A Pathwise Optimization Approach for Reinforcement Learning in Merchant Energy Operations

By

Bo Yang

A thesis submitted to
Tepper School of Business at Carnegie Mellon University
in partial fulfillment of the thesis requirement
for the degree of Doctor of Philosophy
in
Operations Management

Pittsburgh PA, US
April 2022

A Pathwise Optimization Approach for Reinforcement Learning in Merchant Energy Operations

Bo Yang

Abstract

This thesis studies the management of energy conversion assets, such as oil and biorefineries, ethanol production plants, transportation pipelines, and natural gas processing and storage facilities. The merchant operations approach formulates the management of these assets as real option models, which provide a convenient way to maximize the market value of the conversion assets. However, the real option approach gives rise to an intractable Markov decision process (MDP) formulation due to intertemporal linkages between decisions and the high dimensionality of the market state variables when using realistic price models. We consider reinforcement learning (RL) approaches that rely on basis-function-based value function approximations to compute both a feasible policy and a lower bound on the optimal policy value of this MDP, as well as a dual (upper) bound on the latter quantity. In particular, we focus on pathwise optimization (PO), because, in theory, it generates the tightest dual bound for a given set of basis functions.

We first extend PO from optimal stopping to merchant energy production. It is known that applying least squares Monte Carlo (LSM), a state-of-the-art RL approach for financial and real option valuation, in conjunction with information relaxation and duality techniques to realistic merchant ethanol production instances results in sizable (about 13%) average optimality gaps. Our research aims at reducing these gaps by extending the PO applicability from optimal stopping to merchant energy production. This extension rests on formulating a novel PO linear program (PLP), which is difficult to solve optimally because it is both ill conditioned and large scale. We develop an effective preconditioning approach based on principal component analysis (PCA). We mitigate the dimensionality concern by proposing a provably convergent block coordinate descent (BCD) technique. On the known set of merchant ethanol production instances, PO leads to considerably smaller (roughly 7%) average optimality gaps compared to LSM but entails a noticeably larger average computational effort (eleven hours rather than one hour). The optimality gap reduction is almost entirely attributable to the stronger PO-based dual bounds relative to the LSM-based ones. We thus bring to light the near optimality of the LSM-based policy.

Next, we put forth a constraint generation method to solve PLP. The proposed method iterates between a master and a subproblem. The master problem combines subsets of

the constraints and variables of the model to obtain a relaxed linear program that an off-the-shelf solver can handle. The subproblem strengthens this relaxation by efficiently identifying constraint violations in the original linear program. This method provably converges. It can be stopped once the current solution is sufficiently close to the optimal one, which we can check using a bound that we compute when solving the subproblem, or it leads to good-enough bounds on our MDP optimal policy value. We have verified both the efficiency and effectiveness of this version of our method on the benchmark merchant ethanol production instances employed in the previous chapter.

Finally, we extend PO to merchant energy operations and related real option models. The extension includes both modeling and solution aspects. The original modeling approach leads to unbounded PLPs in those applications if no feasible decision can terminate the MDP at each controllable state, e.g., “stop” in optimal stopping. We propose a novel pseudo action scheme to deal with this issue. The pseudo action scheme provides effective bounds to the LP by adding artificial actions from nonanticipative policies, i.e., policies that do not utilize future information. We also develop a new solution approach, the coordination decomposition and regression approach, to reduce the computational complexities of solving PLPs in those applications. Our solution approach (i) solves PLP dual via the coordinated decomposition (alternating direction method of multipliers) and (ii) recovers an associated primal solution by approximately enforcing complementary slackness via two norm regression. Compared to the extant approaches, our approach features low per iteration computational complexity because it exploits the problem's decomposition structure. We conduct numerical studies in realistic energy storage and production to demonstrate the use of our approaches. Our pseudo action scheme extends PO to energy storage and generates comparable results to the benchmark method. In merchant energy production, our solution technique solves both existing instances with substantially less computational efforts (85% reduction in memory and 50% reduction in CPU time) than the BCD and constraint generation approaches and larger instances that were out of reach, achieving near optimal performance and dominating a standard competitor in terms of solution quality.

Acknowledgements

I acknowledge the financial support from the Scott Institute for Energy Innovation at Carnegie Mellon University and from NSF grant CMMI 1761742.

First and foremost, I thank my advisors, Prof. Nicola Secomandi and Prof. Selvaprabu Nadarajah. It is hard to put into words how valuable their guidance, patience and support has been throughout my PhD journey. I am very lucky to be advised by them, and I am truly grateful.

I thank the other members of my dissertation committee, Prof. Alan Scheller Wolf, Prof. Duane Seppi, and Prof. Javier Pena, for their time and effort in reading my dissertation and providing valuable feedback on my work.

I thank Lawrence Rapp and Laila Lee for taking care of all the administrative needs of the PhD program at Tepper. Life would be a lot harder if it was not for their professionalism and generosity.

I thank the classmates at CMU: Arash Haddadan, Franco Berbeglia, Mehmet and Neda Mirzaeian. Whether discussing research or just hanging out, it has always been a pleasure. Special thanks to my first year roomates: Jun Shi, Zhou Yu, and Qi Zhang. You are the best roommates I have ever had. Withou you, my first year in Pittsburgh would be much harder and more lonely.

I thank my best friend Wenlei Zhang.

I thank my partner Sang Wu.

I thank my parents in China. Their love and support throughout my journey has been a great source of comfort and inspiration. I would not be here without them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

We study the management of energy conversion assets, in particular, the energy production and storage facility, and the methodology arising in this context. In §1.1, we introduce the background of merchant operations of energy conversion assets. In §1.2 and §1.3, we describe a Markov decision process (MDP) formulation of the problem and its dual version, respectively. We then briefly review two streams of the extant reinforcement learning (RL) approaches in the literature. We summarize our contributions in §1.5.

