finite analogue of the weighting function \mathbf{B} in FEP and $V(s; \beta)$ can be viewed as an approximation constructed using a functional extension of Monte Carlo sampling applied to $V(s; \beta)$. The resulting ALP with N random basis functions, denoted by FALP_N , is

$$\begin{aligned} \sup_{\beta} \quad & \beta_0 + \sum_{i=1}^N \beta_i \mathbb{E}_{\nu}[\varphi(s; \theta^i)] \\ \text{s.t.} \quad & (1 - \gamma)\beta_0 + \sum_{i=1}^N \beta_i \left(\varphi(s; \theta^i) - \gamma \mathbb{E}[\varphi(s'; \theta^i) \mid s, \mathbf{a}] \right) \leq c(s, \mathbf{a}), \quad (s, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}_s. \end{aligned}$$

This model is a semi-infinite linear program with $N + 1$ variables and an infinite number of constraints. We assume the existence of a solution to FALP_N . This is mild because we can always bound the absolute value of the elements of β by a large constant to ensure the existence of a finite optimal solution without affecting our results. We show this formally in §2.10.2.

Assumption 3 *A finite optimal solution to FALP_N exists.*

Theorem 1 establishes key properties of FALP_N and relies on the constant

$$\Omega := 5(D_s + 1)L\sqrt{\mathbb{E}_\rho[\|\theta\|_2^2]},$$

where $\|\cdot\|_2$ denotes the 2-norm, $D_s := \max_{s \in \mathcal{S}} \|s\|_2$, L is the Lipschitz constant of random basis $\varphi(\cdot)$ defined in Assumption 2, and \mathbb{E}_ρ is the expectation under the distribution ρ . Let $\beta_N^{\text{FA}} := (\beta_{N,0}^{\text{FA}}, \dots, \beta_{N,N}^{\text{FA}})$ represent an optimal solution to FALP_N .

Theorem 1 *The following hold:*

- (i) *For a given N , we have $V(s; \beta_N^{\text{FA}}) \leq V^*(s)$ for all $s \in \mathcal{S}$.*
- (ii) *Suppose there exists a $\underline{\rho} > 0$ such that $\rho(\theta) \geq \underline{\rho}$ for all $\theta \in \Theta$. Given $\delta \in (0, 1]$, we have that any finite optimal FALP_N solution β_N^{FA} satisfies*

$$\|V^* - V(\beta_N^{\text{FA}})\|_{1,\mathbf{v}} \leq \frac{2\|\mathbf{B}^*/\rho\|_{2,\rho}}{(1-\gamma)\underline{\rho}\sqrt{N}} \left(\Omega + \sqrt{2 \ln \left(\frac{1}{\delta} \right)} \right),$$

with a probability of at least $1 - \delta$.

Part (i) of this theorem shows that FALP_N is well-defined and provides a lower bound on the MDP value function V^* at all states. The latter is a known result in approximate linear programming (see, e.g., §2 in De Farias and Van Roy 2003). Part (ii) establishes a high probability $(1, \mathbf{v})$ -norm error bound for this VFA. This bound holds for every choice of \mathbf{v} and decreases at the dimension-independent rate of $1/\sqrt{N}$ akin to Monte Carlo sampling, which are both

encouraging properties. The magnitude of the error increases only logarithmically to obtain a more stringent probability guarantee, that is, as δ is decreased. Its growth also depends on the dimension of the state space as is the case with Monte Carlo sampling. The exact nature of this dependence is captured through Ω and $\underline{\rho}$. In the definition of Ω , both the diameter of the state space D_s and the term $\mathbb{E}_\rho[\|\theta\|_2^2]$ may change as we move to higher dimensions. For example, it can be verified that D_s increases at the rate of \sqrt{d} when the state space is a d -dimensional unit hypercube and $\mathbb{E}_\rho[\|\theta\|_2^2] = 1$ for ReLU bases, that is, it does not change with the state space dimension. The analogous change for the parameter $\underline{\rho}$ depends on the choice of the basis function φ . It can be verified that for Fourier bases and a given probability level, there is a constant $c > 0$ depending on this probability level such that $\underline{\rho}^{-1} = (c/c_\rho)^d$. This suggests that $\underline{\rho}^{-1}$ can be super- or sub- linear in d depending on whether c is larger or smaller than c_ρ , respectively.

Indeed, the nature of the MDP value function V^* also affects the error and this factor is signaled by the presence of the term $\|\mathbf{B}^*\|_{2,\rho}$ in the error bound. When the representation of $V^*(\cdot) = \beta_0^* + \int_{\Theta} \mathbf{B}^*(\theta)\varphi(\cdot;\theta)$ is not unique, one can select $(\beta_0^*, \mathbf{B}^*)$ such that norm $\|\mathbf{B}^*\|_{2,\rho}$ is minimized and this minimum can be viewed as the approximation difficulty associated with V^* when using a class of random basis functions. The condition in Theorem 1(ii) of $\rho(\cdot) \geq \underline{\rho}$ is needed to avoid a situation where random basis functions with a certain set of θ values are needed to approximate the value function well but are not sampled because $\rho(\cdot)$ is zero in this set. This requirement is fairly mild. Sampling distributions with bounded support (e.g., uniform) clearly satisfy it. Since \mathbf{N} is finite, distributions with support over an unbounded set,

such as the normal distribution, satisfy it with high probability because the sampled θ vectors highly likely come from a truncated version of the distribution, which has bounded support.

The error bound in Theorem 1 extends to ALP the random basis function sampling results in Rahimi and Recht (2008), which proposes a functional form of Monte Carlo sampling in the regression setting and assumes knowledge of the function being approximated. If \mathbf{V}^* is known, we can regress N random basis functions against \mathbf{V}^* to compute a VFA defined by the weight vector $\beta_N^{\text{reg}} := \arg \min_{\beta \in \mathbb{R}^{N+1}} \|\mathbf{V}(\beta) - \mathbf{V}^*\|_{1,\mathbf{v}}$. It follows from Proposition 7 that this VFA satisfies the following error bound with a probability of $1 - \delta$:

$$\|\mathbf{V}^* - \mathbf{V}(\beta_N^{\text{reg}})\|_{1,\mathbf{v}} \leq \frac{\|\mathbf{B}^*/\rho\|_{2,\rho}}{\rho\sqrt{N}} \left(\Omega + \sqrt{2 \ln \left(\frac{1}{\delta} \right)} \right). \quad (2.4)$$

The $(2, \rho)$ -norm term in (2.4) involving \mathbf{B}^* improves on an ∞ -norm variant of this term in the original bound of Rahimi and Recht (2008) because we employ in the proofs a solution construction that differs from the one used in that paper. The error bound in (2.4) is unattainable because \mathbf{V}^* is unknown. FALP_N provides a mechanism to compute a VFA without the knowledge of \mathbf{V}^* at the cost of incurring the higher approximation error shown in Theorem 1 compared to (2.4). This increase in error occurs because FALP_N is equivalent to the following constrained regression, a result derived from Lemma 1 in De Farias and Van Roy (2003):

$$\begin{aligned} \min_{\beta} \quad & \|\mathbf{V}(\beta) - \mathbf{V}^*\|_{1,\mathbf{v}} \\ \text{s.t.} \quad & \mathbf{V}(\mathbf{s}; \beta) - \gamma \mathbb{E}[\mathbf{V}(\mathbf{s}'; \beta) \mid \mathbf{s}, \mathbf{a}] \leq \mathbf{c}(\mathbf{s}, \mathbf{a}), \quad \forall (\mathbf{s}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}_{\mathbf{s}}. \end{aligned} \quad (2.5)$$

Proposition 2 establishes that satisfying the constraints in FALP_N worsens the error bound in (2.4) by a factor of $2/(1-\gamma)$, which is precisely what we observe in Theorem 1. In other words, to avoid the need for knowing V^* , which is assumed in the definition of β_N^{reg} , FALP_N incurs a cost of feasibility captured by the factor of $2/(1-\gamma)$.

Proposition 2 *If $\|V^* - V(\beta_N^{\text{reg}})\|_{1,v} \leq \varepsilon$, then $\|V^* - V(\beta_N^{\text{FA}})\|_{1,v} \leq (2/(1-\gamma))\varepsilon$.*

We note that the universal random basis functions that underpin the convergence result in Theorem 1 are closely related to universal kernels (Micchelli et al. 2006). For random basis function φ with sampling density ρ , the associated kernel $k : \mathcal{S} \times \mathcal{S} \mapsto \mathbb{R}$ at centroid $\hat{s} \in \mathcal{S}$ is defined as $k(s, \hat{s}) := \int_{\Theta} \rho(\theta) \varphi(s; \theta) \varphi(\hat{s}; \theta) d\theta$ for all $s \in \mathcal{S}$. From this integral relationship, random basis functions can be viewed as a way of using samples from the known distribution ρ to approximate kernels. Rahimi and Recht (2008) show that the class of functions \mathcal{R} spanned by random basis functions coincides with the space of all finite linear combinations of their associated kernels, $\{\alpha_0 + \sum_i \alpha_i k(\cdot; \hat{s}_i) : \alpha_i \in \mathbb{R}, \hat{s}_i \in \mathcal{S}\}$. Nevertheless, unlike kernels, random basis functions do not require the specification of centroids \hat{s}_i . Optimizing the centroid locations of a collection of N kernels is, in general, non-convex. Sampling centroids requires a sampling distribution over the state space. Such a distribution that ensures convergence rate guarantees is not readily available, and thus sampling is often done heuristically (see, e.g., Bhat et al. 2012).

2.3.2 Implementation Guidelines

We outline an implementation strategy that utilizes constraint sampling to approximate \mathbf{FALP}_N and calculate VFA weights β . These weights can be used to define the greedy policy $\pi_g(\beta)$ (see, e.g., [Powell 2007](#)). The action $\pi_g(s; \beta)$ taken by this policy at state $s \in \mathcal{S}$ solves

$$\min_{a \in \mathcal{A}_s} \{c(s, a) + \gamma \mathbb{E}[V(s'; \beta) \mid s, a]\}. \quad (2.6)$$

Given VFA weights β , the cost of the greedy (feasible) policy $\pi_g(\beta)$, which we denote by $PC(\beta) \equiv PC(\pi_g(\beta))$ (where $PC(\cdot)$ was defined in §2.2.1), is an upper bound on the optimal policy cost. In addition, these weights can be incorporated into other methods in the literature to obtain a lower bound on the optimal cost, which can be used for benchmarking purposes. Please see §2.12.1 and §2.12.2 for a discussion of two such methodologies.

The key step in constraint sampling to solve \mathbf{FALP}_N is to replace its set of constraints with a subset obtained by sampling K iid state-action pairs $\{(s^k, a^k) \in \mathcal{S} \times \mathcal{A}_s : k = 1, 2, \dots, K\}$ from a probability distribution ψ over the state-action space $\mathcal{S} \times \mathcal{A}_s$ ([Calafiore and Campi 2005](#)). The result is the following linear program with N random basis functions and K constraint samples:

$$\begin{aligned} \max_{\beta} \quad & \beta_0 + \sum_{i=1}^N \beta_i \mathbb{E}_{\nu}[\varphi(s; \theta^i)] \\ \text{s.t.} \quad & (1 - \gamma)\beta_0 + \sum_{i=1}^N \beta_i \left(\varphi(s^k; \theta^i) - \gamma \mathbb{E}[\varphi(s'; \theta^i) \mid s^k, a^k] \right) \leq c(s^k, a^k), \quad k = 1, 2, \dots, K. \end{aligned} \quad (2.7)$$

Proposition 3 is an application of a key result in [Calafiore and Campi \(2006\)](#) and shows that the linear program (2.7) for large enough K provides a good randomized approximation of \mathbf{FALP}_N .

Proposition 3 (Theorem 1 in [Calafiore and Campi 2006](#)) *Given $\delta \in (0, 1]$, if ψ is supported over $\mathcal{S} \times \mathcal{A}_s$, linear program (2.7) is bounded, and*

$$K \geq \left\lceil \frac{2}{\delta} \ln\left(\frac{1}{\delta}\right) + 2(N+1) + \frac{2(N+1)}{\delta} \ln\left(\frac{2}{\delta}\right) \right\rceil,$$

then for every optimal solution $\hat{\beta}$ to this program, the following inequality holds

$$\psi \left(\left\{ (s, a) \in \mathcal{S} \times \mathcal{A}_s \mid (1 - \gamma)\hat{\beta}_0 + \sum_{i=1}^N \hat{\beta}_i \left(\varphi(s; \theta^i) - \gamma \mathbb{E}[\varphi(s'; \theta^i) | s, a] \right) \leq c(s, a) \right\} \right) \geq 1 - \delta,$$

with a probability of at least $1 - \delta$.

In particular, this proposition shows that as more samples are added, the set of states where the FALP constraints are violated when measured using ψ is at most δ and this holds with a probability of at least $1 - \delta$. Therefore, if one solves the constraint-sampled version of FALP in (2.7) with a large number of samples K , we expect the results in Theorem 1 to hold approximately.

A sharper constraint sampling result specific to ALP can be found in [De Farias and Van Roy \(2004, Theorem 3.1\)](#) when ψ is chosen using information from the optimal policy, which is unknown. During implementation, ψ can be a uniform distribution or based on states visited by a baseline policy. A candidate baseline policy is greedy policy $\pi_g(\beta)$ computed based on an ALP VFA weights β . Expectations in (2.7) are typically replaced by sample average approximations. The number of constraint samples K can be chosen so that the optimal objective function of (2.7) does not decrease significantly as more samples are added. Once these parameters are set, the optimal solution $\hat{\beta}$ to (2.7) defines VFA $V(\hat{\beta})$.

The quality of the VFA obtained using the above procedure depends on how FALP is formulated, in particular, the number of basis function samples N , the choice of random basis functions, and the state relevance distribution \mathbf{v} . We provide some guidance on these choices next.

Similar to standard Monte Carlo sampling, the value of N depends on the computational budget. That is, one determines the largest N for which the sampled version (2.7) of FALP_N can be tackled within a reasonable time limit (and possibly memory limit) using an off-the-shelf commercial solver. The ability to get good VFAs with a small number of basis functions N is thus an important consideration in choosing random basis functions. While multiple universal random basis functions guarantee the same theoretical convergence rate, their empirical rates may differ. A good starting point is to consider random Fourier basis functions (see Table I), as they are known to provide better approximations as the continuous function V^* becomes smoother (please see §2.1.1 of [Canuto et al. 2012](#) and [Nersessian 2019](#) for recent examples). The non-smoothness of the MDP value function in several applications is localized, that is, even these value functions are smooth in most neighborhoods. Given a choice of random basis functions, the tunable parameters are few and do not depend on the application. The random bases examples in Table I have at most one such parameter (i.e., c_ρ). We recommend tuning this parameter according to the following steps: (i) choose multiple candidate values for c_ρ , (ii) sample N random basis functions for each candidate, (iii) solve the constraint-sampled version (2.7) of FALP_N based on each candidate, and (iv) choose the candidate with the highest optimal objective value. Fourier basis functions, which depend on a single bandwidth parameter, tuned using the aforementioned simple tuning strategy, worked well in both applications in our numerical experiments.

The state-relevance distribution \mathbf{v} plays an important role in linking the quality of the VFA with weights β to the performance of its associated greedy policy $\pi_g(\beta)$ (De Farias and Van Roy 2003, Desai et al. 2012a, Sun et al. 2014). Proposition 4 formalizes this link using the state-visit frequency $\mu_\chi(\beta)$ of this greedy policy, which defines the following probability of visiting a subset of states $\mathcal{S}_1 \subseteq \mathcal{S}$ (see, e.g., pages 132–133 in Hernández-Lerma and Lasserre 1996):

$$\mu_\chi(\mathcal{S}_1; \beta) := \chi(\mathcal{S}_1) + \sum_{t=0}^{\infty} \gamma^{t+1} \mathbb{E} \left[\mathbb{P}(s_{t+1}^{\pi_g(\beta)} \in \mathcal{S}_1 \mid s_t, \pi_g(s_t; \beta)) \right], \quad (2.8)$$

where state $s_{t+1}^{\pi_g(\beta)}$ and transition probability distribution \mathbf{P} retain their definitions from §2.2, and $\chi(\mathcal{S}_1)$ is the probability of the initial state belonging to \mathcal{S}_1 . The expectation \mathbb{E} is taken with respect to control policy $\pi_g(\beta)$ and the distribution χ over initial the state s_0 .

Proposition 4 (Theorem 1 in De Farias and Van Roy 2003) *For a VFA $V(\beta)$ such that $V(\beta) \leq V^*$, we have*

$$\text{PC}(\beta) - \text{PC}(\pi^*) \leq \frac{1}{1-\gamma} \|V(\beta) - V^*\|_{1, \mu_\chi(\beta)}.$$

Proposition 4 shows that for a VFA $V(\beta)$ that lower bounds V^* (e.g., the FALP VFA), the additional cost incurred by using the greedy policy $\pi_g(\beta)$ instead of the optimal policy π^* is bounded above by the $(1, \mu_\chi(\beta))$ -norm difference between the VFA $V(\beta)$ and the MDP value function V^* . This result motivates the search for good VFAs.

If \mathbf{v} and $\mu_\chi(\beta_N^{\text{FA}})$ are identical, Proposition 4 and the reformulation (2.5) imply that FALP VFA $V(\beta_N^{\text{FA}})$ with a small $(1, \mathbf{v})$ -norm error also guarantees good performance for greedy policy

$\pi_g(\beta_N^{\text{FA}})$. However, one does not know $\mu_\chi(\beta_N^{\text{FA}})$ before solving FALP, which makes this choice challenging (De Farias and Van Roy 2003). Heuristics in the literature can be interpreted as approximating the expression (2.8) for $\mu_\chi(\beta_N^{\text{FA}})$. They either consider a static choice of \mathbf{v} or dynamically update it. Popular examples of static choices of \mathbf{v} are (i) the initial state distribution χ , which ignores the second term in (2.8) capturing the effect of states visited by the policy in the future; (ii) a uniform distribution, which can be interpreted as acknowledging that we do not have any information about \mathbf{v} ; and (iii) the state-visit frequency of a baseline policy π , which can be estimated by simulating this policy.

De Farias and Van Roy (2003) and Farias and Van Roy (2006) describe a dynamic approach to update the state-relevance distribution by iteratively applying the third static choice for \mathbf{v} mentioned above. Algorithm 1 summarizes this approach used in conjunction with FALP to guide the choice of the state-relevance distribution. To ease exposition, we make the dependence of FALP_N on \mathbf{v} explicit by writing $\text{FALP}_N[\mathbf{v}]$ and assume this refers to the constraint-sampled version (2.7). The initial iteration $q = 0$ starts by solving $\text{FALP}_N[\mathbf{v}^0]$ based on an initial state-relevance distribution choice \mathbf{v}^0 to obtain the VFA weights β^0 . Then, it simulates the greedy policy $\pi_g(\beta^0)$ to obtain the state-visit distribution $\mu_\chi(\beta^0)$. This distribution is chosen as the new state-relevance distribution \mathbf{v}^1 . Iteration $q = 1$ starts by solving $\text{FALP}_N[\mathbf{v}^1]$ and so on. A total of Q iterations are performed, after which the VFA weights β^{Q-1} is returned. The algorithm thus updates the state state-relevance distribution $Q - 1$ times, while retaining the same random basis functions, that is, the same randomized one-shot approximation. We refer to Algorithm 1 as policy-guided FALP. As Algorithm 1 iterates, one hopes that the state-relevance distribution

Algorithm 1: Policy-guided FALP

Receive: number of random basis functions N , random basis function φ with sampling density ρ , initial state-relevance distribution \mathbf{v}^0 , and maximum number of iterations Q .

Initialize: formulate $\text{FALP}_N[\mathbf{v}^0]$ using \mathbf{v}^0 and random basis function φ with N iid θ samples from ρ .

for $q = 0, 1, \dots, Q - 1$ **do**

(i) Solve $\text{FALP}_N[\mathbf{v}^q]$ to obtain VFA weights β^q .

(ii) Simulate greedy policy $\pi_g(\beta^q)$ to estimate $\mu_\chi(\beta^q)$, and then set $\mathbf{v}_{q+1} \leftarrow \mu_\chi(\beta^q)$.

Return: VFA weights β^{Q-1} .

$\mu_\chi(\beta^q)$ overlaps more with states visited under greedy policy $\pi_g(\beta^q)$, but there is no guarantee that this will happen. In addition, this dynamic approach is more costly than a static choice of \mathbf{v} . As N becomes larger, the time for a single iteration of Algorithm 1 increases, which includes solving $\text{FALP}_N[\mathbf{v}^q]$ to compute VFA weights β^q and simulating the greedy policy $\pi_g(\beta^q)$. This is because $\text{FALP}_N[\mathbf{v}^q]$ will have more variables, so we need to evaluate expectations of a larger number of random basis functions during policy simulation. A sequential strategy is to first select N such that the per iteration cost allows for choosing Q such that a few iterations can be performed within an acceptable time limit.

2.4 Randomized Multi-Shot Approximation

In this section, we introduce a randomized multi-shot approximation approach for dynamically updating the state-relevance distribution that leverages our ability to inexpensively sample

new random basis functions. We present the model and algorithm in §2.4.1, interpret it in §2.4.2, provide supporting theory in §2.4.3, and discuss implementation guidelines in §2.4.4.

2.4.1 Model and Algorithm

Our randomized multi-shot approximation scheme gradually increases the number of basis functions in FALP by sampling new batches of random basis functions of size B and adds guiding constraints to FALP that link the VFAs across consecutive iterations. For a given N , we refer to this modification of FALP_N as $\text{FALP}_N^{\text{SG}}$ (SG stands for self-guiding).

Consider sampling random basis functions in batches of size B iteratively. At iteration $q \in \{0, 1, \dots, Q\}$, model $\text{FALP}_N^{\text{SG}}$ with $N = qB$ random basis functions is

$$\begin{aligned} \max_{\beta} \quad & \beta_0 + \sum_{i=1}^N \beta_i \mathbb{E}_{\mathbf{v}}[\varphi(\mathbf{s}; \theta^i)] \\ \text{s.t.} \quad & (1 - \gamma)\beta_0 + \sum_{i=1}^N \beta_i \left(\varphi(\mathbf{s}; \theta^i) - \gamma \mathbb{E}[\varphi(\mathbf{s}'; \theta^i) | \mathbf{s}, \mathbf{a}] \right) \leq c(\mathbf{s}, \mathbf{a}), \quad \forall (\mathbf{s}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}_{\mathbf{s}}, \quad (2.9) \end{aligned}$$

$$\beta_0 + \sum_{i=1}^N \beta_i \varphi(\mathbf{s}; \theta^i) \geq V(\mathbf{s}; \beta_{N-B}^{\text{SG}}), \quad \forall \mathbf{s} \in \mathcal{S}. \quad (2.10)$$

The only difference between $\text{FALP}_N^{\text{SG}}$ and FALP_N is that the former linear program includes additional constraints (2.10) that require its VFA to be a state-wise upper bound on the past VFA $V(\beta_{N-B}^{\text{SG}})$, which is computed in the previous iteration $q - 1$ by solving $\text{FALP}_{N-B}^{\text{SG}}$. Note that the N -dimensional vector $(\beta_{N-B}^{\text{SG}}, 0, \dots, 0)$ obtained by appending B zeros to the VFA weights β_{N-B}^{SG} is feasible to $\text{FALP}_N^{\text{SG}}$. At the first iteration (i.e., $q = 0$), the $\text{FALP}_0^{\text{SG}}$ VFA becomes a constant function that only includes an intercept term. We assume $V(\mathbf{s}; \beta_{-B}^{\text{SG}}) \equiv -\infty$ for all $\mathbf{s} \in \mathcal{S}$, which implies that the guiding constraints (2.10) are redundant in the first iteration. We refer to the

Algorithm 2: Self-guided FALP

Receive: sampling batch size B , random basis function φ with sampling density ρ , state-relevance distribution ν , and maximum number of iterations Q .

Initialize: the set ϑ of sampled θ vectors to $\{\}$.

for $q = 0, 1, \dots, Q - 1$ **do**

- (i) Set $N \leftarrow qB$.
- (ii) Compute coefficients β_N^{SG} by solving $\text{FALP}_N^{\text{SG}}$ formulated using N random basis functions with parameters in set ϑ , the state-relevance distribution ν , and the past VFA $V(\beta_{N-B}^{\text{SG}})$.
- (iii) Draw B iid samples $\{\theta^1, \dots, \theta^B\}$ from ρ and update $\vartheta \leftarrow \vartheta \cup \{\theta^1, \dots, \theta^B\}$.

Return: VFA weights β_N^{SG} .

resulting iterative scheme summarized in Algorithm 2 as self-guided FALP because the new constraints (2.10) use its own past VFA (hence the label “self”) to shape the current VFA (hence the label “guided”).

The inputs to Algorithm 2 are similar to Algorithm 1, except for the batch size B , which replaces the apriori fixed number of basis functions N across iterations. At each iteration $q \geq 0$, Algorithm 2 (i) sets the number of random basis function N to qB , (ii) solves a revised $\text{FALP}_N^{\text{SG}}$ model formulated with B additional random basis functions compared to $\text{FALP}_{N-B}^{\text{SG}}$, and (iii) samples a batch of θ vectors of size B and includes them in the current set ϑ of such vectors. After Q iterations, it returns the VFA weights β_N^{SG} , where $N = (Q - 1)B$.

Proposition 5 establishes a key property of the VFAs generated by Algorithm 2.

Proposition 5 *At any iteration $q \geq 1$ of Algorithm 2 with $N = qB$, it holds that*

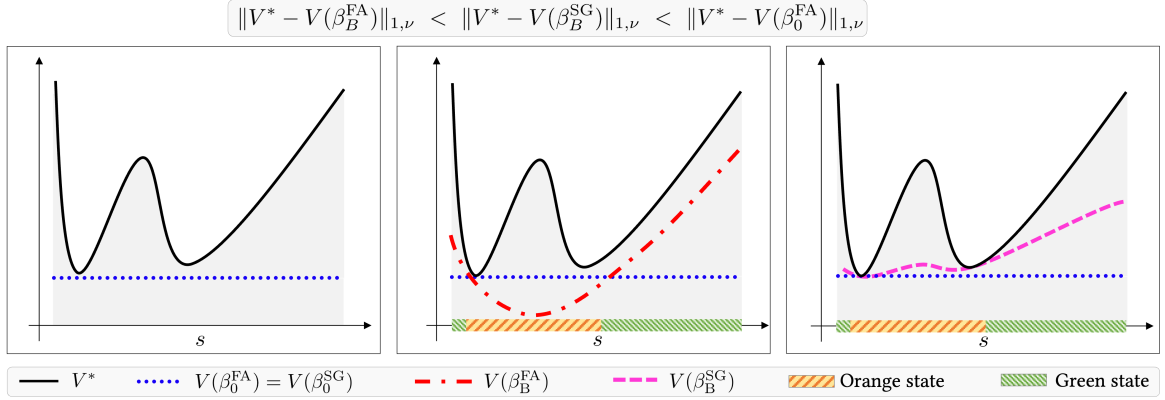
$$V(s; \beta_0^{\text{FA}}) = V(s; \beta_0^{\text{SG}}) \leq V(s; \beta_B^{\text{SG}}) \leq \dots \leq V(s; \beta_N^{\text{SG}}) \leq V^*(s), \quad \forall s \in \mathcal{S}. \quad (2.11)$$

The equality in (2.11) follows from our assumption that $V(\cdot; \beta_{-B}^{\text{SG}}) = -\infty$. For every iteration $q \geq 0$ with $N = qB$, the relationship $V(s; \beta_N^{\text{SG}}) \leq V^*(s)$ holds for all $s \in \mathcal{S}$. This follows from Part (i) of Theorem 1 because β_N^{SG} is feasible to $\text{FALP}_N^{\text{SG}}$, and thus, it is also feasible to FALP_N . The inequalities of the type $V(s; \beta_{N-B}^{\text{SG}}) \leq V(s; \beta_N^{\text{SG}})$ are directly implied by the guiding constraints (2.10).

An important consequence of Proposition 5 is that Algorithm 2 generates a sequence of VFAs that draws (weakly) closer to V^* at all states. Therefore, two consecutive VFAs with $N - B$ and N random basis functions satisfy

$$\|V(\beta_N^{\text{SG}}) - V^*\|_{1, \mu} \leq \|V(\beta_{N-B}^{\text{SG}}) - V^*\|_{1, \mu},$$

for any proper distribution μ defined over the state space and, in particular, when μ is the state-visit frequency $\mu_v(\beta_N^{\text{SG}})$ associated with the greedy policy $\pi_g(\beta_N^{\text{SG}})$. As a result, for any fixed iteration index $\bar{q} \geq 1$ and its corresponding state-visit frequency $\mu_\chi(\beta_{\bar{q}B}^{\text{SG}})$, it follows that the sequence of VFAs $\{V(\beta_{qB}^{\text{SG}}) : q = 0, 1, \dots, \bar{q}\}$ generated by Algorithm 2 improves the worst-case performance bound of greedy policies in Proposition 4, that is $\|V(\beta_{qB}^{\text{SG}}) - V^*\|_{1, \mu_\chi(\beta_{\bar{q}B}^{\text{SG}})}$ is non-increasing in q .

Figure 1: Illustration of self-guiding mechanism with ν equal to a uniform distribution.

2.4.2 Understanding the Self-guiding Mechanism

We begin by shedding light on a connection between the guiding constraints and the greedy policy performance using the illustrative example in Figure 1. Consider the left panel in this figure, where the MDP value function V^* is represented by a (black) solid line. The lowest-cost state of V^* corresponds to the global minimum of this function. The (blue) dotted line represents an intercept-only VFA $V(\beta_0^{\text{FA}})$ obtained from FALP_N with $N = 0$ random basis functions and a uniform state-relevance distribution. Leveraging the constrained regression equivalence of FALP in (2.5), it follows that the intercept-only VFA (i) must be below V^* (i.e., gray region) because of the constraints and (ii) must equal the value of V^* at the lowest-cost state because the objective minimizes the $(1, \nu)$ -norm distance to V^* . This results in our first observation:

(O1) *The intercept-only VFA $V(\beta_0^{\text{FA}})$ provides a constant approximation across all states that equals V^* at the lowest-cost state.*

However, since all states receive an identical value under the intercept-only VFA, its greedy policy is driven by the immediate cost (i.e., myopic). This is undesirable and motivates adding random basis functions to obtain richer VFAs that direct their greedy policies towards the low-cost state.

The middle panel of Figure 1 shows in (red) dotted-and-dashed line the richer VFA $V(\beta_B^{\text{FA}})$ that is computed by FALP_N with $N = B$ basis functions and a uniform state-relevance distribution. This VFA assigns low values to states having high cost under V^* , thus incorrectly directing its greedy policy. To elaborate, vector β_B^{FA} defining this VFA is optimal to FALP_B because we assume that the chosen B basis functions only represent specific shapes within the gray region and cannot entirely span it. Particularly, these basis functions do not represent functions in the gray region that are both above the intercept-only VFA at all states (i.e., visually above), and have a lower $(1, \nu)$ -norm than the VFA $V(\beta_B^{\text{FA}})$ (i.e., better objective value). Our second observation is the following:

(O2) *The FALP_B VFA $V(\beta_B^{\text{FA}})$ provides a better $(1, \nu)$ -norm approximation than $V(\beta_0^{\text{FA}})$ but can result in assigning its lowest value to states that have high cost under V^* .*

The right panel illustrates the VFA $V(\beta_B^{\text{SG}})$ by a (pink) dashed line. This VFA is obtained from $\text{FALP}_N^{\text{SG}}$ with $N = B$ random basis functions and a uniform state relevance distribution. It does not suffer from the issue outlined in (O2) because the guiding constraints require $V(\beta_B^{\text{SG}})$ to be above the intercept-only VFA $V(\beta_0^{\text{FA}})$ at all states. This restriction results in $V(\beta_B^{\text{SG}})$ not

only having an improved $(1, \nu)$ -norm distance to V^* compared to $V(\beta_0^{\text{FA}})$ but also continuing to assign the lowest value to the lowest-cost state under V^* . Our third observation follows.

(O3) *The FALP_B VFA $V(\beta_B^{\text{SG}})$ provides a better $(1, \nu)$ -norm approximation than $V(\beta_0^{\text{FA}})$ and, in addition, assigns its lowest value to the lowest-cost state under V^* .*

Overall, the key takeaway is that good policies are more likely to visit states where V^* is small rather than large. As noted in (O1), the FALP_0 VFA $V(\beta_0^{\text{FA}})$ is exact at the lowest-cost state. The $\text{FALP}_B^{\text{SG}}$ VFA $V(\beta_B^{\text{SG}})$ improves this approximation such that the lowest-cost state under V^* remains the lowest-cost state under $V(\beta_B^{\text{SG}})$, which is (O3). Hence, its greedy policy moves the system toward “real” low-cost states under V^* . In contrast, as mentioned in (O2), the FALP_B VFA $V(\beta_B^{\text{FA}})$ may worsen the approximation quality at the lowest-cost state under V^* , which certainly occurs in Figure 1. Hence, its greedy policy moves the system towards the lowest-cost state under $V(\beta_B^{\text{FA}})$ whose actual cost is higher under V^* . While this illustration considers a single iteration of Algorithm 2, analogous behavior continues in future iterations, with the approximation quality at low-cost states under V^* not being compromised to improve the approximation quality at high-cost states with respect to V^* . We also verified this behavior on a small instance of the inventory control problem tested in our numerical study and provide details in §2.13.1.

Next, to provide insight into the self-guiding mechanism in $\text{FALP}_N^{\text{SG}}$, we dualize constraints (2.10). Specifically, let $\mathbf{y}^*(s) \geq 0$ denote the optimal dual value associated with the constraint (2.10) at state $s \in \mathcal{S}$ and define a state-relevance distribution \mathbf{v}' that evaluates at this state to

$$\mathbf{v}'(s) := \frac{\mathbf{v}(s) + \mathbf{y}^*(s)}{1 + \int_{\mathcal{S}} \mathbf{y}^*(s) \, d s}. \quad (2.12)$$

If strong duality holds, it can be easily verified that an optimal solution of $\text{FALP}_N^{\text{SG}}$ solves

$$\max_{\beta} \beta_0 + \sum_{i=1}^N \beta_i \mathbb{E}_{\mathbf{v}'}[\varphi(s; \theta^i)] \quad \text{s.t.} \quad (2.9).$$

For brevity, we do not discuss the technical conditions for strong duality here (see, e.g., [Shapiro 2009](#), Theorem 2.3, and [Basu et al. 2017](#), Theorem 4.1) because the constraints will be sampled during implementation, in which case standard strong duality for finite linear programs will apply. The above reformulation shows that $\text{FALP}_N^{\text{SG}}$ can be viewed as a modification of the FALP_N static state-relevance distribution using its own past VFA information, that is, the $\text{FALP}_{N-B}^{\text{SG}}$ VFA.

We revisit the example in Figure 1 to illustrate the states where guiding constraints will be binding and the impact of these states on the updated state-relevance distribution \mathbf{v}' used by self-guided FALP (i.e., (2.12)). The guiding constraints must be binding at some of the states colored in orange in the middle panel because $V(\beta_B^{\text{FA}})$ is below $V(\beta_0^{\text{FA}})$, hence β_B^{FA} violates the guiding constraints of $\text{FALP}_B^{\text{SG}}$. By virtue of complementary slackness, the dual variables \mathbf{y}^* in (2.12) take positive values at the subset of orange states where the guiding constraints (2.10) are binding. That is, the updated state-relevance distribution \mathbf{v}' assigns higher values at these

states such that the new VFA $V(\beta_B^{\text{SG}})$ (i) provides a better approximation of V^* at the orange states, and (ii) is above the previous VFA $V(\beta_0^{\text{SG}}) \equiv V(\beta_0^{\text{FA}})$ at all states. This can be seen in the right panel of Figure 1.

2.4.3 Theoretical Guarantees

Studying the quality of the VFAs generated by Algorithm 2 is challenging because consecutive VFAs in this sequence are linked by the guiding constraints (2.10). We propose a new approach to establish an error bound for self-guided FALP VFAs. Specifically, we use the $\text{FALP}_N^{\text{SG}}$ VFA composed of $N \geq 1$ random basis functions as a baseline and analyze the rate at which the $(1, \mathbf{v})$ -norm distance between the $\text{FALP}_{N+H}^{\text{SG}}$ VFA $V(\beta_{N+H}^{\text{SG}})$ and V^* decreases as H new random basis functions are added. We proceed in two steps.

Step 1: Effect of $\text{FALP}_N^{\text{SG}}$ VFA. Consider the set of functions spanned by an intercept plus a linear combination of N random basis functions in set $\{\varphi(\cdot; \theta^1), \varphi(\cdot; \theta^2), \dots, \varphi(\cdot; \theta^N)\}$:

$$\mathcal{W}_N := \left\{ V \in \mathcal{R} \mid \exists (\beta_0, \beta_1, \dots, \beta_N) \in \mathbb{R}^{N+1} \text{ s.t. } V(\cdot) = \beta_0 + \sum_{i=1}^N \beta_i \varphi(\cdot; \theta^i) \right\}.$$

A strategy to account for the impact of $V(\beta_N^{\text{SG}})$ on the number of additional basis functions H is to ask if V^* is a part of the functional space \mathcal{W}_N containing $V(\beta_N^{\text{SG}})$. If $V^* \in \mathcal{W}_N$, then it would not be possible to improve the incumbent VFA $V(\beta_N^{\text{SG}})$ via additional sampling. If $V^* \notin \mathcal{W}_N$, then V^* intuitively has a (projected) component in the functional space \mathcal{W}_N , as well as a nonzero (projected) component in the orthogonal complement of this space. We approximate this orthogonal component using the H additional random basis functions.

Formally, we decompose V^* as $V^* = V(\beta_N^{*,o}) + V(\beta_N^{*,\perp})$, where functions $V(\beta_N^{*,o})$ and $V(\beta_N^{*,\perp})$ are the projections of V^* onto \mathcal{W}_N and its orthogonal complement, respectively (to be precise, these projections are performed onto the closures of these sets). We design an idealized VFA $V(\hat{\beta}_{N+H}) \in \mathcal{W}_{N+H}$ with associated vector $\hat{\beta}_{N+H} \in \mathbb{R}^{N+H+1}$ that retains the approximation quality of $V(\beta_N^{*,o})$ and uses the H additional random basis functions to approximate $V(\beta_N^{*,\perp})$. As H increases, this VFA $V(\hat{\beta}_{N+H})$ becomes increasingly close to V^* with high probability (specifically, probability $1 - \delta$) as shown by the error bound $\|V(\hat{\beta}_{N+H}) - V^*\|_{1,v} \leq E_{(N,H)}$, where

$$E_{(N,H)} := \frac{\|\mathbf{B}_N^{*,\perp}/\rho\|_{2,\rho}}{\rho\sqrt{H}} \left(\Omega + 2\sqrt{2\ln\left(\frac{1}{\delta}\right)} \right).$$

This bound is unattainable since constructing $V(\hat{\beta}_{N+H})$ involves a direct regression on $V(\beta_N^{*,\perp})$, the knowledge of which is unavailable (see Theorem 7 for details). It is thus similar to the unattainable bound (2.4) that regresses on V^* but with an important difference. The term $E_{(N,H)}$ contains the norm $\|\mathbf{B}_N^{*,\perp}/\rho\|_{2,\rho}$ in lieu of $\|\mathbf{B}^*/\rho\|_{2,\rho}$ in (2.4). It is easy to verify that $\|\mathbf{B}_N^{*,\perp}/\rho\|_{2,\rho} < \|\mathbf{B}^*/\rho\|_{2,\rho}$ if the projection of V^* onto \mathcal{W}_N is nonzero. The difference between these norms signals the quality of the most recently computed VFA $V(\beta_N^{\text{SG}})$. This suggests that the number of additional samples H needed to obtain a good approximation of V^* decreases with $\|\mathbf{B}_N^{*,\perp}/\rho\|_{2,\rho}$, that is, when $V(\beta_N^{\text{SG}})$ is itself closer to V^* .

Step 2: Cost of Feasibility. We now turn to update the unattainable error bound from Step 1 to one that relates to the $\text{FALP}_{N+H}^{\text{SG}}$ VFA. The key challenge in doing so is that the idealized VFA $V(\hat{\beta}_{N+H})$ may not belong to the set of feasible solutions \mathcal{F}_{N+H} defined by

constraints (2.9) and (2.10) of $\text{FALP}_{N+H}^{\text{SG}}$ – we can show that $\hat{\beta}_{N+H}$ violates these constraints by at most $(1 + \gamma)E_{(N,H)}$. Indeed, the projection of $\hat{\beta}_{N+H}$ onto \mathcal{F}_{N+H} , denoted by $\text{proj}_{N+H}(\hat{\beta}_{N+H})$, has zero violation. It also satisfies $\|V^* - V(\beta_{N+H}^{\text{SG}})\|_{1,\nu} \leq \|V^* - V(\text{proj}_{N+H}(\hat{\beta}_{N+H}))\|_{1,\nu}$ for any optimal solution β_{N+H}^{SG} to $\text{FALP}_{N+H}^{\text{SG}}$. We thus need to upper bound $\|V^* - V(\text{proj}_{N+H}(\hat{\beta}_{N+H}))\|_{1,\nu}$, which we do via the following triangle inequality:

$$\|V^* - V(\text{proj}_{N+H}(\hat{\beta}_{N+H}))\|_{1,\nu} \leq \|V^* - V(\hat{\beta}_{N+H})\|_{1,\nu} + \|V(\hat{\beta}_{N+H}) - V(\text{proj}_{N+H}(\hat{\beta}_{N+H}))\|_{1,\nu}.$$

The first term is bounded above by $E_{(N,H)}$ from Step 1. For the second term, we show that the inequality $\|V(\hat{\beta}_{N+H}) - V(\text{proj}_{N+H}(\hat{\beta}_{N+H}))\|_{1,\nu} \leq \|\hat{\beta}_{N+H} - \text{proj}_{N+H}(\hat{\beta}_{N+H})\|_1$ holds because random basis function evaluations are no more than 1 (Assumption 8). The 1-norm difference $\|\hat{\beta}_{N+H} - \text{proj}_{N+H}(\hat{\beta}_{N+H})\|_1$ can itself be upper bounded directly from the error bound condition (EBC) used in the optimization literature (Lewis and Pang 1998, Drusvyatskiy and Lewis 2018, Van Ngai et al. 2010, Lin et al. 2022), which is stated as Assumption 4. Define function $\omega : \mathbb{R}^{N+H+1} \mapsto [0, \infty)$ at $\beta \in \mathbb{R}^{N+H+1}$ as follows:

$$\omega(\beta) := \max \left\{ 0, \max_{s \in \mathcal{S}} \{V(s; \beta_{N-B}^{\text{SG}}) - V(s; \beta)\}, \max_{(s,a) \in \mathcal{S} \times \mathcal{A}_s} \{V(s; \beta) - \gamma \mathbb{E}[V(s'; \beta)|s, a] - c(s, a)\} \right\}.$$

Given VFA $V(\beta_{N-B}^{\text{SG}})$, the value of $\omega(\beta)$ measures the maximum violation in $\text{FALP}_{N+H}^{\text{SG}}$ constraints by a vector β , where $\omega(\beta) = 0$ if β is feasible to $\text{FALP}_{N+H}^{\text{SG}}$ and $\omega(\beta) > 0$ otherwise.

Assumption 4 (EBC) Fix $N \geq 1$ and $H \geq 1$. There exists a constant $G > 0$ and an exponent $m \geq 1$ such that for every vector $\beta \in \mathbb{R}^{N+H+1}$, it holds that $\omega(\beta) \geq \frac{1}{G} \|\text{proj}_{N+H}(\beta) - \beta\|_1^m$.

EBC ensures that $\omega(\beta)$ is lower bounded by a degree m polynomial of the 1-norm distance between the vectors $\text{proj}_{N+H}(\beta)$ and β , which is mild. For instance, it is known that m equals 1 for finite linear programs, which applies to $\text{FALP}_{N+H}^{\text{SG}}$ with constraint sampling. For a general semi-infinite linear system, both cases of $m = 1$ and $m > 1$ can occur, with [Van Ngai et al. \(2010\)](#) providing technical conditions under which $m = 1$ holds.

The consequence of steps 1 and 2 described above is the high probability $(1, \nu)$ -norm error bound for the self-guided FALP VFA stated in Theorem 2. Indeed the difference between $E_{(N,H)}$ and the error bound in this theorem can be interpreted as (i) the cost of feasibility to overcome the lack of knowledge of V^* by satisfying (2.9) and (ii) an additional cost of feasibility to ensure improvement in the worst-case greedy policy performance by satisfying the guiding constraints (2.10).

Theorem 2 Suppose Assumption 4 holds and $\rho(\theta) \geq \underline{\rho}$ for all $\theta \in \Theta$ and let β_{N+H}^{SG} be any $\text{FALP}_{N+H}^{\text{SG}}$ optimal solution. Given $\delta \in (0, 1]$ and $N \geq 1$, for any $H \geq 1$, it holds that

$$\|V^* - V(\beta_{N+H}^{\text{SG}})\|_{1,\nu} \leq E_{(N,H)} \left[1 + G ((1 + \gamma) E_{(N,H)})^{(1-m)/m} \right],$$

with a probability of at least $1 - \delta$.

This bound consists of two terms: the first term is the idealized rate from Step 1, and the second term is the worsening of this rate as a result of ensuring feasibility. The convergence

rates with respect to N of the first and second terms are $1/\sqrt{N}$ and $1/\sqrt[m]{N}$, respectively. If ensuring feasibility is difficult for an instance (i.e., $m > 1$), then the dominant rate is $1/\sqrt[m]{N}$. However, if ensuring feasibility is easy (i.e., $m = 1$), then the $\text{FALP}_{N+H}^{\text{SG}}$ VFA error rate is $1/\sqrt{N}$, which is similar to the error rate associated with FALP in Theorem 1. It is also worth noting that m equals 1 and G equals $1/(1 - \gamma)$ for FALP_{N+H} , which is easy to verify based on the proof of Theorem 1 and prior results in ALP. The rate in Theorem 2 essentially reduces to the rate of Theorem 1 under these choices for m and G . Our convergence rate can thus be seen as a way of generalizing the feasibility analysis in the ALP literature when additional constraints are added to its formulation.

2.4.4 Implementation Guidelines

We discuss the implementation guidelines for Algorithm 2, focusing on parameter choices and solution issues that were not already discussed in §2.3.2. Specifically, we need to choose the batch size B and the number of iterations Q . These choices become easier if we fix a target number of basis functions $N = (Q - 1)B$ following the logic discussed for FALP in §2.3.2. Then, smaller values of B entail solving linear programs with fewer decision variables and doing so more often. In other words, the per iteration cost is lower with smaller B , but more iterations are needed and the improvement between iterations will likely be smaller. Therefore, the value of B can be selected to balance improvement in the self-guided FALP objective function value and the per-iteration cost. Solving self-guided FALP requires handling both constraints (2.9) and (2.10). We suggest replacing these constraints with a sampled subset, as done for FALP in

§2.3.2. Under such replacement, analogues of Proposition 5 and the discussion following it hold over the sampled states (please see §2.11 for details).

Although we consider an iteration limit as the stopping criterion in Algorithm 1, several alternatives are possible. For instance, the iteration limit can be replaced by a time limit, or both types of limits can be imposed together. Another strategy is to look at the improvement of consecutive policies and stop when these improvements are smaller than a certain threshold. If a lower bound on the optimal policy cost is available, these improvements can be converted to optimality gaps, and a termination gap can be set.

2.5 Extensions

Although we have assumed continuous state spaces and value functions thus far, the random basis function sampling approach underpinning our models can be readily extended to handle discounted-cost MDPs with finite state spaces. A special structure that arises in important applications is a state space with a low dimensional discrete component and a high dimensional continuous component (e.g., financial and real options pricing). In this case, it is common to define a separate continuous VFA for each discrete state value, and our results directly apply. Next, we handle the more general case when such a strategy may not be computationally feasible.