1.1 Background

Renewable energy such as biofuels, hydrogen, and solar power plays essential roles in economic and environmental activities. For instance, ethanol has been extensively blended in gasoline (e.g., nearly all US gasoline contains 10% - 15% of ethanol) due to the urgency of acting on climate change, which contributes 35 Billion dollars to the US economy in 2021. According to Renewable Fuels Association (2021), ethanol can reduce 70% greenhouse gas (GHG) emission by 2030 compared to gasoline and achieve net-zero GHG emissions by 2050. The worldwide biofuel production and storage capacities are also expected to be growing in recent years as international energy companies like Exxon Mobile, Calumet, and Kinder Morgan are expanding their renewable energy business (Brelsford 2021a,b, Oil & Gas Journal 2021).

Due to the importance of renewable energy, managing energy conversion assets that generate, process, transport, and store renewable energy is thus a crucial task. In practice, this task requires an energy company to obtain (i) a physical operating policy to generate maximal cash flows with the assets and (ii) a financial hedging policy to reduce the uncertainty in those cash flows. Roughly speaking, the physical operating policy aims at improving the expected total revenues brought by the asset. In contrast, the hedging policy tries to lock in the expected value by reducing its variability. In literature, the energy company is referred to as *merchant* and the two aspects are referred to as *merchant energy operations* (Secomandi and Seppi 2014).

This thesis studies (i), i.e., managing the physical operation of energy conversion assets, because it directly impacts the revenue and lays the foundation for hedging. (i) involves formulating the management of energy and commodity conversion assets as real options and solving the corresponding models to obtain an operating policy. The real option technique is commonly used in literature (Dixit and Pindyck 1994, Trigeorgis et al. 1996, Smith and McCardle 1998, 1999, Eydeland and Wolyniec 2003, Geman 2005, Secomandi and Seppi 2016) because it bridges the expected total cash flows and the market value of the asset by modeling commodity prices under the risk neutral measure. Based on this connection, maximizing the expected total cash flows in real option models is equivalent to maximizing the market value of the assets.

The real option formulation of the problem typically leads to an intractable MDP. The source of the intractability comes from the intertemporal linkages between decisions and the high dimensionality of the commodity prices when using realistic financial models. Merchants thus use various approximation approaches to obtain a lower and a dual (upper) bound on the optimal policy value as well as a feasible operating policy. The value of the feasible policy provides a lower bound on the optimal policy value. The dual bound serves as a metric for the quality of the lower bound and the feasible policy.

The value function approximation (VFA, Powell 2007, Bertsekas 2019) based RL approaches are commonly used in merchant energy operation to obtain a lower bound and a feasible policy. The information relaxation technique is the state of the art approach to generate a dual bound (Brown et al. 2010, Brown and Smith 2014, 2011, Brown and Haugh 2017, Brown and Smith 2021, Nadarajah and Secomandi 2018a, Secomandi 2017, Secomandi and Seppi 2016, Lai et al. 2010). However, practice-based RL approaches, such as LSM, in conjunction with the information relaxation techniques, may lead to large gaps between these two bounds in energy operation settings like the energy production (Nadarajah and Secomandi 2020). It is unclear which bound is weak and most contributes to the large gap. Thus, narrowing the gap and finding a dual bound that can provide a more accurate assessment of the policy is crucial in both practice and research.

1.2 MDP

Merchant operations of energy conversion assets can be modeled as discrete time MDPs with finite horizons. Suppose the merchant manages an energy conversion asset over I horizons. The stage number indexed by i is an element of the index set $\mathcal{I} := \{0, ..., I-1\}$. In stage i, the state of the MDP consists of both the endogenous and exogenous components. The endogenous state component describes the status of the asset, such as the inventory level and the operational mode of the plant. The exogenous state tracks the evolution of commodities' futures prices in the wholesale market. We denote the endogenous and exogenous state components as x_i and F_i, respectively. Their feasible sets are respective $\mathcal{X}_i$ and $\mathcal{F}_i$. Clearly, we have $(x_i, F_i) \in \mathcal{X}_i \times \mathcal{F}_i$ for each i.

In each stage, the merchant first observes F_i from the commodity markets and then chooses an action a_i based on x_i. We denote as $\mathcal{A}_i(x_i)$ the feasible action set for $x_i \in \mathcal{X}_i$, i.e., $a_i \in \mathcal{A}_i(x_i)$. Each action a_i has an immediate payoff given by the function $r(x_i, F_i, a_i) : \mathcal{X}_i \times \mathcal{F}_i \times \mathcal{A}_i(x_i) \to \mathbb{R}$. The decision rule $D_i : \mathcal{X}_i \times \mathcal{F}_i \to \mathcal{A}(x_i)$ is a mapping from the state space to the action set for each stage i. A policy π is a collection of these decision rules, i.e., $\pi := \{D_0, D_1, ..., D_{I-1}\}$. We let Π be the set of all feasible policies.

We assume the merchant is a small player compared to the entire commodity market; thus, the action a_i in each stage can only influence the transition of the endogenous state components. This assumption, also known as the price taker assumption, is common in merchant energy operation and real option literature. So we denote as $f(x_i, a_i)$ the endogenous state transition function that characterizes the transition of x_i under the action a_i. The evolution of F_i is governed by a predetermined stochastic process independent of the decision a_i.

We use a constant risk-free discount factor $\delta \in (0, 1)$ to calculate the current value of cash flows in the future. Suppose the initial state is (x_0, F_0), the optimal policy π^* that maximizes the expected total discounted cash flows can be obtained by solving:

$$\max_{\pi \in \Pi} \mathbb{E}\left[\sum_{i=0}^{I-1} \delta^i r(x_i^\pi, F_i, a_i^\pi) \middle| x_0, F_0\right] \tag{1.1}$$

where $\mathbb{E}$ is the expectation taken w.r.t. F_i. The expectation and stochastic process that governs the transition of F_i are typically under the risk neutral measure (Shreve 2004) so (1.1) also maximizes the market value of the asset. Define $V(x_i, F_i)$ as the value function for each state $(x_i, F_i) \in \mathcal{X}_i \times \mathcal{F}_i$. The stochastic dynamic programming (SDP) formulation of (1.1) is

$$V_i(x_i, F_i) = \max_{a_i \in \mathcal{A}(x_i)} \left\{ r(x_i, F_i, a_i) + \delta \mathbb{E}\left[V_{i+1}(f(x_i, a_i), F_{i+1}) \middle| F_i \right] \right\}, \tag{1.2}$$

with the terminal condition $V_I(x_I, F_I) = 0, \ \forall (x_I, F_I) \in \mathcal{X}_I \times \mathcal{F}_I$. In general, (1.2) is computationally intractable due to the well known "curse of dimentionality" (Powell 2007). The source of the intractability is primarily the exogenous state space in merchant energy operations such as energy production, and storage because the exogenous state space tracks the price information that includes both the spot and futures prices of all commodities.

1.3 MDP Dual

There is also a dual version of (1.1) in which decisions are made based on future information but the benefit of this foresight is eliminated by a so-called dual penalty (Brown et al. 2010). The dual MDP generates a dual (upper) bound on the optimal policy value of (1.1).

Let $\bar{F}$ be a sample path that includes exogenous states from stages 0 through $I-1$ starting with F_0. The set $\bar{\mathcal{F}}$ is the collection of all such paths. We denote by $\bar{F}_i$ the stage i exogenous state corresponding to sample path $\bar{F}$. The decision rule for the dual model is denoted as $\bar{D}_i : \mathcal{X}_i \times \bar{\mathcal{F}} \to \mathcal{A}_i(x_i)$ which prescribes a feasible action for stage i, endogenous state x_i, and sample path $\bar{F}$. The dual policy $\bar{\pi}$ is the collection of such decision rules $\{\bar{D}_i, i \in \mathcal{I}\}$. The set of such policies is $\bar{\Pi}$.

Ideal dual penalties depend on the value function associated with (1.2). Consider stage $i \neq I-1$ and suppose we take feasible action a_i for endogenous state x_i and sample path $\bar{F}$. The ideal penalty is the additional value of knowing the stage $i+1$ information $\bar{F}_{i+1}$ at stage i relative to only having knowledge of the information $\bar{F}_i$ at this stage, which corresponds to the discounted difference

$$\delta\Bigg(V_{i+1}\left(f(x_i,a_i),\bar{F}_{i+1}\right) - \mathbb{E}\Bigg[V_{i+1}\left(f(x_i,a_i),F_{i+1}\right)\Bigg|\bar{F}_i\Bigg]\Bigg) \tag{1.3}$$

We use (1.3) to reduce the cash flow that ensues in the stage $i \neq I-1$ from applying the decision rule $\bar{D}_i^{\bar{\pi}}$ to the pair $(x_i, \bar{F})$. The resulting dual MDP is

$$\begin{aligned}\mathbb{E}\Bigg[\max_{\bar{\pi}\in\bar{\Pi}}\Bigg\{\sum_{i\in\mathcal{I}\setminus\{I-1\}}\delta^i\Bigg[&r(x_i^{\bar{\pi}},\bar{F}_i(\bar{F}),\bar{A}_i^{\bar{\pi}}) - \delta\big(V_{i+1}(f(x_i^{\bar{\pi}},\bar{A}_i^{\bar{\pi}}),\bar{F}_{i+1})\\ &-\mathbb{E}[V_{i+1}(f(x_i^{\bar{\pi}},\bar{A}_i^{\bar{\pi}}),F_{i+1})|\bar{F}_i]\big)\Bigg] + \delta^{I-1}r(x_{I-1}^{\bar{\pi}},s_{I-1}(\bar{F}),\bar{A}_{I-1}^{\bar{\pi}})\Bigg\}\Bigg|x_0,F_0\Bigg],\end{aligned} \tag{1.4}$$

where we use the shorthand notation $\bar{D}_i^{\bar{\pi}}$ instead of $\bar{D}_i^{\bar{\pi}}(x_i^{\bar{\pi}},\bar{F}_i)$. This model differs from (1.1) in two key ways: (i) The maximization is inside the expectation because dual policies depend on sample paths, and (ii) its objective function is the sum of the discounted ideally penalized rewards and the last stage reward. Let $V_0(x_0,F_0)$ be the value function for stage 0 and the given state (x_0,F_0), which is obtained in a manner analogous to (1.2) for this stage and state. At optimality, (1.4) equals $V_0(x_0,F_0)$ for each sample path (Brown et al. 2010). It follows that the (1.1) and its dual version (1.4) are equivalent at the optimality. Similar to (1.1), the SDP formulation of (1.4) for each $\bar{F}$ are

$$\begin{aligned} U_i(x_i, F) &= \max_{a_i \in \mathcal{A}_i(x_i)} \Big\{ r_i(x_i, s_i, a_i) - \delta \Big(V_{i+1}\left(f(x_i, a_i), F_{i+1}\right) \\ &\qquad - \mathbb{E}\left[V_{i+1}\left(f(x_i, a_i), F_{i+1}\right) | F_i \right] \Big) + \delta U_{i+1}\left(f(x_i, a_i), F\right) \Big\}, \end{aligned} \tag{1.5}$$

for stage 0 and the given x_0 and each stage $i \in \mathcal{I} \setminus \{0\}$ and $x_i \in \mathcal{X}$, where $U_i(\cdot, F)$ is the stage i dual value function, with $U_I(\cdot, F) := 0$.

Solving (1.4) is no easier than solving (1.1) as the ideal dual penalty relies on the value function $V(x_i, F_i)$ for each state (x_i, F_i). We thus need approximation methods to solve (1.1) and (1.4).

1.4 Approximation Strategies

We can broadly classify the literature about the approximation strategies of solving (1.1) and (1.4) into two streams: (i) primal approximation and (ii) dual approximation. We will briefly review each of them in this section.

Primal Approximation: Most of the approximation strategies belong to this stream. The basic idea is to use VFAs constructed with a set of pre-determined basis functions to approximate the true value function in (1.1). Since the number of basis functions is substantially less than the state space dimension, the resulting model is in a lower and more tractable dimensional space than the original one.