Consider the analogue of the MDP in §2.2.1 with a discrete state space $\mathcal{S} := \{\mathbf{s}^m \in \mathbb{R}^d : m \in \mathcal{M}\}$, where \mathcal{M} is a finite index set and each state \mathbf{s}^m is a bounded real-valued vector. We denote by V^* the MDP value function. Proposition 6 provides a bound on the ∞ -norm error between the FALP_N VFA and V^* , which decreases at a rate of $1/\sqrt{N}$ as more random basis functions are sampled. Such a bound is possible because we can construct a continuous extension of V^* , as

discussed next. Let \mathcal{S}^C be the smallest continuous and compact set containing \mathcal{S} . It is easy to verify that the following continuous function defined for each $s \in \mathcal{S}^C$ coincides with V^* at all the discrete states:

$$V^C(s) := \sum_{m \in \mathcal{M}} V^*(s^m) \max \left\{ 0, 1 - \frac{\|s - s^m\|_2}{\underline{s}} \right\},$$

where $\underline{s} := \min \{ \|s^m - s^{m'}\|_2 : s^m, s^{m'} \in \mathcal{S}, s^m \neq s^{m'} \}$ is a positive constant. We assume $V^C \in \mathcal{R}$, in which case, we have $V^C(\cdot) = \beta_0^C + \int_{\Theta} \mathbf{B}^C(\theta) \varphi(\cdot; \theta) d\theta$ for some $\beta^C := (\beta_0^C, \mathbf{B}^C)$ (the results extend to the case when $V^C \notin \mathcal{R}$, as explained in §2.2 and in §2.10.1). Compared with Theorem 1 in the continuous state space case, the weighting function \mathbf{B}^* is replaced by \mathbf{B}^C , and the constant Ω is instead $\Omega^C := 5(D_s^C + 1)L\sqrt{\mathbb{E}_{\rho}[\|\theta\|_2^2]}$, where $D_s^C := \max_{s \in \mathcal{S}^C} \|s\|_2$. Here, we will continue to use the notation related to \mathbf{FALP}_N from §2.3.1 and define $\|V^* - V(\beta_N^{\text{FA}})\|_{1,\nu}$ to denote the $(1, \nu)$ -norm distance over the discrete state space, which is $\|V^* - V(\beta_N^{\text{FA}})\|_{1,\nu} = \sum_{m \in \mathcal{M}} \nu(s^m) |V^*(s^m) - V(s^m; \beta_N^{\text{FA}})|$.

Proposition 6 *Suppose Assumption 2 with V^* replaced by V^C and Assumption 3 hold, and in addition, $\rho(\theta) \geq \underline{\rho} > 0$ for all $\theta \in \Theta$. Given $\delta \in (0, 1]$, we have that any finite \mathbf{FALP}_N optimal solution β_N^{FA} satisfies*

$$\|V^* - V(\beta_N^{\text{FA}})\|_{1,\nu} \leq \frac{2\|\mathbf{B}^C/\rho\|_{2,\rho}}{(1-\gamma)\underline{\rho}\sqrt{N}} \left(\Omega^C + \sqrt{2 \ln \left(\frac{1}{\delta} \right)} \right),$$

with a probability of at least $1 - \delta$.

When the action space is finite for all states, we can drop Assumption 3 and establish the existence of a finite optimal solution, although as discussed in §2.3.1, this assumption is already mild. We highlight that the construction of FALP_N does not change based on the structure of the state space since the sampling distribution $\rho(\cdot)$ does not depend on this structure. Therefore, the same procedures for generating basis functions apply in the discrete state space case. Using the arguments here, we can also handle state spaces with a mixture of discrete and continuous elements.

Our results also extend to handle MDPs with a finite horizon $T < \infty$ by considering time to be in the state; that is, we can define the state as (t, s) . Because the options pricing application in §2.7 gives rise to a finite-horizon MDP, we formulate FALP_N next in the more familiar notation of such MDPs. Let the index set of stages in the horizon be $\mathcal{T} := \{0, 1, \dots, T\}$. The MDP value function at stage $t \in \mathcal{T} \setminus \{T\}$ is V_t^* , and we assume without a loss of generality that $V_T^* \equiv 0$. At stage $t \in \mathcal{T}$, the state space is \mathcal{S}_t , and the action space at this stage and state $s \in \mathcal{S}_t$ is $\mathcal{A}_t(s)$. Then, the finite horizon analogue of FALP_N computes VFAs that approximate V_t^* at each stage by sampling $\{\theta^1, \theta^2, \dots, \theta^N\}$:

$$V_t^* \approx V(\beta_t) = \beta_{t,0} + \sum_{i=1}^N \beta_{t,i} \varphi(\cdot; \theta^i),$$

where $\beta_t := (\beta_{t,0}, \beta_{t,1}, \dots, \beta_{t,N})$ are the stage t VFA weights. Because the sampling distribution $\rho(\cdot)$ does not depend on the stages or state space, the set of random basis functions can be the same across stages, which also provides the flexibility to use the same basis function weights

across stages if needed. Assuming that the state-relevance distribution ν is defined over the stage 0 state space \mathcal{S}_0 (it could easily be defined over the state spaces at all stages), FALP_N in the finite horizon setting is

$$\begin{aligned} \max_{\beta} \quad & \beta_{0,0} + \sum_{i=1}^N \beta_{0,i} \mathbb{E}_{\nu}[\varphi(s; \theta^i)] \\ \text{s.t.} \quad & (\beta_{t,0} - \gamma \beta_{t+1,0}) + \sum_{i=1}^N \left(\beta_{t,i} \varphi(s; \theta^i) - \gamma \beta_{t+1,i} \mathbb{E}_t[\varphi(s'; \theta^i) \mid s, \mathbf{a}] \right) \leq c_t(s, \mathbf{a}), \\ & \forall (t, s, \mathbf{a}) \in \mathcal{T} \setminus \{T\} \times \mathcal{S}_t \times \mathcal{A}_t(s), \end{aligned}$$

where $c_t(s, \mathbf{a})$ and \mathbb{E}_t are the stage t cost function and expectation under the state transition function from stage t to $t+1$, respectively. We omit the terminal condition for brevity. Theoretical guarantees that are analogous to the infinite horizon case for FALP and self-guided FALP can be derived in the finite horizon setting as well.

2.6 Perishable Inventory Control

We perform a numerical study on the perishable inventory control problem considered in [Lin et al. \(2020\)](#), henceforth abbreviated LNS). We discuss the infinite-horizon discounted-cost MDP formulation of the problem and instances in §2.6.1, the experimental setup in §2.6.2, and numerical findings in §2.6.3.

2.6.1 MDP Formulation and Instances

Managing the inventory of a perishable commodity is a fundamental and challenging problem in Operations Management ([Karaesmen et al. 2011](#), [Chen et al. 2014](#), [Sun et al. 2014](#), and LNS). We study a variant of this problem with partial backlogging and lead time from §7.3 in LNS.

Consider a perishable commodity with $l \geq 0$ and $J \geq 0$ periods of lifetime and ordering lead time, respectively. Ordering decisions are made over an infinite planning horizon. At each decision epoch, the state vector is $s = (s_0, s_1, \dots, s_{l-1}, u_1, u_2, \dots, u_{J-1})$ of size $l + J - 1$. The state element u_i for $i = 1, 2, \dots, J - 1$ is the previously ordered quantity that will be received i periods from now. If $s_0 \geq 0$, s_i for $i = 0, 1, \dots, l - 1$ is the amount of available commodity with i periods of life remaining. If $s_0 < 0$, the values of these state elements are notional quantities to compute the total on-hand inventory, which is $s_0 + \sum_{i=1}^{l-1} s_i$. Inventories s_i and u_i take values in the interval $[0, \bar{a}]$ for all $i = 1, \dots, l - 1$ and $j = 1, 2, \dots, J - 1$, respectively, where $\bar{a} \geq 0$ denotes the maximum ordering level. If $s_0 \in [-\sum_{i=1}^{l-1} s_i, \bar{a}]$, then the on-hand inventory is non-negative. Instead, if $s_0 < -\sum_{i=1}^{l-1} s_i$, then the on-hand inventory $s_0 + \sum_{i=1}^{l-1} s_i$ is negative and represents the amount of backlogged orders.

The demand for the commodity is governed by a random variable. In each period, we assume that the demand is realized before the arrival of order and is satisfied in a first-in-first-out manner. Given a demand realization D , taking an ordering decision (i.e., action) a from a state s results in the system transitioning to a new state

$$s' := \left(\max \left\{ s_1 - (D - s_0)_+, \underline{s} - \sum_{i=2}^{l-1} s_i \right\}, s_2, \dots, s_{l-1}, u_1, u_2, \dots, u_{J-1}, a \right),$$

where $(\cdot)_+ := \max\{\cdot, 0\}$ and $\underline{s} \leq 0$ is a maximum limit on the amount of backlogged orders, beyond which we treat unsatisfied orders as lost sales. The updating logic in the first element of s' ensures that the backlogging limit is enforced. This can be understood as follows: If there was no backlogging limit, then the on-hand inventory after demand realization and before order

arrival would be $s_1 - (D - s_0)_+ + \sum_{i=2}^{l-1} s_i$; instead, in the presence of the maximum backlog limit \underline{s} , this total on-hand inventory of $s_1 - (D - s_0)_+ + \sum_{i=2}^{l-1} s_i$ is greater than or equal to \underline{s} if and only if $s_1 - (D - s_0)_+ \geq \underline{s} - \sum_{i=2}^{l-1} s_i$. The remaining elements of s' are shifted elements of s , with the last element accounting for the latest order \mathbf{a} .

The immediate cost associated with a transition from a state-action pair (s, \mathbf{a}) is

$$c(s, \mathbf{a}) := \gamma^J c_o \mathbf{a} + \mathbb{E}_D \left[c_h \left[\sum_{i=1}^{l-1} s_i - (D - s_0)_+ \right]_+ + c_d (s_0 - D)_+ + c_b \left[D - \sum_{i=0}^{l-1} s_i \right]_+ + c_l \left[\underline{s} + D - \sum_{i=0}^{l-1} s_i \right]_+ \right],$$

where expectation \mathbb{E}_D is given with respect to the demand distribution. The per-unit ordering cost $c_o \geq 0$ is discounted by γ^J because we assume payments for orders are made only upon receipt. The holding cost $c_h \geq 0$ penalizes leftover inventory $(\sum_{i=1}^{l-1} s_i - (D - s_0)_+)_+$, while the per-unit disposal and backlogging costs $c_d \geq 0$ and $c_b \geq 0$ factor in, respectively, the costs associated with disposing $(s_0 - D)_+$ units and backlogging $(D - \sum_{i=0}^{l-1} s_i)_+$ units. Finally, each unit of lost sales $(\underline{s} + D - \sum_{i=0}^{l-1} s_i)_+$ is charged $c_l \geq 0$.

We consider 24 perishable inventory control instances – twelve from LNS with $l = J = 2$ (three-dimensional state space) and twelve new higher-dimensional instances. Six of the new instances have $l = 2$ and $J = 4$ (five-dimensional state space), and the remaining six instances have $l = 5$ and $J = 6$ (ten-dimensional state space). Similar to LNS, across all instances, we fix the demand distribution to a truncated normal distribution with a mean of 5 and support in the range $[0, 10]$. We require the maximum limit on the amount of backlogged orders to equal the maximum ordering level, that is, $\underline{s} = -\bar{\mathbf{a}}$. We vary the cost function parameters, the discount

factor γ , the maximum ordering level \bar{a} , and the demand standard deviation σ . Their specific values are shown in tables II–IV.

2.6.2 Computational Setup

We formulate FALP_N using the guidelines in §2.3.2. We considered both ReLU and Fourier bases and found the latter to perform better as described in §2.13.3. We thus focus on discussing results for Fourier bases. The bandwidth parameter c_ρ is tuned over the candidate set $\{10^5, 10^4, \dots, 10^{-5}\}$. For ν , we consider both the initial MDP state of $s_0 = (5, 5, \dots, 5) \in \mathbb{R}^d$ (i.e., a degenerate initial distribution χ) and a uniform distribution over the hyper-cube $\mathcal{S} = [\underline{s}, \bar{a}] \times [0, \bar{a}]^{d-1}$. The latter choice leads to substantially better policies, so we report the results only for this choice. We use constraint sampling to solve FALP_N and choose $K = 200,000$ state-action pairs sampled from a uniform distribution over the hyper-cube $\mathcal{S} \times \mathcal{A}_s = [\underline{s}, \bar{a}] \times [0, \bar{a}]^d$. The number of basis functions N was set to 150, 300, and 600 for the three-, five-, and ten-dimensional instances, respectively. We approximate expectations in FALP_N using sample average approximations constructed using 2,000 iid samples.

We formulate policy-guided FALP (Algorithm 1) and self-guided FALP (Algorithm 2) using the guidelines in §2.3.2 and §2.4.4, respectively. We denote these methods by $\text{FALP}_{N,Q}^{\text{PG}}$ and $\text{FALP}_{N,Q}^{\text{SG}}$, respectively, to make the number of iterations Q and the number of basis functions N explicit. In both these models, we choose N and the sampled parameters of the basis functions to be the same as FALP_N . We also set Q equal to 7 so that the number of state-relevance distribution updates in both models is $Q - 1 = 6$. At iteration q of $\text{FALP}_{N,Q}^{\text{PG}}$, we solve model $\text{FALP}_N[\nu^q]$ using constraint sampling strategies that create a set of state-action pairs: (i) uniformly as in

FALP_N , (ii) using the greedy policy $\pi_g(\beta^{q-1})$ at iteration $q - 1$, and (iii) by taking the union of the samples from (i) and (ii). We present the results for the best-performing strategy for each instance. For $\text{FALP}_{N,Q}^{\text{SG}}$, we enforce ALP constraints (2.9) at the state-action samples used in FALP_N and the guiding constraints (2.10) at the states in these state-action samples. We also consider the ALP model in LNS as a benchmark, denoted by ALP^{LNS} , with a fixed set of (7d–2) application-specific basis functions that include hinges (i.e., $(\cdot)_+$) to mirror the MDP cost function structure shown in §2.6. We sample the constraints of this model uniformly as in FALP_N .

We use the Gurobi commercial solver to solve linear programs. We simulate the cost of a greedy policy using 500 sample paths. Similar to LNS, we replace the action space $[0, \bar{a}]$ by \bar{a} equally spaced points and find the best action using enumeration. We estimate a lower bound on the optimal cost via a heuristic based on the constraint violation learning approach discussed in §2.12.1. In addition, for each instance and method, we repeat solving linear programs and the simulations of bounds ten times and report averages.

2.6.3 Results

Table II contains results for FALP_{150} and ALP^{LNS} on three-dimensional instances. Columns 1-5 report the parameters of problem instances. Columns 6-7 and 8-9 display the optimality gaps and lower bound gaps, respectively, computed with respect to the best lower bound among these two methods. Both the FALP_{150} and ALP^{LNS} policies are near-optimal with only small differences in their respective optimality gaps. The FALP_{150} lower-bound dominates the one from ALP^{LNS} , which has a lower bound gap between 2.2% and 10.5%. The FALP_{150} results show that random basis functions, which are not designed based on application structure, deliver near-optimal

Table II: Comparison of ALP^{LNS} and FALP on the three-dimensional perishable inventory control instances ($\sigma = 2$ and $c_l = 100$).

γ	c_h	c_d	c_b	\bar{a}	% (UB - best LB)/(best LB)		% (Best LB - LB)/(best LB)	
					ALP^{LNS}	FALP_{150}	ALP^{LNS}	FALP_{150}
0.95	2	5	10	10	0.2	0.1	3.4	0.0
	2	5	10	50	6.3	5.9	2.2	0.0
	5	10	8	10	0.3	0.2	4.0	0.0
	5	10	8	50	0.1	0.2	10.5	0.0
	2	10	10	10	0.3	0.2	3.5	0.0
	2	10	10	30	0.8	1.7	3.1	0.0
0.99	2	5	10	10	0.6	0.2	2.9	0.0
	2	5	10	50	6.2	5.6	2.7	0.0
	5	10	8	10	0.3	0.3	4.1	0.0
	5	10	8	50	1.1	1.5	10.3	0.0
	2	10	10	10	0.6	0.3	3.1	0.0
	2	10	10	30	1.1	1.5	2.9	0.0
Average					1.5	1.5	4.4	0.0

policies and lower bounds. In addition, methods that dynamically update the state-relevance distribution (i.e., policy-guided FALP and self-guided FALP) are not needed for these instances.

Table III displays results for ALP^{LNS} , FALP_{300} , $\text{FALP}_{300,7}^{\text{PG}}$, and $\text{FALP}_{300,7}^{\text{SG}}$ on five-dimensional instances. The policy performance of ALP^{LNS} varies greatly across instances, with a maximum optimality gap of 139.4%. FALP_{300} 's policy performance exhibits less variation to instance primitives but still has a sizeable maximum optimality gap of 21.0%. Unlike the three-dimensional instances, ALP^{LNS} and FALP_{300} policies based on a static state relevance distribution are quite suboptimal on the five-dimensional instances. Among the two methods that update the state relevance distribution, $\text{FALP}_{300,7}^{\text{PG}}$ is highly sensitive to the three constraint sampling strategies described in §2.6.2. In particular, the best sampling strategy changes by instance (see §2.13.2 for more details). The policy-guided FALP performance can deteriorate substantially if the best-

Table III: Comparison of ALP^{LNS} , FALP, policy-guided FALP, and self-guided FALP on the five-dimensional perishable inventory control instances ($\gamma = 0.95$ and $c_l = 1000$).

c_h	c_d	c_b	σ	% (UB - best LB)/(best LB)				% (Best LB - LB)/(best LB)			
				ALP^{LNS}	FALP_{300}	$\text{FALP}_{300,7}^{\text{PG}}$	$\text{FALP}_{300,7}^{\text{SG}}$	ALP^{LNS}	FALP_{300}	$\text{FALP}_{300,7}^{\text{PG}}$	$\text{FALP}_{300,7}^{\text{SG}}$
1	8	2	5	139.4	19.6	12.9	13.9	15.0	0.0	0.1	0.4
1	8	2	2	18.0	21.0	11.7	11.5	6.2	0.0	0.2	0.2
1	2	8	5	13.6	15.6	10.6	7.9	7.8	0.0	1.9	0.8
1	2	8	2	6.8	12.1	4.3	4.3	6.2	0.0	0.9	0.7
2	8	5	5	59.4	15.9	7.1	8.4	12.1	0.2	0.0	0.5
2	8	5	2	8.2	16.1	7.0	7.7	7.6	0.0	0.1	0.5
Average				40.9	16.7	8.9	9.0	9.1	0.0	0.5	0.5

performing strategy is not used, with optimality gaps and lower bound gaps reaching 82.3% and 17.1%, respectively. Moreover, constraint sampling strategy (ii) in §2.6.2 leads to unbounded linear programs, rendering Algorithm 1 unable to be fully executed when employing this strategy. The $\text{FALP}_{300,7}^{\text{SG}}$ instead exhibits stable performance and competitive optimality gaps to $\text{FALP}_{300,7}^{\text{PG}}$. These findings underscore the value of the self-guiding mechanism underpinning self-guided FALP in computing near-optimal policies. In contrast, all the methods, except ALP^{LNS} with pre-selected application-specific basis functions, deliver excellent lower bounds, which is consistent with the discussions in §2.4 that FALPs providing good lower bounds may not provide good policies due to a poor state relevance distribution choice.

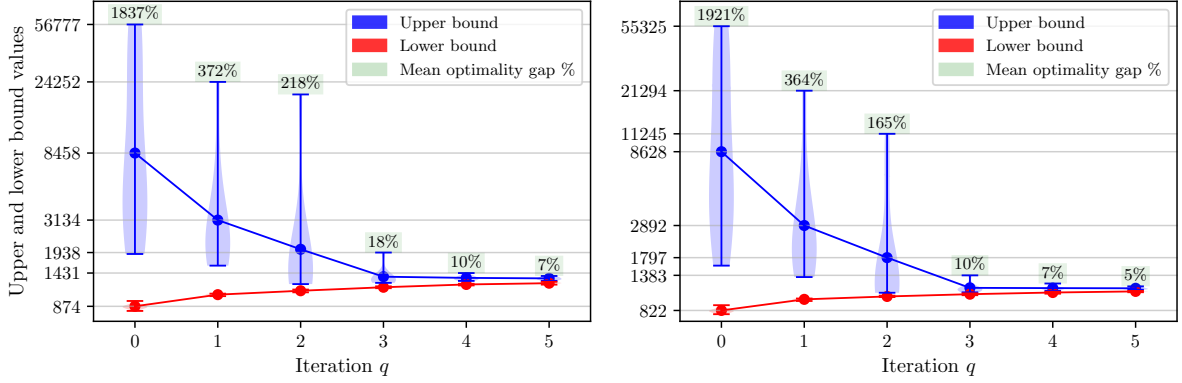
Table IV reports results for ALP^{LNS} , FALP_{600} , FALP_{1000} , and $\text{FALP}_{600,7}^{\text{SG}}$ on ten-dimensional instances. The methods with the worst policies are ALP^{LNS} , followed by FALP_{1000} , and then FALP_{600} . The poor performance of the ALP^{LNS} policies with fixed basis functions is consistent with results

Table IV: Comparison of ALP^{LNS} , FALP, and self-guided FALP on the ten-dimensional perishable inventory control instances ($\gamma = 0.95$ and $c_l = 1000$).

c_h	c_d	c_b	σ	% (UB - best LB)/(best LB)				% (Best LB - LB)/(best LB)			
				ALP^{LNS}	FALP_{600}	FALP_{1000}	$\text{FALP}_{600,7}^{\text{SG}}$	ALP^{LNS}	FALP_{600}	FALP_{1000}	$\text{FALP}_{600,7}^{\text{SG}}$
1	8	2	5	45.6	13.0	33.0	7.4	26.7	0.9	1.3	0.0
1	8	2	2	43.6	6.1	14.6	4.8	19.2	2.2	1.4	0.0
1	2	8	5	110.0	11.4	31.9	7.1	24.1	1.5	1.1	0.0
1	2	8	2	48.3	7.0	10.6	5.1	17.9	1.5	1.4	0.0
2	8	5	5	43.4	14.5	38.4	8.1	31.3	1.5	1.1	0.0
2	8	5	2	8.9	9.1	13.9	6.5	25.1	2.6	2.1	0.0
Average				50.0	10.2	23.7	6.5	24.1	1.7	1.4	0.0

on the five-dimensional instances. A new observation emerges in the ten-dimensional instances when comparing FALP_{600} and FALP_{1000} : ALPs with random basis functions, despite increasing the number of basis functions, can lead to poor policies. This behavior is reasonable because near-optimal policies cover smaller portions of the state space in higher dimensions and this makes it more important to align ALP VFA error minimization and policy performance via the update of the state relevance distribution. Self-guided FALP helps achieve this alignment, as witnessed by its low optimality gaps, while policy-guided FALP exhibits more erratic behavior than on the five-dimensional instances. Specifically, we encountered unbounded linear programs for constraint sampling strategy strategy (ii), which led to unboundedness on the five-dimensional instances, as well as other strategies. We were thus unable to obtain meaningful results for $\text{FALP}_{600,7}^{\text{PG}}$ on ten-dimensional instances. This finding is consistent with the behavior reported by [Farias and Van Roy \(2006\)](#) on a Tetris application: improved policy performance in

Figure 2: $\text{FALP}_{600,7}^{\text{SG}}$ upper and lower bounds on two representative ten-dimensional perishable inventory control instances with (c_h, c_d, c_b, σ) equal to $(1, 8, 2, 5)$ and $(1, 8, 2, 2)$ in the left and right panels, respectively.



the first few iterations followed by an unexplainable drop-off. With regards to the lower bound gaps reported in Table IV, $\text{FALP}_{600,7}^{\text{SG}}$ delivers the best lower bounds, suggesting that the self-guiding mechanism can help tighten lower bounds when used in conjunction with our heuristic based on constraint violation learning on the ten-dimensional instances.

To understand the self-guiding mechanism in $\text{FALP}_{600,7}^{\text{SG}}$, we display in Figure 2 the violin plot of upper and lower bound distributions computed from $\text{FALP}_{600,7}^{\text{SG}}$ over ten trials on two representative ten-dimensional instances, as iteration q increases in Algorithm 2. We also show the optimality gap percentages based on the average of the upper and lower bounds for each q . This figure shows how the upper and lower bounds converge as $\text{FALP}_{600,7}^{\text{SG}}$ iterates. The lower bounds are increasing as the algorithm iterates and they exhibit a relatively small variation across trials. The upper bounds exhibit a decrease in their mean but have high initial variability (q equals 1

to 3) with a subsequent rapid decrease in variation (q equals 4 onwards). The converged policies are near optimal.

As expected, the average run times increase as we employ more basis functions and move to higher dimensional instances. Here, run time refers to the total time in minutes required to solve linear programs, simulate policies, and estimate lower bounds. On the three-dimensional instances, the average run times (over instances and trials) of ALP^{LNS} and FALP_{150} were 1 and 7 minutes, respectively. For instances with a five-dimensional state space, ALP^{LNS} , FALP_{300} , $\text{FALP}_{300,7}^{\text{PG}}$, and $\text{FALP}_{300,7}^{\text{SG}}$ take on average 3, 15, 118, and 42 minutes, respectively. The average run times of ALP^{LNS} , FALP_{600} , FALP_{1000} , and $\text{FALP}_{600,7}^{\text{SG}}$ were 5, 33, 106, and 74 minutes, respectively, on the ten-dimensional instances. Thus, the computational times of self-guided FALP to obtain the policy improvements discussed earlier are encouraging.

2.7 Bermudan Options Pricing

We consider the pricing of a Bermudan call option that provides the holder the option (right but not the obligation) to sell the underlying once over a pre-specified set of future dates. The option-exercise payoff is based on the maximum of the prices of multiple assets, where each price evolves stochastically over time. Specifically, the payoff occurs if the holder exercises the options and the maximum price does not exceed a threshold known as the barrier price or knock-out price; otherwise, the option is worthless. This version is referred to as a knock-out Bermudan option and arises in practice because the knock-out feature limits the risk exposure of the seller and also makes the price of the option lower than its counterpart without this feature. Our numerical study of this problem is based on [Desai et al. \(2012b\)](#), henceforth abbreviated DFM).

In §2.7.1, we present the finite-horizon discounted MDP formulation. In §2.7.2, we describe our computational setup. In §2.7.3, we discuss results and findings.

2.7.1 MDP Formulation

We model the Bermudan call option in DFM which depends on the prices of J assets and formulate it as a finite-horizon MDP based on the notation in §2.5, except for using a reward function $r_t(s_t, a_t)$ instead of a cost function $c_t(s_t, a_t)$. The option has T exercise opportunities over Y years; that is, exercise is possible at times $\{\tau, 2\tau, \dots, T\tau\}$, where $\tau := Y/T$. The asset prices at stage $t \in \mathcal{T} = \{0, 1, \dots, T\}$ are $p_t := (p_{t,1}, p_{t,2}, \dots, p_{t,J})$, where $p_{t,j}$ is the price of the j -th asset at this time. Prices evolve according to a multi-asset geometric Brownian motion. The option is knocked out and becomes worthless any time the maximum of the J asset prices exceeds a pre-specified barrier price p^B . We use the binary variable $y_t \in \{0, 1\}$ to indicate if the option is knocked out at time t . It takes the value of one in this case and is zero otherwise. The transition equations governing y_t are $y_0 = \delta\{\max_j p_{0,j} \geq p^B\}$ and $y_t = \max\{y_{t-1}, \delta\{\max_j p_{t,j} \geq p^B\}\}$ for $t > 0$, where $\delta\{a\}$ equals one if a is true and zero otherwise. At time t , the MDP state is given by the vector $s_t = (p_{t,1}, p_{t,2}, \dots, p_{t,J}, y_t)$ that belongs to the state space $\mathcal{S} = [0, p^B]^J \times \{0, 1\}$. The MDP action a_t is binary, with values of one and zero corresponding to “stop” and “continue,” respectively. Stopping at stage t yields the reward $r_t(s_t, 0) = \gamma^t g(s_t)$, where the discount factor $\gamma = \exp(-r\tau)$, r is the risk-free interest rate, and the payoff function $g(\cdot) : \mathbb{R}^{J+1} \mapsto \mathbb{R}$ with respect to a pre-specified strike price p^S is $g(s_t) := \max\{\max_j\{p_{t,j} - p^S\}, 0\}(1 - y_t)$. A continue decision at state s_t has zero reward, that is, $r_t(s_t, 1) = 0$. The objective is to find an exercise policy that maximizes the discounted expected reward.

Our experiments use nine instances from DFM, for which Y , T , p^S , p^B , and r are 3, 54, 100, 170, and 5%, respectively. The geometric Brownian motion driving the prices has zero correlation and volatilities equal to 20%. All assets share the same initial price $p^I > 0$, that is, $p_{0,1} = p_{0,2} = \dots = p_{0,J} = p^I$. This price takes values from 90, 100, and 110, and the number of assets J takes on the values 4, 8, and 16. Although the asset prices can take values greater than the barrier price p^B , they need not be included in the state space because the option becomes worthless at all such prices. Thus, the range of each price relevant to the MDP belongs to the interval $[0, p^B]$.

2.7.2 Computational Setup and Benchmarks

We formulate the finite-horizon version of FALP_N given in §2.5 using $N = 500$ random Fourier basis functions, with its bandwidth parameter c_p tuned over the candidate set $\{10^5, 10^4, \dots, 10^{-5}\}$. (The focus on random Fourier basis functions is based on this choice outperforming random ReLU basis functions in experiments discussed in §2.13.3.) The strategy of using a policy to obtain a state-relevance distribution in §2.3.2 is simplified because the exercise decisions do not affect prices. Therefore, the price-portion of the state evolves according to the geometric Brownian motion model, regardless of the policy used. Motivated by this property, we use a lognormal state-relevance distribution of prices. We find that FALP_{500} performs much better with this choice than a uniform distribution. We do not consider policy-guided FALP given its unstable behavior. For self-guided FALP, we set $Q = 6$. We sample the constraints of both FALP_{500} and $\text{FALP}_{500,6}^{\text{SG}}$ by generating 3,000 trajectories of prices from the geometric Brownian motion model. We approximate the expected values by sampling 500 transitions from this model.

We consider two application-specific benchmarks. The first is least squares Monte Carlo (LSM), which is popular for financial and real option valuation ([Carriere 1996](#), [Longstaff and Schwartz 2001](#), [Glasserman and Yu 2004](#), and see [Nadarajah and Secomandi 2022](#) for a recent review) and provides very good policies on the instances we consider. This method approximates the optimal continuation function $C_t(s_t) := \mathbb{E}[V_{t+1}^*(p_{t+1})y_{t+1}|p_t]$ with the boundary condition $C_T(s_T) \equiv 0$ using a backward recursive scheme that uses a regression. To construct the continuation function approximation, we use the same application-specific $J + 2$ basis functions considered in DFM, which are $\phi_1(s_t) = 1 - y_t$, $\phi_2(s_t) = g(s_t)$, and $\phi_j(s_t) = (1 - y_t)p_{t,j}$ for $j = 1, 2, \dots, J$. We use 100,000 sample paths to estimate the weights of these basis functions at each time t . Our second benchmark is an ALP with the same $J + 2$ basis functions as LSM. We denote this model by ALP^{DFM} . We construct the constraints of this model using the same price trajectories and transitions used in the construction of FALP_{500} .

We simulate 20,000 price trajectories to evaluate the reward of each greedy policy, which provides a lower bound on the optimal policy value (because we are maximizing reward). The maximum standard error of these estimates is 0.4%. We embed the value/continuation function approximation from each method within the information relaxation and duality framework ([Brown et al. 2010](#)) to estimate an upper bound on the optimal reward (see §2.12.2 for details).

2.7.3 Results

Table V reports the performance of LSM, ALP^{DFM} , FALP_{500} , and $\text{FALP}_{500,6}^{\text{SG}}$ on nine Bermudan option pricing instances in DFM. This table follows the same structure as the tables in §2.6.3. The performance of the FALP_N policy is within 1% of the one from $\text{FALP}_{500,6}^{\text{SG}}$ on six of the nine

Table V: Comparison of optimality gaps on the Bermudan options pricing application.

J	p^{init}	% (Best UB - LB)/(best UB)				% (UB - best UB)/(best UB)			
		LSM	ALP^{DFM}	FALP_{500}	$\text{FALP}_{500,6}^{\text{SG}}$	LSM	ALP^{DFM}	FALP_{500}	$\text{FALP}_{500,6}^{\text{SG}}$
4	90	6.6	4.8	0.9	0.8	0.0	13.3	1.5	1.3
4	100	6.4	6.4	1.9	1.9	0.7	7.5	0.0	0.0
4	110	6.6	8.0	8.4	5.3	0.0	2.6	992.0	3.6
8	90	6.2	6.1	4.1	4.0	0.0	4.1	5.8	5.4
8	100	5.5	7.0	7.8	4.2	0.0	0.7	6.0	0.6
8	110	3.9	6.4	9.5	3.1	0.6	0.1	172.5	0.0
16	90	4.9	6.3	3.3	3.3	0.0	0.0	0.5	0.4
16	100	3.4	5.5	2.4	2.3	0.6	0.0	0.3	0.0
16	110	2.8	5.2	2.4	2.1	0.6	1054.4	0.2	4.0
Average		5.2	6.2	4.5	3.0	0.3	9.1	130.8	1.6

instances but 3.1%, 3.6%, and 6.4% worse on the remaining instances. Once again, we see significant value in updating the state-relevance distribution using the logic in $\text{FALP}_{500,6}^{\text{SG}}$. There is no clear ordering between the policies of ALP^{DFM} and LSM – the average optimality gap of the LSM method across all the instances is 1% smaller than ALP^{DFM} . The $\text{FALP}_{500,6}^{\text{SG}}$ policy is significantly better than the LSM policy, with improvements of less than 2% on six instances and greater than 2% on the remaining three. The largest such improvement is 5.8%.

The upper-bound gaps show that LSM and $\text{FALP}_{500,6}^{\text{SG}}$ lead to the tightest upper bounds on five and four instances, respectively. The upper bounds from ALP^{DFM} and FALP_{500} vary from being near-optimal to highly sub-optimal. ALP^{DFM} and FALP_{500} provide substantially weak upper bounds on one and two instances, respectively, where they also deliver their worst policies relative to other methods. This observation suggests that the ALP^{DFM} and FALP_{500} VFAs on these

instances are far from V^* not only at states visited by good policies but more broadly at other states as well.

The superior self-guided FALP policies come at a computational cost. The average runtime of LSM, ALP^{DFM} , FALP_{500} and $\text{FALP}_{500,6}^{\text{SG}}$ across trials and instances are, respectively, 2.42, 5.1, 99.5, and 117.9 minutes. There is thus an additional, albeit manageable, computational overhead to obtain the improved $\text{FALP}_{500,6}^{\text{SG}}$ policies.

A broader takeaway from these experiments is that an application-agnostic ALP model with random basis functions and a guided state-relevance distribution can provide near-optimal policies and bounds for a challenging option pricing problem, also improving on application-specific benchmarks.

2.8 Conclusions

We revisit the approximate linear programming approach for computing value function approximations (VFAs) of discounted-cost Markov decision processes (MDPs). We focus on the key elements needed to formulate an approximate linear program (ALP). The first is the selection of the basis functions defining the ALP VFA, which we address by cheaply sampled random basis functions. We call the resulting randomized one-shot approximation as feature-based ALP (FALP). The second element is the choice of a state-relevance distribution in the ALP objective. We propose a randomized multi-shot approximation scheme, which we dub self-guided FALP, to guide the state-relevance distribution in FALP using its past VFA information. We develop error bounds showing that self-guided FALP has desirable theoretical properties not shared by existing ALP-based models. We test FALP and self-guided FALP on challenging perishable

inventory control and options pricing applications. Self-guided FALP outperforms FALP and application-specific benchmarks. Our findings showcase the potential for our procedure to (i) reduce the implementation burden of using ALP and (ii) provide an application-agnostic policy and lower bound for MDPs that can be used to benchmark other methods.

Our research suggests several interesting directions for future work, of which we state two. The first is to study the possibility and value of a guided sampling mechanism for ALP where the new samples of random basis functions leverage information from past VFAs. Approaches for the data-dependent sampling of random basis functions in machine learning (see, e.g., [Sinha and Duchi 2016](#), [Shahrampour et al. 2018](#)) can query the function being approximated, which is the unknown MDP value function in our setting. It is unclear how to develop inexpensive and approximate queries of the MDP value function that still provide useful information, which would be needed to obtain an effective and efficient sampling approach. The second is to investigate the value of random basis functions and multi-shot approximations in other approximate dynamic programming methods, also comparing against neural networks and deep learning that attempt to mitigate tuning but lead to nonlinearly parametrized VFAs, which are typically harder to train.

APPENDICES

In §2.9, we provide the proofs of all statements in Chapter 2. In §2.10, we discuss how the assumptions used in §2.2.2 and §2.3.1 can be relaxed. In §2.11, we introduce a constraint sampling bound for self-guided FALP. In §2.12, we discuss two methods for computing optimistic bounds on the optimal policy cost. In §2.13, we provide additional numerical results that supplement the numerical experiments discussed in Chapter 2.

2.9 Proofs

We define a constant $\Gamma := (1 + \gamma)/(1 - \gamma)$ which we will use in various proofs. We also use the notation $\delta\{\mathfrak{a}\}$ to show the indicator function that is 1 when \mathfrak{a} is true and 0 otherwise.

2.9.1 Additional Details of Assumption 1

Assumptions 1 and 2 will hold for all proofs in the electronic companions. In particular, Assumption 1 ensures the existence of an optimal policy solving program (2.1). There are known conditions in the literature that guarantee such existence. We provide an example of these conditions in Assumption 5.

Assumption 5 *It holds that (i) the MDP cost function is bounded over $\mathcal{S} \times \mathcal{A}_s$ and function $c(s, \cdot) : \mathcal{A}_s \mapsto \mathbb{R}$ is lower semicontinuous for all $s \in \mathcal{S}$. (ii) For every bounded and measurable function $V : \mathcal{S} \mapsto \mathbb{R}$, the mapping $(s, \mathfrak{a}) \mapsto \int_{\mathcal{S}} V(s')P(ds'|s, \mathfrak{a})$ is bounded and continuous over $\mathcal{S} \times \mathcal{A}_s$. (iii) There exists a finite-cost policy π such that $PC(s, \pi) < \infty$ for all $s \in \mathcal{S}$.*

Assumption 5 is adopted from assumptions 4.2.1 and 4.2.2 in [Hernández-Lerma and Lasserre 1996](#), henceforth abbreviated as [HL](#). Specifically, in Part (a) of Assumption 4.2.1 in [HL](#), the cost function $c(s, \cdot)$ is assumed to be lower semi-continuous, non-negative, and inf-compact (defined

in Condition 3.3.3 in [HL](#)) whereas, in our setting, non-negativity is replaced by boundedness and the inf-compactness is guaranteed by virtue of $c(s, \cdot)$ being lower semi-continuous and its domain \mathcal{A}_s being either a continuous compact real-valued set or a finite set (please see Assumption 1). Part (b) of Assumption 4.2.1 and Assumption 4.2.2 in [HL](#) are equivalent to parts (ii) and (iii) of Assumption 5, respectively. Note that the condition specified in part (iii) of Assumption 5 is the definition of the strong continuity of the MDP stochastic kernel P (see Condition 3.3.3 in [HL](#)). Under the aforementioned technical conditions, Part (b) of Theorem 4.2.3 in [HL](#) guarantees the existence of a deterministic and stationary policy that is “ γ -discount optimal”. In other words, $\pi^* \in \Pi$ solves (2.1) in our setting.

2.9.2 Proofs of Statements in §2.2

Proof of Proposition 1.

Since $V^* \in \mathcal{R}$, there exists $(\beta_0^*, \mathbf{B}^*)$ such that $V^*(s) = \beta_0^* + \int_{\Theta} \mathbf{B}^*(\theta) \varphi(s; \theta) d\theta$ for all $s \in \mathcal{S}$. We show that $(\beta_0^*, \mathbf{B}^*)$ is our desirable solution. This solution is feasible to FEP since V^* satisfies constraints (2.2). It is also optimal because V^* satisfies the optimality equations $V^*(s) = \min_{a \in \mathcal{A}_s} \{c(s, a) + \gamma \mathbb{E}[V'(s') | s, a]\}$ for every $s \in \mathcal{S}$ which indicates that all the constraints of (2.2) hold as equality. ■

2.9.3 Proofs of Statements in §2.3

To prove Theorem 1, we require the following lemmas and propositions.

Lemma 1 *Any continuous function $V : \mathcal{S} \mapsto \mathbb{R}$ that is feasible to constraints (2.2) satisfies $V(s) \leq V^*(s)$ for all $s \in \mathcal{S}$.*

Proof. The proof follows from Part (b) of Lemma 4.2.7 in [HL](#), which requires four assumptions to hold. We now show that these assumptions are true in our setting. (i) Since V is continuous, it is measurable; (ii) the Bellman operator $TV(s) := \min_{a \in \mathcal{A}_s} \{c(s, a) + \gamma \mathbb{E}[V(s')|s, a]\}$ is well defined for every continuous function V , i.e. the minimum over \mathcal{A}_s is attained since \mathcal{A}_s is either a real-valued continuous compact set or a finite set from Assumption 1, $c(\cdot, \cdot)$ is bounded, and the expectation $\mathbb{E}[V(s')|s, a] = \int_{\mathcal{S}} V(s')P(ds'|s, a)$ is finite by Assumption 5; (iii) since V is feasible to constraints (2.2), we have

$$V(s) \leq \min_{a \in \mathcal{A}_s} \{c(s, a) + \gamma \mathbb{E}[V(s')|s, a]\} = TV(s), \quad \forall s \in \mathcal{S};$$

(iv) finally, the continuity of V and the compactness of \mathcal{S} imply $\max_{s \in \mathcal{S}} |V(s)| < \infty$ and thus

$$\lim_{n \rightarrow \infty} \gamma^n \mathbb{E} \left[\sum_{t=0}^n V(s_t^\pi) \middle| s_0 = s \right] \leq \max_{s \in \mathcal{S}} |V(s)| \lim_{n \rightarrow \infty} (n+1)\gamma^n = 0, \quad \forall (s, \pi) \in \mathcal{S} \times \Pi,$$

where expectation \mathbb{E} and the notation s_t^π retain their definitions from §2.2. These indicate that the function V fulfills the four assumptions of Part (b) of Lemma 4.2.7 in [HL](#) and hence $V(s) \leq V^*(s)$ for all $s \in \mathcal{S}$. ■

Proposition 7 *Suppose $\rho(\theta) \geq \underline{\rho}$, for all $\theta \in \Theta$ and Assumption 2 holds. Consider $\delta \in (0, 1]$ and a function $V(s; \beta) = \beta_0 + \int_{\Theta} \mathbf{B}(\theta) \varphi(s; \theta) d\theta$ with $\|\mathbf{B}/\rho\|_{2, \rho} < \infty$. Given N iid samples $\{\theta^i : i = 1, 2, \dots, N\}$ from ρ , there exist finite coefficients $\bar{\beta}_i, i = 0, 1, 2, \dots, N$, such that*

$$\left\| \mathbf{V}(\boldsymbol{\beta}) - \left(\bar{\beta}_0 + \sum_{i=1}^N \bar{\beta}_i \varphi(\cdot; \boldsymbol{\theta}^i) \right) \right\|_{\infty} \leq \frac{\|\mathbf{B}/\rho\|_{2,\rho}}{\underline{\rho}\sqrt{N}} \left(\Omega + \sqrt{2 \ln \left(\frac{1}{\delta} \right)} \right) \quad (2.13)$$

with a probability of at least $1 - \delta$.

Proof. The proof of this proposition follows similar steps to the proof of Theorem 3.2 in [Rahimi and Recht \(2008\)](#). In particular, given a constant $r > 0$ and N iid samples $\boldsymbol{\vartheta} := (\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \dots, \boldsymbol{\theta}^N)$, we first define random variable $\bar{\mathbf{V}}_{\boldsymbol{\vartheta}}(s) := \beta_0 + \frac{1}{N} \sum_{i=1}^N \mathbf{V}_{i,\boldsymbol{\vartheta}}(s)$ where $\mathbf{V}_{i,\boldsymbol{\vartheta}}(s) := \beta_i^r \varphi(s; \boldsymbol{\theta}^i)$ and $\beta_i^r := \frac{1}{\rho(\boldsymbol{\theta}^i)} \int_{\Theta} \mathbf{B}(\boldsymbol{\theta}) \delta \{ \boldsymbol{\theta} : \|\boldsymbol{\theta} - \boldsymbol{\theta}^i\|_2 \leq r \} d\boldsymbol{\theta}$. Let

$$g(\boldsymbol{\vartheta}) := \|\mathbf{V}(\boldsymbol{\beta}) - \bar{\mathbf{V}}_{\boldsymbol{\vartheta}}\|_{\infty}.$$

We provide an upper bound on $g(\boldsymbol{\vartheta})$ that is decreasing in N and holds with high probability.

To do so, we take the following steps:

Step (i): We first prove

$$\mathbb{E}[g(\boldsymbol{\vartheta})] \leq L(1 + D_s) \|\mathbf{B}/\rho\|_{2,\rho} \left[r + \frac{4}{\underline{\rho}} \sqrt{\frac{\mathbb{E}_{\rho}[\|\boldsymbol{\theta}\|_2^2]}{N}} \right]. \quad (2.14)$$

Step (ii): We then use McDiarmid's inequality to show the inequality

$$g(\boldsymbol{\vartheta}) \leq \mathbb{E}[g(\boldsymbol{\vartheta})] + \frac{\|\mathbf{B}/\rho\|_{2,\rho}}{\underline{\rho}} \sqrt{\frac{2}{N} \ln \left(\frac{1}{\delta} \right)}, \quad (2.15)$$

holds with a probability of at least $1 - \delta$.

The inequality (2.13) then follows from combining (2.14) and (2.15), using the definitions of $g(\cdot)$ and Ω , and setting $\bar{\beta}_0 = \beta_0$ and $\bar{\beta}_i = \frac{1}{N}\beta_i^r, i$ for $r := \sqrt{\mathbb{E}_\rho [\|\theta\|_2^2]} / (\underline{\rho}\sqrt{N})$.

Proof of Step (i): The inequality (2.14) can be easily derived from the following two inequalities:

$$\mathbb{E} [\|\mathbf{V}(\boldsymbol{\beta}) - \mathbb{E}_\rho [\bar{\mathbf{V}}_\vartheta]\|_\infty] \leq Lr(1 + D_s)\|\mathbf{B}/\rho\|_{2,\rho}. \quad (2.16)$$

and

$$\mathbb{E} [\|\bar{\mathbf{V}}_\vartheta - \mathbb{E}_\rho [\bar{\mathbf{V}}_\vartheta]\|_\infty] \leq \frac{4L}{\underline{\rho}\sqrt{N}}\|\mathbf{B}/\rho\|_{2,\rho}(1 + D_s)\sqrt{\mathbb{E}_\rho [\|\theta\|_2^2]} \quad (2.17)$$

In particular, using these two inequalities we get

$$\begin{aligned} \mathbb{E} [g(\vartheta)] &= \mathbb{E} [\|\mathbf{V}(\boldsymbol{\beta}) - \bar{\mathbf{V}}_\vartheta\|_\infty] \\ &= \mathbb{E} [\|\mathbf{V}(\boldsymbol{\beta}) - \mathbb{E}_\rho [\bar{\mathbf{V}}_\vartheta] + \mathbb{E}_\rho [\bar{\mathbf{V}}_\vartheta] - \bar{\mathbf{V}}_\vartheta\|_\infty] \\ &\leq \mathbb{E} [\|\mathbf{V}(\boldsymbol{\beta}) - \mathbb{E}_\rho [\bar{\mathbf{V}}_\vartheta]\|_\infty] + \mathbb{E} [\|\bar{\mathbf{V}}_\vartheta - \mathbb{E}_\rho [\bar{\mathbf{V}}_\vartheta]\|_\infty] \\ &\leq Lr(1 + D_s)\|\mathbf{B}/\rho\|_{2,\rho} + \frac{4L}{\underline{\rho}\sqrt{N}}\|\mathbf{B}/\rho\|_{2,\rho}(1 + D_s)\sqrt{\mathbb{E}_\rho [\|\theta\|_2^2]} \\ &= L(1 + D_s)\|\mathbf{B}/\rho\|_{2,\rho} \left[r + \frac{4}{\underline{\rho}} \sqrt{\frac{\mathbb{E}_\rho [\|\theta\|_2^2]}{N}} \right] \end{aligned} \quad (2.18)$$

We next prove (2.16) and (2.17).