The VFA can be either linear or nonlinear in basis functions. Although a nonlinear model like the deep neural network can potentially capture the true value function better in a more complex state space than the linear model, it is not easy to provide theoretical guarantees for its performance. Thus, the most common choice, in particular for the valuation of financial and real options, is still the linear VFA. Specifically, we define the VFA, denoted as $\hat{V}(x_i, F_i)$, for stage $i \in \mathcal{I} \setminus \{0, I-1\}$ and state $(x_i, F_i) \in \mathcal{X}_i \times \mathcal{F}_i$ as the linear combination

$$\hat{V}(x_i, F_i) := \sum_{b \in \mathcal{B}_i} \beta_{i,x_i,b} \phi_{i,b}(F_i), \tag{1.6}$$

where $\mathcal{B}_i := \{1, ..., B_i\}$ is the index set of B_i basis functions $\phi_{i,b}(F_i)$'s of F_i and $\beta_{i,x_i,b}$ is the weight associated with the b-th such function when the endogenous state is x_i. The associated dual penalty is constructed by replacing $V(x_i, F_i)$ with (1.6).

There is a vast literature on obtaining VFAs defined as (1.6). The state of the art approaches in energy operations and other financial option values include least squares Monte Carlo (LSM, Eydeland and Wolyniec 2003, Glasserman and Yu 2004, Boogert and De Jong 2008, 2011, Gyurkó et al. 2015, Longstaff and Schwartz 2001, Carriere 1996, Tsitsiklis and Van Roy 2001) and approximate linear programming (ALP, De Farias and

Van Roy 2003, 2004, Desai et al. 2012a, Wang et al. 2015, Nadarajah 2014, Adelman 2003, 2004, 2007, Lin et al. 2020). There is a long history of developing these methods, and they have been successfully applied to many applications. A common feature of these methods is that they converge to the true value function from the primal side, i.e., they aim at improving the lower bound.

Once VFA is known, using the VFA (1.6) in lieu of the value function in (1.2) for all endogenous states and solving the resulting SDP gives the stage i decision rule A_i^β of the feasible policy that is greedy with respect to this VFA, that is, the greedy policy π^β.

$$A_i^\beta := \underset{a_i \in \mathcal{A}_i(x_i)}{\arg\max} \left\{ r(x_i, F_i, a_i) + \delta \mathbb{E}[\hat{V}(f(x_i, a_i), F_{i+1}) | F_i] \right\}, \tag{1.7}$$

The associated policy value at (x_i, F_i) is a pointwise estimate of $V(x_i, F_i)$. The sample average of such estimates in a Monte Carlo simulation gives a lower bound on $V(x_i, F_i)$. For the dual bound, we can replace the value function in (1.5) with (1.6) and solve the resulting dual Bellman equations via the backward induction. A dual bound estimator can be obtained analogously to that of the lower bound (See examples in Brown et al. 2010, Brown and Smith 2014, Brown and Haugh 2017, Balseiro and Brown 2019, Lai et al. 2010, Nadarajah and Secomandi 2015, Nadarajah et al. 2015, Secomandi et al. 2015a, Nadarajah and Secomandi 2018b). However, since most methods do not have mechanisms to improve the dual bound, there is no guarantee for its quality.

Dual Approximation: The second stream of literature focuses on computing approximations to the dual value function and penalty. Similar to the primal approximation approach, the dual VFA can be linear, nonlinear in the basis functions or even model free. The methods in this category typically improve the approximated dual value function and the penalty and show that these values converge to their optimal ones. These methods also generate feasible policies by replacing $\hat{V}(f(x_i, a_i), F_{i+1})$ in (1.7) with the approximated dual value functions, though additional steps may be required. In contrast to the primal approximation approach which finds good penalties based on trials and errors, the dual approximation approach formalizes the process and generates tight dual bounds. The dual approach may also improve feasible policies because it generates feasible policies by adjusting anticipated policies, i.e., policies with hindsight information (Brown and Smith 2014, 2021). Pathwise optimization (PO, Desai et al. 2012a) is the state of the art approach. A salient feature of PO is that it generates an approvably tightest dual bound than any other dual approximation approach with a given set of basis functions. To obtain a lower bound and a feasible operating policy, an additional regression step is required. Despite its conceptual appeal, the application of PO is very limited due to the difficulty of solving the underlying linear program in complicated settings.

Compared to the primal approximation approach, the literature for the dual approximation is scant, which makes this area quite active given the importance of generating tight dual bounds (and potentially tight lower bounds based on the approximated dual value function). Some recent developments on this direction include the primal-dual approach in Chen et al. (2020), the reoptimization approach in Trivella et al. (2019), and the reinforcement learning approaches in El Shar and Jiang (2020) and Min et al. (2019).

1.5 Thesis Contributions and Outline

This thesis focuses on using PO to compute lower and dual bounds on the optimal policy value of the MDPs arising in merchant energy operations. Our work extends PO from optimal stopping to MDPs with respective small and large endogenous and exogenous state spaces as well as a finite action space. We identify two computational hurdles for the underlying LP in PO and propose solution methods to deal with those difficulties. The proposed methods significantly improve the efficiency of solving the LP by leveraging preconditioning and convex optimization techniques. We provide analytical supports for each of the proposed methods and compare their performance with LSM, the state of the art approach, under realistic ethanol production and natural gas storage instances. We have three main insights based on our numerical studies: (i) PO generates substantially tighter dual bounds than LSM, which is not known in literature before our study, (ii) based on (i), both PO and LSM generate near optimal lower bounds and feasible operating policies, thus the large gap between the LSM lower and dual bound is almost entirely due to the looseness of its dual bound, and (iii) PO also generates slightly tighter lower bounds than LSM. These findings provide clear answers to the conjecture discussed in §1.1. They also highlight the potential of PO in generating tight bounds for merchant energy operations.

Although we focus on merchant energy production and storage in this thesis, our approaches have potential applicability in problems like swing options and chooser caps, inventory control, network revenue management, and portfolio optimization (see Brown and Smith 2021 and the reference therein).

We list the three chapters contained in this thesis and introduce the contributions of each chapter in detail.

Chapter 2: This chapter studies merchant energy production modeled as a classical real option. LSM combined with information relaxation and duality is a state-of-the-art reinforcement learning methodology to obtain operating policies and optimality gaps for related models. PO is a competing technique developed for optimal stopping settings, in which it typically provides superior results compared to this approach, albeit with a larger computational effort. We apply these procedures to merchant energy production. Employing PO requires methodological extensions. We use principal component analysis and block coordinate descent in novel ways to respectively precondition and solve the ensuing ill-conditioned and large-scale linear program, which even a cutting-edge commercial solver is unable to handle directly. Both techniques yield near optimal operating policies on realistic ethanol production instances. However, at the cost of both considerably longer run times and greater memory usage, PO leads to substantially tighter dual bounds compared to LSM, even when specified in a simple fashion, complementing it in this case.

Chapter 3: This chapter studies a constraint generation method to solve the PO linear program. The proposed method iterates between master and subproblems. The master problem combines subsets of the constraints and variables of the model to obtain a relaxed linear program that an off-the-shelf solver can handle. The subproblem strengthens this relaxation by efficiently identifying constraint violations in the original linear program. This method provably converges. It can be stopped once the current solution is sufficiently close to the optimal one, which we can check using a bound that we compute when solving the subproblem, or it leads to good-enough bounds on our MDP optimal policy value. We have verified both the efficiency and effectiveness of this version of our method on the benchmark merchant ethanol production instances employed in Chapter 2. Compared to the first chapter, the constraint generation method generates similar results with less memory but longer CPU times. However, the way the constraint generation deals with the large exogenous state space can be applied to instances with large endogenous state. Thus, it motivates further research into its extension.

Chapter 4: In this chapter, we generalize PO from merchant energy production to general energy operations and real option models. The considered MDPs feature respectively high and low dimensional exogenous and endogenous state components as well as a finite action space. The original PO potentially has unbounded optimal solutions and objective value if applied to these MDPs. The state of the art approach in dealing with PO also has an excessive memory requirement in complicated settings. We fix the unboundedness issue by adding constraints based on nonanticipated policies and develop a decomposition and regression approach to reduce the memory requirement. We test the effectiveness of these techniques on respective natural gas storage and ethanol production. The extended PO generates tight dual bound and near optimal lower bound for natural gas storage. In ethanol production instances, the proposed algorithms can solve both existing instances faster with significantly less memory than the state-of-the-art method and new larger-size ones that were out of reach, achieving near-optimal performance and dominating a standard competitor in terms of solution quality.

Conclusion and Appendices A-B: We summarize the contributions and discuss the future research in Conclusion. Appendices A-B include proofs and supporting materials for Chapter 2-4, respectively.

Chapter 2

Pathwise Optimization for Reinforcement Learning in Merchant Energy Production

2.1 Introduction

We study the merchant management of energy production assets, such as power and natural-gas-processing plants, oil and bio refineries, and ethanol manufacturing facilities (Tseng and Barz 2002, Tseng and Lin 2007, Devalkar et al. 2011, Thompson 2013, Dong et al. 2014, Boyabatli et al. 2017, Nadarajah and Secomandi 2018b), whereby these assets are operated by trading their inputs and outputs in wholesale markets to take advantage of favorable prices. Modeling as a portfolio of real options the ability of the managers of these assets to dynamically adapt their operating policies to changing market conditions provides a convenient approach to maximize their market values (Dixit and Pindyck 1994, Trigeorgis et al. 1996, Smith and McCardle 1998, 1999, Eydeland and Wolyniec 2003, Geman 2005, Guthrie 2009, Secomandi and Seppi 2014, 2016).

Managing an ethanol factory in wholesale markets (Guthrie 2009, Chapter 17) exemplifies the main ideas underlying merchant energy production. Operating the plant is desirable when the spread between the output and input wholesale prices net of the conversion cost is attractive. Temporary or prolonged periods of unappealing spreads can be dealt with by suspending production or mothballing the plant. In the latter case, reactivating or abandoning operations is advisable when the spread improves or worsens. This managerial flexibility can be modeled as a compound switching and timing option on the uncertain evolution of the prices of ethanol and of corn and natural gas (the raw materials).

As is typical in merchant energy operations (Secomandi and Seppi 2014, 2016), real option models of energy production give rise to intractable Markov decision processes

(MDPs). In each stage the MDP state contains the status of both the plant and the market. The choices of the merchant producer determine the evolution of the former component. Given stochastic processes govern the dynamics of the latter one. Further, they are independent of these decisions based on small plant size and price taking assumptions. That is, these actions do not affect market prices because they are small relative to the depth of the markets in which they take place, so that these prices are taken as fixed when quantifying cash flows. Intertemporal linkages between operational conditions and high dimensional market information (input and output current futures curves) lead to (some of) the well-known "curses of dimensionality" (Powell 2007, §1.2). Reinforcement learning (RL) methods are thus used to obtain (operating) policies and optimality gaps.

Combining least squares Monte Carlo (LSM) and information relaxation and duality techniques (Carriere 1996, Longstaff and Schwartz 2001, Smith 2005, Cortazar et al. 2008, Brown et al. 2010, Secomandi and Seppi 2016, Nadarajah et al. 2017, Secomandi 2017, Nadarajah and Secomandi 2017, 2018a) is a state-of-the-art RL approach for intractable merchant operations MDPs. LSM uses Monte Carlo sample paths of the market uncertainty and regression to compute a value function approximation (VFA) expressed as a linear combination of basis functions that defines both a policy and dual penalties on hindsight information, thus enabling the estimation of both lower and upper (dual) bounds on the optimal policy value. Pathwise optimization (PO, Desai et al. 2012b, Chandramouli 2019) is an RL approach developed for optimal stopping models that first solves a linear program formulated on analogous sample paths to find the best dual penalties constructed using a VFA specified as in LSM. It then obtains both a VFA that determines a policy and its optimality gap. Desai et al. (2012b) show that PO dominates LSM in terms of solution quality at the expense of longer run times, whereas the bound-related findings of Chandramouli (2019) are less conclusive than the ones of these authors.

We extend the limited extant research that compares LSM and PO from optimal stopping to merchant energy production. We formulate an MDP that relies on a model for the evolution of the prices of futures contracts (term structures) on the inputs and the output. Using LSM in this context is standard whereas the application of PO necessitates methodological development.