To prove (2.16), we first note that $\mathbb{E}_\rho [\bar{\mathbf{V}}_\vartheta] = \beta_0 + \mathbb{E}_\rho [V_{1,\vartheta}]$ holds because $\theta^i, i = 1, \dots, N$, are iid samples. In addition, since $\mathbf{B} : \Theta \mapsto \mathbb{R}$ is $(2, \rho)$ -integrable function and thus measurable, it can be written by its positive and negative parts as follows: $\mathbf{B} = \mathbf{B}_+ - \mathbf{B}_-$ where $\mathbf{B}_+ := \max(0, \mathbf{B})$

and $\mathbf{B}_- := \max(0, -\mathbf{B})$. It is also known that both positive and negative parts of a measurable function are measurable. Hence, for every $s \in \mathcal{S}$ we can write

$$\begin{aligned}
\mathbb{E}_\rho [\bar{\mathbf{V}}_\vartheta(s)] &= \beta_0 + \mathbb{E}_\rho [\mathbf{V}_{1,\vartheta}(s)] \\
&= \beta_0 + \int_{\Theta} \rho(\theta^1) \left[\frac{\varphi(s; \theta^1)}{\rho(\theta^1)} \int_{\Theta} \mathbf{B}(\theta) \delta\{\theta : \|\theta - \theta^1\|_2 \leq r\} d\theta \right] d\theta^1 \\
&= \beta_0 + \int_{\Theta} (\mathbf{B}_+(\theta) - \mathbf{B}_-(\theta)) \left[\int_{\Theta} \varphi(s; \theta^1) \delta\{\theta : \|\theta - \theta^1\|_2 \leq r\} d\theta^1 \right] d\theta \\
&\leq \beta_0 + \int_{\Theta} \mathbf{B}_+(\theta) \left[\int_{\Theta} \left(\varphi(s; \theta) + L\|(1, s)\|_2 \|\theta^1 - \theta\|_2 \right) \delta\{\theta : \|\theta - \theta^1\|_2 \leq r\} d\theta^1 \right] d\theta \\
&\quad - \int_{\Theta} \mathbf{B}_-(\theta) \left[\int_{\Theta} \left(\varphi(s; \theta) - L\|(1, s)\|_2 \|\theta^1 - \theta\|_2 \right) \delta\{\theta : \|\theta - \theta^1\|_2 \leq r\} d\theta^1 \right] d\theta \\
&\leq \beta_0 + \int_{\Theta} (\mathbf{B}_+(\theta) - \mathbf{B}_-(\theta)) \varphi(s; \theta) d\theta + L\|(1, s)\|_2 r \int_{\Theta} [\mathbf{B}_+(\theta) + \mathbf{B}_-(\theta)] d\theta \\
&\leq V(s; \beta) + Lr\|(1, s)\|_2 \int_{\Theta} \sqrt{\left(\frac{\beta(\theta)}{\rho(\theta)} \right)^2} \rho(d\theta) \\
&\leq V(s; \beta) + Lr(1 + D_s) \|\mathbf{B}/\rho\|_{2,\rho}, \tag{2.19}
\end{aligned}$$

where the second equality follows from the definition of $\mathbf{V}_{1,\vartheta}(s)$ and $\mathbb{E}_\rho[\mathbf{V}_{1,\vartheta}(s)]$; the third equality from the Fubini's theorem on the exchange of integrals and using $\mathbf{B} = \mathbf{B}_+ - \mathbf{B}_-$; the first inequality from the Lipschitz continuity of φ (by Assumption 2), Cauchy-Schwartz inequality, and the fact that both functions \mathbf{B}_+ and \mathbf{B}_- are non-negative; the second inequality from the fact that the indicator function is less than one and θ is considered in a ball of radius r ; the third inequality from the definition of $V(\beta)$ and the Jensen's inequality $\mathbb{E}[\sqrt{\cdot}] \leq \sqrt{\mathbb{E}[\cdot]}$; and the last inequality from the definitions of D_s and $\|\mathbf{B}/\rho\|_{2,\rho}$. Recalling that (2.19) holds for every $s \in \mathcal{S}$, taking expectation from both sides and rearranging the terms, we obtain (2.16).

To prove (2.17), we consider a sequence of Rademacher random variables $(\epsilon_1, \dots, \epsilon_N)$, where each ϵ_i is a uniform sample from $\{-1, 1\}$. It is easy to see the function $\beta_i^r \varphi(\cdot)$ is $(L/\underline{\rho})\|\mathbf{B}/\rho\|_{2,\rho}$ -Lipschitz and $\beta_i^r \varphi(0) = 0$. This follows from the fact that the function φ is L -Lipschitz continuous (by Assumption 2) and

$$\begin{aligned} \sup_{\theta^i} |\beta_i^r(\theta^i)| &= \sup_{\theta^i} \left\{ \frac{1}{\rho(\theta^i)} \int_{\Theta} |\mathbf{B}(\theta)| \delta \left\{ \theta : \|\theta - \theta^i\|_2 \leq r \right\} d\theta \right\} \\ &\leq \frac{1}{\underline{\rho}} \int_{\Theta} \sqrt{\left(\frac{\mathbf{B}(\theta)}{\rho(\theta)} \right)^2} \rho(d\theta) \\ &= \frac{1}{\underline{\rho}} \|\mathbf{B}/\rho\|_{2,\rho}, \end{aligned} \tag{2.20}$$

where the first equality holds by the definition of β_i^r ; the first inequality by our assumption that $\rho(\cdot)$ is bounded below by $\underline{\rho}$, and the fact that the indicator function is less than one.

Using Theorem 12(4) of [Bartlett and Mendelson \(2002\)](#), Cauchy-Schwartz inequality, and Jensen's inequality, we get

$$\begin{aligned} \mathbb{E}_{\rho} [\|\bar{V}_{\vartheta} - \mathbb{E}_{\rho} [\bar{V}_{\vartheta}]\|_{\infty}] &= \mathbb{E}_{\rho} \left[\sup_s |\bar{V}_{\vartheta} - \mathbb{E}_{\rho} [\bar{V}_{\vartheta}]] \right] \\ &\leq \frac{2}{N} \mathbb{E}_{\rho, \epsilon} \left[\sup_s \left| \sum_{i=1}^N \epsilon_i \beta_i^r \varphi(s; \theta^i) \right| \right] \\ &\leq \frac{4L}{\underline{\rho}N} \|\mathbf{B}/\rho\|_{2,\rho} \mathbb{E}_{\rho, \epsilon} \left[\sup_s \left| \sum_{i=1}^N \epsilon_i (1, s)^{\top} \theta^i \right| \right] \\ &\leq \frac{4L}{\underline{\rho}N} \|\mathbf{B}/\rho\|_{2,\rho} (1 + D_s) \mathbb{E}_{\rho, \epsilon} \left\| \sum_{i=1}^N \epsilon_i \theta^i \right\|_2 \\ &\leq \frac{4L}{\underline{\rho}\sqrt{N}} \|\mathbf{B}/\rho\|_{2,\rho} (1 + D_s) \sqrt{\mathbb{E}_{\rho} [\|\theta\|_2^2]}. \end{aligned}$$

Note that the above inequalities follow similar steps as in inequalities (21) - (24) in [Rahimi and Recht \(2008\)](#).

Proof of Step (ii): Observe that g is stable under any perturbation of its arguments. In particular, for an arbitrary $\ell \in \{1, 2, \dots, N\}$, let $\hat{\vartheta} := (\theta^1, \theta^2, \dots, \hat{\theta}^\ell, \dots, \theta^N)$ be the same as ϑ , except its ℓ -th component. That is, $\hat{\theta}^i = \theta^i$, for all $i \neq \ell$ and $\hat{\theta}^\ell \neq \theta^\ell$. We then have

$$\begin{aligned}
|g(\vartheta) - g(\hat{\vartheta})| &= \left| \left\| V(\boldsymbol{\beta}) - \beta_0 - \frac{1}{N} \sum_{i \neq \ell} V_{i, \vartheta}(s) - \frac{1}{N} V_{\ell, \vartheta}(s) \right\|_\infty - \right. \\
&\quad \left. \left\| V(\boldsymbol{\beta}) - \beta_0 - \frac{1}{N} \sum_{i \neq \ell} V_{i, \hat{\vartheta}}(s) - \frac{1}{N} V_{\ell, \hat{\vartheta}}(s) \right\|_\infty \right| \\
&\leq \frac{1}{N} \|V_{\ell, \vartheta}(s) - V_{\ell, \hat{\vartheta}}(s)\|_\infty \\
&= \frac{1}{N} \|\beta_\ell^r(\theta^\ell) \varphi(s; \theta^\ell) - \beta_\ell^r(\hat{\theta}^\ell) \varphi(s; \hat{\theta}^\ell)\|_\infty \\
&\leq \frac{2}{N} \sup_{\theta^\ell} |\beta_\ell^r(\theta^\ell)| \\
&\leq \frac{2}{N \underline{\rho}} \|\mathbf{B}/\rho\|_{2, \rho}, \tag{2.21}
\end{aligned}$$

where the first equality follows from the definition of $g(\cdot)$; the first inequality from the triangle inequality; the second equality from the definition of $V_{\ell, \vartheta}(s)$; the second inequality from $\|\varphi\|_\infty \leq 1$ (by Assumption 2); and the last inequality from (2.20).

Given $\varepsilon > 0$ and (2.21), McDiarmid's concentration inequality guarantees that

$$\Pr[g(\vartheta) - \mathbb{E}[g(\vartheta)] \geq \varepsilon] \leq \exp\left(\frac{-N \underline{\rho}^2 \varepsilon^2}{2 \|\mathbf{B}/\rho\|_{2, \rho}^2}\right),$$

where $\Pr(\cdot)$ denotes the probability over the samples $\vartheta = (\theta^1, \dots, \theta^N)$. This inequality indicates that

$$g(\vartheta) \leq \mathbb{E}[g(\vartheta)] + \frac{1}{\underline{\rho}} \|\mathbf{B}/\rho\|_{2,\rho} \sqrt{\frac{2}{N} \ln \left(\frac{1}{\delta} \right)},$$

with a probability of at least $1 - \delta$. ■

Definition 2 Let $r := \sqrt{2 \ln(1/\delta)}/(L(1 + D_s)\sqrt{N})$. Given an optimal solution $\beta^* = (\beta_0^*, \mathbf{B}^*)$ to FEP, for N iid samples $\{\theta^i, i = 1, 2, \dots, N\}$ from ρ , we define $\beta^\theta \in \mathbb{R}^{N+1}$ as follows:

$$\beta_i^\theta := \begin{cases} \beta_0^* & \text{for } i = 0; \\ \frac{1}{N\rho(\theta^i)} \int_{\Theta} \mathbf{B}^*(\theta) \delta\{\theta : \|\theta - \theta^i\|_2 \leq r\} d\theta & \text{for } i = 1, 2, \dots, N, \end{cases}$$

and $V(\beta^\theta) = \beta_0^\theta + \sum_{i=1}^N \beta_i^\theta \varphi(\cdot; \theta^i)$.

Lemma 2 Suppose $\rho(\theta) \geq \underline{\rho}$, for all $\theta \in \Theta$ and Assumption 2 holds. Given $\varepsilon > 0$ and $\delta \in (0, 1]$, let $(\beta_0^*, \mathbf{B}^*)$ denote an optimal solution to FEP with value function V^* and β^θ be the corresponding vector defined in Definition 2. Define

$$N_\varepsilon := \left\lceil \frac{\|\mathbf{B}^*/\rho\|_{2,\rho}^2}{\underline{\rho}^2 \varepsilon^2} \left(\Omega + \sqrt{2 \ln \left(\frac{1}{\delta} \right)} \right)^2 \right\rceil. \quad (2.22)$$

(i) If $N \geq N_\varepsilon$, with a probability of at least $1 - \delta$, it holds that $\|V^* - V(\beta^\theta)\|_\infty \leq \varepsilon$.

(ii) If $N \geq N_\varepsilon$, with a probability of at least $1 - \delta$, the vector $(\beta_0^\theta - \Gamma\varepsilon, \beta_1^\theta, \dots, \beta_N^\theta)$ is feasible to FALP_N and

$$\|\mathbf{V}^* - (\mathbf{V}(\beta^\theta) - \Gamma\varepsilon)\|_\infty \leq \frac{2\varepsilon}{(1-\gamma)}.$$

Proof. Part (i). First notice that the vector β^θ defined in the Definition 2 is the same vector of coefficients $(\bar{\beta}_0, \bar{\beta}_1, \dots, \bar{\beta}_N)$ defined in Proposition 7 corresponding to $\mathbf{V}(\beta^*) = \beta_0^* + \int_{\Theta} \mathbf{B}^*(\theta)\varphi(s; \theta) d\theta$. Following similar steps as in the proof of this proposition, we guarantee that with a probability of at least $1 - \delta$

$$\|\mathbf{V}^* - \mathbf{V}(\beta^\theta)\|_\infty \leq \frac{\|\mathbf{B}^*/\rho\|_{2,\rho}}{\underline{\rho}\sqrt{N}} \left(\Omega + \sqrt{2 \ln \left(\frac{1}{\delta} \right)} \right).$$

For $N \geq N_\varepsilon$, this inequality indicates that $\|\mathbf{V}^* - \mathbf{V}(\beta^\theta)\|_\infty \leq \varepsilon$ holds with a probability of at least $1 - \delta$.

Part (ii). If $N \geq N_\varepsilon$, the vector $(\beta_0^\theta - \Gamma\varepsilon, \beta_1^\theta, \dots, \beta_N^\theta)$ is feasible to FALP_N with a probability of at least $1 - \delta$ since

$$\begin{aligned} & (1 - \gamma)(\beta_0^\theta - \Gamma\varepsilon) + \sum_{i=1}^N \beta_i^\theta (\varphi(s; \theta_i) - \gamma \mathbb{E}[\varphi(s'; \theta_i) | s, \mathbf{a}]) \\ &= \mathbf{V}(s; \beta^\theta) - \varepsilon - \gamma \mathbb{E}[\mathbf{V}(s'; \beta^\theta) + \varepsilon | s, \mathbf{a}] \\ &\leq \mathbf{V}^*(s) - \gamma \mathbb{E}[\mathbf{V}^*(s') | s, \mathbf{a}] \\ &= (1 - \gamma)\beta_0^* + \int_{\Theta} \mathbf{B}^*(\theta) (\varphi(s) - \gamma \mathbb{E}[\varphi(s') | s, \mathbf{a}]) d\theta \\ &\leq \mathbf{c}(s, \mathbf{a}), \end{aligned} \tag{2.23}$$

where the first equality comes from the definitions of $V(\beta^\theta)$ and Γ ; the first inequality holds because $|V^*(s) - V(s; \beta^\theta)| \leq \|V^* - V(\beta^\theta)\|_\infty \leq \varepsilon$ for all $s \in \mathcal{S}$ with a probability of at least $1 - \delta$ by Part (i) of this lemma; the second equality results from using the definition of V^* ; and the second inequality holds because $(\beta_0^*, \mathbf{B}^*)$ is an optimal (hence feasible) solution of FEP.

Moreover, if $N \geq N_\varepsilon$, by Part (i) of this lemma and the definition of Γ , we get

$$\|V^* - (V(\beta^\theta) - \Gamma\varepsilon)\|_\infty \leq \|V^* - V(\beta^\theta)\|_\infty + \Gamma\varepsilon \leq \varepsilon + \Gamma\varepsilon = \frac{2\varepsilon}{(1-\gamma)}$$

with a probability of at least $1 - \delta$. ■

Proof of Theorem 1.

Part (i). The function $V(\cdot; \beta_N^{\text{FA}})$ is continuous due to the continuity of the class of basis functions φ (by Assumption 2), and is feasible to constraints (2.2) due to the feasibility of β_N^{FA} to FALP_N . Hence, Lemma 1 guarantees $V(s; \beta_N^{\text{FA}}) \leq V^*(s)$ for all $s \in \mathcal{S}$.

Part (ii). Consider $\varepsilon > 0$. Given $\beta^\theta = (\beta_0^\theta, \beta_1^\theta, \dots, \beta_N^\theta)$ and N_ε defined in Definition 2 and Lemma 2, respectively, part (ii) of Lemma 2 ensures that when $N \geq N_\varepsilon$, the vector $(\beta_0^\theta - \Gamma\varepsilon, \beta_1^\theta, \dots, \beta_N^\theta)$ is a feasible solution to FALP_N with a probability of at least $1 - \delta$ and hence

$$\|V^* - V(\beta_N^{\text{FA}})\|_{1,v} \leq \|V^* - (V(\beta^\theta) - \Gamma\varepsilon)\|_{1,v} \leq \|V^* - (V(\beta^\theta) - \Gamma\varepsilon)\|_\infty \leq \frac{2\varepsilon}{1-\gamma},$$

where we used the optimality of β_N^{FA} to obtain the first inequality, the relationship between $(1, \nu)$ - and ∞ -norms to obtain the second inequality, and part (ii) of Lemma 2 for the last one. Since $N \geq N_\varepsilon$, the proof is complete if we choose

$$\varepsilon \leq \frac{\|\mathbf{B}^*/\rho\|_{2,\rho}}{\underline{\rho}\sqrt{N}} \left(\Omega + \sqrt{2 \ln \left(\frac{1}{\delta} \right)} \right).$$

■

Proof of Proposition 2.

Recall the definition of vector $\beta_N^{\text{reg}} := \arg \min_{\beta \in \mathbb{R}^{N+1}} \|\mathbf{V}(\beta) - \mathbf{V}^*\|_{1,\nu}$. While this vector may not be feasible to FALP_N constraints, it is easy to verify that if we deduct term $(1 + \gamma)\varepsilon/(1 - \gamma)$ from the first element of $\beta_N^{\text{reg}} \in \mathbb{R}^{N+1}$, the resulting vector, which we denote by β_N^{feas} , is feasible to the constraints in (2.5) and thus feasible to FALP_N . Hence, we have

$$\begin{aligned} \|\mathbf{V}^* - \mathbf{V}(\beta_N^{\text{FA}})\|_{1,\nu} &\leq \|\mathbf{V}^* - \mathbf{V}(\beta_N^{\text{feas}})\|_{1,\nu} \\ &\leq \|\mathbf{V}^* - \mathbf{V}(\beta_N^{\text{reg}})\|_{1,\nu} + \|\mathbf{V}(\beta_N^{\text{reg}}) - \mathbf{V}(\beta_N^{\text{feas}})\|_{1,\nu} \\ &\leq \varepsilon + \frac{1 + \gamma}{1 - \gamma} \varepsilon \\ &= \frac{2}{1 - \gamma} \varepsilon. \end{aligned}$$

The first inequality is derived from the feasibility of β_N^{feas} to (2.5), and the second one from the triangle inequality. The last inequality is a result of assumption $\|\mathbf{V}^* - \mathbf{V}(\beta_N^{\text{reg}})\|_{1,\nu} \leq \varepsilon$ and equality $\|\mathbf{V}(\beta_N^{\text{reg}}) - \mathbf{V}(\beta_N^{\text{feas}})\|_{1,\nu} = (1 + \gamma)\varepsilon/(1 - \gamma)$, which is based on the definition of β_N^{feas} . ■

Proof of Proposition 3.

The proof follows from the Corollary 1 and Theorem 1 in [Calafiore and Campi 2006](#), abbreviated by [CC](#), applied to the program (2.7), which is a random relaxation of FALP_N . Under assumptions 1 and 2 in [CC](#), Corollary 1 and Theorem 1 guarantee that with a probability of at least $1 - \delta$, the optimal solution $\hat{\beta}$ of problem (2.7) satisfies:

$$\psi \left(\{(s, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}_s : h^{\text{FA}}(\hat{\beta}; s, \mathbf{a}) \leq 0\} \right) \geq 1 - \delta,$$

where given $\beta = (\beta_0, \beta_1, \dots, \beta_N) \in \mathbb{R}^{N+1}$, the function $h^{\text{FA}} : \mathbb{R}^{N+1} \times \mathcal{S} \times \mathcal{A}_s \mapsto \mathbb{R}$ is defined as follows:

$$h^{\text{FA}}(\beta; s, \mathbf{a}) := (1 - \gamma)\beta_0 + \sum_{i=1}^N \beta_i \left(\varphi(s; \theta^i) - \gamma \mathbb{E}[\varphi(s'; \theta^i) \mid s, \mathbf{a}] \right) - c(s, \mathbf{a}).$$

We only need to show that Assumptions 1 and 2 of [CC](#) hold in our setting. First notice that we use the notations h^{FA} , β , \mathbb{R}^{N+1} , $N+1$, (s, \mathbf{a}) , and $\mathcal{S} \times \mathcal{A}_s$ in Chapter 2 instead of f , θ , Θ , n_θ , δ , and Δ , respectively, in [CC](#). Assumption 1 in [CC](#) requires the function $h^{\text{FA}}(\beta; \cdot, \cdot)$ to be convex in β and continuous. This clearly holds in our setting since $h^{\text{FA}}(\beta; \cdot, \cdot)$ is linear in β , and we assume $\varphi(\cdot)$ is a Lipschitz continuous function. We use a relaxation of Assumption 2 in [CC](#) as stated in Appendix A. In particular, we only show that the program (2.7) is feasible and forgo the uniqueness assumption of the optimal solution to FALP_N . Define $\underline{c} := \min_{s, \mathbf{a}} c(s, \mathbf{a}) / (1 - \gamma)$ which is well-defined since $c(\cdot, \cdot)$ is bounded by Assumption 5. It is straightforward to verify

that $(\underline{c}, 0, \dots, 0) \in \mathbb{R}^{N+1}$ is feasible to FALP_N and hence feasible to program (2.7) for all samples $\{(s^k, \mathbf{a}^k) \in \mathcal{S} \times \mathcal{A}_s : k = 1, 2, \dots, K\}$. ■

2.9.4 Proofs of Statements in §2.4

Proof of Proposition 5.

For every iteration $q \geq 0$, self-guided FALP VFA $V(\cdot; \beta_N^{\text{SG}})$ with $N = qB$ basis functions is a continuous function because of the Lipschitz continuity of φ in Assumption 2. Moreover, this function is feasible to constraints (2.2) because vector β_N^{SG} is feasible to the constraints (2.9) of $\text{FALP}_N^{\text{SG}}$. As a result, Lemma 1 guarantees $V(s; \beta_N^{\text{SG}}) \leq V^*(s)$ for all $q \geq 0$ and $s \in \mathcal{S}$. In addition, guiding constraints (2.10) in $\text{FALP}_{N+B}^{\text{SG}}$ imply $V(\cdot; \beta_N^{\text{SG}}) \leq V(\cdot; \beta_{N+B}^{\text{SG}})$ for every $q \geq 0$, where identity $N = qB$ holds. ■

The proof of Theorem 2 relies on the following definition, propositions 8–10, and Theorem 3.

Definition 3 Given N iid samples $\{\theta^i : i = 1, 2, \dots, N\}$ from ρ , we define

$$\mathcal{B}_N := \left\{ \mathbf{B} : \Theta \mapsto \mathbb{R} \mid \exists (\beta_1, \dots, \beta_N) \in \mathbb{R}^N, \sum_{i=1}^N \beta_i^2 < \infty, \mathbf{B}(\theta) = \sum_{i=1}^N \beta_i \delta\{\theta = \theta^i\}, \right\}.$$

Moreover, let $\bar{\mathcal{B}}_N$ and $\bar{\mathcal{B}}_N^\perp$ denote the closure of \mathcal{B}_N and the perpendicular complement of $\bar{\mathcal{B}}_N$, respectively. In addition, suppose $\mathcal{B} := \{\mathbf{B} : \Theta \mapsto \mathbb{R} : \|\mathbf{B}/\rho\|_{2,\rho} < \infty\}$ denotes the space of all $(2, \rho)$ -integrable functions equipped with the following inner product

$$\langle \mathbf{B}, \mathbf{B}' \rangle_{\mathcal{B}} := \int_{\Theta} \frac{\mathbf{B}(\theta) \mathbf{B}'(\theta)}{\rho(\theta)} d\theta, \quad \text{for } \mathbf{B}, \mathbf{B}' \in \mathcal{B}.$$

Proposition 8 *It follows that*

- (i) *The space \mathcal{B} defined in Definition 3 equipped with inner product $\langle \cdot, \cdot \rangle_{\mathcal{B}}$ is a Hilbert space.*
- (ii) *The set $\bar{\mathcal{B}}_{\mathbf{N}}$ is a closed subset of \mathcal{B} under addition and scalar multiplication.*
- (iii) *Let $(\beta_0^*, \mathbf{B}^*)$ be the optimal solution associated with \mathbf{V}^* . There exist $\mathbf{B}_{\mathbf{N}}^{*,o} \in \bar{\mathcal{B}}_{\mathbf{N}}$ and $\mathbf{B}_{\mathbf{N}}^{*,\perp} \in \bar{\mathcal{B}}_{\mathbf{N}}^{\perp}$ such that $\mathbf{B}^* = \mathbf{B}_{\mathbf{N}}^{*,o} + \mathbf{B}_{\mathbf{N}}^{*,\perp}$ and $\|\mathbf{B}^*/\rho\|_{2,\rho} = \|\mathbf{B}_{\mathbf{N}}^{*,o}/\rho\|_{2,\rho} + \|\mathbf{B}_{\mathbf{N}}^{*,\perp}/\rho\|_{2,\rho}$.*

Proof. Part (i): The space \mathcal{B} is a Hilbert space by Example 4.5 in [Rudin \(1987\)](#).

Part (ii): The set $\bar{\mathcal{B}}_{\mathbf{N}}$ is a closed subset of \mathcal{B} since for every $\mathbf{B} \in \bar{\mathcal{B}}_{\mathbf{N}}$ with $\mathbf{B}(\theta) = \sum_{i=1}^N \beta_i \delta\{\theta = \theta^i\}$, we have $\|\mathbf{B}/\rho\|_{2,\rho} \leq \sum_i \beta_i^2/\rho < \infty$. In addition, $\bar{\mathcal{B}}_{\mathbf{N}}$ is closed under addition since for every $\mathbf{B}, \mathbf{B}' \in \bar{\mathcal{B}}_{\mathbf{N}}$, we have $\mathbf{B} + \mathbf{B}' \in \bar{\mathcal{B}}_{\mathbf{N}}$. It is also closed under scalar multiplication because for every $\mathbf{B} \in \bar{\mathcal{B}}_{\mathbf{N}}$ and $\alpha \in \mathbb{R}$, we have $\alpha\mathbf{B} \in \bar{\mathcal{B}}_{\mathbf{N}}$.

Part (iii): Since $\mathbf{B}^* \in \mathcal{B}$, using parts (i) and (ii) and the orthogonal projection theorem of Hilbert spaces (Theorem 5.23 in [Folland 1999](#)), there exist functions $\mathbf{B}_{\mathbf{N}}^{*,o} \in \bar{\mathcal{B}}_{\mathbf{N}}$ and $\mathbf{B}_{\mathbf{N}}^{*,\perp} \in \bar{\mathcal{B}}_{\mathbf{N}}^{\perp}$ such that $\mathbf{B}^* = \mathbf{B}_{\mathbf{N}}^{*,o} + \mathbf{B}_{\mathbf{N}}^{*,\perp}$ and $\|\mathbf{B}^*/\rho\|_{2,\rho} = \|\mathbf{B}_{\mathbf{N}}^{*,o}/\rho\|_{2,\rho} + \|\mathbf{B}_{\mathbf{N}}^{*,\perp}/\rho\|_{2,\rho}$. ■

Proposition 9 *Consider $\zeta > 0$ and N iid samples $\{\theta^i : i = 1, 2, \dots, N\}$ from ρ . Let $(\beta_0^*, \mathbf{B}^*)$ denote an optimal solution to FEP with $\mathbf{B}^* = \mathbf{B}_{\mathbf{N}}^{*,o} + \mathbf{B}_{\mathbf{N}}^{*,\perp}$ for some $\mathbf{B}_{\mathbf{N}}^{*,o} \in \bar{\mathcal{B}}_{\mathbf{N}}$ and $\mathbf{B}_{\mathbf{N}}^{*,\perp} \in \bar{\mathcal{B}}_{\mathbf{N}}^{\perp}$ (see Proposition 8). Define $\beta_{\mathbf{N}}^{*,\perp} := (0, \mathbf{B}_{\mathbf{N}}^{*,\perp})$. There exists a coefficient function $\mathbf{B}_{\mathbf{N}}^{\zeta} \in \mathcal{B}_{\mathbf{N}}$ such that for $\beta_{\mathbf{N}}^{\zeta} := (\beta_0^*, \mathbf{B}_{\mathbf{N}}^{\zeta})$, we get*

$$\left\| \mathbf{V}^* - \left(\mathbf{V}(\beta_{\mathbf{N}}^{\zeta}) + \mathbf{V}(\beta_{\mathbf{N}}^{*,\perp}) \right) \right\|_{\infty} \leq \zeta. \quad (2.24)$$

Moreover, $V(\boldsymbol{\beta}_N^\zeta)$ can be represented as $V(\cdot; \boldsymbol{\beta}_N^\zeta) = \beta_0^* + \sum_{i=1}^N \beta_i^\zeta \varphi(\cdot; \theta^i)$ for some coefficients $\beta_i^\zeta \in \mathbb{R}, i = 1, 2, \dots, N$.

Proof. Given $\zeta > 0$, since $\mathbf{B}_N^{*,o} \in \bar{\mathcal{B}}_N$ and $\bar{\mathcal{B}}_N$ is the closure of \mathcal{B}_N , there exists a function $\mathbf{B}_N^\zeta \in \mathcal{B}_N$ such that $\|(\mathbf{B}_N^{*,o} - \mathbf{B}_N^\zeta)/\rho\|_{2,\rho} \leq \zeta^2$. Therefore, for all $s \in \mathcal{S}$, we have

$$\begin{aligned}
& \left| V^*(s) - \left(V(s; \boldsymbol{\beta}_N^\zeta) + V(s; \boldsymbol{\beta}_N^{*,\perp}) \right) \right|^2 \\
&= \left(\int_{\Theta} \frac{1}{\rho(\theta)} \left[\mathbf{B}^*(\theta) - (\mathbf{B}_N^\zeta(\theta) + \mathbf{B}_N^{*,\perp}(\theta)) \right] \varphi(s; \theta) \rho(d\theta) \right)^2 \\
&\leq \int_{\Theta} \frac{1}{\rho(\theta)^2} \left[\mathbf{B}_N^{*,o}(\theta) + \mathbf{B}_N^{*,\perp}(\theta) - (\mathbf{B}_N^\zeta(\theta) + \mathbf{B}_N^{*,\perp}(\theta)) \right]^2 \rho(d\theta) \\
&= \|(\mathbf{B}_N^{*,o} - \mathbf{B}_N^\zeta)/\rho\|_{2,\rho}^2 \\
&\leq \zeta^2,
\end{aligned} \tag{2.25}$$

where the first equality follows from the definitions of $V^*(s)$ and $V(s; \cdot)$ evaluated at $\boldsymbol{\beta}_N^\zeta$ and $\boldsymbol{\beta}_N^{*,\perp}$; the first inequality from the Jensen's inequality $(\mathbb{E}[\cdot])^2 \leq \mathbb{E}[(\cdot)^2]$ and $\|\Phi\|_\infty \leq 1$ from Assumption 2; and the second equality from the definition of the $(2, \rho)$ -norm. Since the expression (2.25) holds for all $s \in \mathcal{S}$, we have

$$\left\| V^* - \left(V(\boldsymbol{\beta}_N^\zeta) + V(\boldsymbol{\beta}_N^{*,\perp}) \right) \right\|_\infty = \sup_{s \in \mathcal{S}} \left| V^*(s) - \left(V(s; \boldsymbol{\beta}_N^\zeta) + V(s; \boldsymbol{\beta}_N^{*,\perp}) \right) \right| \leq \zeta.$$

Finally, since $\boldsymbol{\beta}_N^\zeta := (\beta_0^*, \mathbf{B}_N^\zeta)$ with $\mathbf{B}_N^\zeta \in \mathcal{B}_N$, the VFA $V(\boldsymbol{\beta}_N^\zeta)$ can be represented as $V(\cdot; \boldsymbol{\beta}_N^\zeta) = \beta_0^* + \sum_{i=1}^N \beta_i^\zeta \varphi(\cdot; \theta^i)$ for some coefficients $\beta_i^\zeta \in \mathbb{R}, i = 1, 2, \dots, N$. ■

Proposition 10 *Suppose there exists a constant $\underline{\rho} > 0$ such that $\rho(\theta) \geq \underline{\rho}$ for all $\theta \in \Theta$. Consider $\zeta > 0$, $\delta \in (0, 1]$, and N iid samples $\{\theta^i : i = 1, 2, \dots, N\}$ from ρ . Let $(\beta_0^*, \mathbf{B}^*)$ denote an optimal solution to FEP and $(\beta_1^\zeta, \dots, \beta_N^\zeta)$ and $\beta_N^{*\perp} := (0, \mathbf{B}_N^{*\perp})$ be the coefficients described in Proposition 9. For every $H \geq 1$ iid samples $\{\theta^i : i = N+1, N+2, \dots, N+H\}$, there exist $(\beta_0^\perp, \beta_{N+1}^\perp, \beta_{N+2}^\perp, \dots, \beta_{N+H}^\perp) \in \mathbb{R}^H$ such that the vector*

$$\tilde{\beta} := (\beta_0^* + \beta_0^\perp, \beta_1^\zeta, \dots, \beta_N^\zeta, \beta_{N+1}^\perp, \beta_{N+2}^\perp, \dots, \beta_{N+H}^\perp) \in \mathbb{R}^{N+H+1}$$

satisfies

$$\left\| \mathbf{V}^* - \mathbf{V}(\tilde{\beta}) \right\|_\infty \leq \zeta + \frac{\|\mathbf{B}_N^{*\perp}/\rho\|_{2,\rho}}{\underline{\rho}\sqrt{H}} \left(\Omega + \sqrt{2 \ln \left(\frac{1}{\delta} \right)} \right),$$

with a probability of at least $1 - \delta$.

Proof. Since $\mathbf{B}_N^{*\perp} \in \mathcal{B}$, it is easy to see that $\mathbf{V}(\beta_N^{*\perp}) \in \mathcal{R}$. Then, Proposition 7 applied to the function $\mathbf{V}(\beta_N^{*\perp})$ and H samples $\{\theta^i : i = N+1, N+2, \dots, N+H\}$ guarantees that there are H coefficients $(\beta_0^\perp, \beta_{N+1}^\perp, \beta_{N+2}^\perp, \dots, \beta_{N+H}^\perp) \in \mathbb{R}^{H+1}$, such that

$$\left\| \mathbf{V}(\beta_N^{*\perp}) - \left(\beta_0^\perp + \sum_{i=N+1}^{N+H} \beta_i^\perp \varphi(\cdot; \theta^i) \right) \right\|_\infty \leq \frac{\|\mathbf{B}_N^{*\perp}/\rho\|_{2,\rho}}{\underline{\rho}\sqrt{H}} \left(\Omega + \sqrt{2 \ln \left(\frac{1}{\delta} \right)} \right), \quad (2.26)$$

with a probability of at least $1 - \delta$. Using Proposition 9 and the triangle inequality, with the same probability, we obtain

$$\left\| \mathbf{V}^* - \mathbf{V}(\tilde{\beta}) \right\|_\infty$$

$$\begin{aligned}
&\leq \left\| \mathbf{V}^* - \left(\mathbf{V}(\boldsymbol{\beta}_N^\zeta) + \mathbf{V}(\boldsymbol{\beta}_N^{*,\perp}) \right) \right\|_\infty + \left\| \left(\mathbf{V}(\boldsymbol{\beta}_N^\zeta) + \mathbf{V}(\boldsymbol{\beta}_N^{*,\perp}) \right) - \mathbf{V}(\tilde{\boldsymbol{\beta}}) \right\|_\infty \\
&\leq \zeta + \left\| \left(\mathbf{V}(\boldsymbol{\beta}_N^\zeta) + \mathbf{V}(\boldsymbol{\beta}_N^{*,\perp}) \right) - \left(\boldsymbol{\beta}_0^* + \sum_{i=1}^N \beta_i^\zeta \varphi(\cdot; \theta^i) + \boldsymbol{\beta}_0^\perp + \sum_{i=N+1}^{N+H} \beta_i^\perp \varphi(\cdot; \theta^i) \right) \right\|_\infty \\
&\leq \zeta + \left\| \mathbf{V}(\boldsymbol{\beta}_N^\zeta) - \boldsymbol{\beta}_0^* - \sum_{i=1}^N \beta_i^\zeta \varphi(\cdot; \theta^i) \right\|_\infty + \left\| \mathbf{V}(\boldsymbol{\beta}_N^{*,\perp}) - \boldsymbol{\beta}_0^\perp - \sum_{i=N+1}^{N+H} \beta_i^\perp \varphi(\cdot; \theta^i) \right\|_\infty \\
&\leq \zeta + \frac{\|\mathbf{B}_N^{*,\perp}/\rho\|_{2,\rho}}{\underline{\rho}\sqrt{H}} \left(\Omega + \sqrt{2 \ln \left(\frac{1}{\delta} \right)} \right),
\end{aligned}$$

where we used (2.24) and the definition of $\mathbf{V}(\tilde{\boldsymbol{\beta}})$ to obtain the second inequality; the triangle inequality for the third inequality; and $\mathbf{V}(\boldsymbol{\beta}_N^\zeta) = \boldsymbol{\beta}_0^* + \sum_{i=1}^N \beta_i^\zeta \varphi(\cdot; \theta^i)$ and (2.26) for the last one. ■

Recall that

$$E_{(N,H)} = \frac{\|\mathbf{B}_N^{*,\perp}/\rho\|_{2,\rho}}{\underline{\rho}\sqrt{H}} \left(\Omega + 2\sqrt{2 \ln \left(\frac{1}{\delta} \right)} \right).$$

Theorem 3 *Suppose there exists a constant $\underline{\rho} > 0$ such that $\rho(\theta) \geq \underline{\rho}$ for all $\theta \in \Theta$. Given $N \geq 1$ and $\delta \in (0, 1]$, for every $H \geq 1$, there exists a vector $\boldsymbol{\beta} \in \mathbb{R}^{N+H+1}$ such that with a probability of at least $1 - \delta$*

$$(i) \quad \|\mathbf{V}^* - \mathbf{V}(\boldsymbol{\beta})\|_\infty \leq E_{(N,H)} \text{ and}$$

$$(ii) \quad \omega(\boldsymbol{\beta}) \leq (1 + \gamma)E_{(N,H)}.$$

Proof. Let $\zeta := \|\mathbf{B}_N^{*,\perp}/\rho\|_{2,\rho} \sqrt{2 \ln(1/\delta)} / \underline{\rho}\sqrt{H}$ and $\tilde{\boldsymbol{\beta}}$ be the coefficient vector described in Proposition 10 for this specific choice of ζ . We claim that $\tilde{\boldsymbol{\beta}}$ is the desired vector in Theorem 3.

Part (i). Proposition 10 indicates, with a probability of at least $1 - \delta$, that

$$\left\| \mathbf{V}^* - \mathbf{V}(\tilde{\beta}) \right\|_{\infty} \leq \zeta + \frac{\|\mathbf{B}_N^{*,\perp}/\rho\|_{2,\rho}}{\underline{\rho}\sqrt{H}} \left(\Omega + \sqrt{2 \ln \left(\frac{1}{\delta} \right)} \right) = E_{(N,H)}.$$

Part (ii). The inequality $\|\mathbf{V}^* - \mathbf{V}(\tilde{\beta})\|_{\infty} \leq E_{(N,H)}$ from Part (i) indicates that with a probability of at least $1 - \delta$, we have

$$\mathbf{V}(s; \tilde{\beta}) - E_{(N,H)} \leq \mathbf{V}^*(s) \quad \text{and} \quad \mathbf{V}(s; \tilde{\beta}) + E_{(N,H)} \geq \mathbf{V}^*(s), \quad \forall s \in \mathcal{S}. \quad (2.27)$$

Hence, with the same probability, it follows that

$$\begin{aligned} (1 - \gamma)\tilde{\beta}_0 + \sum_{i=1}^{N+H} \tilde{\beta}_i (\varphi(s; \theta_i) - \gamma \mathbb{E} [\varphi(s'; \theta_i) | s, \mathbf{a}]) \\ = \mathbf{V}(s; \tilde{\beta}) - \gamma \mathbb{E} [\mathbf{V}(s'; \tilde{\beta}) | s, \mathbf{a}] \\ \leq \mathbf{V}^*(s) + E_{(N,H)} - \gamma \mathbb{E} [\mathbf{V}^*(s') | s, \mathbf{a}] + \gamma E_{(N,H)} \\ = c(s, \mathbf{a}) + (1 + \gamma)E_{(N,H)}, \end{aligned} \quad (2.28)$$

where the first equality follows from the definition of $\tilde{\beta}$ and the inequality from (2.27). The second equality holds since \mathbf{V}^* is an optimal solution to ELP. In addition, Proposition 5 and (2.27) imply that, with a probability of at least $1 - \delta$, we have

$$\mathbf{V}(s; \beta_N^{\text{SG}}) \leq \mathbf{V}^*(s) \leq \mathbf{V}(s; \tilde{\beta}) + E_{(N,H)}. \quad (2.29)$$

Inequalities (2.28) and (2.29) ensure that $\tilde{\beta}$ is $((1 + \gamma)E_{(N,H)})$ -feasible to constraints (2.9) and $E_{(N,H)}$ -feasible to constraints (2.10) of $\text{FALP}_{N+H}^{\text{SG}}$ with a probability of at least $1 - \delta$, respectively. Therefore, we can conclude $\tilde{\beta}$ satisfies

$$\omega(\tilde{\beta}) \leq \max \{ (1 + \gamma)E_{(N,H)}, E_{(N,H)} \} = (1 + \gamma)E_{(N,H)},$$

with a probability of at least $1 - \delta$. ■

Proof of Theorem 2.

Let $\beta \in \mathbb{R}^{N+H+1}$ be the vector in Theorem 3. Since the feasible set of $\text{FALP}_{N+H}^{\text{SG}}$ is a closed convex set, the 1-norm projection of β onto this set, which we denote by $\hat{\beta} := \text{proj}_{N+H}(\beta)$, is well defined. From Assumption 4, we have

$$\|\hat{\beta} - \beta\|_1 \leq G \cdot \omega(\beta)^{1/m} \leq G \left((1 + \gamma)E_{(N,H)} \right)^{1/m},$$

with a probability of at least $1 - \delta$. Considering VFAs with respect to $\hat{\beta}$ and β , we have

$$\begin{aligned} \|V(\hat{\beta}) - V(\beta)\|_{\infty} &= \left\| (\hat{\beta}_0 - \beta_0) + \sum_{i=1}^{N+H} (\hat{\beta}_i - \beta_i) \varphi(s; \theta^i) \right\|_{\infty} \\ &\leq \sum_{i=0}^{N+H} |\hat{\beta}_i - \beta_i| \\ &\leq G \left((1 + \gamma)E_{(N,H)} \right)^{1/m}, \end{aligned}$$

where the first inequality holds since we have $\|\varphi\|_\infty \leq 1$ from Assumption 2 and the second inequality follows from $\|\hat{\beta} - \beta\|_1 \leq G((1 + \gamma)E_{(N,H)})^{1/m}$. Using the triangle inequality and Part (i) of Theorem 3, we can show that

$$\|V^* - V(\hat{\beta})\|_\infty \leq \|V^* - V(\beta)\|_\infty + \|V(\beta) - V(\hat{\beta})\|_\infty \leq E_{(N,H)} + G((1 + \gamma)E_{(N,H)})^{1/m},$$

holds with a probability of at least $1 - \delta$. Hence, for any optimal solution β_{N+H}^{SG} to $\text{FALP}_{N+H}^{\text{SG}}$, with a probability of at least $1 - \delta$, it holds that

$$\|V^* - V(\beta_{N+H}^{\text{SG}})\|_{1,\nu} \leq \|V^* - V(\hat{\beta})\|_{1,\nu} \leq \|V^* - V(\hat{\beta})\|_\infty \leq E_{(N,H)} \left[1 + G((1 + \gamma)E_{(N,H)})^{(1-m)/m} \right],$$

where the first inequality holds since $\hat{\beta}$ is feasible to $\text{FALP}_{N+H}^{\text{SG}}$ and β_{N+H}^{SG} is optimal. ■

2.9.5 Proofs of Statements in §2.5

Proof of Proposition 6.

Applying Proposition 7 to $V^C(\cdot) = \beta_0^C + \int_{\Theta} \mathbf{B}^C(\theta) \varphi(\cdot; \theta) d\theta$ with $\|\mathbf{B}^C/\rho\|_{2,\rho} < \infty$ and replacing Ω with Ω^C , we get that for N iid samples $\{\theta^i : i = 1, 2, \dots, N\}$ from ρ , there exist coefficients $\bar{\beta} := (\bar{\beta}_0, \bar{\beta}_1, \dots, \bar{\beta}_N)$ such that

$$\sup_{s \in \mathcal{S}^C} |V^C(s) - V(s; \bar{\beta})| = \|V^C - V(\bar{\beta})\|_\infty \leq E_N := \frac{\|\mathbf{B}^C/\rho\|_{2,\rho}}{\underline{\rho}\sqrt{N}} \left(\Omega^C + \sqrt{2 \ln \left(\frac{1}{\delta} \right)} \right), \quad (2.30)$$

with a probability of at least $1 - \delta$. Using the definition of V^C (see §2.5), it is straightforward to see that $V^C(s^m) = V^*(s^m)$ for all $s^m \in \mathcal{S}$. Hence, the inequality (2.30) indicates that with a probability of at least $1 - \delta$,

$$\sup_{s^m \in \mathcal{S}} |V^*(s^m) - V(s^m; \bar{\beta})| = \sup_{s^m \in \mathcal{S}} |V^C(s^m) - V(s^m; \bar{\beta})| \leq \sup_{s \in \mathcal{S}^C} |V^C(s) - V(s; \bar{\beta})| \leq E_N, \quad (2.31)$$

where we used the fact that $\mathcal{S} \subseteq \mathcal{S}^C$ to obtain the first inequality.

In addition, since $V^*(s^m)$ satisfies FALP constraints, i.e., $V^*(s^m) - \gamma \mathbb{E}[V^*(s') | s^m, \mathbf{a}] \leq c(s^m, \mathbf{a})$ for all $(s^m, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}_s$, following similar steps as in (2.23), the inequality (2.31) indicates that the solution $\hat{\beta} := (\bar{\beta}_0 - \Gamma E_N, \bar{\beta}_1, \dots, \bar{\beta}_N)$ is feasible to \mathbf{FALP}_N with a probability of at least $1 - \delta$. Hence, we have

$$\begin{aligned} \|V^* - V(\beta_N^{\text{FA}})\|_{1, \mathbf{v}} &\leq \|V^* - V(\hat{\beta})\|_{1, \mathbf{v}} \\ &= \|V^* - V(\bar{\beta})\|_{1, \mathbf{v}} + \Gamma E_N \\ &= \sum_{m \in \mathcal{M}} \mathbf{v}(s^m) |V^*(s^m) - V(s^m, \bar{\beta})| + \Gamma E_N \\ &\leq (1 + \Gamma) E_N \\ &= \frac{2\|\mathbf{B}^C/\rho\|_{2, \rho}}{(1 - \gamma)\underline{\rho}\sqrt{N}} \left(\Omega^C + \sqrt{2 \ln \left(\frac{1}{\delta} \right)} \right), \end{aligned}$$

where the first inequality follows from the feasibility of $\hat{\beta}$ and optimality of β_N^{FA} to \mathbf{FALP}_N ; the first equality from the definition of $\hat{\beta}$; the second equality from the $(1, \mathbf{v})$ -norm definition; the second inequality from (2.31); and the last equality from the definition of E_N . ■

2.10 Relaxing Assumptions

In §2.10.1 and §2.10.2, we discuss how our theory extends when assumptions $V^* \in \mathcal{R}$ and 3 fail to hold, respectively.

2.10.1 Relaxing Assumption of $V^* \in \mathcal{R}$

In this section, Proposition 11 shows that there exists a feasible solution to FEP such that its VFA is ε -close to V^* under the ∞ -norm for every choice of $\varepsilon > 0$, given that Assumptions 1 and 2 are satisfied (refer to §2.9.1). Consider the sequence of these feasible solutions when $\varepsilon \rightarrow \infty$. While each element in this sequence is feasible to FEP and gets closer to V^* , the limit point of this sequence may not be feasible to FEP because the set \mathcal{R} is not a closed set. Therefore, whenever $V^* \in \mathcal{R}$ fails to hold, FEP may not attain an optimal solution. In this case, one can replace the constraint $\|\mathbf{B}/\rho\|_{2,\rho} < \infty$ in the formulation of FEP by the constraint $\|\mathbf{B}/\rho\|_{2,\rho} \leq C_\varepsilon$ for a sufficiently large finite constant $C_\varepsilon > 0$ to ensure FEP attains an optimal solution with a VFA that is ε -close to V^* .

Proposition 11 *Assume $V^* \notin \mathcal{R}$. Given $\varepsilon > 0$, there exists a feasible solution, $\beta_\varepsilon^{\text{FE}} = (\beta_{0,\varepsilon}^{\text{FE}}, \mathbf{B}_\varepsilon^{\text{FE}})$ to FEP such that*

$$\|V^* - V(\beta_\varepsilon^{\text{FE}})\|_\infty \leq \frac{2\varepsilon}{1-\gamma}.$$

Proof. Since MDP value function V^* is continuous (by Assumption 1) and the class of random basis function φ is universal (by Assumption 2), there is $\hat{V} \in \mathcal{R}$ such that $\|V^* - \hat{V}\|_\infty \leq \varepsilon$. Since \hat{V} belongs to \mathcal{R} , it can be written as $\hat{V}(s, \hat{\beta}) = \hat{\beta}_0 + \int_{\Theta} \hat{\mathbf{B}}(\theta) \varphi(s; \theta) d\theta$ for some $\hat{\beta} = (\hat{\beta}_0, \hat{\mathbf{B}})$ with $\|\hat{\mathbf{B}}/\rho\|_{2,\rho} < \infty$. Recall that $\Gamma = (1+\gamma)/(1-\gamma)$. We now show that $\beta_\varepsilon^{\text{FE}} = (\beta_{0,\varepsilon}^{\text{FE}}, \mathbf{B}_\varepsilon^{\text{FE}}) :=$

$(\hat{\beta}_0 - \Gamma\epsilon, \hat{\mathbf{B}})$ is the desired feasible FEP solution. This is because $\|\mathbf{B}_\epsilon^{\text{FE}}/\rho\|_{2,\rho} = \|\hat{\mathbf{B}}/\rho\|_{2,\rho} < \infty$ and for any $(s, a) \in \mathcal{S} \times \mathcal{A}_s$, we have

$$\begin{aligned}
(1 - \gamma)\beta_{0,\epsilon}^{\text{FE}} + \int_{\Theta} \mathbf{B}_\epsilon^{\text{FE}}(\theta)(\varphi(s) - \gamma\mathbb{E}[\varphi(s') | s, a]) d\theta \\
&= (1 - \gamma)(\hat{\beta}_0 - \Gamma\epsilon) + \int_{\Theta} \hat{\mathbf{B}}(\theta)(\varphi(s) - \gamma\mathbb{E}[\varphi(s') | s, a]) d\theta \\
&= -(1 + \gamma)\epsilon + \hat{V}(s) - \gamma\mathbb{E}[\hat{V}(s') | s, a] \\
&\leq -(1 + \gamma)\epsilon + V^*(s) + \epsilon - \gamma\mathbb{E}[V^*(s') - \epsilon | s, a] \\
&= V^*(s) - \gamma\mathbb{E}[V^*(s') | s, a] \\
&\leq c(s, a),
\end{aligned}$$

where the first inequality is valid since $\|V^* - \hat{V}\|_\infty \leq \epsilon$, which ensures $\hat{V}(s) \leq V^*(s) + \epsilon$ and $-\hat{V}(s) \leq -V^*(s) + \epsilon$ for all $s \in \mathcal{S}$. Thus, $\beta_\epsilon^{\text{FE}}$ is feasible to FEP. In addition, the VFA $V(\beta_\epsilon^{\text{FE}}) = \hat{V}(\hat{\beta}) - \Gamma\epsilon$ belongs to \mathcal{R} and $\|V^* - V(\beta_\epsilon^{\text{FE}})\|_\infty \leq \|V^* - \hat{V}\|_\infty + \Gamma\epsilon \leq \epsilon + \Gamma\epsilon = 2\epsilon/(1 - \gamma)$, which completes the proof. \blacksquare

2.10.2 Relaxing Assumption 3

For a given $\alpha > 0$, define vector $\hat{\beta} \in \mathbb{R}^{N+1}$ as an optimal solution to the following program:

$$\begin{aligned}
\max_{\beta} \quad & \beta_0 + \sum_{i=1}^N \beta_i \mathbb{E}_v[\varphi(s; \theta^i)] \\
\text{s.t.} \quad & (1 - \gamma)\beta_0 + \sum_{i=1}^N \beta_i \left(\varphi(s; \theta^i) - \gamma\mathbb{E}[\varphi(s'; \theta^i) | s, a] \right) \leq c(s, a), \quad \forall (s, a) \in \mathcal{S} \times \mathcal{A}_s \quad (2.32) \\
& |\beta_i| \leq \alpha, \quad \forall i = 1, 2, \dots, N.
\end{aligned}$$

Although there are explicit bounds on $\beta_1, \beta_2, \dots, \beta_N$, the constraints of the problem also imply

$$\beta_0 \leq \max_{\{(\beta_1, \dots, \beta_N) : |\beta_i| \leq \alpha\}} \max_{(s, a) \in \mathcal{S} \times \mathcal{A}_s} \left\{ \frac{1}{1 - \gamma} \left[c(s, a) - \sum_{i=1}^N \beta_i \left(\varphi(s; \theta^i) - \gamma \mathbb{E}[\varphi(s'; \theta^i) \mid s, a] \right) \right] \right\},$$

where the right-hand side is upper bounded by a constant because the state and action spaces are compact, and the cost function evaluations are finite because V^* is bounded, which follows from it being a continuous function defined over a compact set. Program (2.32) always attains its maximum since it optimizes a continuous function with $N + 1$ decision variables over a compact convex set.

Proposition 12 develops an error bound for the VFA associated with (2.32).

Proposition 12 *Suppose $\rho(\theta) \geq \underline{\rho}$ for all $\theta \in \Theta$. Given $\delta \in (0, 1]$, we have that any optimal solution $\hat{\beta} \in \mathbb{R}^{N+1}$ to linear program (2.32) with $\alpha \geq \|B^*/\rho\|_{2,\rho}/(N\underline{\rho})$ satisfies*

$$\|V^* - V(\hat{\beta})\|_{1,v} \leq \frac{\|B^*/\rho\|_{2,\rho}}{\underline{\rho}\sqrt{N}} \left(\Omega + \sqrt{2 \ln \left(\frac{1}{\delta} \right)} \right),$$

with a probability of at least $1 - \delta$.

Proof. (i) Any feasible solution β to (2.32) satisfies $V(s; \beta) \leq V^*(s)$ for all $s \in \mathcal{S}$ by Lemma 1 since $V(\cdot; \beta)$ is continuous by Assumption 2. From this it follows that $\mathbb{E}_v[V(\beta)] \leq \mathbb{E}_v[V^*]$. By Assumption 1, V^* is a continuous function over a compact domain and is thus bounded by a

finite constant, which implies that the optimal objective function value of (2.32) is also bounded above by this constant. Therefore, FALP_N has a finite optimal objective function value.

Let β^* be an optimal solution to (2.32). Then $\beta_1^*, \beta_2^*, \dots, \beta_N^*$ are finite because of the bounding constraints. The next proposition develops a VFA error rate for this program.

(ii) Consider $\varepsilon > 0$. Given $\beta^\theta = (\beta_0^\theta, \beta_1^\theta, \dots, \beta_N^\theta)$ and N_ε respectively defined in Definition 2 and Lemma 2, part (ii) of Lemma 2 ensures that when $N \geq N_\varepsilon$, the vector $(\beta_0^\theta - \Gamma\varepsilon, \beta_1^\theta, \dots, \beta_N^\theta)$ is a feasible solution to FALP_N with a probability of at least $1 - \delta$. From the definition of each element β_i^θ , we have that

$$|\beta_i^\theta| \leq \frac{\|\mathbf{B}^*/\rho\|_{2,\rho}}{N\rho}.$$

Hence, vector $(\beta_0^\theta - \Gamma\varepsilon, \beta_1^\theta, \dots, \beta_N^\theta)$ is a feasible solution to (2.32) with a probability of at least $1 - \delta$, and we thus have

$$\|V^* - V(\hat{\beta})\|_{1,\nu} \leq \|V^* - (V(\beta^\theta) - \Gamma\varepsilon)\|_{1,\nu} \leq \|V^* - (V(\beta^\theta) - \Gamma\varepsilon)\|_\infty \leq \frac{2\varepsilon}{1-\gamma}.$$

In above, we used the optimality of $\hat{\beta}$ to obtain the first inequality, the relationship between $(1, \nu)$ - and ∞ -norms to obtain the second inequality, and part (ii) of Lemma 2 for the last one.

Since $N \geq N_\varepsilon$, by choosing choose ε according to

$$\varepsilon \leq \frac{\|\mathbf{B}^*/\rho\|_{2,\rho}}{\rho\sqrt{N}} \left(\Omega + \sqrt{2 \ln \left(\frac{1}{\delta} \right)} \right),$$

we complete the proof. ■

2.11 Constraint Sampling Bound for Self-guided FALP

Let $\{(s^k, a^k) \in \mathcal{S} \times \mathcal{A}_s : k = 1, 2, \dots, K\}$ be a set of K state-action pairs sampled from a probability distribution ψ over the state-action space $\mathcal{S} \times \mathcal{A}_s$. The constraint-sampled self-guided FALP is given by the following linear program that has N random basis functions and $2K$ constraints:

$$\begin{aligned} & \max_{\beta} \beta_0 + \sum_{i=1}^N \beta_i \mathbb{E}_{\nu}[\varphi(s; \theta^i)] \\ & \text{s.t. } (1 - \gamma)\beta_0 + \sum_{i=1}^N \beta_i \left(\varphi(s^k; \theta^i) - \gamma \mathbb{E}[\varphi(s'; \theta^i) | s^k, a^k] \right) \leq c(s^k, a^k), \quad k = 1, 2, \dots, K, \quad (2.33) \\ & \beta_0 + \sum_{i=1}^N \beta_i \varphi(s^k; \theta^i) \geq V(s^k; \beta_{N-B}^{\text{SG}}), \quad k = 1, 2, \dots, K. \end{aligned}$$

The following proposition develops a probabilistic bound on the number of samples K to ensure that the volume of state-action pairs that an optimal solution to (2.33) satisfies their corresponding constraints of $\text{FALP}_N^{\text{SG}}$ is high.

Proposition 13 *Given $\delta \in (0, 1]$, if ψ is supported over $\mathcal{S} \times \mathcal{A}_s$, linear program (2.33) is bounded, and the number of samples K satisfies*

$$K \geq \left\lceil \frac{2}{\delta} \ln \left(\frac{1}{\delta} \right) + 2(N + 1) + \frac{2(N + 1)}{\delta} \ln \left(\frac{2}{\delta} \right) \right\rceil,$$

then for every optimal solution $\hat{\beta}$ to (2.33), the following inequality holds

$$\psi \left(\left\{ (s, a) \in \mathcal{S} \times \mathcal{A}_s \mid V(s; \hat{\beta}) - \gamma \mathbb{E}[V(s'; \hat{\beta}) | s, a] \leq c(s, a), \quad V(s; \hat{\beta}) \geq V(s; \beta_{N-B}^{\text{SG}}) \right\} \right) \geq 1 - \delta.$$

with a probability of at least $1 - \delta$.

Proof. The proof of this proposition is similar to the proof of Proposition 3, where we leverage the theoretical results in CC. Following the program called “prototype control problem” introduced in §II of CC, we define three functions $h^{\text{FA}}, h^{\text{SG}}, h : \mathbb{R}^{N+1} \times \mathcal{S} \times \mathcal{A}_s \mapsto \mathbb{R}$ at $\beta = (\beta_0, \beta_1, \dots, \beta_N) \in \mathbb{R}^{N+1}$, $s \in \mathcal{S}$, and $\mathbf{a} \in \mathcal{A}_s$ as follows:

$$\begin{aligned} h^{\text{FA}}(\beta; s, \mathbf{a}) &:= (1 - \gamma)\beta_0 + \sum_{i=1}^N \beta_i \left(\varphi(s; \theta^i) - \gamma \mathbb{E}[\varphi(s'; \theta^i) \mid s, \mathbf{a}] \right) - c(s, \mathbf{a}), \\ h^{\text{SG}}(\beta; s, \mathbf{a}) &:= V(s; \beta_{N-B}^{\text{SG}}) - \beta_0 - \sum_{i=1}^N \beta_i \varphi(s; \theta^i), \\ h(\beta; s, \mathbf{a}) &:= \max \{ h^{\text{FA}}(\beta; s, \mathbf{a}), h^{\text{SG}}(\beta; s, \mathbf{a}) \}. \end{aligned}$$

Note that given $s \in \mathcal{S}$, the function $h^{\text{SG}}(\beta; s, \mathbf{a})$ is constant across all actions $\mathbf{a} \in \mathcal{A}$. Thus, program $\text{FALP}_N^{\text{SG}}$ can be reformulated as:

$$\max_{\beta} \beta_0 + \sum_{i=1}^N \beta_i \mathbb{E}_{\nu}[\varphi(s; \theta^i)] \quad \text{s.t.} \quad h(\beta; s, \mathbf{a}) \leq 0, \quad \forall (s, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}_s. \quad (2.34)$$

Taking assumptions 1 and 2 in CC to hold, if we apply Corollary 1 and Theorem 1 of CC to program (2.33), which is a random relaxation of (2.34), we obtain a guarantee that the optimal solution $\hat{\beta}$ of (2.33) satisfies:

$$\psi \left(\{ (s, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}_s : h(\hat{\beta}; s, \mathbf{a}) \leq 0 \} \right) \geq 1 - \delta,$$

with a probability of at least $1 - \delta$. To complete the proof, we show that assumptions 1 and 2 of CC hold in our setting.

First notice that we use the notations \mathbf{h} , β , \mathbb{R}^{N+1} , $N+1$, (s, \mathbf{a}) , and $\mathcal{S} \times \mathcal{A}_s$ in Chapter 2 instead of f , θ , Θ , \mathbf{n}_θ , δ , and Δ , respectively, in CC. Assumption 1 in CC requires the function $\mathbf{h}(\beta; \cdot, \cdot)$ to be convex in β and continuous. In our setting, $\mathbf{h}(\beta; \cdot, \cdot) = \max\{\mathbf{h}^{\text{FA}}(\beta; \cdot, \cdot), \mathbf{h}^{\text{SG}}(\beta; \cdot, \cdot)\}$ is convex because it is the pointwise maximum of two convex (linear) functions \mathbf{h}^{FA} and \mathbf{h}^{SG} . It is also continuous in β since we assume $\varphi(\cdot)$ is Lipschitz continuous. We use a relaxation of Assumption 2 in CC as stated in Appendix A. In particular, we only show that the program (2.33) is feasible and forgo the uniqueness assumption of the optimal solution to $\text{FALP}_N^{\text{SG}}$. By appending B zeros to the past self-guided FALP solution $\beta_{N-B}^{\text{SG}} \in \mathbb{R}^{N+1-B}$, we define vector $(\beta_{N-B}^{\text{SG}}, 0, 0, \dots, 0) \in \mathbb{R}^{N+1}$ which is feasible to $\text{FALP}_N^{\text{SG}}$ and thus is feasible to (2.33) for all samples $\{(s^k, \mathbf{a}^k) \in \mathcal{S} \times \mathcal{A}_s : k = 1, 2, \dots, K\}$. ■

2.12 Optimistic Bound Estimation

In §2.12.1, we discuss the estimation of optimistic bounds on the MDP optimal policy value using the constraint violation learning approach in Lin et al. (2020). In §2.12.2, we discuss the information relaxation and duality approach from Brown et al. (2010) that can be used to estimate optimistic bounds on the optimal policy value of an MDP, where the exogenous state space is large and the controllable part of the state space is low dimensional.

2.12.1 Constraint Violation Learning

We first discuss a heuristic approach based on the constraint violation learning approach (CVL; Lin et al. 2020) for estimating a lower bound on the optimal policy cost (or upper

bounds on the optimal policy reward). We then explain how we use this method to estimate lower bounds for the perishable inventory control instances studied in §2.6. CVL utilizes primal and dual updates to approximate the MDP value function while simultaneously learning which constraints of ALP are being violated by the current VFA weights. Our CVL-based heuristic only performs the dual update to obtain a valid lower bound. Specifically, for a given VFA, this approach uses the ideas in CVL to learn a distribution that assigns high values to state-action pairs where ALP constraints are violated and then employs this information to define a valid lower bound.

CVL-based heuristic. For any VFA $V(\beta)$ with $\beta \in \mathbb{R}^{N+1}$, define function $y(\cdot, \cdot; \beta) : \mathcal{S} \times \mathcal{A}_s \mapsto \mathbb{R}$ as

$$y(s, a; \beta) := \mathbb{E}_\chi[V(\beta)] + \frac{1}{1-\gamma} \left(c(s, a) + \gamma \mathbb{E}[V(s'; \beta) \mid s, a] - V(s; \beta) \right),$$

where the second term encodes the slack in the FALP constraint for a given β at a state-action pair (s, a) . The coefficient β may not be feasible for all FALP constraints. We observe that minimizing the function $y(s, a; \beta)$ over state-action pairs corresponds to finding the most violated constraint in FALP_N since the term $\mathbb{E}_\chi[V(\beta)]$ is independent of the state and action and the term $(c(s, a) + \gamma \mathbb{E}[V(s'; \beta) \mid s, a] - V(s; \beta)) / (1 - \gamma)$ is the constraint slack. Thus, if the minimum value of function $y(s, a; \beta)$ over state-action pairs is strictly less than $\mathbb{E}_\chi[V(\beta)]$, then β violates a constraint of FALP_N . Otherwise, β is feasible to FALP_N .

Lemma 3 is directly based on Lemma EC.3 in Lin et al. (2020) and provides a lower bound on the optimal cost. For a given VFA $V(\beta)$ and $\lambda \in (0, 1]$, we define a density Y on $\mathcal{S} \times \mathcal{A}_s$ as

$$Y(s, \mathbf{a}; \beta, \lambda) := \frac{\exp(-y(s, \mathbf{a}; \beta)/\lambda)}{\int_{\mathcal{S} \times \mathcal{A}_s} \exp(-y(s, \mathbf{a}; \beta)/\lambda) d(s, \mathbf{a})}. \quad (2.35)$$

Lemma 3 (Lemma EC.3 in Lin et al. 2020) *Suppose y is Lipschitz continuous with constant $L_y > 0$. For any $\lambda \in (0, 1]$ and $\beta \in \mathbb{R}^{N+1}$, we have*

$$\mathbb{E}_Y[y(s, \mathbf{a}; \beta)] + \lambda(\Lambda + d_{(s, \mathbf{a})} \ln(\lambda)) \leq \text{PC}(\pi^*),$$

where constant Λ is defined as follows:

$$\Lambda := -\ln \left[\bar{\Gamma} \left(1 + \frac{d_{(s, \mathbf{a})}}{2} \right) \left(R_{\mathcal{S} \times \mathcal{A}_s} \sqrt{\pi} \right)^{-d_{(s, \mathbf{a})}} \int_{\mathcal{A}_s} d(s, \mathbf{a}) \right] - L_y (R_{\mathcal{S} \times \mathcal{A}_s} + D_{(s, \mathbf{a})}).$$

Moreover, $d_{(s, \mathbf{a})}$ is the dimension of the space $\mathcal{S} \times \mathcal{A}_s$. Function $\bar{\Gamma}$ is the standard gamma function, π is the Archimedes constant, $R_{\mathcal{S} \times \mathcal{A}_s} > 0$ is the radius of the largest ball contained in $\mathcal{S} \times \mathcal{A}$, and $D_{(s, \mathbf{a})}$ is the diameter of $\mathcal{S} \times \mathcal{A}$.

Given a solution β and its VFA $V(\beta)$, Lemma 3 shows that a valid lower bound on optimal cost $\text{PC}(\pi^*)$ can be computed as the sum of the expected value $\mathbb{E}_Y[y(s, \mathbf{a}; \beta)]$ and a constant term.

Applying CVL-based heuristic to perishable inventory control instances. Estimating a lower bound using the aforementioned CVL-based heuristic requires generating samples $\{(s^i, \mathbf{a}^i) : i = 1, 2, \dots, I\}$ from distribution Y in (2.35). Computing the denominator of this

distribution is intractable but it is known that there are MCMC methods that can generate samples from un-normalized distributions. For example, the Metropolis-Hastings algorithm can be used to generate samples $\{(s^i, \mathbf{a}^i) : i = 1, 2, \dots, I\}$ from the un-normalized density $\exp(-\mathbf{y}(s, \mathbf{a}; \beta)/\lambda)$ in the numerator of Y , which is proportional to $Y(s, \mathbf{a}; \beta, \lambda)$. Upon generating samples $\{(s^i, \mathbf{a}^i) : i = 1, 2, \dots, I\}$, we obtain the following lower bound estimate based on a sample average approximation:

$$\text{LB}(\beta) := \mathbb{E}_{\chi}[\mathbf{V}(\beta)] + \frac{1}{I(1-\gamma)} \sum_{i=1}^I \left[c(s^i, \mathbf{a}^i) + \gamma \mathbb{E}[\mathbf{V}(s'; \beta) | s^i, \mathbf{a}^i] - \mathbf{V}(s^i; \beta) \right] + \lambda(\Lambda + \mathbf{d}_{(s, \mathbf{a})} \ln(\lambda))$$

The two additional expectations here can be also replaced by sample average approximations. In our numerical experiments in §2.6, we estimate $\text{LB}(\beta)$ using the Metropolis-Hastings method with $I = 4000$ samples. These samples are obtained by generating 8 Markov Chains with the length of 1500 in parallel, burning the first 1000 samples, and then using the last 500 samples. Parameter Λ can be easily evaluated for the instances studied in §2.6. The perishable inventory control application cost function is Lipschitz with constant $L_c > 0$, where $L_c = 2(\gamma^L c_o \bar{a} + c_h \bar{a} + c_b \underline{s} + c_d \bar{a} + c_l \bar{a})$. From this we infer that the Lipschitz constant associated with \mathbf{y} is $L_y = (4\|\beta\|_1 + L_c)/(1-\gamma)$. We choose the other parameters defining Λ as follows: $\mathbf{d}_{(s, \mathbf{a})}$ is given by the summation of the dimensions of MDP state and action spaces that depends on each instance; $\mathbf{R}_{S \times \mathcal{A}_s}$ is $\frac{\bar{a}}{2}$ and $\mathbf{D}_{(s, \mathbf{a})} = 3\bar{a}^2 + (\underline{s} - \bar{a})^2$; and λ is set to $1/(\Lambda + \mathbf{d}_{(s, \mathbf{a})})$. One can tune the last parameter to possibly obtain tighter bounds.

2.12.2 Information Relaxation and Duality

We switch from cost minimization to reward maximization here to be consistent with the Bermudan option pricing problem that we apply it to. Information relaxation and duality (IR; [Brown et al. 2010, 2022](#)) is a general framework to compute upper bounds on the optimal policy reward of MDPs. This approach relies on allowing the decision maker to observe realizations of future uncertainties when making a decision at the current time and then penalizing the knowledge of such information. We discuss how this method can be used for Bermudan options pricing to derive upper bounds on the optimal policy reward. For applying this method beyond this application, please see [Brown et al. \(2010\)](#), [Nadarajah et al. \(2017\)](#), and [Brown et al. \(2022\)](#).

Let $\{(s_0^i, s_1^i, \dots, s_T^i) : i = 1, 2, \dots, I\}$ be a set of I sample paths generated from a fixed initial state s_0 . At time t , $s_t^i = (p_{t,1}^i, p_{t,2}^i, \dots, p_{t,J}^i, y_t^i)$ encodes prices of J assets at time t on the i -th sample path as well as the binary variable y_t^i that shows if the option is knocked-out or not (please see §2.7.1). A perfect information relaxation with zero dual penalty requires solving the following deterministic dual dynamic program on each sample path i :

$$V_t^D(s_t^i) = \begin{cases} g(s_t^i), & t = T \\ \max \{g(s_t^i), \gamma V_{t+1}^D(s_{t+1}^i)\}, & t = T-1, T-2, \dots, 0. \end{cases}$$

The average of the dual value function at the initial state along each sample path defines the upper bound estimate $UB = (\sum_{i=1}^I V_0^D(s_0^i))/I$. This bound based on a zero dual bound is typically loose.

VFAs/CFAs can be used to define dual penalties that can be incorporated into the above dynamic program for improving the bound quality. Let $V_t : \mathcal{S} \mapsto \mathbb{R}$ be the time- t VFA. Define the VFA-based dual penalty $z_t(s_{t+1}^i, s_t^i, \mathbf{a}_t; V_{t+1})$ at time t , action \mathbf{a}_t , and the state pair (s_{t+1}^i, s_t^i) on the i -th sample path as follows:

$$z_t(s_{t+1}^i, s_t^i, \mathbf{a}_t; V_{t+1}) = \gamma(1 - \mathbf{a}_t) \left(V_{t+1}(s_{t+1}^i) - \frac{1}{M} \sum_{m=1}^M V_{t+1}(s_{t+1}^{m|i}) \right),$$

where the next state $s_{t+1}^{m|i}$ is drawn from stochastic kernel $P(\cdot | s_t^i)$. The set of samples $\{s_{t+1}^{m|i} : m = 1, 2, \dots, M\}$ are called inner samples. For the action $\mathbf{a}_t = 1$ that corresponds to option exercise, we have $z_t(s_{t+1}^i, s_t^i, 1; V_{t+1}) = 0$. Note that $\mathbb{E}[z_t(s_{t+1}, s_t, \mathbf{a}_t; V_{t+1}) | s_t] = 0$, which shows that the VFA-based dual penalties is feasible. We can also construct feasible dual penalties based on CFAs. Let $C_t : \mathcal{S} \mapsto \mathbb{R}$ be the time- t CFA. The dual penalty $z_T(\cdot, \cdot, \cdot; C_{T+1}) \equiv 0$ and define the time- t dual penalty $z_t(s_{t+1}^i, s_t^i, \mathbf{a}_t; C_{t+1})$ with respect to CFA C_{t+1} as follows:

$$z_t(s_{t+1}^i, s_t^i, \mathbf{a}_t; C_{t+1}) = \gamma(1 - \mathbf{a}_t) \left(\max \left\{ g(s_{t+1}^i), C_{t+1}(s_{t+1}^i) \right\} - \frac{1}{M} \sum_{m=1}^M \left[\max \{ g(s_{t+1}^{m|i}), C_{t+1}(s_{t+1}^{m|i}) \} \right] \right).$$

Both VFA-based and CFA-based dual penalties can be used to improve the upper bound obtained from perfect information relaxation with zero dual penalty. Consider the following deterministic dual dynamic program:

$$V_t^D(s_t^i; z) = \begin{cases} g(s_t^i), & t = T, \\ \max \{g(s_t^i), \gamma V_{t+1}^D(s_{t+1}^i; z_{t+1}) - z_t(s_{t+1}^i, s_t^i, 0)\}, & t = T-1, T-2, \dots, 0, \end{cases} \quad (2.36)$$

where dual penalties (z_0, z_1, \dots, z_T) can be defined using VFAs or CFAs as described above.

Note that the maximum term in (2.36) is equivalent to

$$\max \left\{ g(s_t^i) - z_t(s_{t+1}^i, s_t^i, 1), \gamma V_{t+1}^D(s_{t+1}^i; z_{t+1}) - z_t(s_{t+1}^i, s_t^i, 0) \right\}.$$

Because $z_t(s_{t+1}^i, s_t^i, 1) = 0$, the above maximization simplifies to the one in (2.36). Upon solving dual dynamic program (2.36), we obtain the upper bound estimate $UB(z) = (\sum_{i=1}^I V_0^D(s_0^i; z))/I$.

For the Bermudan options pricing instances considered in §2.7, we use $I = 20,000$ sample paths and $M = 500$ inner samples to estimate the upper bound $UB(z)$. We use LSM to compute a CFA and ALP^{DFM} , $FALP_{500}$, and $FALP_{500,6}^{SG}$ to compute VFAs in §2.7.3. In each case, we estimate an upper bound $UB(z)$ using the corresponding CFA- and VFA-based dual penalty definition.

2.13 Addendum to Numerical Study

In §2.13.1, we numerically visualize the self-guiding mechanism on a two-dimensional instance of the perishable inventory control problem. In §2.13.2, we report the performance of

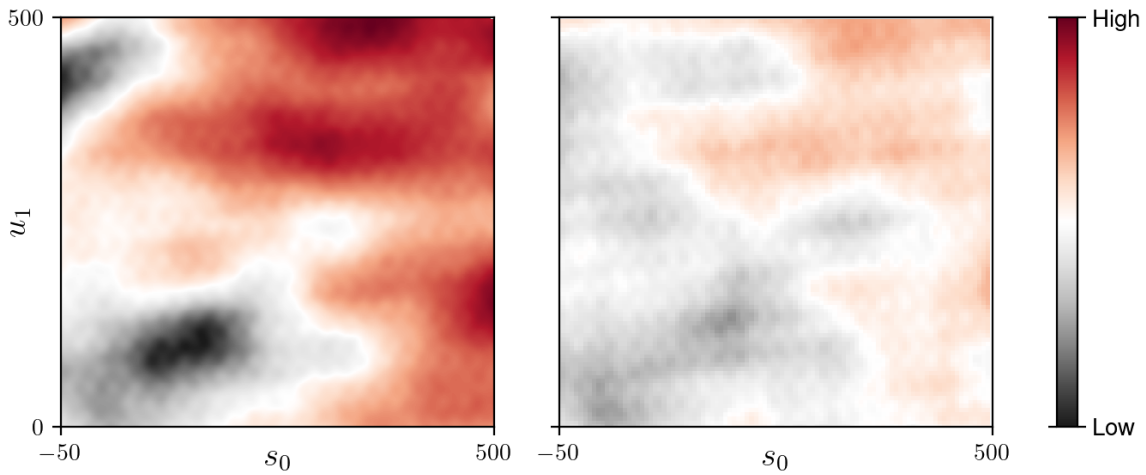
policy-guided FALP when using different constraint sampling strategies. In §2.13.3, we evaluate how our approach performs on perishable inventory control and Bermudan options pricing applications when using ReLU basis functions. In §2.13.4, we report raw lower and upper bound values computed from FALP, self-guided FALP, and other benchmarks we consider in Chapter 2.

2.13.1 Visualization of Self-guiding Mechanism

We consider a two-dimensional instance of the perishable inventory control problem studied in §2.6.1 to visualize the following: (i) VFAs, (ii) relative approximation quality and the states visited by policies, and (iii) the implicit state-relevance distribution used by self-guided FALP. We focus on an instance of the MDP in §2.6.1 with a lifetime of $\mathfrak{l} = 1$ and a lead time of $J = 2$. The state vector of this instance has two elements $\mathbf{s} = (s_0, \mathbf{u}_1) \in \mathbb{R}^2$, where s_0 is the on-hand inventory level expiring in the current period and \mathbf{u}_1 is the order quantity arriving in the next period. Demand follows a truncated normal distribution with a mean of 5, a standard deviation of 2, and has support in the interval $[0, 10]$. These choices are the ones we used for the three-dimensional instances in Table II. We choose the state space diameter of this MDP to be large by setting the maximum ordering level to be $\bar{\mathbf{a}} = 500$ and the maximum limit on the number of backlogged orders to be $\underline{\mathbf{s}} = -50$. We use the following cost function parameters: ordering cost $c_o = 20$, holding cost $c_h = 2$, disposal cost $c_d = 8$, backlogging cost $c_b = 100$, and lost sales cost $c_l = 100$. We set the discount factor to $\gamma = 0.95$. Using the computational setup in §2.6.2, we compute control policies for this MDP using FALP_N with $N = 50$ random basis functions and

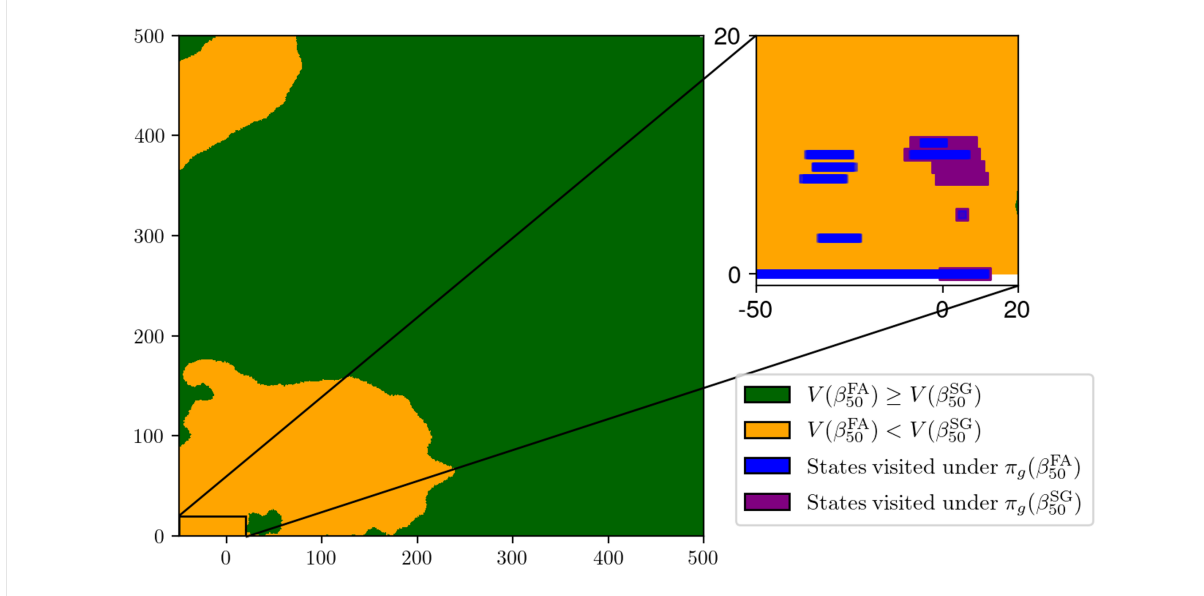
$\text{FALP}_{N,Q}^{\text{SG}}$ with $N = 50$ and $Q = 51$ (implied batch size is $B = 1$). For both these models, we use a uniform \mathbf{v} .

Figure 3: Comparison of FALP VFA $V(\beta_{50}^{\text{FA}})$ (left panel) and self-guided FALP VFA $V(\beta_{50}^{\text{SG}})$ (right panel) on a two-dimensional perishable inventory control instance.



The left panel of Figure 3 shows the VFA $V(\beta_{50}^{\text{FA}})$ and the right panel shows the VFA $V(\beta_{50}^{\text{SG}})$. Both panels share the same x-axis of s_0 and y-axis of u_1 , with the color bar encoding high (low) values as red (black). We observe that the optimal objective values of FALP_{50} and $\text{FALP}_{50}^{\text{SG}}$ are 5610 and 4832, respectively. As expected, $V(\beta_{50}^{\text{FA}})$ is closer to V^* than $V(\beta_{50}^{\text{SG}})$ under the $(1, \mathbf{v})$ -norm. Since $V(\beta_{50}^{\text{FA}})$ is a lower bound on V^* (we say lower bound because we employ constraint sampling such that our samples are very dense in the 2-D state-action space), the

Figure 4: Illustrating the impact of guiding constraints on greedy policy performance.



reddish states likely correspond to the states where V^* has very high values. This suggests that FALP_{50} improves the $(1, \nu)$ -norm by making $V(\beta_{50}^{FA})$ close to V^* at states where V^* has high values. In contrast, $\text{FALP}_{50}^{\text{SG}}$ is distinctly different and this difference affects policy performance. The optimality gaps of $\pi_g(\beta_{50}^{\text{SG}})$ and $\pi_g(\beta_{50}^{\text{FA}})$ are 4.1% and 48.2%, respectively (computed w.r.t the FALP_N lower bound).

Figure 4 shows two regions of the state space: orange states that satisfy the inequality $V(\beta_{50}^{\text{SG}}) > V(\beta_{50}^{\text{FA}})$ (which is analogous to orange states in Figure 1), while green states satisfy $V(\beta_{50}^{\text{SG}}) \leq V(\beta_{50}^{\text{FA}})$ (which is analogous to green states in Figure 1). Since both panels in Figure 3 show low values in the orange states, it is likely that V^* takes low values at these states. In the orange region, it is necessary that some of the guiding constraints with right-hand sides from

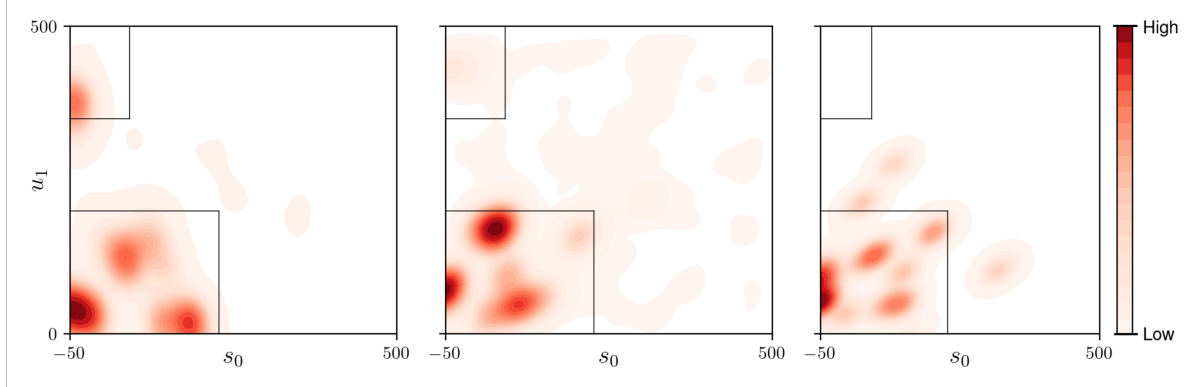
the set $\{V(\beta_q^{SG}) : q = 0, 1, \dots, 50\}$ are binding because $V(\beta_{50}^{SG})$ is closer to V^* in this region. If all of the guiding constraints are redundant, then the optimal objective values of $FALP_{50}$ and $FALP_{50}^{SG}$ must coincide, which is not the case in this example. The smaller subplot in Figure 4 zooms into the bottom-left corner of the state space and depicts the states visited under greedy policies $\pi_g(\beta_{50}^{FA})$ and $\pi_g(\beta_{50}^{SG})$ that are shown in blue and purple, respectively. Since both $V(\beta_{50}^{FA})$ and $V(\beta_{50}^{SG})$ are highly likely to lower bound V^* , it follows that $V(\beta_{50}^{SG})$ provides a better approximation of V^* at the orange states, which also suggests that $V(\beta_{50}^{SG})$ should provide a greedy policy that is better at driving the system to lower cost states under V^* in the orange region than the greedy policy based on $V(\beta_{50}^{FA})$. This provides support for the observed lower optimality gap of $\pi_g(\beta_{50}^{SG})$ compared to $\pi_g(\beta_{50}^{FA})$.

We next visualize the state-relevance distribution under self-guided FALP. Figure 5 plots the state-relevance distributions $\nu'(\beta_{40}^{SG})$, $\nu'(\beta_{45}^{SG})$, and $\nu'(\beta_{50}^{SG})$. It is reassuring to see that all these distributions are concentrated at the bottom left corner of the state space where we expect V^* to take its lowest values.

2.13.2 Analyzing the Impact of Constraint Sampling on Policy-guided FALP

As discussed in §2.6.2, we have implemented three versions of policy-guided FALP that differ only in how their constraints are sampled. Version (i) uses uniformly sampled state-action pairs to define constraints of the linear program $FALP_N[\nu^q]$ solved at iteration q of policy-guided FALP algorithm. Version (ii) constructs the constraints of $FALP_N[\nu^q]$ at iteration q using state-action pairs visited under greedy policy $\pi_g(\beta^{q-1})$ obtained in iteration $q - 1$. For the initial iteration $q = 0$, Version (ii) employs uniformly sampled state-action pairs. Version (iii) integrates both

Figure 5: Self-guided FALP state-relevance distributions $\mathbf{v}'(\beta_{40}^{\text{SG}})$ (left panel), $\mathbf{v}'(\beta_{45}^{\text{SG}})$ (middle panel), and $\mathbf{v}'(\beta_{50}^{\text{SG}})$ (right panel) on a two-dimensional perishable inventory control instance.



uniformly sampled state-action pairs in Version (i) and the pairs visited under the policy-guided FALP greedy policy in Version (ii). Unfortunately, Version (ii) results in severely poor greedy policies, so we did not report its optimality gaps.

Table VI is an extended version of Table III that reports the optimality gaps and lower-bound gaps of Versions (i) and (iii) on the five-dimensional perishable inventory control instances. The optimality-gap ranges for Version (i) and Version (iii) are 4.3%–82.3% and 7.1%–57.8%, respectively. Both versions deliver good policies on some instances and lead to poor policies on a couple of them. These results suggest that the performance of greedy policies obtained from policy-guided FALP is sensitive to the constraint sampling strategy used and the problem instance solved. The lower-bound gap ranges for Versions (i) and (iii) are respectively 0.0% to 7.8% and 0.0% to 17.1%, where we can see Version (i) produces better lower bounds than Version (iii) on five out of six instances.

Table VI: Comparison of the effect of different constraint sampling strategies on policy-guided FALP (extended version of Table III).

c_h	c_d	c_b	σ	% (UB - best LB)/(best LB)					% (Best LB - LB)/(best LB)				
				ALP ^{LNS}	FALP ₃₀₀	FALP _{300,7} ^{PG} (i)	FALP _{300,7} ^{PG} (iii)	FALP _{300,7} ^{SG}	ALP ^{LNS}	FALP ₃₀₀	FALP _{300,7} ^{PG} (i)	FALP _{300,7} ^{PG} (iii)	FALP _{300,7} ^{SG}
1	8	2	5	139.4	19.6	12.9	38.7	13.9	15.0	0.0	0.1	0.3	0.4
1	8	2	2	18.0	21.0	11.7	12.6	11.5	6.2	0.0	0.2	1.7	0.2
1	2	8	5	13.6	15.6	82.3	10.6	7.9	7.8	0.0	7.8	1.9	0.8
1	2	8	2	6.8	12.1	4.3	57.8	4.3	6.2	0.0	0.9	17.1	0.7
2	8	5	5	59.4	15.9	10.6	7.1	8.4	12.1	0.2	0.0	0.3	0.5
2	8	5	2	8.2	16.1	7.0	7.4	7.7	7.6	0.0	0.1	9.8	0.5
Average				40.9	16.7	21.5	22.4	9.0	9.1	0.0	1.5	5.2	0.5

2.13.3 Analyzing ReLU Basis Functions

In this section, we compare the performance of FALP and self-guided FALP models formulated using two different random basis function classes: ReLU bases ($\varphi(\cdot) = \max\{\cdot, 0\}$) and Fourier bases ($\varphi(\cdot) = \cos(\cdot)$). We refer to the formulation of the FALP_N model with ReLU and Fourier basis functions as ReLU FALP_N and Fourier FALP_N, respectively. Similarly, we use ReLU FALP_N^{SG} and Fourier FALP_N^{SG} when ReLU and Fourier basis functions are used to formulate FALP_N^{SG}.

Perishable Inventory Control. We apply ReLU FALP_N with $N = 300$ and $N = 600$ to our three-dimensional perishable inventory control instances studied in Table II. Table VII reports the optimality gap and lower-bound gap values for Fourier FALP₁₅₀ (also considered in Table II), ReLU FALP₃₀₀, and ReLU FALP₆₀₀. These gaps retain their definitions from Table II, except that the best lower bound (LB) in Table VII is the maximum of the lower bounds obtained from

Table VII: Comparison of ReLU FALP and Fourier FALP on the three-dimensional perishable inventory control instances ($\sigma = 2$ and $c_l = 100$).

γ	c_h	c_d	c_b	\bar{a}	% (UB - best LB)/(best LB)			% (Best LB - LB)/(best LB)		
					Fourier	ReLU		Fourier	ReLU	
					FALP ₁₅₀	FALP ₃₀₀	FALP ₆₀₀	FALP ₁₅₀	FALP ₃₀₀	FALP ₆₀₀
0.95	2	5	10	10	0.2	1.4	0.3	0.0	1.5	0.8
	2	5	10	50	6.3	66.4	7.7	0.0	12.5	16.0
	5	10	8	10	0.3	4.4	0.3	0.0	1.5	0.8
	5	10	8	50	0.1	202.2	3.9	0.0	8.6	15.6
	2	10	10	10	0.3	1.3	0.3	0.0	1.5	0.9
	2	10	10	30	0.8	41.7	1.9	0.0	4.0	4.2
0.99	2	5	10	10	0.6	35.9	0.5	0.0	1.4	0.8
	2	5	10	50	6.2	164.0	8.2	0.0	4.3	11.9
	5	10	8	10	0.3	9.0	0.5	0.0	1.4	0.8
	5	10	8	50	1.1	163.6	4.3	0.0	9.8	15.0
	2	10	10	10	0.6	35.3	0.6	0.0	1.4	0.9
	2	10	10	30	1.1	12.7	2.0	0.0	2.4	6.6
Average					1.5	61.5	2.6	0.0	4.2	6.2

these three models. Optimality gap ranges for Fourier FALP₁₅₀, ReLU FALP₃₀₀, and ReLU FALP₆₀₀ are 0.1%–6.3%, 1.3%–202.2%, and 0.3%–8.2%, respectively. We observe that Fourier FALP₁₅₀ provides near-optimal policies on all three-dimensional instances. On four out of six instances with a small state-space diameter ($\bar{a} = 10$), ReLU FALP₃₀₀ also leads to tight optimality gaps. However, on the remaining instances, especially those with a large state-space diameter ($\bar{a} = 50$), ReLU FALP₃₀₀ results in poor policies. Doubling the number of basis functions N , we observe that ReLU FALP₆₀₀ essentially closes the optimality gaps on almost all instances. Therefore, ReLU FALP₆₀₀ has comparable policy performance to Fourier FALP₁₅₀. Fourier FALP₁₅₀ has zero

lower-bound gaps across all instances which indicates it results in the best lower bounds among three models considered in Table VII. For ReLU FALP₃₀₀ and ReLU FALP₆₀₀, the lower-bound gap ranges are 1.4%–12.5% and 0.8%–16.0%, respectively. While both of these models produce excellent lower bounds on instances with small and medium state space diameters, i.e., $\bar{a} = 10$ and $\bar{a} = 30$, they lead to suboptimal lower bounds on four instances with the largest state-space diameter ($\bar{a} = 50$).