Our PO linear program (PLP) is hard to solve in the specified setting despite our use of an advanced commercial solver such as Gurobi Optimization (2021). This difficulty distinguishes even instances with short horizons because PLP is ill-conditioned, that is, has a large condition number. Scale is an additional complication for instances with long horizons for the stated specification. We address these issues by developing (i) an exact preconditioning procedure based on principal component analysis (PCA) and (ii) a block coordinate descent (BCD) optimization method. Our PCA procedure exploits the block diagonal structure of the dual-penalty component of PLP for efficiency and exactly reformulates it by making its columns orthogonal within each block, implicitly leading to new dual penalties that are defined in terms of basis functions that are particular linear combinations of the original ones. We provide some theoretical support for the effectiveness of this approach. Our BCD algorithm solves this preconditioned linear program by iteratively optimizing the values of groups of decision variables, while fixing the ones of the remaining variables. It thus requires less memory than employing a monolithic approach, which is impractical due to its excessive memory requirement for large instances. We establish that an idealized version of our BCD technique converges to an optimal solution.

We numerically assess the performance of both LSM and PO on a set of realistic instances with the futures price model calibrated to market data and values of the operational parameters based on the literature. The estimated optimality gaps of the PO-and LSM-based policies, both obtained using the PO-based dual bounds, are on average 7% and 8%, respectively, whereas the latter ones grow to 11% if the LSM-based dual bounds are used. PO thus plays an important role in establishing the effectiveness of both these policies. Computationally PO is considerably more onerous than LSM, on average taking several hours instead of minutes per instance. When PO is specified with simple basis functions, its run times almost halve and it leads to strong and weak dual and greedy bounds, respectively, thus complementing the LSM ability to determine good greedy bounds. Finally, we analyze the operating choices made by different operating policies. The PO-based policy tends to abandon the plant sooner than the LSM-based one. The optimal static policy, which optimizes the net present value of discounted cash flows ignoring uncertainty, performs poorly on our instances because it quickly ceases operations. This finding brings to light the importance of adopting a good dynamic policy in the considered setting.

Our research is potentially relevant for other commodity merchant operations contexts and related real option models (Secomandi and Seppi 2014, 2016), including oil and natural gas extraction fields, liquefied natural gas facilities, copper mines, and renewable energy plants (Brennan and Schwartz 1985, Smith and McCardle 1998, 1999, Kamrad and Ernst 2001, Cortazar et al. 2008, Rømo et al. 2009, Enders et al. 2010, Lai et al. 2011, Arvesen et al. 2013, Denault et al. 2013, Hinz and Yee 2018, Zhou et al. 2019), because they feature configurations of the MDP states and their dynamics that are analogous to the ones of our MDP. Specifically, these models are finite horizon and discrete time MDPs with states that include a few operating levels and several market variables, dynamics of these components that depend on the decisions made and given stochastic processes, respectively, and a small number of actions (including models with continuous operating levels and choices that can be optimally discretized). Our work motivates additional research aimed at reducing the computational burden of the PO approach.

The MDP that we consider is based on the real option ethanol production model presented in Guthrie (2009, Chapter 17). This author uses a one-factor model of the processing spread and represents its evolution using a binomial tree, which enables the use of stochastic dynamic programming for obtaining a corresponding policy. In contrast, we employ a multifactor model of the dynamics of the input and output futures curves, which is common in the commodity and energy merchant operations literature (see, e.g., Clewlow and Strickland 2000, Eydeland and Wolyniec 2003, Lai et al. 2010), and rely on RL methods to approximately solve the resulting intractable MDP.

Desai et al. (2012b) and Chandramouli (2019) employ PO for optimal stopping, formulating their linear programs by enumerating the payoffs of every possible stopping choice, of which there are multiple ones in Chandramouli (2019). This approach is intractable with more than one stopping time. Similarly, it leads to a linear optimization model with an exponential number of constraints when applied to merchant energy production. We thus specify PLP by exploiting the dynamic programming formulation of the MDP dual model, which allows us to avoid this issue.

Desai et al. (2012b) apply a commercial linear programming solver to readily optimize their PO model. Chandramouli (2019) approximately solves his PO linear program as a sequence of single stopping PO models formulated and optimized as in Desai et al. (2012b). Different from the decomposition approach of this author, we find a near optimal solution to PLP using a BCD algorithm that sequentially solves more manageable PLP versions until a given termination criterion is met. The literature on BCD methods is extensive (see, e.g., Sargent and Sebastian 1973, Grippo and Sciandrone 2000, Tseng 2001, Nesterov 2012, Richtárik and Takáč 2014, Bertsekas 2015, Chapter 6, and references therein). Our use of this methodology in a PO setting is novel. The idealized version of the cyclic BCD algorithm and its theoretical analysis are based on common assumptions (see, e.g., Bertsekas 2015, §6.5).

Desai et al. (2012b) and Chandramouli (2019) do not report any computational instabilities in their applications nor do they arise when we employ LSM in ours. In contrast, we observe that PLP is severely ill-conditioned, which gives rise to numerical issues. Preconditioning is a common technique to facilitate solving mathematical programs that suffer from ill-conditioning, in particular ones with linear constraints (see, e.g., Renegar 1995a,b, Cheung and Cucker 2001, Epelman and Freund 2002, Belloni and Freund 2009, Amelunxen and Burgisser 2012, Peña et al. 2014, and references therein). Employing PCA for preconditioning, as we do, rather than dimensionality reduction, as commonly done in the literature (see, e.g., Jolliffe 2002), appears to be unique. Our theoretical analysis of the effectiveness of this approach is rooted in the literature on condition numbers (see, e.g., Zhang and Adelman 2009, Golub and Van Loan 2012 and references therein).

We are not the first to observe collinearity in an RL setting. For example, (Ariyajunya et al. 2021) describe state space multicollinearity issues that negatively affect the performance of the RL method they use to solve an ozone pollution model. They propose two techniques to rectify this deficiency, one of which is based on orthogonalization through feature extraction. Although our PCA approach is based on a similar idea, it applies to the components of dual penalties related to basis functions associated with a VFA in a PO linear program rather than the state space of an MDP.

The outcome of our numerical comparison of the LSM- and PO-based bounds is directionally similar to the one of Desai et al. (2012b) for optimal stopping, whereas the analogous finding of Chandramouli (2019) for multiple optimal stopping is mixed. However, we observe that PO achieves considerably smaller and larger improvements in the quality of the estimated lower and dual bounds, respectively, relative to LSM than they do.