Our results in Table VII show that our FALP model with both Fourier and ReLU bases leads to very good greedy policies and lower bounds. However, to achieve comparable greedy policies from Fourier FALP_N and ReLU FALP_N, we need to use a significantly larger number of basis functions N in the latter basis function class. This behavior is particularly pronounced when dealing with challenging instances with large state space diameters. We did not apply ReLU FALP_N to our five- and ten-dimensional instances, as we expect that a very large number of ReLU basis functions would be needed to achieve near-optimal policies and lower bounds. This would require us to solve linear programs with a very large number of columns, which is computationally onerous.

Bermudan Options Pricing. We next apply ReLU basis functions to the Bermudan options pricing instances considered in §2.7. The Bermudan options pricing problem is a finite-time horizon MDP, which means that VFA-based models such as FALP and self-guided FALP need to store VFA weights for each stage. As a result, applying VFA-based methods to finite-time horizon MDPs requires significantly more memory than infinite-time horizon MDPs. Thus, unlike the perishable inventory control application where we tested ReLU bases with $N = 300$ and

Table VIII: Comparison of ReLU FALP and ReLU self-guided FALP with Fourier FALP and Fourier self-guided FALP on the Bermudan options pricing instances.

J	p^{init}	% (Best UB - LB)/(best UB)				% (UB - best UB)/(best UB)			
		Fourier		ReLU		Fourier		ReLU	
		FALP ₅₀₀	FALP _{500,6} ^{SG}	FALP ₅₀₀	FALP _{500,6} ^{SG}	FALP ₅₀₀	FALP _{500,6} ^{SG}	FALP ₅₀₀	FALP _{500,6} ^{SG}
4	90	2.1	2.0	3.8	3.8	0.3	0.0	3.8	3.8
4	100	1.9	1.9	4.2	4.2	0.0	0.0	2.3	2.3
4	110	8.0	4.9	5.4	5.4	996.7	4.0	0.0	0.0
8	90	6.7	6.6	9.0	9.0	2.9	2.5	0.0	0.0
8	100	7.9	4.3	6.7	6.7	5.9	0.4	0.0	0.0
8	110	9.5	3.1	5.8	5.6	172.5	0.0	1.1	0.5
16	90	3.7	3.7	7.7	7.2	0.1	0.0	1.8	0.2
16	100	2.5	2.4	7.8	7.3	0.2	0.0	5.1	0.9
16	110	2.4	2.1	7.4	6.7	0.2	0.0	12.3	1.9
Average		5.0	3.4	6.4	6.2	131.0	0.8	2.9	1.1

$N = 600$ on three-dimensional instances, we maintain a fixed number of samples $N = 500$ across Bermudan options pricing instances but consider four-, eight-, and sixteen-dimensional instances.

Table VIII shows the optimality gap and upper-bound gap values of four models: Fourier FALP₅₀₀ and Fourier FALP_{500,6}^{SG} (also considered in Table V), and ReLU FALP₅₀₀ and ReLU FALP_{500,6}^{SG}. Note that the best upper bound (UB) in this table is the smallest upper bound. Specifically, we use the CFA or VFA obtained from each of these models to compute upper bound on the optimal policy payoff using the information relaxation and duality approach discussed in §2.12.2 and as we did in §2.7. Optimality gap ranges for Fourier FALP₅₀₀, Fourier FALP_{500,6}^{SG}, ReLU FALP₅₀₀, and ReLU FALP_{500,6}^{SG} are 1.9%–9.5%, 1.9%–6.6%, 3.8%–9.0%, and 3.8%–

9.0%. We observe that for four-dimensional instances with $J = 4$, both ReLU FALP_{500} and ReLU $\text{FALP}_{500,6}^{\text{SG}}$ yield good policies. However, for the other instances with $J = 8$ and $J = 16$, these methods lead to weaker policies compared to Fourier $\text{FALP}_{500,6}^{\text{SG}}$. The upper-bound gap ranges for Fourier FALP_{500} , Fourier $\text{FALP}_{500,6}^{\text{SG}}$, ReLU FALP_{500} , and ReLU $\text{FALP}_{500,6}^{\text{SG}}$ are 0.0%–996.7%, 0.0%–4.0%, 0.0%–12.3%, and 0.0%–3.8%, respectively. Fourier $\text{FALP}_{500,6}^{\text{SG}}$ and ReLU $\text{FALP}_{500,6}^{\text{SG}}$ deliver excellent upper bounds for most of the instances. But they perform poorly on a few instances

Our results in Table VIII suggest that increasing the dimension of the state space J leads to an increase in the optimality gaps for ReLU FALP_{500} and ReLU $\text{FALP}_{500,6}^{\text{SG}}$, especially when $J = 16$. These observations imply that Fourier bases scale better with the state space dimension, as evidenced by the near-optimal performance of Fourier $\text{FALP}_{500,6}^{\text{SG}}$ on the largest instances with $J = 16$. This finding complements the results presented in Table VII, which indicates that more ReLU bases are required for solving instances with larger state space diameters.

2.13.4 Upper and Lower Bound Values

In this section, we report the upper and lower bound values obtained from all methods studied in Chapter 2 for both perishable inventory control and Bermudan options pricing applications. Table IX presents ALP^{LNS} and FALP_{150} bounds on the three-dimensional perishable inventory control instances studies in Table II. Table X reports ALP^{LNS} , FALP_{300} , $\text{FALP}_{300,7}^{\text{PG}}$, and $\text{FALP}_{300,7}^{\text{SG}}$ bounds on the five-dimensional perishable inventory control instances studies in Table III. Table XI presents ALP^{LNS} , FALP_{600} , FALP_{1000} , and $\text{FALP}_{600,7}^{\text{SG}}$ lower and upper bounds on the ten-dimensional perishable inventory control instances studies in Table IV. Table XII presents lower and upper bounds computed from methods LSM, ALP^{DFM} , FALP_{500} , and $\text{FALP}_{500,6}^{\text{SG}}$ on the

nine DFM instances studied in Table V. Table XIII reports lower and upper bounds that are computed from Versions (i) and (iii) of policy-guided FALP discussed in §2.13.2 and are used to compute optimality gap and lower-bound gap values in Table VI. Table XIV reports lower and upper bounds obtained from Fourier FALP_{150} , ReLU FALP_{300} , and ReLU FALP_{600} on the three-dimensional perishable inventory control instances and are used to create Table VII. Finally, Table XV reports Fourier FALP_{500} , Fourier $\text{FALP}_{500,6}^{\text{SG}}$, ReLU FALP_{500} , and ReLU $\text{FALP}_{500,6}^{\text{SG}}$ lower and upper bounds that are used to in Table VIII. We note that the lower and upper bound values reported in Tables IX–XV are the average values across 10 trials for each method and each instance.

Table IX: Lower bound and upper bounds used to compute optimality and lower-bound gaps in Table II.

γ	c_h	c_d	c_b	\bar{a}	Lower bound		Upper bound (policy cost)	
					ALP ^{LNS}	FALP ₁₅₀	ALP ^{LNS}	FALP ₁₅₀
0.95	2	5	10	10	1974.7	2043.4	2048.2	2046.1
	2	5	10	50	1895.8	1938.4	2060.9	2053.4
	5	10	8	10	2035.7	2120.8	2126.2	2125.7
	5	10	8	50	1906.5	2131.2	2132.3	2135.9
	2	10	10	10	1989.9	2062.7	2069.3	2067.8
	2	10	10	30	1988.4	2052.4	2068.3	2086.4
0.99	2	5	10	10	10883.9	11206.9	11270.4	11231.0
	2	5	10	50	10425.3	10716.1	11379.0	11315.5
	5	10	8	10	11121.1	11590.5	11629.8	11621.2
	5	10	8	50	10335.8	11522.4	11643.6	11689.6
	2	10	10	10	10943.5	11290.0	11355.9	11324.0
	2	10	10	30	10912.4	11233.3	11360.7	11400.1

Table X: Lower bound and upper bounds used to compute optimality and lower-bound gaps in Table III.

c_h	c_d	c_b	\bar{a}	Lower bound				Upper bound (policy cost)			
				ALP ^{LNS}	FALP ₃₀₀	FALP _{300,7} ^{PG}	FALP _{300,7} ^{SG}	ALP ^{LNS}	FALP ₃₀₀	FALP _{300,7} ^{PG}	FALP _{300,7} ^{SG}
1	8	2	5	1024.6	1205.2	1203.7	1200.5	2885.7	1441.8	1361.2	1373.3
1	8	2	2	958.4	1022.0	1020.2	1020.2	1205.8	1236.4	1141.5	1139.7
1	2	8	5	1125.9	1220.4	1196.8	1210.5	1386.1	1410.5	1349.8	1316.4
1	2	8	2	1016.7	1083.8	1074.0	1075.8	1157.9	1215.4	1130.1	1130.7
2	8	5	5	1153.6	1308.7	1311.7	1305.6	2090.3	1520.7	1405.2	1442.4
2	8	5	2	1036.5	1122.1	1120.7	1116.1	1214.6	1302.7	1200.9	1208.7

Table XI: Lower bound and upper bounds used to compute optimality and lower-bound gaps in Table IV.

c_h	c_d	c_b	\bar{a}	Lower bound				Upper bound (policy cost)			
				ALP ^{LNS}	FALP ₆₀₀	FALP ₁₀₀₀	FALP _{600,7} ^{SG}	ALP ^{LNS}	FALP ₆₀₀	FALP ₁₀₀₀	FALP _{600,7} ^{SG}
1	8	2	5	901.9	1220.0	1215.4	1231.0	1792.5	1391.4	1636.7	1322.5
1	8	2	2	880.4	1065.1	1074.3	1089.4	1563.9	1156.2	1248.3	1141.1
1	2	8	5	903.6	1173.1	1177.3	1190.4	2500.0	1326.2	1570.5	1274.8
1	2	8	2	878.7	1054.0	1054.5	1070.0	1587.2	1144.5	1183.7	1124.6
2	8	5	5	1025.7	1471.3	1477.3	1493.8	2141.7	1710.3	2067.5	1614.8
2	8	5	2	981.9	1276.1	1283.8	1310.7	1427.6	1430.2	1493.4	1396.1

Table XII: Lower bound and upper bounds used to compute optimality and lower-bound gaps in Table V.

J	p^{init}	Lower bound (policy payoff)				Upper bound			
		LSM	ALP ^{DFM}	FALP ₅₀₀	FALP _{600,7} ^{SG}	LSM	ALP ^{DFM}	FALP ₅₀₀	FALP _{600,7} ^{SG}
4	90	33.17	33.83	35.20	35.24	35.52	40.24	36.07	35.97
4	100	41.46	41.47	43.43	43.44	44.57	47.60	44.29	44.28
4	110	47.77	47.06	46.85	48.42	51.13	52.45	558.35	52.95
8	90	43.83	43.91	44.85	44.88	46.74	43.91	49.45	49.27
8	100	49.92	49.16	48.74	50.63	52.85	53.22	56.03	53.16
8	110	53.41	52.03	50.30	53.91	55.91	55.68	151.55	55.61
16	90	50.62	49.86	51.48	51.48	53.25	53.22	53.49	53.44
16	100	53.60	52.43	54.18	54.23	55.84	55.49	55.68	55.59
16	110	55.25	53.90	55.46	55.65	57.18	655.99	56.95	56.82

Table XIII: Lower bound and upper bounds used to compute optimality and lower-bound gaps in Table VI.

c_h	c_d	c_b	\bar{a}	Lower bound		Upper bound (policy cost)	
				Version (i)	Version (iii)	Version (i)	Version (iii)
1	8	2	5	1203.7	1201.4	1361.2	1671.7
1	8	2	2	1020.2	1004.7	1141.5	1151.3
1	2	8	5	1125.0	1196.8	2224.5	1349.8
1	2	8	2	1074.0	898.4	1130.1	1710.1
2	8	5	5	1311.7	1307.7	1450.5	1405.2
2	8	5	2	1120.7	1012.1	1200.9	1205.4

Table XIV: Lower bound and upper bounds used to compute optimality and lower-bound gaps in Table VII.

γ	c_h	c_d	c_b	\bar{a}	Lower bound			Upper bound (policy cost)		
					Fourier	ReLU		Fourier	ReLU	
					FALP ₁₅₀	FALP ₃₀₀	FALP ₆₀₀	FALP ₁₅₀	FALP ₃₀₀	FALP ₆₀₀
0.95	2	5	10	10	2043.4	2012.4	2026.5	2048.2	2071.9	2049.4
	2	5	10	50	1938.4	1696.9	1629.0	2060.9	3225.9	2087.4
	5	10	8	10	2120.8	2089.3	2103.1	2126.2	2213.7	2127.8
	5	10	8	50	2131.2	1948.6	1798.2	2132.3	6440.6	2214.7
	2	10	10	10	2062.7	2031.7	2043.9	2069.3	2090.2	2069.8
	2	10	10	30	2052.4	1970.6	1966.4	2068.3	2907.7	2091.2
0.99	2	5	10	10	11206.9	1153.5	11112.8	11270.4	15235.1	11260.6
	2	5	10	50	10716.1	10250.7	9445.1	11379.0	28286.6	11596.5
	5	10	8	10	11590.5	11429.1	11497.2	11629.8	12631.0	11654.0
	5	10	8	50	11522.4	10392.1	9788.5	11643.6	30374.5	12022.8
	2	10	10	10	11290.0	11127.6	11193.8	11355.9	15277.4	11355.1
	2	10	10	30	11233.3	10969.2	10496.4	11360.7	13654.6	11459.2

Table XV: Lower bound and upper bounds used to compute optimality and lower-bound gaps in Table VIII.

J	p^{init}	Lower bound (policy payoff)				Upper bound			
		Fourier		ReLU		Fourier		ReLU	
		FALP ₅₀₀	FALP _{500,6} ^{SG}	FALP ₅₀₀	FALP _{500,6} ^{SG}	FALP ₅₀₀	FALP _{500,6} ^{SG}	FALP ₅₀₀	FALP _{500,6} ^{SG}
4	90	35.2	35.2	34.6	34.6	36.1	36.0	37.3	37.3
4	100	43.4	43.4	42.4	42.4	44.3	44.3	45.3	45.3
4	110	46.8	48.4	48.1	48.1	558.4	52.9	50.9	50.9
8	90	44.8	44.9	43.7	43.7	49.5	49.3	48.1	48.1
8	100	48.7	50.6	49.4	49.4	56.0	53.2	52.9	52.9
8	110	50.3	53.9	52.4	52.5	151.5	55.6	56.2	55.9
16	90	51.5	51.5	49.3	49.6	53.5	53.4	54.4	53.5
16	100	54.2	54.2	51.2	51.5	55.7	55.6	58.4	56.1
16	110	55.5	55.6	52.6	53.0	56.9	56.8	63.8	57.9

CHAPTER 3

RANDOMIZED MULTI-SHOT APPROXIMATION OF AVERAGE COST MARKOV DECISION PROCESSES

(Co-authors: Parshan Pakiman and Selva Nadarajah)

Abstract

Approximate linear programming is a well-established approach for computing control policies for average-cost Markov decision processes (MDPs). This method approximates the MDP bias function using a weighted sum of basis functions. It solves an approximate linear program (ALP) that maximizes a lower bound on the optimal policy cost and produces optimal weight for each basis function. When rich basis functions are selected, the optimal objective value of ALP is a near-optimal lower bound. However, ALP can result in weak bias function approximations (BFAs) and control policies even if bases are rich because ALP formulation does not include any measure of BFA error in its objective function. We propose a new approximate linear programming approach to tackle the challenges of selecting rich basis functions and modifying ALP formulation. We combine a known two-phase ALP model studied in the literature with a randomized multi-shot approximation mechanism recently proposed for discounted-cost MDPs. Our method thus has two phases. First, it defines BFA in ALP using universal random basis functions to mitigate the impact of poor basis functions on ALP lower bound quality. Second, it solves a sequence of ALP models that iteratively refine their formulation using previously

computed BFAs in this sequence to mitigate the impact of poor ALP formulation on policy performance. We establish a probabilistic convergence rate showing our lower bound approaches to the optimal policy cost. In addition, we show that our sequence of ALP models taking multiple shots at randomly approximating MDP bias function results in policies with improving worst-case performance. We apply our method to two inventory management problems, resulting in near-optimal lower bounds and effective control policies.

3.1 Introduction

Average-cost Markov decision processes (MDPs; see, e.g., Chapter 5 of [Hernández-Lerma and Lasserre 1996](#)) provide mathematical models for sequential decision-making problems such as inventory control, capacity allocation, queuing, and hospital management ([Mahadevan 1996](#), [Adelman and Klabjan 2005](#), [De Farias and Van Roy 2006](#), [Adelman and Klabjan 2012](#), [Adelman and Mersereau 2013](#), [Dai and Shi 2019](#)). These MDPs usually feature high-dimensional state and action spaces, making exact solutions intractable.

Approximate linear programming ([Schweitzer and Seidmann 1985](#), [De Farias and Van Roy 2003](#)) is a well-established model-based reinforcement learning method for approximating large-scale average-cost MDPs. This method relies on (i) approximating the MDP bias function using a linear combination of so-called basis functions defined over the MDP state space and (ii) solving an approximate linear program (ALP) to obtain the optimal weight of each basis function in this linear combination. ALP has one decision variable for each basis function weight, in addition to a variable representing a lower bound on the optimal policy cost. ALP maximizes this lower bound and yields optimal weights of basis functions. It is known that if basis functions

are powerful enough to approximate the MDP bias function closely, the ALP lower bound is arbitrarily close to the optimal policy cost. However, because the ALP objective function does not include any BFA error term, ALP BFA and policy qualities can be highly sub-optimal even if basis functions are powerful.

The studies below explored modifying ALP reformulations by integrating a BFA error term into its objective function, thereby improving the qualities of ALP BFA and policies. These reformulations all require a predetermined set of basis functions as input.

- [De Farias and Van Roy \(2002\)](#) proposed a two-phase ALP model. In the first phase, it solves the original ALP formulation to obtain a lower bound on the optimal policy cost. In the second phase, it uses the lower bound value from the first phase and solves a different ALP model that minimizes a surrogate loss for BFA error. This loss is the difference between BFA and an upper bound on the MDP bias function that is weighted based on a state-relevance distribution, which assigns weights to different regions of the state space. This distribution naturally arises in ALP formulations for discounted-cost MDPs, as we saw in Chapter 2. Therefore, the second-phase ALP can be seen as artificially adding the state-relevance distribution to the average-cost ALP formulation in order to control BFA quality. Although this surrogate loss enables controlling BFA quality in the second-phase ALP, a capability lacking in the first-phase model, it may not accurately capture BFA error. In particular, when the first-phase ALP produces a weak lower bound, the upper bound on the MDP bias function arising in the definition of the surrogate loss can be weak and thus lead to a poor BFA in the second-phase ALP. In other words, minimizing this

surrogate loss may not directly translate into minimizing the true loss between BFA and the MDP bias function.

- [De Farias and Van Roy \(2006\)](#) proposed a cost-shaping ALP formulation to control BFA quality. This formulation involves constructing a perturbed MDP with a transition kernel obtained from a convex combination of the original MDP transition kernel and a so-called restart distribution. The cost-shaping ALP formulation is obtained from the original ALP model written for the perturbed MDP with an additional slack variable allowing for constraints violation. The amount of such violation is then managed by adding a penalty term in the cost-shaping ALP objective function. The authors showed that the performance of the greedy policy obtained from the cost-shaping approach is proportional to the least attainable BFA error for a given set of basis functions. However, to deploy this approach, one needs to specify multiple parameters, including a distribution that determines the amount of constraint violation at each state, a penalty factor for constraint violation in the ALP objective, the restart distribution, and basis functions. The authors stated that automatically choosing these parameters is an open question. To the best of our knowledge, there have been no numerical experiments conducted to assess the performance of this approach, possibly due to the computational difficulties associated with tuning its parameters.
- [Veatch \(2013\)](#) builds on the work by [De Farias and Van Roy \(2006\)](#) and utilizes the “smoothed” ALP formulation in [Desai et al. \(2012a\)](#) to design an ALP model for average-cost MDPs. In comparison to the original ALP formulation for average-cost MDPs, the model in [Veatch \(2013\)](#) features a modified objective function with a surrogate loss for

BFA error and a penalty term for constraint violation. Unfortunately, this formulation relies on idealized information based on the MDP optimal policy, limiting the use of this method. Moreover, the numerical performance of this method has not been explored yet.

We develop an ALP method that integrates the two-phase ALP model in [De Farias and Van Roy \(2002\)](#) with the randomized multi-shot approximation mechanism proposed in Chapter 2 for discounted-cost MDPs and relies on random basis functions ([Rahimi and Recht 2008](#), [Rahimi and Recht 2009](#)). Our method is based on two randomized ALP models, namely bound-focused ALP (BALP) and policy-focused ALP (PALP), which are analogous to the first and second phase ALPs in [De Farias and Van Roy \(2002\)](#). BALP uses a batch of random basis functions sampled from a readily available distribution, i.e., a single-shot approximation. We show that BALP lower bound converges to the optimal policy cost with a high probability. Upon solving BALP, our method solves a sequence of PALP models that utilize BALP lower bound and BFA. These PALPs, which are analogous to self-guided ALPs in Chapter 2, have an increasing number of random basis functions that are sampled iteratively in multiple batches. We link BFAs in this sequence of PALP models using “guiding constraints”. They ensure the error of these BFAs and an upper bound on the cost of greedy policies with respect to these PALP BFAs are weakly improving.

In contrast to the approach outlined in [De Farias and Van Roy \(2002\)](#), where a single second-phase ALP is solved, our method solves multiple second-phase ALPs, specifically PALPs. Additionally, our guiding constraints in the context of average-cost MDPs are added to the second-phase ALP formulation in [De Farias and Van Roy \(2002\)](#), whereas our guiding constraints

in the context of discounted-cost MDPs are added directly to the original ALP model. Therefore, our algorithms and their analyses in this chapter are fundamentally different from those in [De Farias and Van Roy \(2002\)](#) and in Chapter 2.

3.1.1 Contributions

- **Model.** We propose new average-cost ALP models, BALP and PALP, that address issues associated with the original ALP formulation as well as the two-phase ALP method in [De Farias and Van Roy \(2002\)](#). These models extend our randomized multi-shot approximation of discounted-cost MDPs in Chapter 2 to average-cost MDPs. This extension is non-trivial because if we directly add guiding constraints to the original average-cost ALP model, as done in Chapter 2 for discounted-cost MDPs, these constraints become redundant and do not guarantee improving worst-case policy performance improvement.
- **Theory.** We show that the gap between optimal policy cost and BALP lower bound converges to zero at a dimension-free rate of one divided by the square root of the number of random bases used to formulate BALP. We also develop an upper bound on the performance of greedy policies obtained from PALP. To this end, we first generalize a previously known bound in [De Farias and Van Roy \(2002\)](#) for finite-state MDPs to the continuous-state MDPs. Then, using this performance bound, we show that our guiding constraints weakly improve the cost of greedy policies based on PALP models as more basis functions are sampled. Moreover, our theoretical findings for BALP and PALP provide a new insight: the number of random basis function samples required to obtain a tight lower bound from BALP can be significantly smaller than the number of samples needed

to achieve near-optimal BFAs from PALP. In other words, learning accurate lower bounds can be much easier than learning accurate BFAs.

- **Numerical experiments.** We apply our method to the generalized joint replenishment problem studied in [Adelman and Klabjan \(2012\)](#) and an average-cost version of the perishable inventory control (PIC) problem studied in Chapter 2. [Adelman and Klabjan \(2012\)](#) showed that affine BFAs provide high-quality greedy policies on their instances without holding cost, but computing tight lower bounds requires using more sophisticated BFAs based on ridge-type basis functions. We benchmark BALP lower bounds against the ones obtained from the algorithm in [Adelman and Klabjan \(2012\)](#). Our application-agnostic BALP method formulated with Stump bases results in near-optimal lower bounds comparable to those obtained from the application-specific benchmark in [Adelman and Klabjan \(2012\)](#) on the instances without holding cost. As shown in Chapter 2, computing near-optimal policies for high-dimensional discounted-cost PIC instances is challenging. We also observe that this is, in fact, the case when considering an average-cost version of these problem instances. We apply our randomized multi-shot approximation method that solves PALPs formulated using Fourier basis functions to these instances and find that it provides near-optimal policies. We demonstrate that PALP greedy policies are substantially better than the ones from BALP and its modified version. Moreover, we show that PALP significantly outperforms several benchmarks. Our numerical results contribute to the limited literature evaluating the numerical performance of ALP models for large-scale average-cost MDPs.

- **Solution of BALP and PALP.** BALP and PALP are semi-infinite linear programs.

These models can be solved using constraint sampling (De Farias and Van Roy 2004, Calafiore and Campi 2006). For generalized joint replenishment (GJR) problem instances studied in Adelman and Klabjan (2012), constraint sampling may not provide good approximations of the original semi-infinite linear programs because of action space high-dimensionality. In addition, the greedy policy optimization method for GJR cannot be approached via discretization, as was done in Chapter 2, because GJR has a high-dimensional action space, unlike applications in Chapter 2. We thus show how BALP and PALP formulated using specific classes of random basis functions can be solved using constraint generation. Specifically, if the random basis function class used to formulate these models is piecewise constant (e.g., Stump bases) or piecewise linear (e.g., ReLU bases) and the MDP cost function and transition kernel have structure, we can use the constraint generation method to solve BALP and PALP. For the GJR problem, because MDP components have linear structures, we can reformulate the separation problem in the constraint generation method and the greedy policy optimization problem as mixed-integer linear programs when Stump basis functions are used to formulate BALP. Constraint generation in conjunction with random basis functions is new in the ALP literature.

3.1.2 Related work

Pakiman et al. (2020), which is the paper underpinning Chapter 2, applies random basis functions to discounted-cost ALP and proposes a “self-guiding” mechanism to mitigate the effect of state-relevance distribution choice on greedy policy performance, where this distribution is

a parameter appearing in the discounted-cost ALP formulation (see Chapter 2). The main difference between our work and [Pakiman et al. \(2020\)](#) is that there is no value in adding guiding constraints proposed by [Pakiman et al. \(2020\)](#) to the standard ALP formulation for average-cost MDPs (i.e., these constraints become redundant). Instead, we demonstrate that by adding analogous guiding constraints to a second-phase ALP model based on [De Farias and Van Roy \(2002\)](#), we can ensure a worst-case measure of greedy policy is improving. Therefore, our work extends the results in [Pakiman et al. \(2020\)](#) to average-cost MDPs in a non-trivial manner. In addition, we show that accessing an approximation of the MDP bias function over a possibly small region of the state space suffices to obtain tight lower bounds, but this is not true if we want to ensure a near-optimal greedy policy. This result is new relative to our findings in [Pakiman et al. \(2020\)](#).

The seminal work by [Klabjan and Adelman \(2007\)](#) proposes a convergent algorithm based on primal-dual linear programs that produce basis functions for average-cost semi-MDPs, albeit requiring the solution of challenging nonlinear programs. [Adelman and Klabjan \(2012\)](#) leverages the structure of the GJR problem and develops a tractable algorithm to perform these primal-dual steps, and they show that this method delivers excellent policies and lower bounds for this application. Our work is similar to both of these papers in terms of dynamically updating basis functions. The main difference is that our basis functions are sampled inexpensively from known distributions and do not require optimization or domain knowledge. [Adelman and Klabjan \(2012\)](#) use information based on flow-balance constraints in dual ALP and problem

structure to generate bases, but we leverage the primal ALP formulation. Moreover, we focus on MDPs, but [Adelman and Klabjan \(2012\)](#) focus on deterministic semi-MDPs.

The structure of the paper is as follows. In §3.2, we provide background material on MDPs. In §3.3 and §3.4, we discuss BALP and PALP models, respectively. In §3.5, we present our main algorithm that combines BALP and PALP models and explain how to solve them with constraint sampling and constraint generation methods. In §3.6 and §3.7, we present our numerical experiments on GJR and PIC problems, respectively. We conclude in §3.8. All proofs and supporting materials are available in §§3.9-3.10.

3.2 Markov Decision Processes

We consider an MDP with the state space of $\mathcal{S} \subseteq \mathbb{R}^d$ and the action space of $\mathcal{A} \subseteq \mathbb{R}^{d_a}$. We denote by $\mathcal{A}_s \subseteq \mathcal{A}$ the set of feasible actions from state $s \in \mathcal{S}$. Taking action $\mathbf{a} \in \mathcal{A}_s$ in state $s \in \mathcal{S}$ results in the immediate cost of $c(s, \mathbf{a})$ and in the transition of the system to the next state s' with the probability of $P(s'|s, \mathbf{a})$. The expected average cost per stage of a given (deterministic and stationary) policy $\pi : \mathcal{S} \mapsto \mathcal{A}$ from an initial state $s = s_0 \in \mathcal{S}$ is:

$$\text{AC}(s; \pi) := \limsup_{n \rightarrow \infty} \mathbb{E}_s^\pi \left[\frac{1}{n} \sum_{t=0}^{n-1} c(s_t, \pi(s_t)) \right], \quad (3.1)$$

where $\{(s_n, \pi(s_n)) : n = 0, 1, \dots\}$ is an infinite sequence of states and actions under policy π when starting from initial state $s_0 = s$. For each policy π and initial state s , the expectation operator $\mathbb{E}_s^\pi[\cdot]$ over infinite sequences of states and actions is well-defined by the Ionescu-Tulcea theorem (see, e.g., Proposition C.10 in [Hernández-Lerma and Lasserre 1996](#)).

The goal of the system is to find an optimal control policy π^* with the minimum expected long-run average cost when starting from an initial state $s \in \mathcal{S}$. Formally, this goal requires solving the following policy optimization problem:

$$\inf_{\pi: \mathcal{S} \rightarrow \mathcal{A}} \text{AC}(s; \pi). \quad (3.2)$$

When the Markov process defined by the stochastic kernel $P(\cdot | \cdot, \pi(\cdot))$ over \mathcal{S} is positive Harris-recurrent for every policy π (see, e.g., Theorem 2.5 in [Saldi et al. 2017](#)), the average cost $\text{AC}(s; \pi)$ is a constant independent of the initial state s . Thus, under this condition, if a policy is optimal to (3.2) in a specific state s , then it will also be optimal in all other states. Harris-recurrence means that every state $s \in \mathcal{S}$ can be reached in a finite number of transitions when starting from an arbitrary initial state s and taking actions according to π . The positiveness means that the Markov process of states under π admits a unique invariant probability measure $\mu(\cdot; \pi)$ that satisfies $\int_{\mathcal{S}} \mu(s; \pi) \, ds = 1$ and

$$\int_{\mathcal{S}} P(s' \in \mathcal{X} | s, \pi(s)) \mu(s; \pi) \, ds = \mu(\mathcal{X}; \pi), \quad \forall \mathcal{X} \subseteq \mathcal{S}. \quad (3.3)$$

In the context of MDPs with a finite state space, when a so-called weak accessibility assumption holds, the optimal average cost is the same for all initial states (see Proposition 5.2.3 in [Bertsekas 2015](#)). This assumption is similar to the positive Harris-recurrence assumption in our setting for MDPs with continuous state space.

Assumption 6 requires the MDP state and action spaces to be compact continuous sets and the Markov process generated by every policy to be positive Harris-recurrent, which is widely used in the literature for analyzing average-cost MDPs. For example, please see Assumption 2.3 in Gordienko and Hernández-Lerma (1995) and Theorem 3.3 in Vega-Amaya (2003).

Assumption 6 *State space $\mathcal{S} \subseteq \mathbb{R}^d$ and each feasible action set $\mathcal{A}_s \subseteq \mathbb{R}^{d_a}$ are compact continuous sets. Moreover, for each policy π , the Markov process defined by the transition kernel $P(\cdot|\cdot, \pi(\cdot))$ over the state space is positive Harris-recurrent.*

Under Assumption 6, the average cost of a policy π is the constant $\eta^\pi \in \mathbb{R}$ that satisfies the following identities (as shown in Theorem 2.5 of Saldi et al. 2017):

$$\eta^\pi = \text{AC}(\hat{s}; \pi) = \int_{\mathcal{S}} c(s, \pi(s)) \mu(s; \pi) \, ds, \quad \forall \hat{s} \in \mathcal{S}.$$

Therefore, the average cost minimization problem (3.2) can be written as $\inf_{\pi} \eta^\pi$ that finds an optimal π^* (if exists) with the smallest cost η^{π^*} .

The cost minimization problem (3.2) over the set of all policies for finding π^* is related to the following average cost optimality equation:

$$u(s) = \inf_{a \in \mathcal{A}} \{c(s, a) - \eta + \mathbb{E}[u(s')|s, a]\}, \quad \forall s \in \mathcal{S}, \quad (3.4)$$

where constant $\eta \in \mathbb{R}$ and function $\mathbf{u} : \mathcal{S} \mapsto \mathbb{R}$ are variables in (3.4). In the following assumption, we require both an optimal policy π^* that solves (3.2) and a pair (η^*, \mathbf{u}^*) that solves (3.4) exist. We also clarify how (3.2) and (3.4) are linked.

Assumption 7 *There exists triplet $(\pi^*, \eta^*, \mathbf{u}^*)$ such that (i) the optimal policy π^* solves (3.2), (ii) pair (η^*, \mathbf{u}^*) solves (3.4), (iii) function $\mathbf{u}^* : \mathcal{S} \mapsto \mathbb{R}$ is continuous over \mathcal{S} , and (iv) following identities hold for all $s \in \mathcal{S}$:*

$$\eta^* = \eta^{\pi^*} = \text{AC}(s; \pi^*) = \inf_{\pi} \text{AC}(s; \pi), \text{ and } \mathbf{u}^*(s) = \mathbf{c}(s, \pi^*(s)) - \eta^* + \mathbb{E}[\mathbf{u}^*(s') | s, \pi^*(s)].$$

Assumption 7 states that (i) an optimal policy π^* solving (3.2) exists, i.e., the “inf” in (3.2) can be replaced by a “min”; (ii) solution (η^*, \mathbf{u}^*) to the optimality equation (3.4) exists, where $\mathbf{u}^* : \mathcal{S} \mapsto \mathbb{R}$ is known as the *MDP bias function*; (iii) the MDP bias function is continuous over \mathcal{S} ; (iv) η^* obtained from optimality equation is the cost of the optimal policy cost, i.e., $\eta^{\pi^*} = \text{AC}(s; \pi^*)$, and the optimal policy π^* selects action $\pi^*(s)$ that minimizes the expression given in (3.4) for every state s . Another way to express this final property is that π^* is greedy with respect to \mathbf{u}^* . Specifically, define the greedy policy $\pi_g(s; \mathbf{u})$ with respect to $\mathbf{u} : \mathcal{S} \mapsto \mathbb{R}$ at state $s \in \mathcal{S}$ as,

$$\pi_g(s; \mathbf{u}) := \arg \min_{\mathbf{a} \in \mathcal{A}} \{\mathbf{c}(s, \mathbf{a}) + \mathbb{E}[\mathbf{u}(s') | s, \mathbf{a}]\}. \quad (3.5)$$

The above objective function is based on optimality equation (3.4) without including the constant term η because removing it from the minimization does not change the optimal action obtained in optimization problem (3.5). Assumption 7 (iv) guarantees that the identity

$\pi^*(s) = \pi_g(s; \mathbf{u}^*)$ holds for all $s \in \mathcal{S}$, meaning π^* is greedy with respect to \mathbf{u}^* . In other words, the policy optimization problem (3.2) is equivalent to solving optimality equation (3.4) to obtain pair (η^*, \mathbf{u}^*) and then plug in \mathbf{u}^* into the greedy optimization problem (3.5) to recover π^* . A body of work studies conditions on the MDP primitives under which Assumption 7 holds. For example, see Theorem 5.5.4 of [Hernández-Lerma and Lasserre \(1996\)](#) and Theorem 2.5 of [Saldi et al. \(2017\)](#). See also [Klabjan and Adelman \(2006\)](#) for analogous results for semi-MDPs. In §3.9, we discuss a set of such conditions for the completeness.

Although Assumption 7 ensures existence of triplet $(\pi^*, \eta^*, \mathbf{u}^*)$, it is known that every shift of \mathbf{u}^* by a constant $\mathbf{c} \in \mathbb{R}$ results in the updated pair $(\eta^*, \mathbf{u}^* + \mathbf{c})$ that is also a solution to (3.4). Theorem 10.3.7 of [Hernández-Lerma and Lasserre \(1999\)](#) states that all pairs satisfying optimality equation (3.4) has the form of $(\eta^*, \mathbf{u}^* + \mathbf{c})$ for some constant \mathbf{c} . For a given solution (η^*, \mathbf{u}^*) to (3.4), if we define $\mathbf{c} = -\mathbf{u}^*(\bar{s})$ for a fixed reference state $\bar{s} \in \mathcal{S}$, then $(\eta^*, \mathbf{u}^* - \mathbf{u}^*(\bar{s}))$ becomes the unique solution to (3.4) that satisfies $\mathbf{u}^*(s) - \mathbf{u}^*(\bar{s}) = 0$ at $s = \bar{s}$. Hereafter, notation (η^*, \mathbf{u}^*) thus refers to the unique solution of the optimality equation, where the MDP bias function \mathbf{u}^* satisfies condition $\mathbf{u}^*(s) = 0$ at the reference state $s = \bar{s}$. We define \mathcal{U} as the collection of all continuous functions $\mathbf{u} : \mathcal{S} \mapsto \mathbb{R}$ that satisfy condition $\mathbf{u}^*(\bar{s}) = 0$. Thus, \mathbf{u}^* belong to \mathcal{U} .

3.3 Bound-Focused Programs

In §3.3.1, we present a linear programming reformulation of the optimality equation. In §3.3.2, we present an alternative reformulation based on random basis functions. In §3.3.3, we construct BALP using the reformulation in §3.3.2, and we develop a probabilistic error bound for BALP.

3.3.1 Bound-Focused Exact Linear Program

Linear programming provides a well-established approach to reformulate optimality equation (3.4). The computation of the optimal average cost η^* can be reformulated according to the following bound-focused exact linear program (BELP):

$$\begin{aligned} \sup_{(\eta, \mathbf{u}) \in \mathbb{R} \times \mathcal{U}} \quad & \eta \\ & \eta + \mathbf{u}(s) - \mathbb{E}[\mathbf{u}(s')|s, \mathbf{a}] \leq c(s, \mathbf{a}) \quad \forall (s, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}_s. \end{aligned} \quad (3.6)$$

BELP has a decision variable $\mathbf{u}(s)$ per state $s \in \mathcal{S}$, where $\mathbf{u} \in \mathcal{U}$, and an additional variable $\eta \in \mathbb{R}$. It also has a constraint for each state-action pair. Therefore, BELP is an infinite-dimensional linear program. To better understand BELP and the use of the term “bound-focused”, consider the following proposition.

Proposition 14 *Every BELP feasible solution (η, \mathbf{u}) satisfies $\eta \leq \eta^*$, and pair (η^*, \mathbf{u}^*) is an optimal solution to BELP.*

Proposition 14 shows that BELP maximizes a lower bound η on η^* , suggesting the name “bound-focused”. Also, it shows that pair (η^*, \mathbf{u}^*) solving (3.4) is an optimal solution to BELP. This means that BELP is an exact reformulation of optimality equation (3.4), and the “sup” in BELP can be replaced by “max” because BELP attains its optimal solution (η^*, \mathbf{u}^*) . Moreover, this proposition suggests that BELP is equivalent to the regression problem

$$\min_{(\eta, \mathbf{u})} \{|\eta - \eta^*| : (\eta, \mathbf{u}) \in \mathbb{R} \times \mathcal{U} \text{ satisfies (3.6)}\},$$

that minimizes the absolute deviation between η and the optimal cost η^* . Proposition 14, which suggests this regression-based reformulation of BLP, extends the results in Lemma 1 of De Farias and Van Roy (2002) to the MDPs with continuous state space. An important implication of this proposition is that if one can solve BLP, then the optimal cost η^* can be recovered from the optimal objective value of this program, but solving BLP to obtain η^* is, unfortunately, intractable because this program is infinite-dimensional. In the following section, we use random basis functions to derive a BLP reformulation that enables closely approximating it.

3.3.2 Bound-Focused Feature-based Exact Program

Proposition 14 suggests that BLP can be seen as a non-parametric regression model to find η^* . This model is non-parametric because it has decision variables belonging to the non-parametric set of all continuous functions $u \in \mathcal{U}$ satisfying $u(\bar{s}) = 0$. We discuss an alternative representation of the elements in \mathcal{U} based on random basis functions that allow us to develop an exact but parametric reformulation of BLP. The resulting parametric model enables closely approximating η^* .

Random basis functions are a popular tool in Machine Learning for approximating functions and tackling supervised learning problems. Let $s \in \mathbb{R}^d$ be a state vector, $\theta \in \mathbb{R}^{d+1}$ be a parameter vector that belongs to a parameter space $\Theta \subseteq \mathbb{R}^{d+1}$, and ρ be a probability distribution over Θ . Given $\theta = (\theta_0, \theta_1, \dots, \theta_d)$ sampled from ρ , we can define the random linear feature map $\theta_0 + \sum_{i=1}^d s_i \theta_i$ that is obtained by taking the inner product of $(1, s)$ and θ . A key benefit of random features is that θ does not need to be chosen or optimized. Instead, new feature maps

can be generated inexpensively by sampling θ from ρ . A nonlinear version of this feature map can be constructed if we apply nonlinear (activation) function $\varphi : \mathbb{R} \mapsto \mathbb{R}$ to the inner product $\theta_0 + \sum_{i=1}^d s_i \theta_i$. For example, random Fourier feature maps are defined based on $\varphi(\cdot) = \cos(\cdot)$. A random Fourier feature maps pair (s, θ) to the value of $\cos(\theta_0 + \sum_{i=1}^d s_i \theta_i)$. Random Fourier features have sampling distribution ρ defined over $\Theta = \mathbb{R}^{d+1}$ for which θ_0 is sampled from a uniform distribution over interval $[-\pi, \pi]$ and each θ_i is drawn from a standard normal distribution with the standard deviation of $c_\rho > 0$, which is a tunable parameter.

In the ALP literature, a basis function is a mapping from MDP state space to the real line. Therefore, we can interpret mapping $(s, \theta) \mapsto \varphi(\theta_0 + \sum_{i=1}^d s_i \theta_i)$ as a random basis function with parameter θ . Other examples of random bases are Stump and ReLU basis functions specified by piecewise constant and piecewise linear functions $\varphi(\cdot) = \text{sgn}\{\cdot, 0\}$ and $\varphi(\cdot) = \max\{\cdot, 0\}$, respectively. The signum function $\text{sgn}\{a, 0\}$ evaluates to -1 , 0 , or 1 if a is negative, zero, and positive, respectively. The parameter θ for ReLU bases can be sampled from a uniform distribution over a unit sphere in \mathbb{R}^{d+1} . For Stump basis functions, θ_0 is sampled from a uniform distribution with support over an interval $[-c_\rho, c_\rho]$, where $c_\rho > 0$ is a tunable parameter, and the remaining elements of θ are sampled from a uniform distribution defined on the discrete set $\{e^1, \dots, e^d\}$, where e^i for $i \in \{1, 2, \dots, d\}$ is the d -dimensional unit vector with 1 in the i -th coordinate and zero elsewhere. Parameter c_ρ needs to be chosen such that $[-c_\rho, c_\rho]^d \supseteq \mathcal{S}$ holds.

Random basis functions provide parametric representations of functions in the non-parametric set \mathcal{U} . Given random basis functions identified by (φ, ρ) , consider an integrable weighting function $\beta : \Theta \mapsto \mathbb{R}$ and define its $(2, \rho)$ -norm as follows:

$$\|\beta/\rho\|_{2,\rho} := \int_{\Theta} \left(\frac{\beta(\theta)}{\rho(\theta)} \right)^2 \rho(d\theta) = \int_{\Theta} \frac{(\beta(\theta))^2}{\rho(\theta)} d\theta.$$

Given β , define function $u(\cdot; \beta) : \mathcal{S} \mapsto \mathbb{R}$ parameterized by $\beta(\theta)$ as follows:

$$u(s; \beta) := \int_{\Theta} \beta(\theta) \varphi(s; \theta) d\theta, \quad (3.7)$$

and further let \mathcal{R} be the class of all continuous functions in \mathcal{U} admitting representation (3.7) and having a finite $(2, \rho)$ -norm, i.e.,

$$\mathcal{R} := \left\{ u \in \mathcal{U} \mid \exists \beta \text{ s.t. } u(\cdot) = u(\cdot; \beta), u(\bar{s}; \beta) = 0, \|\beta/\rho\|_{2,\rho} < \infty \right\}.$$

When random bases have a “universality” property, \mathcal{R} becomes a dense subset of \mathcal{U} . That is, every non-parametric function in \mathcal{U} can be approximated closely using a function in \mathcal{R} that is parameterized by $\beta(\theta)$. Formally, if random basis function identified by (φ, ρ) is universal, then for each $u \in \mathcal{U}$ and $\varepsilon > 0$, there is a $u(\beta) \in \mathcal{R}$ such that $\|u - u(\beta)\|_{\infty} := \sup_s |u(s) - u(s; \beta)| \leq \varepsilon$.

Assumption 8 *Random basis function φ is universal, and its sampling distribution ρ has a finite second moment. Also, φ has a Lipschitz constant $L > 0$ and satisfies $\|\varphi\|_{\infty} \leq 1$ and*

$\varphi(0) = 0$. Moreover, we require $\mathbf{u}^* \in \mathcal{R}$ that entails existence of $\boldsymbol{\beta}^*$ such that $\|\boldsymbol{\beta}^*/\rho\|_{2,\rho} < \infty$ and $\mathbf{u}^* = \mathbf{u}(\boldsymbol{\beta}^*)$.

The universality requirement in Assumption 8 is non-restrictive as Fourier, ReLU, and Stump basis functions are all universal. This assumption also requires Lipschitz continuity of φ , $\|\varphi\|_\infty \leq 1$, and $\varphi(0) = 0$, which are standard assumptions in the literature (see, e.g., Theorem 3.2 in [Rahimi and Recht 2008](#)). Fourier and ReLU bases meet these requirements, but Stump does not due to its discontinuity at zero. Requirement $\mathbf{u}^* \in \mathcal{R}$ is non-restrictive because if it does not hold, then \mathcal{R} includes an arbitrarily close approximation of \mathbf{u}^* (please see Chapter 2 for a similar discussion).

Random basis functions provide a natural reformulation of BLP based on the parametric form (3.7). Specifically, when the random basis function is universal, we can replace decision variable $\mathbf{u} \in \mathcal{U}$ with $\mathbf{u}(\boldsymbol{\beta}) \in \mathcal{R}$ without incurring a significant loss. Performing this replacement, we obtain the bound-focused feature-based exact program (BFEP):

$$\sup_{(\eta, \boldsymbol{\beta}) \in \mathbb{R} \times \mathcal{B}} \eta + \int_{\Theta} \boldsymbol{\beta}(\theta) \left(\varphi(s; \theta) - \mathbb{E}[\varphi(s'; \theta) | s, \mathbf{a}] \right) d\theta \leq c(s, \mathbf{a}), \quad \forall (s, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}_s,$$

where $\mathcal{B} := \{\boldsymbol{\beta} : \Theta \mapsto \mathbb{R} : \|\boldsymbol{\beta}/\rho\|_{2,\rho} < \infty, \mathbf{u}(\bar{s}; \boldsymbol{\beta}) = 0\}$ is the set of all weighting functions $\boldsymbol{\beta}$ with a finite $(2, \rho)$ -norm such that their associated function $\mathbf{u}(s; \boldsymbol{\beta})$ is zero at $s = \bar{s}$, similar to all continuous functions $\mathbf{u} \in \mathcal{U}$. Note that BFEP is not a linear program because of constraint $\|\boldsymbol{\beta}/\rho\|_{2,\rho} < \infty$.