Trivella et al. (2019) study a version of the merchant commodity production model of Nadarajah and Secomandi (2018b, 2020) that discourages abandonment using LSM and duality methods. The LSM-based bounds that we obtain are looser than the ones of these authors. Further, they are not as tight as the using LSM-based bounds computed by Nadarajah et al. (2017) and Nadarajah and Secomandi (2017) for energy storage, Nadarajah et al. (2017) for swing options, and Nadarajah and Secomandi (2018a) for networks of energy storage and transport assets.

Section 2.2 presents the merchant energy production MDP that we study. We introduce the approximate solution approach that we adopt to obtain policies and dual bounds based on VFAs in §2.3. Section §2.4 discusses how to use LSM and PO to obtain VFAs. We present the PCA and BCD algorithms we develop to solve PLP in §2.5. Section 2.6 reports the results of our numerical study. We conclude in §2.7. An appendix includes the proofs of formal results.

2.2 Model

This section presents a real option model of the merchant management of energy production. We provide a simple description of this formulation in §2.2.1 and formulate an MDP in §2.2.2. The contents of §2.2.1 and §2.2.2 largely rely on material available in Guthrie (2009, Chapter 17) and Nadarajah and Secomandi (2020, §4), which is itself in part based on Guthrie (2009, Chapter 17).

2.2.1 Informal Overview

For concreteness we consider a plant that converts corn and natural gas into ethanol. A merchant manages this facility by periodically making operating choices and transacting on a spot basis in wholesale markets to buy and sell, respectively, the inputs and output of the manufacturing process. The current and anticipated conditions of the conversion spread, which is the difference of the ethanol spot price and the sum of both the spot prices of corn and natural gas scaled by their respective requirements and the marginal production cost, play an important role in determining these decisions. Suppose the plant is operational. If the spot conversion spread is positive then it is beneficial to source, produce, and sell at full capacity. If this spread is negative then the merchant can avoid receiving it by suspending production, mothballing the plant, or abandoning it. Suspension keeps the plant operational at a cost, which is incurred every time this decision is taken. Mothballs changes the status of the plant from operational to mothballed. Compared to suspension, this choice requires paying a one time cost, a cost for each period during which the plant is kept mothballed, which is however lower than the cost of suspending production, and a one time cost if the plant is reactivated later on, that is, it becomes operational again. Abandonment is associated with a salvage value and foregoes all cash flows that the merchant may otherwise earn in the future. Intuitively, the merchant can suspend production or mothball the plant when anticipating short or long lived unappealing conversion spreads. Further, reactivating or abandoning the plant once it is mothballed can take advantage or limit the negative effect of improved or worsened anticipated conversion spreads. Abandonment does not require that the asset be first mothballed. It is compulsory at the end of the facility lifetime. Whereas deciding when to produce is easy, the challenge lies in deciding which market conditions warrant adopting one of the available switching (suspension, mothballs, and reactivation) or timing (abandonment) decisions with the goal of maximizing the residual market value of the asset.

2.2.2 MDP

The residual life of the plant includes I decision dates. The stage set $\mathcal{I} := \{0, 1, \ldots, I-1\}$ includes their indices.

The operational and mothballed operating modes are O and M, respectively. Deciding to mothball or reactivate the plant in a given stage leads to a mothballed or operational facility in the next stage (longer such transitions can be accommodated as in Guthrie (2009, Chapter 17), and Nadarajah and Secomandi (2020, §4.1). The abandoned operating mode is denoted by A. The stage i operating mode set is $\mathcal{X}_i$ and x_i is an element of this set. This set equals $\{\mathsf{O}\}$ if $i = 0$, that is, the plant is initially operational, and $\{\mathsf{A}, \mathsf{M}, \mathsf{O}\}$ otherwise.

We denote as P the decision to produce Q gallons of ethanol using γ_{C} bushels of corn and γ_{N} mmBTU of natural gas each per gallon of output. We label as S, M, R, and A the suspension, mothballing, reactivation, and abandonment actions, respectively. The feasible action set corresponding to the operating mode x_i is $\mathcal{A}_i(x_i)$. It is defined as follows:

- If $x_i = \mathsf{O}$ then $\mathcal{A}_i(x_i)$ equals $\{\mathsf{A}, \mathsf{M}, \mathsf{P}, \mathsf{S}\}$ if $i \neq I-1$ and $\{\mathsf{A}\}$ if $i = I-1$.

- If $x_i = \mathsf{M}$ then $\mathcal{A}_i(x_i)$ equals $\{\mathsf{A}, \mathsf{M}, \mathsf{R}\}$ if $i \notin \{0, I-1\}$ and $\{\mathsf{A}\}$ if $i = I-1$.

- If $x_i = \mathsf{A}$ then $\mathcal{A}_i(x_i) = \{\mathsf{A}\}$.

The function $f(x_i, a_i)$ gives the next stage operating mode that results from executing feasible action a_i in the current stage when the operating mode is x_i. In particular, its value is O if the pair (x_i, a_i) belongs to the set $\{(\mathsf{O}, \mathsf{P}), (\mathsf{O}, \mathsf{S}), (\mathsf{M}, \mathsf{R})\}$ and ai in all other cases. Figure 2.1 illustrates these transitions.