In Proposition 15, we show that pair (η^*, β^*) is an optimal BFEP solution. Thus, BFEP attains an optimal solution, and the “sup” in its formulation can be replaced by a “max”.

Proposition 15 *Pair (η^*, β^*) is an optimal solution to BFEP.*

Proposition 15 suggests that BFEP is an exact parametric reformulation of BELP. The parametric form and the exactness of BFEP allow us to construct ALP models whose optimal objective values are arbitrarily close to η^* , as we discuss in the subsequent section.

3.3.3 Bound-Focused Approximate Linear Program

Consider the integral form (3.7) for the BFEP decision variable $u(\beta)$. We can approximate it to obtain a BFA using sample average approximation with N randomly sampled parameters $\theta^1, \theta^2, \dots, \theta^N$ from ρ as follows:

$$u(s; \beta) := \beta_0 + \sum_{i=1}^N \beta_i \varphi(s; \theta^i),$$

where β is the finite weight vector $\beta = (\beta_0, \beta_1, \dots, \beta_N) \in \mathbb{R}^{N+1}$. Coefficients β is a finite analogue of the weighting function β in FELP and $u(s; \beta)$ can be viewed as a randomized BFA constructed using a functional extension of Monte Carlo sampling applied to $u(s; \beta)$. The addition of intercept β_0 is solely for the purpose of ensuring BFA satisfies constraint $u(\bar{s}; \beta) = 0$ for an appropriate choice of β_0 .

Replacing BFA $u(s; \beta)$ with bias function $u(s; \boldsymbol{\beta})$ in BFEP results in the following bound-focused ALP (BALP), denoted BALP_N :

$$\begin{aligned} \sup_{(\eta, \beta) \in \mathbb{R}^{N+2}} \quad & \eta \\ \eta + \sum_{i=1}^N \beta_i \left(\varphi(s; \theta^i) - \mathbb{E}[\varphi(s'; \theta^i) | s, \mathbf{a}] \right) & \leq c(s, \mathbf{a}) \quad \forall (s, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}_s, \\ \beta_0 + \sum_{i=1}^N \beta_i \varphi(\bar{s}; \theta^i) & = 0. \end{aligned}$$

This model is a semi-infinite linear program with $N+2$ variables and a continuum of constraints. If BALP_N attains an optimal solution, we denote it by $(\eta_N^{\text{BA}}, \beta_N^{\text{BA}}) \in \mathbb{R}^{N+1}$. Otherwise, we can add a constraint to this program requiring a norm of β to be finite. The resulting restriction of BALP_N always attains an optimal solution. The formulation of BALP_N raises fundamental questions: Does the optimal objective value of BALP_N converge to η^* as we increase the value of N ? And if so, at what rate does this convergence occur? We show that the answer to the first question is yes, and the optimal objective value of BALP_N converges to η^* at the rate of $1/\sqrt{N}$. To motivate our theoretical analyses formalizing answers to these questions, we use the following example.

Example 1 Consider an MDP with a finite number of S states in the set $\mathcal{S} := \{1, 2, \dots, S\}$ and two actions $\mathbf{a} \in \{0, 1\}$. This MDP is shown in Figure 6 and its transition probabilities are depicted in gray boxes. Specifically, we have $P(s+1|s, \mathbf{a}) = \mathbf{a}$ and $P(s-1|s, \mathbf{a}) = 1 - \mathbf{a}$, where $s-1 \equiv 1$ for $s = 1$ and $s+1 \equiv S$ for $s = S$. The MDP immediate cost function $c(s, \mathbf{a}) = s$ for all $s \in \mathcal{S}$, which is independent of the action. The optimal policy that minimizes the long-run average cost per

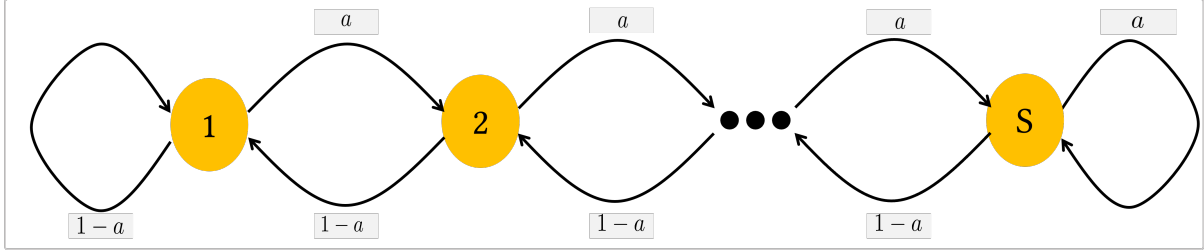
stage selects the optimal action $\mathbf{a} = 0$ at all states and results in the optimal long-run expected average cost of $\eta^* = 1$. Solving the optimality equation $\mathbf{u}(s) = \min\{s-1+\mathbf{u}(s-1), s-1+\mathbf{u}(s+1)\}$, the MDP bias function becomes $\mathbf{u}^*(s) = s(s-1)/2$. We choose state \bar{s} to be $s = 1$ for which $\mathbf{u}^*(\bar{s}) = 0$. The BELP in this example can be written as follows:

$$\begin{aligned} \max_{(\eta, \mathbf{u})} \quad & \eta \\ & \eta + \mathbf{u}(s) - \mathbf{u}(s-1) \leq s \quad \forall s = 1, 2, \dots, S, \\ & \eta + \mathbf{u}(s) - \mathbf{u}(s+1) \leq s \quad \forall s = 1, 2, \dots, S, \\ & \mathbf{u}(1) = 0. \end{aligned} \tag{3.8}$$

Assume we use the constant BFA of the form $\hat{\mathbf{u}}(s) = \mathbf{k}$ to approximate the MDP bias function $\mathbf{u}^*(s) = s(s-1)/2$, where $\mathbf{k} \in \mathbb{R}$. It is easy to see that for $\mathbf{k} = 0$, this BFA is feasible to (3.8). In fact, pair $(1, \hat{\mathbf{u}})$ with $\mathbf{k} = 0$ is an optimal solution to (3.8). This observation suggests that to learn the optimal cost $\eta^* = 1$, one may only need a simple BFA such as $\hat{\mathbf{u}}(s) = 0$, which is a poor approximation of $\mathbf{u}^*(s) = s(s-1)/2$. In other words, it may not be needed to find a BFA that closely approximates the MDP bias function $\mathbf{u}^*(s) = s(s-1)/2$ at all states in order to recover the optimal average cost $\eta^* = 1$.

In Example 1, we recover η^* using BFA $\hat{\mathbf{u}}$ that satisfies $\hat{\mathbf{u}}(s) = \mathbf{u}^*(s)$ at state $s = 1$. This particular state is the only point visited by the optimal policy in the long run. Extending this insight, we can likely derive an approximation for η^* as long as we access to a BFA $\hat{\mathbf{u}}$ that closely approximates \mathbf{u}^* within the region of the MDP state space that are visited by the optimal policy. This region can be smaller than \mathcal{S} and thus approximating \mathbf{u}^* over this region can be

Figure 6: Illustrating the connection between lower bound quality and BFA quality on a toy MDP.



much easier than approximating it over the entire state space \mathcal{S} . In Example 1, the function that matches u^* at $s = 1$ is a constant function, yet the MDP bias function is a quadratic bias function $u^*(s) = s(s - 1)/2$, which is harder to approximate than a constant function.

To formalize this concept, let \mathcal{S}^* denote the largest subset of \mathcal{S} to which the invariant probability distribution $\mu^*(\cdot) \equiv \mu(\cdot; \pi^*)$ assigns a positive mass. Specifically, define $\mathcal{S}^* := \{s \in \mathcal{S} : \mu^*(s) > 0\}$ that satisfies $\mu^*(\mathcal{S}^*) = 1$ and $\mu^*(\mathcal{S} \setminus \mathcal{S}^*) = 0$. Additionally, introduce the following hypothetical idealized math program (IMP):

$$\sup_{(\eta, \beta) \in \mathbb{R} \times \mathcal{B}} \quad \eta$$

$$\eta + u(s; \beta) - \mathbb{E}[u(s'; \beta) | s, \pi^*(s)] \leq c(s, \pi^*(s)) \quad \forall s \in \mathcal{S}^*.$$

The term “idealized” is used for IMP since its formulation depends on the knowledge of the optimal policy π^* . We show in Proposition 16 that η^* can be obtained from IMP, a considerably smaller model than BELP. Define β^{IMP} as follows

$$\beta^{\text{IMP}} := \arg \min \{ \|\beta/\rho\|_{2,\rho} : \beta \in \mathcal{B}, u(s; \beta) = u^*(s), \forall s \in \mathcal{S}^* \},$$

that is the weight function with the smallest $(2, \rho)$ -norm among all weight functions whose associated bias functions match u^* over \mathcal{S}^* . Note that β^{IMP} is well-defined because the minimum in its definition is attainable, and it possesses a smaller $(2, \rho)$ -norm compared to β^* , i.e., $\|\beta^{\text{IMP}}\|_{2,\rho} \leq \|\beta^*\|_{2,\rho}$.

Proposition 16 *Pair $(\eta^*, \beta^{\text{IMP}})$ is an optimal solution to IMP.*

The implication of Proposition 16 is that obtaining η^* does not require knowing u^* at all states. Instead, knowing u^* at \mathcal{S}^* suffices to recover η^* . Thus, approximating the optimal average cost η^* , which is possible through $u(\beta^{\text{IMP}})$, is simpler than approximating the MDP bias function $u^* = u(\beta^*)$.

To establish an error bound on the difference between η^* and the optimal objective value of BALP_N , denoted as η_N^{BA} , we can either approximate u^* or $u(\beta^{\text{IMP}})$. Approximating the former function seems excessive, as recovering η^* doesn’t require the full knowledge of u^* across the entire state space. Thus, approximating the latter function appears to be reasonable. However, pair $(\eta^*, \beta^{\text{IMP}})$ based on this latter function might not be feasible to BFEP constraints because IMP constraints are a subset of BFEP constraints. Thus, approximating $u(\beta^{\text{IMP}})$ may result

in a BFA that does not satisfy constraints (3.6). As a solution, we introduce a third function, different from \mathbf{u}^* and $\mathbf{u}(\boldsymbol{\beta}^{\text{IMP}})$, that is simpler to approximate compared to \mathbf{u}^* and is feasible to BFEP constraints. Define the following set of weight functions,

$$\mathcal{B}^{\text{AC}} := \left\{ \boldsymbol{\beta} \in \mathcal{B} \mid (\eta^*, \mathbf{u}(\boldsymbol{\beta})) \text{ is feasible to (3.6), } \mathbf{u}(s; \boldsymbol{\beta}) = \mathbf{u}^*(s), \forall s \in \mathcal{S}^* \right\},$$

and the weight function $\boldsymbol{\beta}^{\text{AC}} := \arg \min \{ \|\boldsymbol{\beta}/\rho\|_{2,\rho} : \boldsymbol{\beta} \in \mathcal{B}^{\text{AC}} \}$. Set \mathcal{B}^{AC} includes the weighting function $\boldsymbol{\beta}$ of each BFA $\mathbf{u}(\boldsymbol{\beta}) \in \mathcal{U}$, where (i) pair $(\eta^*, \mathbf{u}(\boldsymbol{\beta}))$ is feasible to BELP constraints when η is set to η^* , and (ii) BFA $\mathbf{u}(s; \boldsymbol{\beta})$ matches \mathbf{u}^* at all states within \mathcal{S}^* . Because $\mathbf{u}^* = \mathbf{u}(\boldsymbol{\beta}^*) \in \mathcal{R}$, we have $\boldsymbol{\beta}^* \in \mathcal{B}^{\text{AC}}$, i.e., \mathcal{B}^{AC} is non-empty and thus the weighting function $\boldsymbol{\beta}^{\text{AC}}$ is well-defined. Moreover, $\boldsymbol{\beta}^{\text{AC}}$ has the following property:

$$\|\boldsymbol{\beta}^{\text{IMP}}\|_{2,\rho} \leq \|\boldsymbol{\beta}^{\text{AC}}\|_{2,\rho} \leq \|\boldsymbol{\beta}^*\|_{2,\rho}.$$

The weight function $\boldsymbol{\beta}^{\text{AC}}$ thus possesses the smallest $(2, \rho)$ -norm among all other weight functions whose corresponding BFA $\mathbf{u}(\boldsymbol{\beta}) \in \mathcal{R}$ is feasible to constraints (3.6) when $\eta = \eta^*$ and is identical to \mathbf{u}^* at all states visited under the optimal policy π^* . Proposition 17 formalizes an important property of pair $(\eta^*, \boldsymbol{\beta}^{\text{AC}})$.

Proposition 17 *Pair $(\eta^*, \boldsymbol{\beta}^{\text{AC}})$ is an optimal solution to BFEP.*

Comparing propositions 15 and 17, we observe that $(\eta^*, \beta^{\text{AC}})$ and (η^*, β^*) are both optimal to BFEP. To develop our error bound, we use the former pair with the associated bias function $\mathbf{u}(\beta^{\text{AC}})$ that can be easier to approximate compared to the MDP bias function $\mathbf{u}(\beta^*) = \mathbf{u}^*$.

Theorem 4 establishes that BALP_N optimal objective value η_N^{BA} converges to η^* at a dimension-free rate of $1/\sqrt{N}$ with a high probability. This theorem relies on the following definition:

$$\text{Err}(N, \delta; \beta^{\text{AC}}) := \frac{\|\beta^{\text{AC}}/\rho\|_{2,\rho}}{\underline{\rho}\sqrt{N}} \left(5(D_s + 1)L\sqrt{\mathbb{E}_\rho[\|\theta\|_2^2]} + \sqrt{2 \ln\left(\frac{1}{\delta}\right)} \right).$$

The error rate $\text{Err}(N, \delta; \beta^{\text{AC}})$ is defined for every integer $N \geq 1$ and probability threshold $\delta \in (0, 1]$, given weight function β^{AC} . It depends on the constant $\underline{\rho} > 0$, state space diameter $D_s := \max_{s \in \mathcal{S}} \|s\|_2$, Lipschitz constant L of random basis $\varphi(\cdot)$ defined in Assumption 8, and term $\sqrt{\mathbb{E}_\rho[\|\theta\|_2^2]}$ that signals the standard deviation of ρ .

Theorem 4 *Suppose $\rho(\theta) \geq \underline{\rho} > 0$ for all $\theta \in \Theta$. Given $\delta \in (0, 1]$, we have that every finite optimal BALP_N solution $(\eta_N^{\text{BA}}, \beta_N^{\text{BA}})$ satisfies*

$$0 \leq \eta^* - \eta_N^{\text{BA}} \leq 4\text{Err}(N, \delta; \text{Err}(N, \delta; \beta^{\text{AC}})),$$

with a probability of at least $1 - \delta$.

Theorem 4 suggests that BALP is convergent, i.e., if we sample a sufficiently large number of random bases, then lower bound η_N^{BA} becomes arbitrarily close to the optimal policy cost η^* with a high probability.

3.4 Policy-Focused Programs

In §3.4.1, we present a general performance bound for greedy policies. In §3.4.2, we discuss how a discounted-cost ALP delivers BFA for average-cost MDPs. In §3.4.3, we introduce an alternative exact linear program that relies on random basis functions and includes a BFA error term in its objective function. In §3.4.4, we present PALP and analyze its theoretical properties.

3.4.1 Policy Performance Bound

Recall Proposition 14 and Example 1. They suggest that while (η^*, \mathbf{u}^*) is an optimal solution to BELP, there might be an alternative optimal solution to this program. That is, if we denote this alternative solution by \mathbf{u}^{BE} , then we have $\mathbf{u}^* \neq \mathbf{u}^{\text{BE}}$. This degeneracy imposes an issue when computing greedy policy with respect to \mathbf{u}^{BE} . Specifically, although both (η^*, \mathbf{u}^*) and $(\eta^{\text{BE}}, \mathbf{u}^{\text{BE}})$ are both optimal to BELP, the cost of greedy policy $\pi_g(\mathbf{u}^{\text{BE}})$ can be substantially worse than the cost of the greedy policy $\pi_g(\mathbf{u}^*) = \pi^*$ with respect to \mathbf{u}^* . We show this degeneracy issue arises in Example 1.

Example 2 (Revisiting Example 1) *Recall the MDP bias function $\mathbf{u}^*(s) = s(s-1)/2$ in Example 1. Define $\hat{\mathbf{u}}$ with $\hat{\mathbf{u}}(s) = 0$ at $s = \bar{s} = 1$ and $\hat{\mathbf{u}}(s) = 1/s$ for $s \geq 2$. It is easy to verify that the pair $(1, \hat{\mathbf{u}})$ is an optimal solution to BELP (3.8). However, the greedy policy with respect to $\hat{\mathbf{u}}$ always chooses the worst action $\mathbf{a} = 1$ at all states, i.e., $\pi_g(s; \hat{\mathbf{u}}) = 1$ for all $s \in \mathcal{S}$. This greedy policy has the average cost of $\eta^{\pi_g(\hat{\mathbf{u}})} = S$, which is S times larger than the optimal policy cost $\eta^* = 1$. Therefore, greedy policies based on bias functions obtained from BELP may lead to highly suboptimal greedy policies.*

The above example thus suggests that if we are lucky such that solving BELP results in the MDP bias function \mathbf{u}^* , then the greedy policy with respect to this BELP optimal solution coincides with the optimal policy. Nevertheless, there can be alternative BELP optimal solutions with sub-optimal greedy policies.

In Theorem 5, we show that to compute good greedy policies, we need to access a BFA that is close to the MDP bias function \mathbf{u}^* at all states. In Example 2, because $\hat{\mathbf{u}}(s)$ is a poor approximation of $\mathbf{u}^*(s)$ at all states, except $s = 1$, it led to a poor greedy policy, matching the insight from Theorem 5. Let $\mu(\cdot; \beta) \equiv \mu(\cdot; \pi_g(\beta))$ be the invariant probability measure defined in (3.3) for greedy policy $\pi_g(\beta)$.

Theorem 5 *Given $\beta \in \mathbb{R}^{N+1}$, the cost of the greedy policy $\pi_g(\beta)$ is upper bounded as follows:*

$$AC(\pi_g(\beta)) \leq \eta^* + \|\mathbf{u}(\beta) - \mathbf{u}^*\|_\infty + \|\mathbf{u}^* - \mathbf{u}(\beta)\|_{1, \mu(\beta)} \leq \eta^* + 2\|\mathbf{u}(\beta) - \mathbf{u}^*\|_\infty. \quad (3.9)$$

Theorem 5 generalizes Theorem 2 in De Farias and Van Roy (2002) to MDPs with a continuous state space. Performance bound (3.9) suggests that the cost of a greedy policy depends on the ∞ -norm quality of BFA. Consequently, for the optimal pair $(\eta_N^{\text{BA}}, \beta_N^{\text{BA}})$ derived from BALP_N , the ∞ -norm gap $\|\mathbf{u}(\beta_N^{\text{BA}}) - \mathbf{u}^*\|_\infty$ may be substantial, leading to its greedy policy $\pi_g(\eta_N^{\text{BA}}, \beta_N^{\text{BA}})$ being sub-optimal. It is worth noting that the performance bound (3.9) can be improved to $AC(\pi_g(\beta)) \leq \eta^* + \|\mathbf{u}^* - \mathbf{u}(\beta)\|_{1, \mu(\beta)}$ if β is such that $\mathbf{u}(s; \beta) \leq \mathbf{u}^*(s)$ for every state $s \in \mathcal{S}$.

3.4.2 Discounted-cost Approach to Average-Cost MDPs

In this section, we discuss a strategy to compute control policies such that bound (3.9) on their average cost performance can be potentially low. This strategy relies on approximately solving the discounted-cost version of the MDP and has the following steps: (i) formulate a discounted-cost variant of the MDP, (ii) solve an ALP model to obtain a value function approximation (VFA) for the discounted-cost model, (iii) recover a BFA from this VFA, and (iv) construct the greedy policy with respect to the resulting BFA.

For the average-cost MDP model in §3.2, we define the expected discounted cost of a policy $\pi_\alpha : \mathcal{S} \mapsto \mathcal{A}$ as follows similar to Chapter 2:

$$\text{DC}(s; \pi_\alpha) := \mathbb{E}_s^{\pi_\alpha} \left[\sum_{t=0}^{\infty} \alpha^t c(s_t, \pi(s_t)) \right], \quad (3.10)$$

where $\alpha \in (0, 1)$ is a discount factor. Comparing the two measures for evaluating the effectiveness of a policy, namely average and discounted costs in (3.10) and (3.1), respectively, we consider the optimization problem $\inf_{\pi_\alpha} \text{DC}(s; \pi_\alpha)$ that seeks to identify a policy that minimizes latter measure, unlike the optimization problem (3.2) that focuses on the former measure. There are several known conditions under which there exists a discounted-cost optimal policy π_α^* that solves problem $\inf_{\pi_\alpha} \text{DC}(s; \pi_\alpha)$ at all states, meaning that identity $\text{DC}(s; \pi_\alpha) = \text{DC}(s; \pi_\alpha^*)$ holds

for every state $s \in \mathcal{S}$. In addition, there exists an MDP value function, denoted $V_\alpha^* : \mathcal{S} \mapsto \mathbb{R}$, that is a solution to the following discounted-cost optimality equation:

$$V_\alpha^*(s) := \min_{a \in \mathcal{A}} \{c(s, a) + \alpha \mathbb{E}[V_\alpha^*(s') | s, a]\}, \quad \forall s \in \mathcal{S}.$$

The optimal policy π_α^* and the MDP value function $V_\alpha^*(s)$ are linked as follows: the action $\pi_\alpha^*(s) \in \mathcal{A}_s$ chosen in state s by the optimal policy π_α^* minimizes the objective function inside the above optimality equation. That is, for every $s \in \mathcal{S}$, the following equality holds:

$$V_\alpha^*(s) := c(s, \pi_\alpha^*(s)) + \alpha \mathbb{E}[V_\alpha^*(s') | s, \pi_\alpha^*(s)].$$

Similar to our discussion in Chapter 2, there are known assumptions under which the optimal pair $(\pi_\alpha^*, V_\alpha^*)$ for the discounted-cost objective exists. We thus assume such conditions hold and the optimal pair $(\pi_\alpha^*, V_\alpha^*)$ exists.

The discounted-cost and average-cost objectives are connected. Given discount factor α and MDP value function V_α^* , define the average-cost as $\eta_\alpha := (1 - \alpha)m_\alpha$ and the bias function as $u_\alpha(s) := V_\alpha^*(s) - m_\alpha$, where $m_\alpha := V_\alpha^*(\bar{s})$. It is easy to confirm that pair (η_α, u_α) is a solution to the following optimality equation (see pages 84–55 in [Hernández-Lerma and Lasserre 1996](#)):

$$u_\alpha(s) = \min_{a \in \mathcal{A}} \{c(s, a) - \eta_\alpha + \alpha \mathbb{E}[u_\alpha(s') | s, a]\}, \quad \forall s \in \mathcal{S}.$$

The above equation resembles the average-cost optimality equation (3.4), except it has an additional α before the expectation term such that if $\alpha = 1$, the above equation boils down to (3.4). In addition, if α is close to 1, we may expect the MDP bias function \mathbf{u}^* to be close to the bias function \mathbf{u}_α .

Motivated by the construction of the bias function \mathbf{u}_α from value function V_α^* , we can construct a BFA if we access a VFA approximating V_α^* . A VFA is defined as a linear combination of basis functions, e.g., random basis functions. Specifically, given N random basis functions with parameters $\theta^1, \theta^2, \dots, \theta^N$, we can define VFA $V_\alpha(s; \beta) := \beta_0 + \sum_{i=0}^N \beta_i \varphi(s; \theta^i)$, as we did in Chapter 2. The coefficients of this VFA can be optimized using the following discounted-cost ALP model:

$$\sup_{\beta \in \mathbb{R}^{N+1}} \{ \mathbb{E}_\nu[V_\alpha(s; \beta)] : V_\alpha(s; \beta) \leq c(s, \mathbf{a}) + \alpha \mathbb{E}[V_\alpha(s'; \beta) | s, \mathbf{a}], \forall (s, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}_s \}, \quad (3.11)$$

where distribution ν over \mathcal{S} is called the state-relevance distribution that assigns a non-negative value to each state. It is known that the objective of (3.11) can be equivalently written as $\min_\beta \|V_\alpha^* - V_\alpha(\beta)\|_{1, \nu}$, where $\|V_\alpha^* - V_\alpha(\beta)\|_{1, \nu} := \mathbb{E}_\nu[|V_\alpha^* - V_\alpha|]$. This reformulation of (3.11) means that the ALP model minimizes the $(1, \nu)$ -norm distance between the MDP value function V_α^* and VFA $V_\alpha(\beta)$. Note that we labeled ALP model (3.11) as the “feature-based ALP” in Chapter 2.

Let β_N^{DC} be a finite optimal solution to (3.11). We can define BFA $\mathbf{u}_\alpha(s; \beta_N^{\text{DC}})$ using VFA weights β_N^{DC} at state $s \in \mathcal{S}$ as $\mathbf{u}_\alpha(s; \beta_N^{\text{DC}}) := V_\alpha(s; \beta_N^{\text{DC}}) - m_\alpha(\beta_N^{\text{DC}})$, where $m_\alpha(\beta_N^{\text{DC}}) := V_\alpha(\bar{s}; \hat{\beta})$.

We also define $\pi_g(\beta_N^{\text{DC}}; \alpha)$ as the greedy policy with respect to BFA $\mathbf{u}_\alpha(\beta_N^{\text{DC}})$. Using Theorem 5, it is easy to derive the following upper bound on the performance of $\pi_g(\beta; \alpha)$:

$$\text{AC}(\pi_g(\beta_N^{\text{DC}}; \alpha)) \leq \eta^* + 2|\mathbf{m}_\alpha - \mathbf{m}_\alpha(\beta_N^{\text{DC}})| + 2\|\mathbf{V}_\alpha^* - \mathbf{V}_\alpha(\beta_N^{\text{DC}})\|_\infty + 2\|\mathbf{u}_\alpha - \mathbf{u}^*\|_\infty \quad (3.12)$$

Two terms $|\mathbf{m}_\alpha - \mathbf{m}_\alpha(\beta_N^{\text{DC}})|$ and $\|\mathbf{V}_\alpha(\beta) - \mathbf{V}_\alpha^*\|_\infty$ are errors incurred due to approximating MDP value function, and the term $\|\mathbf{u}_\alpha - \mathbf{u}^*\|_\infty$ is the error resulted by solving a discounted cost objective in lieu of the average-cost objective.

To shed light on (3.12), we use Theorem 1 from Chapter 2. Note that ALP model (3.11) minimizes the $(1, \nu)$ -norm gap, but we observe in (3.12) the ∞ -norm gap. Therefore, this model does not necessarily minimize the third term in (3.12). Temporarily assume that ν is selected to make the reduction in $(1, \nu)$ -norm gap translates to a reduction in the ∞ -norm gap. From Theorem 1, we can infer that inequality $\|\mathbf{V}_\alpha(\beta) - \mathbf{V}_\alpha^*\|_{1, \nu} \leq C/(1 - \alpha)\sqrt{N}$ holds with a high probability, where $C > 0$ represents a non-negative constant. This upper bound suggests that as α approaches 1, a substantially larger number of random basis functions N is needed to maintain the $(1, \nu)$ -norm gap $\|\mathbf{V}_\alpha(\beta) - \mathbf{V}_\alpha^*\|_{1, \nu}$ below a fixed threshold. Specifically, to ensure $\|\mathbf{V}_\alpha(\beta) - \mathbf{V}_\alpha^*\|_{1, \nu} \leq \varepsilon$, we need to require $N \geq (C/(1 - \alpha)\varepsilon)^2$, indicating that N must grow superlinearly in $1 - \alpha$. Thus α plays a trade-off between making either $\|\mathbf{V}_\alpha(\beta) - \mathbf{V}_\alpha^*\|_{1, \nu}$ or $\|\mathbf{u}_\alpha - \mathbf{u}^*\|_\infty$ small. That is, when α is close to zero, it is easy to reduce $\|\mathbf{V}_\alpha(\beta) - \mathbf{V}_\alpha^*\|_{1, \nu}$ but $\|\mathbf{u}_\alpha - \mathbf{u}^*\|_\infty$ is large, whereas when α is close to one, $\|\mathbf{u}_\alpha - \mathbf{u}^*\|_\infty$ is small but minimizing $\|\mathbf{V}_\alpha(\beta) - \mathbf{V}_\alpha^*\|_{1, \nu}$ is hard because of needing a large N .

The error term $|\mathbf{m}_\alpha - \mathbf{m}_\alpha(\beta_N^{\text{DC}})|$ relies on the choices of both α and \bar{s} . Specifically, if the VFA error $|V_\alpha^*(s) - V\alpha(s; \hat{\beta})|$ at state $s = \bar{s}$ is substantial, then the constant $\mathbf{m}_\alpha(\beta_N^{\text{DC}})$ will deviate significantly from the true value of $V_\alpha^*(\bar{s})$. Therefore, when using the discounted-cost approach to average-cost MDP, the choices of the discount factor α and the reference state \bar{s} can substantially impact policy performance. Regrettably, determining the optimal values for these parameters remains a practical challenge.

3.4.3 Policy-Focused Exact Programs

In this section, we present a direct strategy to compute policies such that the upper bound (3.9) on their performance is small. As discussed, BELP suffers from a critical degeneracy issue. [De Farias and Van Roy \(2002\)](#) identified this issue in an average-cost ALP model and used a queuing example to illustrate why this issue impacts greedy policy quality. In fact, as we already see, the root cause of this issue is in the formulation of the exact model BELP. Thus, if we directly approximate BELP to obtain an ALP, the resulting ALP model will also suffer from the same degeneracy issue in BELP. To mitigate this issue, [De Farias and Van Roy \(2002\)](#) suggested solving a different ALP formulation, called second-phase ALP, that has the expected value of its BFA with respect to a state-relevance distribution in its objective function. This distribution assigns weights to different regions of the state space and thus controls the quality of BFA, similar to the role of ν in the formulation of (3.11). State-relevance distribution directly arises in the objective function ALPs for discounted-cost MDPs, e.g., see (3.11), but this is not the case in the average-cost ALP. Therefore, [De Farias and Van Roy \(2002\)](#) suggested artificially adding state-relevance distribution to an ALP model for average-cost MDPs. Motivated by their

formulation, we introduce below an exact linear program involving a state-relevance distribution in its objective function.

Let ν be a state-relevance distribution defined over \mathcal{S} . We define the policy-focused exact linear program (PELP) as follows:

$$\begin{aligned} \sup_{\mathbf{u} \in \mathcal{U}} \quad & \mathbb{E}_\nu[\mathbf{u}] \\ & \mathbf{u}(s) - \mathbb{E}[\mathbf{u}(s')|s, \mathbf{a}] \leq c(s, \mathbf{a}) - \eta^*, \quad \forall (s, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}_s. \end{aligned}$$

The bias function \mathbf{u} is the only decision variable of the PELP, unlike BELP with decision variables of both η and \mathbf{u} . In the former model, η is essentially fixed to the optimal objective value of the latter model. At state \bar{s} , both models require $\mathbf{u}(\bar{s}) = 0$. Comparing PELP with (3.11), both models incorporate the state-relevance distribution ν in their objective functions. However, PELP focuses on the direct optimization of the bias function, whereas (3.11) performs VFA optimization for a discounted-cost version of the MDP. As we show in Proposition 18, PELP is a regression problem that minimizes $(1, \nu)$ -norm gap between the decision variable \mathbf{u} and \mathbf{u}^* . In addition, we show that \mathbf{u}^* is the unique solution to the PELP if ν is positive almost everywhere.

Proposition 18 *Assume state-relevance distribution ν assigns a positive mass to all non-zero measure subsets of \mathcal{S} . Then, PELP is equivalent to the following regression model:*

$$\begin{aligned} \min_{\mathbf{u} \in \mathcal{U}} \quad & \|\mathbf{u} - \mathbf{u}^*\|_{1, \nu} \\ & \mathbf{u}(s) - \mathbb{E}[\mathbf{u}(s')|s, \mathbf{a}] \leq c(s, \mathbf{a}) - \eta^*, \quad \forall (s, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}_s. \end{aligned}$$

Moreover, \mathbf{u}^* is the unique PELP optimal solution.

Proposition 18 indicates that PELP aims to minimize the $(1, \nu)$ -norm distance between a bias function $\mathbf{u} \in \mathcal{U}$ and the MDP bias function \mathbf{u}^* . Because PELP is an exact model (i.e., it does not include any approximation), minimizing the $(1, \nu)$ -norm results in minimizing the ∞ -norm, as can be inferred from Proposition 18.

Proposition 18 highlights that PELP minimizes the $(1, \nu)$ -norm distance between a bias function $\mathbf{u} \in \mathcal{U}$ and the MDP bias function \mathbf{u}^* . Specifically, since PELP is an exact model without any approximations, minimizing the $(1, \nu)$ -norm translates into minimizing the ∞ -norm. This is established by Proposition 18 because \mathbf{u}^* is the unique PELP optimal solution. Because of this ∞ -norm reduction, PELP can be seen as a model that reduces the ∞ -norm error term in performance bound (3.9). Therefore, we refer to it as “policy-focused”. Moreover, Proposition 18 suggests that computing (η^*, \mathbf{u}^*) , in principle, is equivalent to first solving BELP to obtain η^* and then using η^* in PELP to obtain \mathbf{u}^* . In fact, if we use a uniform state-relevance distribution ν in PELP, then Proposition 18 suggests that the greedy policy with respect to PELP optimal solution \mathbf{u}^* is the optimal policy. For the MDP in Example 1, if we use a uniform state relevance distribution in PELP, then it is easy to verify that the MDP bias function $\mathbf{u}^*(s) = s(s-1)/2$ becomes its optimal solution, as we expect by Proposition 18.

We next define the policy-focused feature-based exact program (PFEP) as follows:

$$\begin{aligned} \sup_{\beta \in \mathcal{B}} \int_{\Theta} \beta(\theta) \mathbb{E}_{\nu}[\varphi(s; \theta)] \, d\theta \\ \int_{\Theta} \beta(\theta) \left(\varphi(s; \theta) - \mathbb{E}[\varphi(s'; \theta) | s, \mathbf{a}] \right) \, d\theta \leq c(s, \mathbf{a}) - \eta^*, \quad \forall (s, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}_s. \end{aligned}$$

Unlike BERP and PELP directly optimizing the non-parametric bias function $\mathbf{u} \in \mathcal{U}$, PFEP optimizes the parametric weighting function $\boldsymbol{\beta} \in \mathcal{B}$, similar to BFEP. In the following proposition, we show that $\boldsymbol{\beta}^*$ is an optimal solution to PFEP. Therefore, PFEP attains an optimal solution, and its “sup” can be replaced by a “max”.

Proposition 19 *Assume state-relevance distribution ν assigns a positive mass to all non-zero measure subsets of \mathcal{S} . Weighting function $\boldsymbol{\beta}^*$ is an optimal solution to PFEP.*

Proposition 19 implies that PFEP is a parametric reformulation of PELP, which is a result of the universality of random bases, our mild assumption $\mathbf{u}^* \in \mathcal{R}$ in §3.3.2, and Proposition 18. An important implication of this proposition is provided in the following remark.

Remark 1 *As we observed in Proposition 17, pair $(\eta^*, \boldsymbol{\beta}^{\text{AC}})$ is an optimal solution to BFEP. However, the weighting function $\boldsymbol{\beta}^{\text{AC}}$ may not be an optimal solution for PFEP. Specifically, if*

$$\nu(\{s : \mathcal{S} \mid \mathbf{u}(s; \boldsymbol{\beta}^{\text{AC}}) \neq \mathbf{u}^*(s)\}) > 0 \quad \text{and} \quad \nu(\mathcal{S} \setminus \mathcal{S}^*) > 0,$$

then $\boldsymbol{\beta}^{\text{AC}}$ is suboptimal to PFEP. In other words, PFEP finds a bias function that is close to \mathbf{u}^ at all states $s \in \mathcal{S}$, but BFEP only focuses on those states visited by π^* , which is \mathcal{S}^* . Thus, PFEP is an appropriate model for computing BFAs and reducing ∞ -norm. Thus, PFEP is an appropriate model for computing BFAs and reducing ∞ -norm. Thus, PFEP is an appropriate model for computing BFAs with small ∞ -norm errors.*

3.4.4 Policy-Focused Approximate Linear Program

Recall PFEP that minimizes the $(1, \nu)$ -norm distance of $\mathbf{u}(\boldsymbol{\beta})$ to \mathbf{u}^* . As we discussed, the $(1, \nu)$ -norm reduction in PFEP and PELP translates to ∞ -norm BFA error reduction since these models are exact. However, if we approximate PFEP decision variable $\mathbf{u}(\boldsymbol{\beta})$ by sampling random basis functions, similar to the derivation of BALP from BFEP, then the reduction in $(1, \nu)$ -norm distance to \mathbf{u}^* does not necessarily translate to reduction in the ∞ -distance to \mathbf{u}^* , which is essential to ensure the performance of the greedy policy obtained from the approximate model (see Theorem 5). Therefore, to achieve BFAs with low ∞ -norm errors, we use a similar idea to our “self-guiding mechanism” in Chapter 2. We design a method that includes (i) iteratively sampling random basis functions in batches and solving a sequence of ALP models obtained upon approximating BFEP and (ii) connecting BFAs in this sequence using guiding constraints that ensure an upper bound on the BFA ∞ -norm error is weakly decreasing.

Let η_N^{BA} be the optimal objective value of BALP_N that includes N random basis functions with parameters $\{\theta^1, \theta^2, \dots, \theta^N\}$. We construct a sequence of ALP models with $N, N+B, \dots, N+QB$ basis functions, where B is a sampling batch size and Q is the number of times random bases are sampled. In other words, we sample random bases in batches of size B for Q iterations to construct these ALP models and obtain the total of $N+QB$ basis functions. At iteration $q \geq 1$, we obtain $N+qB$ bases with parameters $\{\theta^1, \dots, \theta^N\} \cup \{\theta^{N+1}, \dots, \theta^{N+qB}\}$ and solve policy-focused approximate linear program (PALP_{N+qB}):

$$\sup_{\boldsymbol{\beta} \in \mathbb{R}^{N+qB}} \quad \beta_0 + \sum_{i=1}^{N+qB} \beta_i \mathbb{E}_{\nu}[\varphi(s; \theta^i)]$$

$$\begin{aligned}
\sum_{i=1}^{N+qB} \beta_i \left(\varphi(s; \theta^i) - \mathbb{E}[\varphi(s'; \theta^i) | s, \mathbf{a}] \right) &\leq c(s, \mathbf{a}) - \eta_N^{\text{BA}}, \quad \forall (s, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}_s, \\
\beta_0 + \sum_{i=1}^{N+qB} \beta_i \varphi(s; \theta^i) &\geq \mathbf{u}(s; \beta_{N+(q-1)B}^{\text{PA}}), \quad \forall s \in \mathcal{S}, \\
\beta_0 + \sum_{i=1}^{N+qB} \beta_i \varphi(\bar{s}; \theta^i) &= 0.
\end{aligned} \tag{3.13}$$

In the above formulation, β_{N+qB}^{PA} denotes a finite optimal solution of PALP_{N+qB} . We use initialization $\mathbf{u}(\beta_N^{\text{PA}}) \equiv \mathbf{u}(\beta_N^{\text{BA}})$ at $q = 1$. We refer to constraints (3.13) as guiding constraints that ensure the BFA with $N + qB$ basis functions is a state-wise upper bound on the past BFA with $N + (q - 1)B$ bases, where our initial BFA used for guiding is the BALP VFA $\mathbf{u}(\beta_N^{\text{BA}})$. PALP_{N+qB} has the objective with respect to the state-relevance distribution \mathbf{v} to control the quality of the BFA, similar to PELP and PFEP.

To understand why PALP results in BFAs with a small ∞ -norm gap (i.e., strong greedy policies), we present an analysis that relies on an idealized bias function. This analysis generalized the one in [De Farias and Van Roy \(2002\)](#) for finite-state MDPs to continuous-state MDPs. Informally, given lower bound η_N^{BA} on η^* , there is an idealized bias function, denoted $\mathbf{u}^{\text{ID}}(\eta_N^{\text{BA}})$, that is a state-wise upper bound on every feasible solution to PALP_{N+qB} for all q . We will show that PALP minimizes the $(1, \mathbf{v})$ -gap between its BFA and this idealized bias function. We will also show that the idealized bias function $\mathbf{u}^{\text{ID}}(\eta_N^{\text{BA}})$ gets closer to the MDP bias function \mathbf{u}^* as η_N^{BA} gets closer to η^* upon increasing N . Therefore, PALP indirectly minimizes the ∞ -norm gap between BFA and the MDP bias function \mathbf{u}^* .

To define the idealized bias function, we revisit the optimality equation (3.4). Let $Tu : \mathcal{S} \mapsto \mathbb{R}$ be the transformation of u under mapping T . We define the evaluation of Tu at state $s \in \mathcal{S}$ as $Tu(s) := \mathbb{E}[u(s')|s, \pi^*(s)]$. Under Assumption 9, we can show that T is a bounded linear transformation over the Banach space of all continuous functions defined over \mathcal{S} (please see Proposition 23). In the literature, this assumption is referred to as the strong continuity of the MDP transition kernel (see, e.g., Condition 3.3.3 in [Hernández-Lerma and Lasserre 1996](#)).

Assumption 9 *For every measurable bounded function u , mapping $(s, a) \mapsto \int_{\mathcal{S}} u(s')P(ds'|s, a)$ is bounded and continuous over $\mathcal{S} \times \mathcal{A}_s$.*

Recall pair (η^*, u^*) solves optimality equation (3.4). We can rewrite this equation in terms of T and $g^*(s) := c(s, \pi^*(s)) - \eta^*$ as follows $u^* = g^* + Tu^*$. Iterating this equation for K times, which requires replacing Tu^* with its definition recursively, results in the following version of the optimality equation:

$$u^*(s) = \sum_{k=0}^K T^k g^*(s) + T^{K+1} u^*(s), \quad \forall s \in \mathcal{S}, \quad (3.14)$$

where function $T^0 := I$ is the identity transformation that satisfies $Iu = u$. For each k , transformation T^k applied to an arbitrary function u results in function $T^k u : \mathcal{S} \mapsto \mathbb{R}$ that evaluates to $T^k u(s) := \int_{\mathcal{S}} P^k(s'|s, \pi^*) u(s') ds'$ at state s . Here, $P^k(\cdot|s, \pi^*)$ denotes the k -step transition probabilities from initial state s when using the optimal policy π^* (see, e.g., Page 21 in [Hernández-Lerma and Lasserre 1996](#)). Taking the limit of (3.14) when $K \rightarrow \infty$, we obtain $u^* = \lim_{K \rightarrow \infty} \sum_{k=0}^K T^k g^*$.

From our discussion in §3.2, function \mathbf{u}^* satisfying the additional condition $\mathbf{u}^*(s) = 0$ at the reference state $s = \bar{s}$ is the unique solution of the optimality equation (3.4). The recent formulation of \mathbf{u}^* , which is $\mathbf{u}^* = \lim_{K \rightarrow \infty} \sum_{k=0}^K T^k \mathbf{g}^*$, does not reflect condition $\mathbf{u}^*(\bar{s}) = 0$. To incorporate this condition into this definition, we utilize the following identities:

$$\mathbf{u}^*(s) = \mathbf{u}^*(s) - \mathbf{u}^*(\bar{s}) = \lim_{K \rightarrow \infty} \sum_{k=0}^K T^k \mathbf{g}^*(s) - T^k \mathbf{g}^*(\bar{s}) = \lim_{K \rightarrow \infty} \sum_{k=0}^K F^k \mathbf{g}^*(s),$$

where the evaluation of transformation F^k applied to an arbitrary function \mathbf{u} at state s is:

$$F^k \mathbf{u}(s) = \int_{\mathcal{S}} (\mathbf{P}^k(s'|s, \pi^*) - \mathbf{P}^k(s'|\bar{s}, \pi^*)) \mathbf{u}(s') \, d s'.$$

Therefore, unlike the former definition of \mathbf{u}^* based on T^k , which is $\mathbf{u}^* = \lim_{K \rightarrow \infty} \sum_{k=0}^K T^k \mathbf{g}^*$, the latter definition based on F^k , which is $\mathbf{u}^* = \lim_{K \rightarrow \infty} \sum_{k=0}^K F^k \mathbf{g}^*(s)$, reflects the condition $\mathbf{u}^*(\bar{s}) = 0$. Once again, because $\mathbf{u}^* = \lim_{K \rightarrow \infty} \sum_{k=0}^K F^k \mathbf{g}^*(s)$ satisfying condition $\mathbf{u}^*(\bar{s}) = 0$ is the unique solution of the optimality equation (3.4), limit $\lim_{K \rightarrow \infty} \sum_{k=0}^K F^k \mathbf{g}^*$ must exist. Therefore, \mathbf{u}^* can be explicitly written as $\mathbf{u}^* = F^\infty \mathbf{g}^*$, where transformation F^∞ is defined as $F^\infty := \sum_{k=0}^\infty F^k$.

We are now ready to define the idealized bias function. Define $\mathbf{u}^{\text{ID}}(s; \eta_N^{\text{BA}}) := F^\infty \mathbf{g}_N^{\text{BA}}$, where $\mathbf{g}_N^{\text{BA}}(s) := (c(s, \pi^*(s)) - \eta_N^{\text{BA}})$. Note that \mathbf{g}_N^{BA} is similar to \mathbf{g}^* , except \mathbf{g}_N^{BA} includes the value of η_N^{BA} obtained by BALP_N and not the optimal average cost η^* . The following proposition establishes that the ∞ -norm difference between the idealized bias function $\mathbf{u}^{\text{ID}}(\eta_N^{\text{BA}})$ and the MDP bias

function \mathbf{u}^* linearly scales on the error term $\boldsymbol{\eta}^* - \boldsymbol{\eta}_N^{\text{BA}}$, where the slope of this linear function depends on a norm of F^∞ defined as $\|F^\infty\| := \sup\{\|F^\infty \mathbf{u}\|_\infty : \|\mathbf{u}\|_\infty \leq 1, \mathbf{u} : \mathcal{S} \mapsto \mathbb{R}\}$.

Proposition 20 *Given N , it holds that $\|\mathbf{u}^{\text{ID}}(\boldsymbol{\eta}_N^{\text{BA}}) - \mathbf{u}^*\|_\infty \leq \|F^\infty\|(\boldsymbol{\eta}^* - \boldsymbol{\eta}_N^{\text{BA}})$. Moreover, for every $q = 1, 2, \dots, Q$ and each solution $\beta \in \mathbb{R}^{N+qB}$ feasible to PALP_{N+qB} , it holds that*

$$\mathbf{u}(s; \beta) \leq \mathbf{u}^{\text{ID}}(s; \boldsymbol{\eta}_N^{\text{BA}}) \leq \mathbf{u}^*(s) + \|F^\infty\|(\boldsymbol{\eta}^* - \boldsymbol{\eta}_N^{\text{BA}}), \quad \forall s \in \mathcal{S}.$$

Using Proposition 20, it is easy to verify that PALP_{N+qB} is equivalent to the following optimization problem:

$$\begin{aligned} \min_{\beta \in \mathbb{R}^{N+qB}} \quad & \|\mathbf{u}(\beta) - \mathbf{u}^{\text{ID}}(\boldsymbol{\eta}_N^{\text{BA}})\|_{1,v} \\ \mathbf{u}(s; \beta) - \mathbb{E}[\mathbf{u}(s'; \beta) | s, \mathbf{a}] \quad & \leq \mathbf{c}(s, \mathbf{a}) - \boldsymbol{\eta}_N^{\text{BA}}, \quad \forall (s, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}_s, \\ \|\mathbf{u}(\beta) - \mathbf{u}^{\text{ID}}(\boldsymbol{\eta}_N^{\text{BA}})\|_\infty \quad & \leq \left\| \mathbf{u}(\beta_{N+(q-1)B}^{\text{PA}}) - \mathbf{u}^{\text{ID}}(\boldsymbol{\eta}_N^{\text{BA}}) \right\|_\infty, \\ \mathbf{u}(\bar{s}; \beta) \quad & = 0. \end{aligned}$$

Therefore, the infinity-norm gap between PALP_{N+qB} BFA and the idealized bias function, which is $\|\mathbf{u}^{\text{ID}}(\boldsymbol{\eta}_N^{\text{BA}}) - \mathbf{u}(\beta_{N+qB}^{\text{PA}})\|_\infty$, is weakly decreases in q , meaning,

$$\left\| \mathbf{u}^{\text{ID}}(\boldsymbol{\eta}_N^{\text{BA}}) - \mathbf{u}(\beta_{N+B}^{\text{PA}}) \right\|_\infty \geq \left\| \mathbf{u}^{\text{ID}}(\boldsymbol{\eta}_N^{\text{BA}}) - \mathbf{u}(\beta_{N+2B}^{\text{PA}}) \right\|_\infty \geq \dots \geq \left\| \mathbf{u}^{\text{ID}}(\boldsymbol{\eta}_N^{\text{BA}}) - \mathbf{u}(\beta_{N+QB}^{\text{PA}}) \right\|_\infty. \quad (3.15)$$

Using the above inequalities, we can show that the worst-case performance of greedy policies obtained from PALP is weakly improving in q . This result is formalized in the following corollary.

Corollary 1 Fix N and $q = 1, 2, \dots, Q$. Given feasible solution $\beta \in \mathbb{R}^{N+qB}$ to PALP_{N+qB} , the average cost of the greedy policy $\pi_g(\beta)$ satisfies:

$$\text{AC}(\pi_g(\beta)) \leq \eta^* + 2(\eta^* - \eta_N^{\text{BA}})\|F^\infty\| + 2\|\mathbf{u}(\beta) - \mathbf{u}^{\text{ID}}(\eta_N^{\text{BA}})\|_\infty$$

Corollary 1 upper bounds the performance of greedy policy $\text{AC}(\pi_g(\beta))$ obtained from any PALP_{N+qB} feasible solution $\beta \in \mathbb{R}^{N+qB}$ based on two error terms: $(\eta^* - \eta_N^{\text{BA}})\|F^\infty\|$ and $\|\mathbf{u}(\beta) - \mathbf{u}^{\text{ID}}(\eta_N^{\text{BA}})\|_\infty$. The first term captures the impact of using η_N^{BA} in lieu of η^* in the formulation PALP_{N+qB} on the greedy policy $\pi_g(\beta)$. The second term captures how close BFA $\mathbf{u}(\beta)$ is to the idealized bias function $\mathbf{u}^{\text{ID}}(\eta_N^{\text{BA}})$, given lower bound η_N^{BA} obtained from BALP_N . The upper bound established in Corollary 1, in conjunction with Theorem 4 and inequalities (3.15), indicate that when both N and Q take large values, our randomized multi-shot approximation approach, which involves two steps of solving BALP_N initially to obtain lower bound η_N^{BA} and then solving a sequence of PALPs to compute BFA $\mathbf{u}(\beta_{N+QB}^{\text{PA}})$, culminates greedy policy $\pi_g(\beta_{N+QB}^{\text{PA}})$ with an average cost close to η^* .

3.5 Algorithm

Our main algorithm, which involves solving BALP and PALP, is summarized in Algorithm 3 and is the average-cost counterpart of the self-guided FALP algorithm (see Algorithm 2 in Chapter 2). In Step 1, we sample N random basis function parameters $\{\theta^i : i = 1, 2, \dots, N\}$ from ρ in one shot and initialize the set of samples ϑ to these N samples. We next formulate BALP_N and solve it to obtain its optimal solution $(\eta_N^{\text{BA}}, \beta_N^{\text{BA}})$. In addition, we initialize vector

Algorithm 3: Randomized Multi-shot Approximation Algorithm for Average-Cost MDPs

Step 1. Sample N random basis parameters $\{\theta^i : i = 1, \dots, N\}$ from ρ , set $\vartheta \leftarrow \{\theta^i : i = 1, \dots, N\}$, and solve BALP_N formulated using parameters in ϑ to compute $(\eta_N^{\text{BA}}, \beta_N^{\text{BA}})$. Also, initialize $\beta_N^{\text{PA}} \leftarrow \beta_N^{\text{BA}}$.

for $q = 1, 2, \dots, Q$ **do**

Step 2. Sample B random basis parameters $\{\theta^{N+(q-1)B+i} : i = 1, 2, \dots, B\}$ from ρ , and update the set of sampled parameters $\vartheta \leftarrow \vartheta \cup \{\theta^{N+(q-1)B+i} : i = 1, 2, \dots, B\}$.

Step 3. Solve PALP_{N+qB} formulated using parameters in ϑ , constant η_N^{BA} from Step 1, and past BFA weights $\beta_{N+(q-1)B}^{\text{PA}}$ from iteration $q-1$ to obtain new BFA weights β_{N+qB}^{PA} .

Step 4. Simulate greedy policy $\pi_g(\beta_{N+qB}^{\text{PA}})$ by solving (3.5) to estimate policy cost $\text{AC}(\pi_g(\beta_{N+qB}^{\text{PA}}))$.

Return: Lower bound η_N^{BA} , upper bound $\text{AC}(\pi_g(\beta_{N+qB}^{\text{PA}}))$, optimal BFA weights β_{N+qB}^{PA} .

β_N^{PA} to BALP_N BFA weights β_N^{BA} , which is used in the right-hand-side of guiding constraints of PALP_{N+B} . Next, we perform steps 2 and 3 iteratively for Q iterations. In Step 2, we sample B additional random basis function parameters $\{\theta^{N+(q-1)B+i} : i = 1, 2, \dots, B\}$ and append them to ϑ . In Step 3, we formulate PALP_{N+qB} via random basis function samples in ϑ and past solution $\beta_{N+(q-1)B}^{\text{PA}}$. Then, we solve PALP_{N+qB} to obtain BFA weights β_{N+qB}^{PA} . In Step 4, we simulate greedy policy $\pi_g(\beta_{N+qB}^{\text{PA}})$ with respect to the terminal BFA weights β_{N+qB}^{PA} . The algorithm returns lower and upper bounds as well as the terminal BFA weights.

In steps 1 and 3, the Algorithm 3 requires solving BALP and PALP , respectively, and Step 4 requires solving greedy policy optimization (3.5) with respect to BFA $u(\beta_{N+qB}^{\text{PA}})$. BALP and PALP can be solved using two commonly used methods for solving semi-infinite linear

programs: constraint sampling and constraint generation. Note that the constraint violation learning approach in [Lin et al. \(2020\)](#) may be also used as an alternative approach to solve these semi-infinite linear programs. Solving greedy policy optimization (3.5) is not trivial. If MDP has a low-dimensional action space, we can solve this program using discretization and enumeration, as we did in Chapter 2. Otherwise, solving it may be approached by leveraging the problem structure. For example, if underlying MDP has a structure such that (3.5) can be reformulated as a known optimization problem (e.g., linear program, mixed-integer program, convex program), then we can apply commercial solvers to such reformulations and get the greedy policy. In §3.6, we discuss that this optimization problem can be cast as a mixed-integer linear program for a generalized joint replenishment problem and can be solved via Gurobi ([Gurobi Optimization 2019](#)). If MDP does not have any structure, then one might use a first-order method to solve the greedy policy optimization problem.

We describe below constraint sampling and constraint generation techniques for solving BALP. These methods can be similarly and directly applied to PALP, but we omit the details for brevity.

Constraint sampling. The constraint sampling approach replaces the continuum of BALP constraints with a finite subset of them obtained from sampling K iid state-action pairs $\{(\mathbf{s}^k, \mathbf{a}^k) \in \mathcal{S} \times \mathcal{A}_s : k = 1, 2, \dots, K\}$ from a probability distribution ψ over the state-action space $\mathcal{S} \times \mathcal{A}_s$.

The result is the following finite linear program with N random basis functions and K constraint samples:

$$\begin{aligned} \max_{(\eta, \beta) \in \mathbb{R}^{N+2}} \quad & \eta \\ \eta + \sum_{i=1}^N \beta_i \left(\varphi(s^k; \theta^i) - \mathbb{E}[\varphi(s'; \theta^i) | s^k, a^k] \right) & \leq c(s^k, a^k) \quad \forall k = 1, 2, \dots, K \\ \beta_0 + \sum_{i=1}^N \beta_i \varphi(\bar{s}; \theta^i) & = 0. \end{aligned} \quad (3.16)$$

If the number of samples K is sufficiently large and ψ is positive almost everywhere in $\mathcal{S} \times \mathcal{A}_s$, the existing theory suggests that (3.16) should provide a good randomized approximation of BALP (De Farias and Van Roy 2004, Calafiore and Campi 2006). Please also see Proposition 3 in Chapter 2. We utilize formulation (3.16) in our numerical experiments in §3.7 and show that it is effective on our perishable inventory control instances. However, in general, (3.16) may be an unbounded model when K is small or a poor choice of ψ is used. In addition, we may need to use a large value of K to obtain a good approximation of BALP from (3.16).

Constraint generation. Constraint generation is a complementary approach to constraint sampling. This method starts by solving the following version of BALP, denoted $\text{BALP}_N[\mathcal{H}_h]$:

$$\begin{aligned} \max_{(\eta, \beta) \in \mathbb{R}^{N+2}} \quad & \eta \\ \eta + \sum_{i=1}^N \beta_i \left(\varphi(s; \theta^i) - \mathbb{E}[\varphi(s'; \theta^i) | s, a] \right) & \leq c(s, a) \quad \forall (s, a) \in \mathcal{H}_h \\ \beta_0 + \sum_{i=1}^N \beta_i \varphi(\bar{s}; \theta^i) & = 0, \end{aligned}$$

where \mathcal{H}_h is a set of h state-action pairs. For example, this set can be constructed by sampling state-action pairs from ψ in constraint sampling. By solving $\text{BALP}_N[\mathcal{H}_h]$ for a small h , we obtain an initial solution (η^0, β^0) . Here, we assume $\text{BALP}_N[\mathcal{H}_h]$ is bounded, so (η^0, β^0) exists. Utilizing (η^0, β^0) , constraint generation method requires solving the following separation problem (SP; e.g., see §3.1 of [Adelman and Klabjan 2012](#)):

$$(\hat{s}^0, \hat{a}^0) = \arg \min_{s, a} \left\{ c(s, a) - \eta - \sum_{i=1}^N \beta_i \left(\varphi(s^i; \theta^i) + \mathbb{E}[\varphi(s'; \theta^i) | s, a] \right) \right\},$$

Upon solving SP and obtaining (\hat{s}^0, \hat{a}^0) , we define the updated set \mathcal{H}_{h+1} as $\mathcal{H}_{h+1} := \mathcal{H}_h \cup \{(\hat{s}^0, \hat{a}^0)\}$ and solve program $\text{BALP}_N[\mathcal{H}_{h+1}]$ that has the following additional (most violating) constraint compared to $\text{BALP}_N[\mathcal{H}_h]$:

$$\eta + \sum_{i=1}^N \beta_i \left(\varphi(\hat{s}^0; \theta^i) - \mathbb{E}[\varphi(s'; \theta^i) | \hat{s}^0, \hat{a}^0] \right) \leq c(\hat{s}^0, \hat{a}^0).$$

This process of iteratively solving $\text{BALP}_N[\mathcal{H}_h]$, solving SP, and updating \mathcal{H}_h is repeated for H iterations. We stop when the optimal objective value of SP is non-negative. If the optimal objective value of SP at iteration h is non-negative, then there is no more “violating” constraint and thus $\text{BALP}_N[\mathcal{H}_h]$ has the same optimal objective value as BALP_N . In this case, we stop the process and use the terminal BFA weights from $\text{BALP}_N[\mathcal{H}_h]$, which is an optimal solution to BALP_N .

The bottleneck in the constraint generation approach is solving SP, which is a nonlinear program in general. There are multiple sources of non-linearity: the MDP cost function, the

MDP transition kernel, and basis functions. Because Stump bases and ReLU bases are piecewise constant and piecewise linear, respectively, we can mitigate non-linearity associated with the basis functions in SP formulation. Therefore, if the MDP has structure, e.g., both cost function and transition kernel are linear, SP can become a linear or mixed-integer program, depending on the basis functions structure. For example, in the generalized joint replenishment problem considered in §3.6, the MDP cost function and transition kernel have linear structures. Thus, we show that SP becomes a mixed-integer linear program for this application when using Stump basis functions, which are piecewise constant. It is easy to see that a similar mixed-integer linear program exists when using ReLU bases to formulate BALP for this application.

3.6 Generalized Joint Replenishment

The generalized joint replenishment (GJR) involves the replenishment of a collection of products that are consumed at a fixed and deterministic rate and are coupled via a shared replenishment capacity [Adelman and Klabjan 2012](#), abbreviated as [AK](#). We present the average-cost deterministic semi-MDP formulation of this problem using the formulation in [AK](#). Note that the methodology we presented in this chapter, which is designed for average-cost MDPs, can be extended to cover deterministic semi-MDPs.

Consider managing the replenishment of inventories across J products over a continuous time horizon. Each product j is consumed at a finite and deterministic rate $\lambda_j > 0$. We denote by $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_J)$ the vector of these rates. A state vector $s = (s_1, s_2, \dots, s_J)$ encodes the inventory levels of these products measured in normalized units, where each component $s_j \geq 0$ is non-negative for all $j \in \{1, 2, \dots, J\}$. A zero value for the j -th state component signals that the

j -th product is stocked out. Since the replenishment time can be postponed when there is no product that is stocked out, we can assume that at least one product has zero inventory in the state vector. Thus, the state space of GJR is given by $\mathcal{S} := \{s : 0 \leq s \leq S, s_j = 0 \text{ for some } j \in \{1, 2, \dots, J\}\}$, where $S = (S_1, S_2, \dots, S_J) \in (0, \infty)^J$ is a vector of maximum inventory levels. The replenishment decision is specified by action $\mathbf{a} \in \mathbb{R}_+^J$. This decision at a given state $s \in \mathcal{S}$ belongs to the set $\mathcal{A}_s := \{\mathbf{a} \in \mathbb{R}_+^J : s + \mathbf{a} \leq S, \sum_{j=1}^J a_j \leq A\}$. Here, constant $A \in \mathbb{R}_+$ denotes a capacity constraint on the total replenishment amount. The immediate MDP cost $c(s, \mathbf{a})$ for GJR has two components. The first one is a fixed value, denoted $c_{\text{supp}(\mathbf{a})}$, that depends on the subset of products replenished, denoted $\text{supp}(\mathbf{a}) := \{j \in \{1, \dots, J\} | a_j > 0\}$. The second one is given by the variable cost $\sum_{j=1}^J (2s_j a_j + a_j^2) h_j / 2\lambda_j$ with $h_j \geq 0$ denoting the holding cost per unit per time of product j . Because the usage rate is deterministic, the time until the next replenishment and the MDP transition kernel are both given by deterministic functions. Specifically, the time until the next replenishment is $\tau(s, \mathbf{a}) := \min_j \{(s_j + a_j) / \lambda_j\}$ if the system is currently in state s and action \mathbf{a} is taken. The system transitions to the new state $s' = s + \mathbf{a} - \tau(s, \mathbf{a})\lambda$ from current state s when taking action \mathbf{a} . The optimality equation for this deterministic semi-MDP is slightly different from (3.4) and is given by:

$$u(s) = \inf_{\mathbf{a} \in \mathcal{A}} \{c(s, \mathbf{a}) - \eta \tau(s, \mathbf{a}) + u(s + \mathbf{a} - \tau(s, \mathbf{a})\lambda)\}, \quad \forall s \in \mathcal{S},$$

where we use the definition of GJR transition function, that is, $s' = s + \mathbf{a} - \lambda \tau(s, \mathbf{a})$, to derive this equation.

[AK](#) approximate $u(s)$ using a (static) affine component $\beta_0 - \sum_{j=1}^J \beta_{1,j} s_j$ and an adaptive component $\sum_{i=1}^I \beta_{2,i} f^i\left(\sum_{j=1}^J r_j^i s_j\right)$ with I terms, where $f^i : \mathbb{R} \mapsto \mathbb{R}$ is a piecewise linear ridge function and $r^i \in \mathbb{R}^J$ is a ridge vector. Putting these two components together results in the following [AK](#) BFA:

$$u(s; \beta) := \beta_0 - \sum_{j=1}^J \beta_{1,j} s_j - \sum_{i=1}^I \beta_{2,i} f^i\left(\sum_{j=1}^J r_j^i s_j\right).$$

[AK](#) decompose η according to $\eta(\lambda) = \hat{\eta} + \sum_{j=1}^J \beta_{1,j} \lambda_j$, where $\hat{\eta}$ is an intercept and each $\beta_{1,j}$ can be interpreted as the marginal value associated with product j . This breakdown is not needed for the tractability of their algorithm but facilitates managerial interpretation. Putting together [AK](#) BFA with their decomposition of η , we obtain the following ALP solved by [AK](#), which we refer to this ALP as AK-ALP,

$$\begin{aligned} \max_{\hat{\eta}, \beta} \quad & \hat{\eta} + \sum_{j=1}^J \beta_{1,j} \lambda_j \\ & \hat{\eta} \tau(s, a) - \sum_{j=1}^J \beta_{1,j} a_j - \sum_{i=1}^I \beta_{2,i} \left[f^i\left(\sum_{j=1}^J r_j^i s'_j\right) - f^i\left(\sum_{j=1}^J r_j^i s_j\right) \right] \leq c(s, a), \quad \forall (s, a) \in \mathcal{S} \times \mathcal{A}_s. \end{aligned}$$

[AK](#) approach the solution of AK-ALP using constraint generation, which involves solving mixed integer programs. In addition, their algorithm dynamically generates ridge basis functions to update BFA via information from dual of AK-ALP formulation for GJR. We implemented AK-ALP as a benchmark following the details in [AK](#).

To be consistent with [AK](#), we use the same approximation $\eta(\lambda)$ for η , and we also define our BFA as follows:

$$u(s; \beta) := \beta_0 - \sum_{j=1}^J \beta_{1,j} s_j - \sum_{i=1}^N \beta_{2,i} \varphi(s; \theta^i), \quad (3.17)$$

where the adaptive basis function component in the AK-ALP BFA has been substituted with random basis functions. We let $\varphi(s; \theta)$ be Stump basis function defined in [§3.4.3](#) (please also see [Table I](#) in [Chapter 2](#)).

3.6.1 Constraint Generation for Stump Basis Functions

We show that constraint generation can be used to solve BALP and PALP formulated using Stump basis functions. Motivated by the mixed integer linear programming reformulation of the SP when holding cost is zero in [§3.1](#) of [AK](#), we discuss the analogous mixed integer linear programming formulation of SP for BALP. Recall that for Stump basis functions with $\varphi(\cdot) = \text{sgn}(\cdot)$ we have that intercept θ_0 is drawn from a uniform distribution over $[-c_\rho, c_\rho]$ and the remaining elements of θ are sampled from a uniform distribution defined on the discrete set $\{e^1, \dots, e^d\}$. For the ease of notation, write each sample θ^i as the pair (ω_i, ℓ_i) , where $\omega_i \in [-c_\rho, c_\rho]$ and $\ell_i \in \{0, 1, \dots, d\}$. Using the transition time $\tau(s, a) = \min_j \{\frac{s_j + a_j}{\lambda_j}\}$ and the

BFA in (3.17), it can be verified that SP in §3.5 for GJR is equivalent to following mixed-integer linear program:

$$\begin{aligned}
 \text{SP} \equiv & \min_{(G, Q, Q', s, a, t, s', Z, Z')} \left[\left(c' + \sum_{j=1}^J c_j'' G_j \right) - \left(\hat{\eta} t + \sum_{j=1}^J \beta_{1,j} a_j + \sum_{i=1}^N \beta_{2,i} (Z'_i - Z_i) \right) \right] \\
 & \sum_{j=1}^J Q_j \geq 1, & j = 1, 2, \dots, J, \\
 & \sum_{j=1}^J G_j \geq 1, & j = 1, 2, \dots, J, \\
 & \sum_{j=1}^J a_j \leq A, \\
 & \sum_{j=1}^J Q'_j \geq 1, \\
 & a_j \leq S_j G_j, & j = 1, 2, \dots, J, \\
 & s'_j = s_j + a_j - \lambda_j t, & j = 1, 2, \dots, J, \\
 & s_j + a_j \leq S_j, & j = 1, 2, \dots, J, \\
 & s_j \leq S_j (1 - Q_j), & j = 1, 2, \dots, J, \\
 & s'_i \leq S_j (1 - Q'_j), & j = 1, 2, \dots, J, \\
 & Q_j \leq G_j, & j = 1, 2, \dots, J, \\
 & Z_i = \text{sgn}(s'_{\ell_i} - \omega_i), & i = 1, \dots, N, \\
 & Z'_i = \text{sgn}(s_{\ell_i} - \omega_i), & i = 1, \dots, N, \\
 & G, Q, Q', \text{ binary}, \\
 & Z, Z', \text{ integer}, \\
 & s, a, t, s', \text{ non-negative.}
 \end{aligned}$$

In the above mixed-integer linear program, the variable G_j is one if product j is replenished and zero otherwise. Constraint $\sum_{j=1}^J G_j \geq 1$ ensures that at least one product is replenished. If $G_j = 1$ for some product $j \in \{1, 2, \dots, J\}$, i.e., it is replenished, then constraint $a_j \leq S_j$ ensures that the replenishment decision a_j can take any feasible replenishment value. If $G_j = 0$, product j is not replenishment and thus constraint $a_j \leq S_j$ enforces $a_j = 0$. Constraint $s'_j = s_j + a_j - \lambda_j t$ models the MDP transition function. Constraints $s_j + a_j \leq S_j$ and $\sum_{j=1}^J a_j \leq A$ guarantee that the state-action pair (s, a) adheres to the inventory and the replenishment capacities, respectively. For product $j \in \{1, 2, \dots, J\}$, if binary variable Q_j is one, product j is stocked out at the current decision time, i.e., $s_j = 0$, and if Q'_j is one, then this product will be stocked out in the next decision epoch, i.e., $s'_j = 0$. Constraints $\sum_{j=1}^J Q_j \geq 1$ and $\sum_{j=1}^J Q'_j \geq 1$ ensure at least a product at the current and one product at the next decision epoch is stocked out. If $G_j = 0$ for some product j , then it should not be replenished, and thus $Q_j = 0$ via constraint $Q_j \leq G_j$; otherwise, $Q_j \in \{0, 1\}$, that is, we can either replenish a stocked-out product or a product with a non-zero inventory level. Integer variables $Z_i \in \{-1, 0, 1\}$ and $Z'_i \in \{-1, 0, 1\}$ model the value of random basis functions $\varphi(s; \theta^i)$ and $\varphi(s'; \theta^i)$, respectively. The sign function defining Stump bases can be implemented in a commercial solver as a piecewise constant function using a big-M formulation or approximately as a piecewise linear function.

The setup of AK-ALP and BALP differ mainly in how adaptive basis functions are generated. In the former approach, ridge basis functions are generated via an application-specific algorithm, whereas, in the latter case, we sample Stump basis functions. Because we solve AK-ALP and PALP via constraint generation, the optimal objective values of these programs provide a lower

bound on the optimal policy cost. To estimate policy cost, we simulate the policy only based on the static part of the BFA because AK showed that this affine BFA suffices to obtain good policies for GJR instances without holding cost. We perform policy simulation by solving K-step greedy policy optimization problem discussed in §3.2 of AK for this affine BFA. We highlight that it is possible to cast this K-step greedy policy optimization problem when using a BFA based on Stump bases as a mixed-integer linear program, similar to SP.

3.6.2 Instances and Computational Setup

We conduct numerical experiments on 14 instances of the GJR problem based on Table 2 of AK. These instances have a zero holding cost, and the number of products (J) is varied between 4 and 6. Because the holding cost is zero, the MDP cost function becomes $c(s, a) = c_{\text{supp}(a)} = c' + \sum_{j \in \text{supp}(a)} c_j''$, where $c' \geq 0$ and $c_j'' \geq 0$ are constant and product-specific fixed costs, respectively. AK set c' to 100 and sample each c_j'' from a uniform distribution over the range $[0, 60]$ independently. The usage rate λ_j is distributed uniformly in the interval $[0, 10]$. The vector of maximum inventory levels S is chosen based on two random variables u_j and α_j associated with each product $j \in \{1, 2, \dots, J\}$ that are distributed uniformly over $[0, 1]$ and $\{2, 4, 8\}$, respectively. These random variables are independent across products. The j -th upper bound S_j on the inventory level is defined in three ways, labeled “random”, “constant”, and “discrete”, as $S_j = 10\lambda_j u_j + \lambda_j$, $S_j = \sum_{k=1}^J \lambda_k (u_k + \frac{1}{J})$, and $S_j = \alpha_j \sum_{k=1}^J \lambda_k (u_k + \frac{1}{J})$, respectively. The joint replenishment capacity A equals the summation of the first $z\%$ of the smallest storage limits $\{S_j : j = 1, 2, \dots, J\}$, where z varies in set $\{50, 60, 67, 75, 80, 100\}$ across instances.

We formulate BALP_N using N Stump bases with parameter $c_\rho := \max_j \{S_j\}$. We compute lower bounds using BALP BFA, AK-ALP BFA, and affine BFA. We numerically observed that performing SP for BALP_N when N is large (e.g., $N \geq 50$) is challenging. We thus increase N in batches of size 10 and solve multiple BALP and SP problems to ensure tractability. No information is shared across multiple iterations in this process. To ensure the tractability of SP, we also add a 1-norm constraint to BALP, which reduces the number of times we need to solve SP. We also simulate the greedy policy with respect to the affine BFA using the K -step greedy policy optimization problem with $K = 5$. For each method and each instance, we compute an optimality gap, which is the difference between the affine BFA upper bound and the lower bound from each method, expressed as a percentage of the lower bound. We stop BALP and AK-ALP when the optimality gap drops below 2%. If this criterion is not met, we stop these programs after 2 hours of runtime.

3.6.3 Results

Table [XVI](#) reports upper bounds obtained from the greedy policy with respect to the affine BFA, in addition to lower bounds from affine BFA, BALP, and AK-ALP. It also reports the optimality gap computed as (upper bound - lower bound) expressed as a percentage of the lower bound, where the upper bound is obtained from affine VFA, and the lower bound can be the lower bound from either of the three models. As explained, affine BFA leads to near-optimal lower and upper bounds on 8 instances for which we did not run algorithms BALP and AK-ALP. Therefore, entries of the table of these methods and such instances are empty. BALP and AK-ALP result in significantly better lower bounds than the affine BFA model. The maximum

Table XVI: Comparison of BALP and AK-ALP lower and upper bounds in generalized joint replenishment problem instances.

J	Instance	UB	Affine		BALP		AK-ALP	
			LB	Gap	LB	Gap	LB	Gap
4	1	184.6	184.3	0.1%				
	2	93.7	92.7	1.0%				
	3	179.3	165.0	8.7%	175.3	2.3%	175.2	2.4%
	4	171.2	160.0	7.0%	170.1	0.7%	169.6	0.9%
	5	69.3	68.7	0.8%				
	6	29.4	29.4	0.0%				
6	7	146.0	145.8	0.2%				
	8	91.8	90.5	1.5%				
	9	91.0	90.2	0.9%				
	10	117.8	114.5	3.0%	115.3	2.2%	115.3	2.2%
	11	107.8	104.9	2.8%	105.9	1.7%	105.9	1.8%
	12	108.2	104.9	3.2%	106.9	1.2%	107.0	1.1%
	13	53.1	52.2	1.6%				
	14	31.9	29.0	9.9%	31.7	0.7%	31.7	0.8%
	15	31.8	29.0	9.5%	31.7	0.4%	31.7	0.5%

improvements from BALP and AK-ALP are 9.3% and 9.2%, respectively. The average of BALP and AK-ALP optimality gaps are 1.4% and 1.7%, respectively, which shows these models are near-optimal. Our results suggest that lower bounds from BALP, which does not exploit the structure of the GJR problem to generate bases, are comparable to the AK-ALP model, which performs basis function selection by exploiting problem structure. In addition, in the case of GJR, because the upper bounds based on affine BFA are near-optimal upper bounds, there is no need for our randomized multi-shot approximation mechanism in Algorithm 3. Therefore, BALP is enough to close the optimality gap, and there is no need to use PALP.

Table XVII: Parameters of five-dimensional perishable inventory control instances.

Instance	Holding cost c_h	Disposal cost c_d	Backlogging cost c_b	Demand STD σ	c_l
1	1	8	2	5	1000
2	1	8	2	2	1000
3	1	2	8	5	1000
4	1	2	8	2	1000
5	2	8	5	5	1000
6	2	8	5	2	1000

3.7 Perishable Inventory Control Problem

We perform a numerical study on an average-cost variant of the perishable inventory control problem considered in §3.7 of Chapter 2. In §3.7.1, we discuss problem instances and benchmarks, and we report our results in §3.7.2.

3.7.1 Instances and Benchmarks

We revisit the perishable inventory control problem studied in §2.6 of Chapter 2. Specifically, we focus on our five-dimensional instances in Table III. We repeat the parameters of these instances in Table XVII. Instead of using a discounted cost function, we use the following average cost function:

$$c(s, \mathbf{a}) := c_o \mathbf{a} + \mathbb{E}_D \left[c_h \left[\sum_{i=1}^{l-1} s_i - (D - s_0)_+ \right]_+ + c_d (s_0 - D)_+ + c_b \left[D - \sum_{i=0}^{l-1} s_i \right]_+ + c_l \left[\underline{s} + D - \sum_{i=0}^{l-1} s_i \right]_+ \right].$$

Compared to the MDP cost function in §2.6, the above cost function does not have any discount factor.

We formulate BALP and PALP using Fourier basis functions. We use our parameter choices in Chapter 2 to set up our experiments in this section. To solve BALP and PALP, we use the constraint sampling approach discussed in §3.5 with $K = 200,000$ state-action pairs sampled from a uniform distribution over the hyper-cube $\mathcal{S} \times \mathcal{A}_s = [\underline{s}, \bar{a}] \times [0, \bar{a}]^d$. We set the number of random basis functions N to 300 for BALP, i.e., we solve BALP_{300} . We also consider a modification of BALP_N that first solves BALP_{300} and then solves PALP_N that has a fixed average-cost value of η_{300}^{BA} obtained from BALP_{300} and includes a uniform state-relevance distribution. We refer to this version of BALP_N as BALP_{300}^* . Moreover, we consider our randomized multi-shot approximation approach in Algorithm 3. We use the notation $\text{ALP}_{150,150}^{\text{MS}}$ to refer to this approach. Specifically, $\text{ALP}_{150,150}^{\text{MS}}$ runs Step 1 of Algorithm 3 with $N = 150$ random bases (the first subscript) and runs steps 2 and 3 of this algorithm for $Q = 6$ iterations using batch size $B = 25$ that results in BQ of 150 (the second subscript). Method $\text{ALP}_{150,150}^{\text{MS}}$ relies on solving PALP. For each PALP solved in $\text{ALP}_{150,150}^{\text{MS}}$, we use a uniform state-relevance distribution ν .

As a benchmark, we considered FALP and self-guided FALP models proposed in Chapter 2. We run discounted-cost model FALP_{300} with $\gamma = 0.999$, but we add constraint $\beta_0 + \sum_{i=1}^N \beta_i \varphi(\bar{s}; \theta^i) = \mathbf{m}_\gamma$ to this model, where \mathbf{m}_γ is some constant. The addition of this constraint is based on the discussion on pages 84–85 of [Hernández-Lerma and Lasserre \(1996\)](#) that constructs a BFA from a VFA. We consider two choices for \mathbf{m}_γ that are $\mathbf{m}_\gamma = 0$ and $\mathbf{m}_\gamma = v_0$ with v_0 being the optimal objective value of FALP with the intercept-only VFA having $N = 0$ random bases. We use notations $\gamma\text{-FALP}_{300}[0]$ and $\gamma\text{-FALP}_{300}[v_0]$ for FALP models with $\mathbf{m}_\gamma = 0$ and $\mathbf{m}_\gamma = v_0$, respectively. Both these models have $N = 300$ random bases. For self-guided

FALP, we run Algorithm 3 in Chapter 2 for $Q = 7$ iterations. We denote this method by $\gamma\text{-FALP}_{300,7}^{\text{SG}}[v_0]$.

We approximate expectations in all aforementioned models using sample average approximations constructed using 2,000 iid samples. We run each model 10 times with freshly sampled random basis function parameters. We simulate policies from initial state $s_0 = (5, 5, \dots, 5) \in \mathbb{R}^d$. We simulate 100 trajectories of length 10,000 to estimate the long-run average cost of each method.

Because we solve BALP using constraint sampling, its optimal objective value does not provide a valid lower bound on the optimal cost. We thus compute a lower bound on η^* based on $\gamma\text{-FALP}_{300}[v_0]$ VFA. Specifically, we plug in $\gamma\text{-FALP}_{300}[v_0]$ VFA into our heuristic based on constraint violation learning approach in §2.12.1 of Chapter Chapter 2 to obtain a lower bound on the optimal cost of the discounted-cost problem. Denote this quantity by LB_γ . It is known that $(1 - \gamma)\text{LB}_\gamma$ is a lower bound on the optimal policy cost of the average-cost problem, i.e., $(1 - \gamma)\text{LB}_\gamma \leq \eta^*$ (please see pages 84–85 of [Hernández-Lerma and Lasserre 1996](#)). Utilizing this lower bound, we compute the optimality gap for each method and instance, similar to the optimality gap we computed in §3.6.

3.7.2 Results

Table XVIII reports upper bounds (UBs) and lower bounds (LBs) obtained from $\gamma\text{-FALP}_{300}[0]$, $\gamma\text{-FALP}_{300}[v_0]$, $\gamma\text{-FALP}_{300,7}^{\text{SG}}[v_0]$, BALP_{300} , BALP_{300}^* , and $\text{ALP}_{150,150}^{\text{MS}}$. We also report the lower bounds based on $\gamma\text{-FALP}_{300}[v_0]$ and optimality gaps computed using this lower bound in Table XVIII. Note that the lower bound values, upper bounds, and optimality gaps are averages across

Table XVIII: Comparison of methods on perishable inventory control problem instances.

Dimension	Instance	γ -FALP ₃₀₀ [0]			γ -FALP ₃₀₀ [v_0]		γ -FALP _{300,7} ^{SG} [v_0]		BALP ₃₀₀		BALP [*] ₃₀₀		ALP ^{MS} _{150,150}	
		LB	UB	Gap	UB	Gap	UB	Gap	UB	Gap	UB	Gap	UB	Gap
5	1	66	115	74%	98	49%	101	53%	86	30%	84	28%	72	8%
	2	57	610	973%	615	982%	1058	1761%	75	32%	85	49%	60	5%
	3	67	1832	2651%	1871	2710%	1070	1506%	666	900%	82	24%	69	4%
	4	60	1442	2300%	1105	1738%	1081	1699%	162	170%	77	29%	61	1%
	5	71	654	816%	639	794%	100	40%	90	26%	89	24%	77	7%
	6	62	1303	2008%	1203	1847%	836	1252%	750	1114%	82	33%	64	4%

10 trials. All discounted-cost models deliver poor policies, suggesting solving the discounted-cost problems does not provide good policies for our average-cost problem instances. BALP₃₀₀ leads to policies with the optimality gap of less than 32% on 3 out of 6 instances but highly sub-optimal policies on the other 3 instances. The worst-case performance of BALP₃₀₀ is on the 6th instance, where its optimality gap is 1114%. Interestingly, when we correct this model and consider BALP^{*}₃₀₀, across all 6 instances, this version of BALP delivers policies of at most 49%. The best method is ALP^{MS}_{150,150}, which provides near-optimal control policies and beats all other models. These results underscore the value of our randomized multi-shot approximation approach relying on PALP. Moreover, we note that γ -FALP₃₀₀ VFA with $m_\gamma = 0$ leads to a great lower bound, as witnessed by the low optimality gap values of γ -FALP₃₀₀.

3.8 Conclusion

Our work focuses on solving large-scale average-cost Markov decision processes (MDPs) by employing an approximate linear programming approach. This method involves approximating MDP bias functions through a linear combination of basis functions and solving an approximate linear program (ALP) to compute the weights of these basis functions. It is known that when basis functions deliver a good approximation of the MDP bias function, ALP generates tight lower bounds on the optimal policy cost. However, this method fails to provide good bias function approximations (BFA) and control policies.

We introduce a new approximate linear programming (ALP) approach by combining a two-phase ALP model in [De Farias and Van Roy \(2002\)](#) with a randomized multi-shot approximation method for discounted-cost MDPs in [Pakiman et al. \(2020\)](#). Our approach has two steps. First, we use universal random basis functions to formulate an ALP that ensures delivering a near-optimal lower bound. We develop a finite probabilistic convergence rate for this lower bound obtained from our method. Second, we solve a sequence of ALP models that iteratively refine their formulations using previously computed BFAs. We show that this iterative randomized multi-shot approximation mechanism ensures a worst-case measure of policy performance is improving. We applied our approach to two inventory management problems, yielding near-optimal lower bounds and effective control policies.

APPENDICES

3.9 Addendum to Assumption 7

There are known conditions documented in the literature that validate Assumption 7. A set of such conditions is available in [Gordienko and Hernández-Lerma \(1995\)](#). Specifically, Theorem 2.8 in [Gordienko and Hernández-Lerma \(1995\)](#) guarantees that our Assumption 7 holds if Assumptions 2.2, 2.3, 2.4, and 2.7 from that paper are satisfied. Another set of conditions relies on a “vanishing-discount” argument outlined in [Hernández-Lerma and Lasserre \(1996\)](#). We present these conditions relative to our setting in Assumptions 10–11 below. When these assumptions hold, Theorem 5.5.4 in [Hernández-Lerma and Lasserre \(1996\)](#) ensures the validity of Assumption 7.

Assumption 10 *The MDP cost function c is bounded below, lower semicontinuous, and inf-compact. The MDP transition kernel Q is strongly continuous: given any measurable bounded function $V : \mathcal{S} \mapsto \mathbb{R}$, the mapping $(s, a) \mapsto \int_{\mathcal{S}} V(s')P(ds'|s, a)$ is bounded and continuous over $\mathcal{S} \times \mathcal{A}_s$.*

The first part of Assumption 10 is the same as Assumption 4.2.1 in [Hernández-Lerma and Lasserre \(1996\)](#). Note that lower semicontinuity and inf-compactness are defined before Condition 3.3.4 and in Condition 3.3.3 of [Hernández-Lerma and Lasserre \(1996\)](#), respectively. The second part of Assumption 10 is the same as Part (b) of Assumption 4.2.1 in [Hernández-Lerma and Lasserre \(1996\)](#).

Define the α -discount value function $V_\alpha : \mathcal{S} \mapsto \mathbb{R}$ for some discount factor $\alpha \in [0, 1)$ at $s \in \mathcal{S}$ as:

$$V_\alpha(s) := \inf_{\pi: \mathcal{S} \mapsto \mathbb{R}} \mathbb{E}^\pi \left[\sum_{t=0}^{\infty} \alpha^t c(s_t, \pi(s_t)) \mid s_0 = s \right].$$

Assumption 11 *There exists a state $\hat{s} \in \mathcal{S}$, a non-negative function $w : \mathcal{S} \mapsto [0, \infty)$, a number $\underline{\alpha} \in (0, 1)$, and constants $M, N \geq 0$ such that (i) inequality $(1 - \alpha)V_\alpha(\hat{s}) \leq M$ holds for every $\alpha \in [\underline{\alpha}, 1)$, and (ii) inequality $-N \leq V_\alpha(s) - V_\alpha(\hat{s}) \leq w(s)$ holds for every $s \in \mathcal{S}$ and $\alpha \in [\underline{\alpha}, 1)$. Moreover, w is measurable and satisfies $\int_{\mathcal{S}} w(ds')P(s'|s, a) < \infty$ for every $(s, a) \in \mathcal{S} \times \mathcal{A}_s$.*

Assumption 11 is an integration of Assumptions 5.4.1 and Assumption 5.5.1 (a) in [Hernández-Lerma and Lasserre \(1996\)](#). It can be verified for different problems, for example, the discounted Linear-Quadratic problem in Example 5.4.2 of [Hernández-Lerma and Lasserre \(1996\)](#). Under Assumption 11, the following lemma, which appears as a proof of Theorem 5.4.3 in [Hernández-Lerma and Lasserre \(1996\)](#), holds.

Lemma 4 *Under Assumption 11, there is a sequence $\{\alpha_n : n = 0, 1, \dots\} \subseteq [0, 1)$ approaching from below to 1 such that $\lim_{n \rightarrow \infty} (1 - \alpha_n)V_{\alpha_n}(s)$ exists and is a constant for all $s \in \mathcal{S}$.*

Using the sequence of functions V_{α_n} in the above lemma, we require the following assumption that is the same as Assumption 5.5.1 (b) in [Hernández-Lerma and Lasserre \(1996\)](#).

Assumption 12 *The family of functions $\{V_{\alpha_n}(s) - V_{\alpha_n}(\hat{s}) : n = 0, 1, \dots\}$, where $\{\alpha_n : n = 0, 1, \dots\}$ and \hat{s} are defined in Lemma 4 and Assumption 11, respectively, is equicontinuous.*

For the formal definition of equicontinuity in Assumption 11, please see Remark 5.5.2 in [Hernández-Lerma and Lasserre \(1996\)](#). Once again, if Assumptions 10–11 hold, Theorem 5.5.4 in [Hernández-Lerma and Lasserre \(1996\)](#) ensures the validity of Assumption 7 in our paper.

3.10 Proofs

Proof of Proposition 14.

We first show that inequality $\eta \leq \eta^*$ holds for every feasible solutions $(\eta, \mathbf{u}) \in \mathbb{R} \times \mathcal{U}$ to BELP. Consider an initial state $s = s_0 \in \mathcal{S}$ and a policy π such that $\eta^\pi = \text{AC}(s_0; \pi) < \infty$. Let s_n and \mathbf{a}_n be the state and action, respectively, reach at stage n when following policy π . For every BELP feasible solution $(\eta, \mathbf{u}) \in \mathbb{R} \times \mathcal{U}$, it holds that

$$\mathbf{u}(s_0) \leq \mathbb{E}[\mathbf{c}(s_0, \mathbf{a}_0) - \eta + \mathbf{u}(s_1) \mid s_0, \mathbf{a}_0].$$

Iterating the above inequality for $n \geq 1$ times, we obtain the following inequality:

$$\mathbf{u}(s_0) \leq \mathbb{E}_s^\pi \left[\sum_{i=0}^n \mathbf{c}(s_i, \mathbf{a}_i) \right] - n\eta + \mathbb{E}[\mathbf{u}(s_{n+1}) \mid s_n, \mathbf{a}_n].$$

If we divide the above inequality by n , rearrange its terms, and take its limit when $n \rightarrow \infty$, we obtain the following upper bound on η :

$$\eta \leq \lim_{n \rightarrow \infty} \left\{ \frac{\mathbb{E}_s^\pi [\sum_{i=0}^n \mathbf{c}(s_i, \mathbf{a}_i)]}{n} + \frac{\mathbb{E}[\mathbf{u}(s_{n+1}) \mid s_n, \mathbf{a}_n] - \mathbf{u}(s_0)}{n} \right\} = \text{AC}(s_0; \pi) = \eta^\pi. \quad (3.18)$$

The first equality above holds since $AC(s_0; \pi) < \infty$ and thus limit $\lim_{n \rightarrow \infty} \frac{\mathbb{E}_s^\pi[\sum_{i=0}^n c(s_i, a_i)]}{n}$ exists and equals $AC(s_0; \pi)$. Also, fraction $\frac{\mathbb{E}[u(s_{n+1})|s_n, a_n] - u(s_0)}{n}$ goes to zero when $n \rightarrow \infty$, noting that the numerator of this fraction is bounded because $u \in \mathcal{U}$ is a continuous function defined over a compact domain. Because (3.18) holds for every policy π with a finite cost $AC(s_0; \pi) < \infty$, it should hold for π^* . Using π^* in conjunction with (3.18) results in the required inequality $\eta \leq \eta^*$. Next, we show that pair $(\eta^*, u^*) \in \mathbb{R} \times \mathcal{U}$ is optimal to BELP. Because this pair solves optimality equation (3.4), it is a feasible solution to BELP. Thus, the optimal objective value of BELP is an upper bound on η^* . On the other hand, we observed that for every feasible solution $(\eta, u) \in \mathbb{R} \times \mathcal{U}$ to the BELP, η is a lower bound on η^* . Hence, the optimal objective value of BELP is, in fact, η^* , and the pair (η^*, u^*) is an optimal solution to BELP. ■

Proof of Proposition 15.

Recall the definition of $\mathcal{B} = \{\beta : \Theta \mapsto \mathbb{R} : \|\beta/\rho\|_{2,\rho} < \infty, u(\bar{s}; \beta) = 0\}$ and $\mathcal{R} = \{u \in \mathcal{U} \mid \exists \beta \text{ s.t. } u(\cdot) = u(\cdot; \beta), u(\bar{s}; \beta) = 0, \|\beta/\rho\|_{2,\rho} < \infty\}$. We can rewrite \mathcal{B} in terms of elements in \mathcal{R} as $\mathcal{B} = \{\beta : \Theta \mapsto \mathbb{R} : u(\beta) \in \mathcal{R}\}$. Therefore, we can rewrite BFEP by replacing its decision variable $\beta \in \mathcal{B}$ with decision variable $u(\beta) \in \mathcal{R}$ as follows:

$$\begin{aligned} & \sup_{(\eta, u(\beta)) \in \mathbb{R} \times \mathcal{R}} \eta \\ & \eta + u(s; \beta) - \mathbb{E}[u(s'; \beta)|s, a] \leq c(s, a), \quad \forall (s, a) \in \mathcal{S} \times \mathcal{A}_s. \end{aligned}$$

Because $\mathcal{R} \subseteq \mathcal{U}$, the feasible set of the above program (which is equivalent to BFEP) is a subset of the BELP feasible set. Thus, we have that the BFEP optimal objective value is upper bounded by η^* . Because pair (η^*, u^*) is feasible to BELP by Proposition 14 and $u^* = u(\beta^*) \in \mathcal{R}$ by

Assumption 8, we observe that pair $(\eta^*, \mathbf{u}(\boldsymbol{\beta}^*))$ is feasible to the above reformulation of BFEP. Thus, pair $(\eta^*, \boldsymbol{\beta}^*)$ is feasible to BELP. Therefore, η^* should be a lower bound on the BFEP optimal objective value. Since η^* is both an upper and a lower bound on the BFEP optimal objective value, it must be the BFEP optimal objective value. Hence, pair $(\eta^*, \boldsymbol{\beta}^*)$ is a BFEP optimal solution. \blacksquare

Proof of Proposition 16.

Let $(\eta, \boldsymbol{\beta}) \in \mathbb{R} \times \mathcal{B}$ be a feasible solution to IMP. It holds that

$$\eta \leq c(s, \pi^*(s)) + \mathbf{u}(s; \boldsymbol{\beta}) - \mathbb{E}[\mathbf{u}(s'; \boldsymbol{\beta}) \mid s, \pi^*(s)], \quad \forall s \in \mathcal{S}^*.$$

Integrating the inequalities above with respect to the invariant probability measure $\mu(\cdot; \pi^*)$ defined in (3.3), and employing the definition of \mathcal{S}^* , encompassing all states $s \in \mathcal{S}$ for which $\mu(s; \pi^*) > 0$, yields the following inequality:

$$\int_{\mathcal{S}} \eta \mu(ds; \pi^*) \leq \int_{\mathcal{S}} c(s, \pi^*(s)) \mu(ds; \pi^*) + \int_{\mathcal{S}} (\mathbf{u}(s; \boldsymbol{\beta}) - \mathbb{E}[\mathbf{u}(s'; \boldsymbol{\beta}) \mid s, \pi^*(s)]) \mu(ds; \pi^*).$$

If we combine the above inequality with properties $\int_{\mathcal{S}} \mu(ds; \pi^*) = \int_{\mathcal{S}^*} \mu(ds; \pi^*) = 1$ and $\eta^* = AC(s; \pi^*) = \int_{\mathcal{S}} c(s, \pi^*(s)) \mu(s; \pi^*) ds$, where the latter one holds for every $s \in \mathcal{S}$ due to Assumption 6, we obtain the following inequality:

$$\eta \leq \eta^* + \int_{\mathcal{S}} \mathbf{u}(s; \boldsymbol{\beta}) \mu(ds; \pi^*) - \int_{\mathcal{S}} \mathbf{u}(s'; \boldsymbol{\beta}) \left(\int_{\mathcal{S}} P(s', \pi^*(s)) \mu(s; \pi^*) ds \right) ds'.$$

Applying identity (3.3) for the choice of $\mathcal{X} = \{s'\}$ to the last term in the above inequality, it boils down to $\int_{\mathcal{S}} u(s'; \boldsymbol{\beta}) \mu(ds'; \pi^*)$. Therefore, we obtain inequality $\eta \leq \eta^*$ that holds for every feasible solution $(\eta, \boldsymbol{\beta}) \in \mathbb{R} \times \mathcal{B}$ to IMP. From the definition of $\boldsymbol{\beta}^{\text{IMP}}$, we have $u(s; \boldsymbol{\beta}^{\text{IMP}}) = u^*(s)$ for all $s \in \mathcal{S}^*$. Because pair (η^*, u^*) is feasible to all BLP constraints and $u(s; \boldsymbol{\beta}^{\text{IMP}}) = u^*(s)$ for all $s \in \mathcal{S}^*$, pair $(\eta^*, \boldsymbol{\beta}^{\text{IMP}})$ is feasible to IMP. Notably, this feasible pair represents an optimal solution to IMP since it yields the maximum attainable objective value of IMP, namely η^* . ■

Proof of Proposition 17.

We show $(\eta^*, \boldsymbol{\beta}^{\text{AC}})$ is an optimal solution to both BFEP. To show this result, we use duality theory for infinite-dimensional linear programs. Define set $\mathcal{K} := \{(s, a) : s \in \mathcal{S}, a \in \mathcal{A}_s\}$. [Klabjan and Adelman \(2006\)](#) provide primal dual linear programs for general semi-MDPs. Specifically, primal-dual pair (6) and (7) in their paper can be written in our setting as the following primal-dual pair:

$$z^p := \sup_{\eta, u} \quad \eta \quad (3.19)$$

$$\eta + u(s) - \mathbb{E}[u(s')|s, a] \leq c(s, a) \quad \forall (s, a) \in \mathcal{S} \times \mathcal{A}_s.$$

$$z^d := \inf_q \int_{\mathcal{K}} c(s, a) q(s, a) d(s, a)$$

$$\int_{\mathcal{K}} P(\mathcal{X}|s, a) q(s, a) d(s, a) = q((\mathcal{X} \times \mathcal{A}_s) \cap \mathcal{K}), \quad \forall \mathcal{X} \subseteq \mathcal{S},$$

$$\int_{\mathcal{K}} q(s, a) d(s, a) = 1,$$

$$q(s, a) \geq 0. \quad (3.20)$$

In above, $\mathbf{u} : \mathcal{S} \mapsto \mathbb{R}$ is a bounded measurable function and $\mathbf{q} : \mathcal{K} \mapsto \mathbb{R}$ is a signed measure with a finite total variation norm. Note that the difference between (3.19) and BELP is that the former is a relaxation of the latter because BELP requires $\mathbf{u} \in \mathcal{U}$. We proceed in two steps.

In Step (i), we show that (3.19) and (3.20) are consistent, i.e., they have feasible solutions. From Proposition 14, pair $(\boldsymbol{\eta}^*, \mathbf{u}^*)$ is feasible to BELP, so it is feasible to (3.19), which is a relaxation of BELP. It thus hold that $\boldsymbol{\eta}^* \leq \mathbf{z}^p$. Next, we show that (3.20) has a feasible solution. Recall Assumption 6. Define probability measure $\hat{\mu}(\mathbf{s}, \mathbf{a}) := \mu^*(\mathbf{s})\delta\{\mathbf{a} = \pi^*(\mathbf{s})\}$, where $\delta\{\mathbf{a} = \pi^*(\mathbf{s})\}$ is the Dirac measure evaluating to one if $\mathbf{a} = \pi^*(\mathbf{s})$ and zero otherwise. From the definition of invariant distribution $\mu^*(\cdot) = \mu(\cdot; \pi^*)$, for each $\mathcal{X} \subseteq \mathcal{S}$, we have

$$\int_{\mathcal{K}} P(\mathcal{X}|\mathbf{s}, \mathbf{a})\hat{\mu}(\mathbf{s}, \mathbf{a})d(\mathbf{s}, \mathbf{a}) = \int_{\mathcal{S}} P(\mathcal{X}|\mathbf{s}, \pi^*(\mathbf{s}))\mu^*(d\mathbf{s}) = \mu^*(\mathcal{X}) = \hat{\mu}((\mathcal{X} \times \mathcal{A}_{\mathbf{s}}) \cap \mathcal{K}).$$

In addition, we know that $\int_{\mathcal{K}} \hat{\mu}(d(\mathbf{s}, \mathbf{a})) = 1$ and $\hat{\mu}(\mathbf{s}, \mathbf{a}) \geq 0$. Hence, $\hat{\mu}$ is feasible to (3.20).

In Step (ii), we use complementary slackness for the primal-dual pair (3.19) and (3.20) (e.g., see Theorem 6.2.4 of Hernández-Lerma and Lasserre 1996). This result states that if triplet $(\boldsymbol{\eta}, \mathbf{u}, \mathbf{q})$ is such that pair $(\boldsymbol{\eta}, \mathbf{u})$ is feasible to primal problem (3.19), \mathbf{q} is feasible to dual problem (3.20), and identity $\int_{\mathcal{K}} \mathbf{q}(\mathbf{s}, \mathbf{a})(\mathbf{c}(\mathbf{s}, \mathbf{a}) - \mathbf{u}(\mathbf{s}) - \boldsymbol{\eta} + \mathbb{E}[\mathbf{u}(\mathbf{s}')|\mathbf{s}, \mathbf{a}]) d(\mathbf{s}, \mathbf{a}) = \mathbf{0}$ holds, then $(\boldsymbol{\eta}, \mathbf{u})$ is optimal to (3.19) and \mathbf{q} is optimal to (3.20). Now consider $((\boldsymbol{\eta}^*, \mathbf{u}(\boldsymbol{\beta}^{\text{AC}})), \hat{\mu})$. From the definition of weighting function $\boldsymbol{\beta}^{\text{AC}}$, pair $(\boldsymbol{\eta}^*, \mathbf{u}(\boldsymbol{\beta}^{\text{AC}}))$ is feasible to BELP and thus feasible to (3.19). As we already saw in Step (i), $\hat{\mu}$ is feasible to dual problem (3.20). Therefore, if

we show identity $\int_{\mathcal{K}} \hat{\mu}(s, a)(c(s, a) - u(s; \beta^{AC}) - \eta^* + \mathbb{E}[u(s'; \beta^{AC})|s, a]) \, d(s, a) = 0$ holds, then $((\eta^*, u(\beta^{AC})), \hat{\mu})$ is an optimal primal-dual solution. This identity holds as we can write:

$$\begin{aligned}
& \int_{\mathcal{K}} \hat{\mu}(s, a)(c(s, a) - u(s; \beta^{AC}) - \eta^* + \mathbb{E}[u(s'; \beta^{AC})|s, a]) \, d(s, a) \\
&= \int_{\mathcal{S}} \mu^*(s)(c(s, \pi^*(s)) - u(s; \beta^{AC}) - \eta^* + \mathbb{E}[u(s'; \beta^{AC})|s, \pi^*(s)]) \, ds \\
&= \int_{\mathcal{S}^*} \mu^*(s)(c(s, \pi^*(s)) - u(s; \beta^{AC}) - \eta^* + \mathbb{E}[u(s'; \beta^{AC})|s, \pi^*(s)]) \, ds \\
&= \int_{\mathcal{S}^*} \mu^*(s)(c(s, \pi^*(s)) - u^*(s) - \eta^* + \mathbb{E}[u^*(s')|s, \pi^*(s)]) \, ds \\
&= 0
\end{aligned}$$

The first and second equalities above follows from the definitions of $\hat{\mu}$ and \mathcal{S}^* , respectively. The third equality holds because $u(s; \beta^{AC}) = u^*(s)$ for all $s \in \mathcal{S}^*$. The last equality because (η^*, u^*) solve optimality equation (3.4). Therefore, $(\eta^*, u(\beta^{AC}))$ is an optimal solution to the primal model (3.19). Since $u(s; \beta^{AC})$ evaluates to zero at $s = \bar{s}$ due to the definition of β^{AC} , and function $u(\beta^{AC}) \in \mathcal{R} \subseteq \mathcal{C}$ is continuous, pair (η^*, β^{AC}) is feasible to BFEP. In fact, this pair is optimal because this feasible solution attains the BFEP optimal objective value η^* , guaranteed by Proposition 15. Hence, pair (η^*, β^{AC}) is an optimal solution of BFEP. ■

Proposition 21 (Proposition 7 in Chapter 2) *Suppose $\rho(\theta) \geq \underline{\rho}$, for all $\theta \in \Theta$ and Assumption 8 holds. Consider $\delta \in (0, 1]$ and a function $\mathbf{u}(\boldsymbol{\beta})$ with $\boldsymbol{\beta} \in \mathcal{B}$. Given N iid samples $\{\theta^i : i = 1, 2, \dots, N\}$ from ρ , there is a vector $\bar{\boldsymbol{\beta}} \in \mathbb{R}^N$ such that*

$$\left\| \mathbf{u}(s; \boldsymbol{\beta}) - \sum_{i=1}^N \bar{\beta}_i \varphi(s; \theta^i) \right\|_{\infty} \leq \text{Err}(N, \delta; \boldsymbol{\beta}), \quad (3.21)$$

with a probability of at least $1 - \delta$.

Corollary 2 *Suppose $\rho(\theta) \geq \underline{\rho}$, for all $\theta \in \Theta$ and Assumption 8 holds. Consider $\delta \in (0, 1]$ and a function $\mathbf{u}(\boldsymbol{\beta})$ with $\boldsymbol{\beta} \in \mathcal{B}$. Given N iid samples $\{\theta^i : i = 1, 2, \dots, N\}$ from ρ , if we let $\bar{\boldsymbol{\beta}} \in \mathbb{R}^N$ be the vector defined in Proposition 21, then there exists a vector*

$$\hat{\boldsymbol{\beta}} := \left(- \sum_{i=1}^N \bar{\beta}_i \varphi(\bar{s}; \theta^i), \bar{\beta}_1, \bar{\beta}_2, \dots, \bar{\beta}_N \right) \in \mathbb{R}^{N+1},$$

such that

$$\left\| \mathbf{u}(\boldsymbol{\beta}) - \mathbf{u}(\hat{\boldsymbol{\beta}}) \right\|_{\infty} \leq 2\text{Err}(N, \delta; \boldsymbol{\beta}), \quad (3.22)$$

with a probability of at least $1 - \delta$. Moreover, we have $\mathbf{u}(\bar{s}; \hat{\boldsymbol{\beta}}) = 0$, i.e., $\mathbf{u}(\hat{\boldsymbol{\beta}}) \in \mathcal{U}$.

Proof. From Proposition 21 and the fact that $\mathbf{u}(\bar{s}; \boldsymbol{\beta}) = 0$ since $\boldsymbol{\beta} \in \mathcal{B}$, we have

$$\begin{aligned} \left\| \mathbf{u}(\boldsymbol{\beta}) - \mathbf{u}(\hat{\boldsymbol{\beta}}) \right\|_{\infty} &\leq |\hat{\beta}_0| + \left\| \mathbf{u}(s; \boldsymbol{\beta}) - \left(\sum_{i=1}^N \bar{\beta}_i \varphi(s; \theta^i) \right) \right\|_{\infty} \\ &= \left| \mathbf{u}(\bar{s}; \boldsymbol{\beta}) - \sum_{i=1}^N \bar{\beta}_i \varphi(\bar{s}; \theta^i) \right| + \left\| \mathbf{u}(s; \boldsymbol{\beta}) - \left(\sum_{i=1}^N \bar{\beta}_i \varphi(s; \theta^i) \right) \right\|_{\infty} \end{aligned}$$

$$\leq 2\text{Err}(\mathbf{N}, \delta; \boldsymbol{\beta})$$

Applying the right hand side of the above inequality to (3.22), we obtain (3.22). Identity $\mathbf{u}(\bar{\mathbf{s}}; \hat{\boldsymbol{\beta}}) = \mathbf{0}$ is trivial given the definition of $\hat{\boldsymbol{\beta}}_0$. \blacksquare

Proof of Theorem 4.

Since pair $(\eta^*, \boldsymbol{\beta}^{\text{AC}})$ is optimal to BFEP by Proposition 17, we have $\mathbf{c}(\mathbf{s}, \mathbf{a}) - \eta^* \geq \mathbf{u}(\mathbf{s}; \boldsymbol{\beta}^{\text{AC}}) - \mathbb{E}[\mathbf{u}(\mathbf{s}'; \boldsymbol{\beta}^{\text{AC}})|\mathbf{s}, \mathbf{a}]$ for every $(\mathbf{s}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}_s$. Applying Corollary 2 to $\boldsymbol{\beta}^{\text{AC}} \in \mathcal{B}$, there exists a vector $\hat{\boldsymbol{\beta}} \in \mathbb{R}^{N+1}$ such that $\|\mathbf{u}(\boldsymbol{\beta}^{\text{AC}}) - \mathbf{u}(\hat{\boldsymbol{\beta}})\|_\infty \leq 2\text{Err}(\mathbf{N}, \delta; \boldsymbol{\beta}^{\text{AC}})$. Therefore, for every $(\mathbf{s}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}_s$, the following inequalities hold with a probability of at least $1 - \delta$:

$$\begin{aligned} \mathbf{c}(\mathbf{s}, \mathbf{a}) - \eta^* &\geq \mathbf{u}(\mathbf{s}; \boldsymbol{\beta}^{\text{AC}}) - \mathbb{E}[\mathbf{u}(\mathbf{s}'; \boldsymbol{\beta}^{\text{AC}})|\mathbf{s}, \mathbf{a}] \\ &\geq \mathbf{u}(\mathbf{s}; \hat{\boldsymbol{\beta}}) - (2\text{Err}(\mathbf{N}, \delta; \boldsymbol{\beta}^{\text{AC}})) - \mathbb{E}[\mathbf{u}(\mathbf{s}'; \hat{\boldsymbol{\beta}})|\mathbf{s}, \mathbf{a}] - (2\text{Err}(\mathbf{N}, \delta; \boldsymbol{\beta}^{\text{AC}})) \end{aligned}$$

Therefore, with this probability, we have

$$\mathbf{c}(\mathbf{s}, \mathbf{a}) - \left(\eta^* - 4\text{Err}(\mathbf{N}, \delta; \boldsymbol{\beta}^{\text{AC}})\right) \geq \mathbf{u}(\mathbf{s}; \hat{\boldsymbol{\beta}}) - \mathbb{E}[\mathbf{u}(\mathbf{s}'; \hat{\boldsymbol{\beta}})|\mathbf{s}, \mathbf{a}], \quad \forall (\mathbf{s}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}_s, \quad (3.23)$$

that together with identity $\mathbf{u}(\bar{\mathbf{s}}; \hat{\boldsymbol{\beta}}) = \mathbf{0}$, which holds by Corollary 2, shows that pair $(\eta^* - 4\text{Err}(\mathbf{N}, \delta; \boldsymbol{\beta}^{\text{AC}}), \hat{\boldsymbol{\beta}}) \in \mathbb{R}^{N+2}$ is feasible to BALP_N . As a result, we obtain bound $0 \leq \eta^* - \eta_N^{\text{BA}} \leq 4\text{Err}(\mathbf{N}, \delta; \boldsymbol{\beta}^{\text{AC}})$. \blacksquare

Proof of Theorem 5.

Consider the following inequalities:

$$\begin{aligned}
& \text{AC}(\pi_g(\beta)) - \eta^* \\
& \stackrel{(i)}{=} \mathbb{E}_{\mu(\beta)} [\mathbf{c}(s, \pi_g(s; \beta))] - \eta^* \\
& \stackrel{(ii)}{=} \mathbb{E}_{\mu(\beta)} [\mathbf{c}(s, \pi_g(s; \beta)) - \eta^* + \mathbb{E}[\mathbf{u}(s'; \beta) | s, \pi_g(s; \beta)] - \mathbf{u}(s; \beta)] \\
& \stackrel{(iii)}{=} \mathbb{E}_{\mu(\beta)} \left[\min_{\mathbf{a} \in \mathcal{A}_s} \left\{ \mathbf{c}(s, \mathbf{a}) - \eta^* + \mathbb{E}[\mathbf{u}(s'; \beta) | s, \mathbf{a}] \right\} - \mathbf{u}(s; \beta) \right] \\
& \stackrel{(iv)}{\leq} \|\mathbf{u}(\beta) - \mathbf{u}^*\|_\infty + \mathbb{E}_{\mu(\beta)} \left[\min_{\mathbf{a} \in \mathcal{A}_s} \left\{ \mathbf{c}(s, \mathbf{a}) - \eta^* + \mathbb{E}[\mathbf{u}^*(s'; \beta) | s, \mathbf{a}] \right\} - \mathbf{u}(s; \beta) \right] \\
& \stackrel{(v)}{=} \|\mathbf{u}(\beta) - \mathbf{u}^*\|_\infty + \mathbb{E}_{\mu(\beta)} [\mathbf{u}^*(s) - \mathbf{u}(s; \beta)] \\
& \stackrel{(vi)}{\leq} \|\mathbf{u}(\beta) - \mathbf{u}^*\|_\infty + \|\mathbf{u}^* - \mathbf{u}(\beta)\|_{1, \mu(\eta, \beta)}
\end{aligned}$$

Identity (i) holds due to Assumption 6 under which identity $\text{AC}(s'; \pi) = \int_{\mathcal{S}} \mathbf{c}(s, \pi(s)) \mu(s; \pi) \, \mathrm{d} s$

holds for every policy π and state $s' \in \mathcal{S}$; equality (ii) relies on the following identity:

$$\mathbb{E}_{\mu(\beta)} [\mathbb{E}[\mathbf{u}(s'; \beta) | s, \pi_g(s; \beta)]] = \mathbb{E}_{\mu(\beta)} [\mathbf{u}(s; \beta)],$$

that holds by virtue of (3.3); equality (iii) is obtained using the definition of the greedy policy;

inequality (iv) is a result of inequality $\mathbf{u}(s; \beta) \leq \mathbf{u}^*(s) + \|\mathbf{u} - \mathbf{u}^*\|_\infty$ that holds for every $s \in \mathcal{S}$;

equality (v) holds since pair (η^*, \mathbf{u}^*) solves optimality equation (3.4); inequality (vi) directly

follows from the definition of $(1, \mu(\eta, \beta))$ -norm. Rearranging the terms in the above inequalities

results in (3.9), which finishes the proof. ■

Proposition 22 *The MDP bias function \mathbf{u}^* admits the following representation for each $s \in \mathcal{S}$:*

$$\mathbf{u}^*(s) = \limsup_{n \rightarrow \infty} \sum_{i=0}^n \mathbb{E}_s^{\pi^*} [c(s_i, \mathbf{a}_i) - \eta^*],$$

where s_n and \mathbf{a}_n are, respectively, the state and action reach at stage n under optimal policy π^* .

Proof. Please see Lemma 4.3 in [Luque-Vásquez and Hernández-Lerma \(1999\)](#).

Proof of Proposition 18.

Proof entails three following steps. First, we show that every feasible solution \mathbf{u} to PELP satisfies $\mathbf{u}(s) \leq \mathbf{u}^*(s)$ for all $s \in \mathcal{S}$. Fix some initial state $s = s_0 \in \mathcal{S}$. Let s_n and \mathbf{a}_n be the state and action, respectively, reach at stage n when following policy π^* . Since \mathbf{u} is feasible to PELP, we have that $\mathbf{u}(s_0) \leq c(s_0, \mathbf{a}_0) - \eta^* + \mathbb{E}[\mathbf{u}(s_1)|s_0, \mathbf{a}_0]$. Iterating this inequality, we obtain that

$$\mathbf{u}(s_0) \leq \sum_{i=0}^n \mathbb{E}_{s_0}^{\pi^*} [c(s_i, \mathbf{a}_i) - \eta^*] + \mathbb{E}[\mathbf{u}(s_{n+1})|s_n, \mathbf{a}_n]$$

Taking the limit of the above inequality and applying Lemma 22, we obtain that

$$\mathbf{u}(s_0) \leq \limsup_{n \rightarrow \infty} \sum_{i=0}^n \mathbb{E}_{s_0}^{\pi^*} [c(s_i, \mathbf{a}_i) - \eta^*] = \mathbf{u}^*(s_0),$$

for every arbitrary choice of $s_0 \in \mathcal{S}$. Next, we show the regression-based model in the proposition is a correct reformulation of PELP. The PELP is equivalent to

$$\min_{\mathbf{u} \in \mathcal{U}} \mathbb{E}_{\mathbf{v}}[\mathbf{u}^*] - \mathbb{E}_{\mathbf{v}}[\mathbf{u}]$$

$$\mathbf{u}(s) - \mathbb{E}[\mathbf{u}(s')|s, \mathbf{a}] \leq c(s, \mathbf{a}) - \eta^*, \quad \forall (s, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}_s.$$

Because of the first step, we have $\mathbb{E}_\nu[\mathbf{u}] \leq \mathbb{E}_\nu[\mathbf{u}^*]$ and thus $\|\mathbf{u} - \mathbf{u}^*\|_{1,\nu} = \mathbb{E}_\nu[\mathbf{u}^*] - \mathbb{E}_\nu[\mathbf{u}]$. Therefore, we can replace the objective function of the above program with $\|\mathbf{u} - \mathbf{u}^*\|_{1,\nu}$, where this replacement shows that the regression-based model in the proposition is a reformulation of PELP. Finally, we show that for every PELP optimal solution \mathbf{u}^{PE} , we have $\mathbf{u}^{\text{PE}}(s) = \mathbf{u}^*(s)$ for all $s \in \mathcal{S}$. From the previous part, we see that $\mathbf{u}^{\text{PE}}(s) \leq \mathbf{u}^*(s)$ for all $s \in \mathcal{S}$ because \mathbf{u}^{PE} is feasible to PELP. Assume there is a state \hat{s} such that $\mathbf{u}^{\text{PE}}(\hat{s}) < \mathbf{u}^*(\hat{s})$. Since \mathbf{u}^* and \mathbf{u}^{PE} are continuous, there must exist a ball around \hat{s} , denoted $\mathcal{N}_{\hat{s}}$, such that $\mathbf{u}^{\text{PE}}(s) < \mathbf{u}^*(s)$ for all $s \in \mathcal{N}_{\hat{s}}$. Since $\nu(\cdot)$ is positive for all non-zero measure subsets of \mathcal{S} , which includes $\mathcal{N}_{\hat{s}}$, we have

$$\mathbb{E}_\nu[\mathbf{u}^{\text{PE}}] = \int_{\mathcal{S} \setminus \mathcal{N}_{\hat{s}}} \mathbf{u}^{\text{PE}}(s) \nu(ds) + \int_{\mathcal{N}_{\hat{s}}} \mathbf{u}^{\text{PE}}(s) \nu(ds) < \mathbb{E}_\nu[\mathbf{u}^*].$$

Because (η^*, \mathbf{u}^*) is feasible to the optimality equation (3.4) and is thus \mathbf{u}^* a feasible to PELP, we have $\mathbb{E}_\nu[\mathbf{u}^*] \leq \mathbb{E}_\nu[\mathbf{u}^{\text{PE}}]$. This contradicts with $\mathbb{E}_\nu[\mathbf{u}^{\text{PE}}] < \mathbb{E}_\nu[\mathbf{u}^*]$ and thus there is no \hat{s} such that $\mathbf{u}^{\text{PE}}(\hat{s}) < \mathbf{u}^*(\hat{s})$. In other words, for all $s \in \mathcal{S}$, it must hold that $\mathbf{u}^{\text{PE}}(s) \geq \mathbf{u}^*(s)$. From the first step of the proof, we saw that $\mathbf{u}^{\text{PE}}(s) \leq \mathbf{u}^*(s)$. Therefore, we obtain equality $\mathbf{u}^{\text{PE}}(s) = \mathbf{u}^*(s)$ for all $s \in \mathcal{S}$. ■

Proof of Proposition 19.

Similar to the proof of Proposition 15, where we rewrite BFEP by substituting its decision variable $\beta \in \mathcal{B}$ with the decision variable $\mathbf{u}(\beta) \in \mathcal{R}$, we can reformulate PFEP as follows:

$$\begin{aligned} & \sup_{\mathbf{u}(\beta) \in \mathcal{R}} \mathbb{E}_{\mathbf{v}}[\mathbf{u}(\beta)] \\ & \mathbf{u}(s; \beta) - \mathbb{E}[\mathbf{u}(s'; \beta) | s, \mathbf{a}] \leq c(s, \mathbf{a}) - \eta^*, \quad \forall (s, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}_s. \end{aligned}$$

Because $\mathcal{R} \subseteq \mathcal{U}$ and \mathbf{u}^* is optimal to PELP by Proposition 18, for every feasible PFEP solution $\mathbf{u} \in \mathcal{R}$, it follows that $\mathbb{E}_{\mathbf{v}}[\mathbf{u}(\beta)] \leq \mathbb{E}_{\mathbf{v}}[\mathbf{u}^*]$. Under Assumption 8, which requires $\mathbf{u}^* = \mathbf{u}(\beta^*)$, the inequality $\mathbb{E}_{\mathbf{v}}[\mathbf{u}(\beta)] \leq \mathbb{E}_{\mathbf{v}}[\mathbf{u}(\beta^*)]$ holds for any feasible PFEP solution $\mathbf{u}(\beta) \in \mathcal{R}$. From Proposition 18, $\mathbf{u}^* = \mathbf{u}(\beta^*)$ is feasible to PELP, so β^* satisfies all PFEP constraints. Since weighting function β^* is a feasible PFEP solution and achieves the maximum attainable objective value $\mathbb{E}_{\mathbf{v}}[\mathbf{u}(\beta^*)]$, it is optimal to BFEP. Next, we show that β ■

Proposition 23 *Let \mathcal{C} be the space of all continuous functions over \mathcal{S} . Mapping $\mathsf{T} : \mathcal{C} \mapsto \mathcal{C}$ is a bounded linear transformation over Banach space \mathcal{C} equipped with ∞ -norm $\|\cdot\|_{\infty}$.*

Proof. We claim that $\mathsf{T} : \mathcal{C} \mapsto \mathcal{C}$ is a linear bounded transformation over \mathcal{C} . First, recall \mathcal{C} which is the space of all real-valued continuous functions $\mathbf{u} : \mathcal{S} \mapsto \mathbb{R}$ defined over compact domain \mathcal{S} . It is known that \mathcal{C} is a vector space and if it is equipped with the infinity-norm

$\|\mathbf{u}\|_\infty := \max_s |\mathbf{u}(s)|$, it becomes a Banach space. Second, T is linear because for every $\mathbf{u}, \mathbf{v} \in \mathcal{C}$ and every $\alpha, \beta \in \mathbb{R}$, we have

$$T(\alpha \mathbf{u} + \beta \mathbf{v}) = \mathbb{E}[\alpha \mathbf{u}(s') + \beta \mathbf{v}(s') \mid \cdot, \pi^*(\cdot)] = \alpha T\mathbf{u} + \beta T\mathbf{v}.$$

Next, T is bounded in essence that for every $\mathbf{u} \in \mathcal{C}$, the following holds:

$$\|T\mathbf{u}\|_\infty \leq \sup_s \left| \int_{\mathcal{S}} P(s'|s, \pi^*(s)) \mathbf{u}(d s') \right| \leq \sup_s \left| \int_{\mathcal{S}} P(d s'|s, \pi^*(s)) \|\mathbf{u}\|_\infty \right| d s' \leq \|\mathbf{u}\|_\infty,$$

We observe that T is a transformation because the MDP stochastic kernel P is strongly continuous by Assumption 9 and thus function $T\mathbf{u}(s) := \mathbb{E}[\mathbf{u}(s') \mid s, \pi^*(s)]$ is continuous over \mathcal{S} for every $\mathbf{u} \in \mathcal{C}$, i.e., $T\mathbf{u} \in \mathcal{C}$. Therefore, $T : \mathcal{C} \mapsto \mathcal{C}$ is a linear bounded transformation over Banach space \mathcal{C} . ■

Proof of Proposition 20.

Let $\varepsilon := \eta^* - \eta_N^{\text{BA}}$. Define function $\mathbf{e} : \mathcal{S} \mapsto \mathbb{R}$ as $\mathbf{e}(\cdot) = 1$. Utilizing identity $\mathbf{u}^* = F^\infty \mathbf{g}^*$ and the definition of $\mathbf{u}^{\text{ID}}(\eta_N^{\text{BA}}) = F^\infty \mathbf{g}_N^{\text{BA}}$, for every $s \in \mathcal{S}$, we have

$$\mathbf{u}^{\text{ID}}(s; \eta_N^{\text{BA}}) - \mathbf{u}^*(s) = F^\infty \mathbf{g}^*(s) - F^\infty \mathbf{g}_N^{\text{BA}}(s) = F^\infty (\mathbf{g}^* - \mathbf{g}_N^{\text{BA}})(s) = \varepsilon F^\infty \mathbf{e}(s).$$

Using the definition of norm $\|F^\infty\|$, we have

$$\|\mathbf{u}^{\text{ID}}(s; \eta_N^{\text{BA}}) - \mathbf{u}^*(s)\|_\infty = \varepsilon \sup_{s \in \mathcal{S}} |F^\infty \mathbf{e}(s)| \leq \varepsilon \sup_{\mathbf{u}} \{\|F^\infty \mathbf{u}\|_\infty : \|\mathbf{u}\|_\infty \leq 1\} = \varepsilon \|F^\infty\|.$$

The first part of the proof is thus complete. We next focus on the second part. Given $q = 1, 2, \dots, Q$, let β be a feasible solution to program PALP_{N+qB} . For $\mathbf{u}(\beta)$, we have

$$\mathbf{u}(s; \beta) - \mathbb{E}[\mathbf{u}(s'; \beta) | s, \mathbf{a}] \leq \mathbf{c}(s, \mathbf{a}) - \boldsymbol{\eta}_N^{\text{BA}}, \quad \forall (s, \mathbf{a}).$$

For the particular choice of actions $\mathbf{a} = \pi^*(s)$ at each state s , the above inequality holds. That is, for β that is feasible to PALP_{N+qB} , we have $\mathbf{u}(s; \beta) - \mathbb{E}[\mathbf{u}(s'; \beta) | s, \pi^*(s)] \leq \mathbf{c}(s, \pi^*(s)) - \boldsymbol{\eta}_N^{\text{BA}}$ for all $s \in \mathcal{S}$. Combining this inequality with the definition of T nad F , we obtain that β satisfies $\mathbf{u}(\beta) \leq \mathbf{g}_N^{\text{BA}} + T\mathbf{u}(\beta)$. Iterating this inequality, we obtain $\mathbf{u}(\beta) \leq \lim_{K \rightarrow \infty} \sum_{k=0}^K T^k \mathbf{g}_N^{\text{BA}}$. Hence, we have

$$\mathbf{u}(s; \beta) = \mathbf{u}(s; \beta) - \mathbf{u}(\bar{s}; \beta) \leq \lim_{K \rightarrow \infty} \sum_{k=0}^K T^k \mathbf{g}_N^{\text{BA}}(s) - T^k \mathbf{g}_N^{\text{BA}}(\bar{s}) = \lim_{K \rightarrow \infty} \sum_{k=0}^K F^k \mathbf{g}_N^{\text{BA}}(s) = F^\infty \mathbf{g}_N^{\text{BA}}(s)$$

Using the definition of the idealized solution $\mathbf{u}^{\text{ID}}(s; \boldsymbol{\eta}_N^{\text{BA}}) = F^\infty \mathbf{g}_N^{\text{BA}}(s)$, the above inequalities, and the first part of the proposition, we obtain that

$$\mathbf{u}(s; \beta) \leq \mathbf{u}^{\text{ID}}(s; \boldsymbol{\eta}_N^{\text{BA}}) \leq \mathbf{u}^*(s) + (\boldsymbol{\eta}^* - \boldsymbol{\eta}_N^{\text{BA}}) \|F^\infty\|,$$

for all $s \in \mathcal{S}$. The above inequalities complete the proof. ■

CITED LITERATURE

Bibliography

- Adelman D (2003) Price-directed replenishment of subsets: methodology and its application to inventory routing. *Manufacturing & Service Operations Management* 5(4):348–371.
- Adelman D, Klabjan D (2005) Duality and existence of optimal policies in generalized joint replenishment. *Mathematics of Operations Research* 30(1):28–50.
- Adelman D, Klabjan D (2012) Computing near-optimal policies in generalized joint replenishment. *INFORMS Journal on Computing* 24(1):148–164.
- Adelman D, Mersereau AJ (2013) Dynamic capacity allocation to customers who remember past service. *Management Science* 59(3):592–612.
- Balseiro SR, Gurkan H, Sun P (2019) Multiagent mechanism design without money. *Operations Research* 67(5):1417–1436.
- Bartlett PL, Mendelson S (2002) Rademacher and Gaussian complexities: risk bounds and structural results. *Journal of Machine Learning Research* 3(Nov):463–482.
- Basu A, Martin K, Ryan CT (2017) Strong duality and sensitivity analysis in semi-infinite linear programming. *Mathematical Programming* 161(1-2):451–485.
- Beevi KS, Nair MS, Bindu GR (2016) Detection of mitotic nuclei in breast histopathology images using localized ACM and Random Kitchen Sink based classifier. *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2435–2439.
- Bertsekas DP (2015) *Dynamic programming and optimal control, 4th Edition*, volume 2 (Athena Scientific).
- Bertsekas DP, Tsitsiklis JN (1996) *Neuro-dynamic Programming*, volume 5 (Athena Scientific).
- Bhat N, Farias V, Moallemi CC (2012) Non-parametric approximate dynamic programming via the kernel method. *Advances in Neural Information Processing Systems*, 386–394.
- Blado D, Toriello A (2019) Relaxation analysis for the dynamic knapsack problem with stochastic item sizes. *SIAM Journal on Optimization* 29(1):1–30.
- Brown DB, Smith JE, Sun P (2010) Information relaxations and duality in stochastic dynamic programs. *Operations Research* 58(4-part-1):785–801.
- Brown DB, Smith JE, et al. (2022) Information relaxations and duality in stochastic dynamic programs: A review and tutorial. *Foundations and Trends® in Optimization* 5(3):246–339.
- Calafiore G, Campi MC (2005) Uncertain convex programs: randomized solutions and confidence levels. *Mathematical Programming* 102(1):25–46.
- Calafiore GC, Campi MC (2006) The scenario approach to robust control design. *IEEE Transactions on automatic control* 51(5):742–753.
- Canuto C, Hussaini MY, Quarteroni A, Thomas Jr A, et al. (2012) *Spectral methods in fluid dynamics* (Springer Science & Business Media).
- Carriere JF (1996) Valuation of the early-exercise price for options using simulations and nonparametric regression. *Insurance: Mathematics and Economics* 19(1):19–30.
- Chen X, Pang Z, Pan L (2014) Coordinating inventory control and pricing strategies for perishable products. *Operations Research* 62(2):284–300.
- Dai JG, Shi P (2019) Inpatient overflow: An approximate dynamic programming approach. *Manufacturing & Service Operations Management* 21(4):894–911.

- De Farias DP, Van Roy B (2002) Approximate linear programming for average-cost dynamic programming. *Advances in Neural Information Processing Systems* 15.
- De Farias DP, Van Roy B (2003) The linear programming approach to approximate dynamic programming. *Operations Research* 51(6):850–865.
- De Farias DP, Van Roy B (2004) On constraint sampling in the linear programming approach to approximate dynamic programming. *Mathematics of Operations Research* 29(3):462–478.
- De Farias DP, Van Roy B (2006) A cost-shaping linear program for average-cost approximate dynamic programming with performance guarantees. *Mathematics of Operations Research* 31(3):597–620.
- Desai VV, Farias VF, Moallemi CC (2012a) Approximate dynamic programming via a smoothed linear program. *Operations Research* 60(3):655–674.
- Desai VV, Farias VF, Moallemi CC (2012b) Pathwise optimization for optimal stopping problems. *Management Science* 58(12):2292–2308.
- Drusvyatskiy D, Lewis AS (2018) Error bounds, quadratic growth, and linear convergence of proximal methods. *Mathematics of Operations Research* 43(3):919–948.
- Farias VF, Van Roy B (2006) Tetris: a study of randomized constraint sampling. *Probabilistic and Randomized Methods for Design Under Uncertainty* 189–201.
- Folland GB (1999) *Real Analysis: Modern Techniques and Their Applications* (New York, NY: John Wiley & Sons).
- Forsell N, Sabbadin R (2006) Approximate linear-programming algorithms for graph-based Markov decision processes. *Proceedings of the 2006 Conference on ECAI 2006: 17th European Conference on Artificial Intelligence August 29–September 1, 2006, Riva del Garda, Italy*, 590–594.
- Franke JK, Koehler G, Biedenkapp A, Hutter F (2021) Sample-efficient automated deep reinforcement learning. *International Conference on Learning Representations*.
- Fujimoto S, Hoof H, Meger D (2018) Addressing function approximation error in actor-critic methods. *International Conference on Machine Learning*, 1587–1596 (PMLR).
- Glasserman P, Yu B (2004) Simulation for American options: regression now or regression later? *Monte Carlo and Quasi-Monte Carlo Methods 2002*, 213–226 (Springer).
- Gordienko E, Hernández-Lerma O (1995) Average cost Markov control processes with weighted norms: existence of canonical policies. *Applicationes Mathematicae* 23(2):199–218.
- Guestrin C, Koller D, Parr R, Venkataraman S (2003) Efficient solution algorithms for factored MDPs. *Journal of Artificial Intelligence Research* 19:399–468.
- Gurobi Optimization L (2019) Gurobi optimizer reference manual. URL <http://www.gurobi.com>.
- Haarnoja T, Ha S, Zhou A, Tan J, Tucker G, Levine S (2019) Learning to walk via deep reinforcement learning. *Robotics: Science and Systems*.
- Haskell WB, Jain R, Sharma H, Yu P (2020) A universal empirical dynamic programming algorithm for continuous state MDPs. *IEEE Transactions on Automatic Control* 65(1):115–129, ISSN 2334-3303.
- Haugh MB, Kogan L (2004) Pricing American options: a duality approach. *Operations Research* 52(2):258–270.
- Hernández-Lerma O, Lasserre JB (1996) *Discrete-time Markov Control Processes: Basic Optimality Criteria*, volume 30 (New York, NY: Springer Science & Business Media).
- Hernández-Lerma O, Lasserre JB (1999) *Further Topics on Discrete-time Markov Control Processes*, volume 42 (New York, NY: Springer Science & Business Media).

- Hua Z, Yu Y, Zhang W, Xu X (2015) Structural properties of the optimal policy for dual-sourcing systems with general lead times. *IIE Transactions* 47(8):841–850.
- Karaesmen IZ, Scheller-Wolf A, Deniz B (2011) *Managing perishable and aging inventories: review and future research directions*, 393–436 (New York, NY: Springer).
- Klabjan D, Adelman D (2006) Existence of optimal policies for semi-Markov decision processes using duality for infinite linear programming. *SIAM Journal on Control and Optimization* 44(6):2104–2122.
- Klabjan D, Adelman D (2007) An infinite-dimensional linear programming algorithm for deterministic semi-Markov decision processes on Borel spaces. *Mathematics of Operations Research* 32(3):528–550.
- Lewis AS, Pang JS (1998) Error bounds for convex inequality systems. *Generalized Convexity, Generalized Monotonicity: Recent Results* 75–110.
- Lin Q, Ma R, Nadarajah S, Soheili N (2022) A parameter-free and projection-free restarting level set method for adaptive constrained convex optimization under the error bound condition. *Working paper*.
- Lin Q, Nadarajah S, Soheili N (2020) Revisiting approximate linear programming: Constraint-violation learning with applications to inventory control and energy storage. *Management Science* 66(4):1544–1562.
- Longstaff FA, Schwartz ES (2001) Valuing American options by simulation: a simple least-squares approach. *The Review of Financial Studies* 14(1):113–147.
- Lu Y, Dhillon P, Foster DP, Ungar L (2013) Faster ridge regression via the subsampled randomized hadamard transform. *Advances in Neural Information Processing Systems*, 369–377.
- Luque-Vásquez F, Hernández-Lerma O (1999) Semi-Markov control models with average costs. *Applcationes mathematicae* 26(3):315–331.
- Mahadevan S (1996) An average-reward reinforcement learning algorithm for computing bias-optimal policies. *AAAI/IAAI, Vol. 1*, 875–880 (Citeseer).
- McGrew JS, How JP, Williams B, Roy N (2010) Air-combat strategy using approximate dynamic programming. *Journal of Guidance, Control, and Dynamics* 33(5):1641–1654.
- McWilliams B, Balduzzi D, Buhmann JM (2013) Correlated random features for fast semi-supervised learning. *Advances in Neural Information Processing Systems*, 440–448.
- Micchelli CA, Xu Y, Zhang H (2006) Universal kernels. *Journal of Machine Learning Research* 7(Dec):2651–2667.
- Mladenov M, Boutilier C, Schuurmans D, Elidan G, Meshi O, Lu T (2017) Approximate linear programming for logistic Markov decision processes. *Proceedings of the Twenty-sixth International Joint Conference on Artificial Intelligence*, 2486–2493.
- Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, et al. (2015) Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533.
- Mohri M, Rostamizadeh A, Talwalkar A (2012) *Foundations of Machine Learning* (Cambridge, MA: MIT press), first edition.
- Nadarajah S, Margot F, Secomandi N (2015) Relaxations of approximate linear programs for the real option management of commodity storage. *Management Science* 61(12):3054–3076.
- Nadarajah S, Margot F, Secomandi N (2017) Comparison of least squares monte carlo methods with applications to energy real options. *European Journal of Operational Research* 256(1):196–204.

- Nadarajah S, Secomandi N (2022) A review of the operations literature on real options in energy. *European Journal of Operational Research* ISSN 0377-2217.
- Nersessian A (2019) Fourier tools are much more powerful than commonly thought. *Lobachevskii Journal of Mathematics* 40(8):1122–1131.
- Osband I, Van Roy B, Russo DJ, Wen Z, et al. (2019) Deep exploration via randomized value functions. *Journal of Machine Learning Research* 20(124):1–62.
- Pakiman P, Nadarajah S, Soheili N, Lin Q (2020) Self-guided approximate linear programs: randomized multi-shot approximation of discounted cost markov decision processes. Available at SSRN: <http://dx.doi.org/10.2139/ssrn.3512665>.
- Peters J, Vijayakumar S, Schaal S (2003) Reinforcement learning for humanoid robotics. *Proceedings of the third IEEE-RAS International Conference on Humanoid Robots*, 1–20.
- Powell WB (2007) *Approximate Dynamic Programming: Solving the Curses of Dimensionality* (Hoboken, NJ: John Wiley & Sons).
- Puterman ML (1994) *Markov Decision Processes: Discrete Stochastic Dynamic Programming* (Hoboken, NJ: John Wiley & Sons).
- Rahimi A, Recht B (2008) Random features for large-scale kernel machines. *Advances in Neural Information Processing Systems*, 1177–1184.
- Rahimi A, Recht B (2008) Uniform approximation of functions with random bases. *2008 46th Annual Allerton Conference on Communication, Control, and Computing*, 555–561.
- Rahimi A, Recht B (2009) Weighted sums of random kitchen sinks: replacing minimization with randomization in learning. *Advances in Neural Information Processing Systems*, 1313–1320.
- Rudin W (1987) *Real and Complex Analysis* (Singapore: McGraw-Hill).
- Saldi N, Yüksel S, Linder T (2017) On the asymptotic optimality of finite approximations to Markov decision processes with Borel spaces. *Mathematics of Operations Research* 42(4):945–978.
- Schweitzer PJ, Seidmann A (1985) Generalized polynomial approximations in Markovian decision processes. *Journal of Mathematical Analysis and Applications* 110(2):568–582.
- Shahrampour S, Beirami A, Tarokh V (2018) On data-dependent random features for improved generalization in supervised learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.
- Shapiro A (2009) Semi-infinite programming, duality, discretization and optimality conditions. *Optimization* 58(2):133–161.
- Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, Hubert T, Baker L, Lai M, Bolton A, et al. (2017) Mastering the game of go without human knowledge. *Nature* 550(7676):354–359.
- Sinha A, Duchi JC (2016) Learning kernels with random features. *Advances in Neural Information Processing Systems* 29:1298–1306.
- Steimle LN, Denton BT (2017) Markov decision processes for screening and treatment of chronic diseases. *Markov Decision Processes in Practice*, 189–222 (Springer).
- Sun P, Wang K, Zipkin P (2014) Quadratic approximation of cost functions in lost sales and perishable inventory control problems. *Fuqua School of Business, Duke University, Durham, NC*.
- Tong C, Topaloglu H (2013) On the approximate linear programming approach for network revenue management problems. *INFORMS Journal on Computing* 26(1):121–134.

- Trick MA, Zin SE (1997) Spline approximations to value functions: linear programming approach. *Macroeconomic Dynamics* 1(1):255–277.
- Van Ngai H, Kruger A, Théra M (2010) Stability of error bounds for semi-infinite convex constraint systems. *SIAM Journal on Optimization* 20(4):2080–2096.
- Veatch MH (2013) Approximate linear programming for average cost MDPs. *Mathematics of Operations Research* 38(3):535–544.
- Vega-Amaya O (2003) The average cost optimality equation: a fixed point approach. *Bol. Soc. Mat. Mexicana* 9(1):185–195.
- Wang D, Zeng J, Lin SB (2020) Random sketching for neural networks with ReLU. *IEEE Transactions on Neural Networks and Learning Systems* 32(2):748–762.
- Wu L, Chen PY, Yen IEH, Xu F, Xia Y, Aggarwal C (2018) Scalable spectral clustering using random binning features. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2506–2515.
- Yang Q, Zhang J, Shi G, Hu J, Wu Y (2019) Maneuver decision of UAV in short-range air combat based on deep reinforcement learning. *IEEE Access* 8:363–378.
- Zipkin P (2008) On the structure of lost-sales inventory models. *Operations Research* 56(4):937–944.

VITA

NAME	Parshan Pakiman
EDUCATION	<p>Ph.D., <i>Information and Decision Sciences</i>, University of Illinois Chicago, Chicago, IL, USA, 2023.</p> <p>M.Sc., <i>Business Analytics</i>, University of Illinois Chicago, Chicago, IL, USA, 2023.</p> <p>B.Sc., <i>Applied Mathematics</i>, University of Tehran, Tehran, Iran, 2016.</p>
PUBLICATIONS	<p>P. Pakiman, S. Nadarajah, N. Soheili, and Q. Lin. “<i>Self-Guided Approximate Linear Programs: Randomized Multi-Shot Approximation of Discounted Cost Markov Decision Processes</i>.” Under review at <i>Management Science</i>.</p> <p>P. Pakiman, and S. Nadarajah. “<i>Randomized Multi-Shot Approximation of Average Cost Markov Decision Processes</i>.” Working paper.</p> <p>A. Chenreddy, P. Pakiman, S. Nadarajah, R. Chandrasekaran, and R. Abens. “<i>SMOILE: A Shopper Marketing Optimization and Inverse Learning Engine</i>.” Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2019).</p>