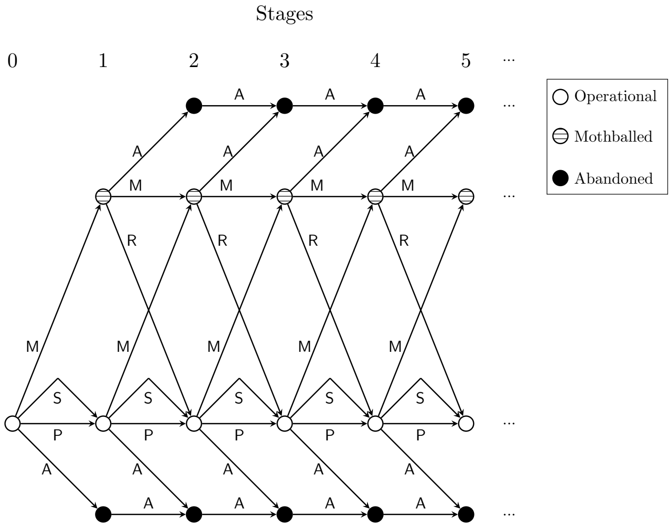


Figure 2.1: Illustration of function $f(x_i, a_i)$

We abbreviate corn, ethanol, and natural gas to C, E, and N, respectively, and include these labels in set $\mathcal{C}$. The spot price of commodity $c \in \mathcal{C}$ in stage i is $s_i^c \in \mathbb{R}_+$. The total spot conversion spread is $(s_i^{\mathrm{E}} - \gamma_{\mathrm{C}} s_i^{\mathrm{C}} - \gamma_{\mathrm{N}} s_i^{\mathrm{N}}) * Q - \mathsf{C}_{\mathsf{P}}$, where C_{P} is the cost of producing Q gallons of ethanol in addition to the cost of purchasing the two inputs; that is, the merchant is a price taker. The respective costs per stage of producing Q gallons of ethanol, suspending production, and keeping the plant fully mothballed are C_{P}, C_{S} $(< \mathsf{C}_{\mathsf{P}})$, and C_{M} $(< \mathsf{C}_{\mathsf{S}})$ dollars. The one time costs of mothballing and reactivating the plant are I_{M} and I_{R}, respectively. Abandoning the plant yields a net salvage value of S (this notation differs from the S label used to denote the suspension action). The per stage reward depends on the operating mode x_i, the spot price vector $s_i := (s_i^c, c \in \mathcal{C})$, and the action $a_i \in \mathcal{A}(x_i)$:

$$r(x_i, s_i, a_i) := \begin{cases} (s_i^{\mathrm{E}} - \gamma_{\mathrm{C}} s_i^{\mathrm{C}} - \gamma_{\mathrm{N}} s_i^{\mathrm{N}})Q - \mathsf{C}_{\mathsf{P}}, & \text{if } (x_i, a_i) \in (\mathsf{O}, \mathsf{P}), \\ -\mathsf{C}_{\mathsf{S}}, & \text{if } (x_i, a_i) = (\mathsf{O}, \mathsf{S}), \\ -\mathsf{I}_{\mathsf{M}}, & \text{if } (x_i, a_i) = (\mathsf{O}, \mathsf{M}), \\ -\mathsf{C}_{\mathsf{M}}, & \text{if } (x_i, a_i) = (\mathsf{M}, \mathsf{M}), \\ -\mathsf{I}_{\mathsf{R}}, & \text{if } (x_i, a_i) = (\mathsf{M}, \mathsf{R}), \\ S, & \text{if } (x_i, a_i) \in \{(\mathsf{O}, \mathsf{A}), (\mathsf{M}, \mathsf{A})\}, \\ 0, & \text{if } (x_i, a_i) \in \{(\mathsf{A}, \mathsf{A})\}. \end{cases}$$

The forward curve for a given commodity includes both its spot price and the prices of the futures for a set of traded maturities. We formulate the evolution of the spot prices of corn, ethanol, and natural gas using a model of the joint dynamics of the forward curves of these commodities. Specifically, as in Cortazar and Schwartz (1994), Clewlow and Strickland (2000, Chapter 8), and Secomandi and Seppi (2014, §4.3), we employ a Markovian multifactor model of the evolution of the vector of these forward curves under the assumption that the merchant's decisions do not affect it; that is, the plant is small relative to the market size. We take the stages to be the maturities of the futures. The price in stage i of the futures for commodity c with delivery in stage $j \geq i$ is $F_{i,j}^c \in \mathbb{R}_+$. If i equals j then $F_{i,i}^c$ and s_i^c coincide. Given $F_{i,j}^c$ the price $F_{i+1,j}^c$, with $j > i$, satisfies

$$F_{i+1,j}^c = F_{i,j}^c \exp\left[-\frac{1}{2}(T_{i+1} - T_i) \sum_{k=1}^{K} \sigma_{c,i,j,k}^2 + \sqrt{T_{i+1} - T_i} \sum_{k=1}^{K} \sigma_{c,i,j,k} W_k \right], \tag{2.1}$$

where T_i is the time corresponding to stage i, K is the number of stochastic factors, which are common to all commodities and maturities, $\sigma_{c,i,j,k}$ is the stage i loading coefficient on the k-th factor for the futures of commodity c with maturity in stage j, and W_k is a standard normal random variables that is uncorrelated with the other $K - 1$ ones (the loading coefficients embed the correlation between price changes). This model is driftless because it is specified under a risk neutral measure. The stage i forward curve of commodity c, F_i^c, is the vector $(F_{i,j}^c, j \in \{i, \ldots, I - 1\})$. The vector of forward curves in stage i is $F_i := (F_i^c, c \in \mathcal{C})$. It takes values in $\mathbb{R}_+^{3(I-i)}$. According to model (2.1), given the vector of forward curves F_i in stage i the stage $i + 1$ vector of forward curves F_{i+1}, which includes the spot price vector s_{i+1}, is jointly lognormally distributed with parameters that depend on F_i, e.g., F_i is the mean of F_{i+1}. In particular, the distribution of the spot price vector s_{i+1} depends on the vector $(F_{i,i+1}^c, c \in \mathcal{C})$.

The state of our MDP in stage i includes the operating mode x_i and the vector of forward curves F_i. The stage i state space is thus $\mathcal{X}_i \times \mathbb{R}^{3(I-i)_+}$. The presence of the vector of forward curves in the state is needed to be able to determine the stochastic transitions of this part of the state. This vector affects the merchant's optimal decision in a particular stage and state because it conditions the evolution of the future market conditions, that is, it is the basis of the merchant's anticipation of such conditions. For example, it is reasonable to surmise that for a given stage the merchant should be less prone to optimally mothball or abandon the plant in a state with vectors of forward curves that correspond to positive expected future spot conversion margins compared to states with forward curves associated with negative such margins.

A feasible policy π is the collection of decision rules $\{A_i^\pi, i \in \mathcal{I}\}$, with $A_i^\pi : \mathcal{X}_i \times \mathbb{R}_+^{3(I-i)} \to \mathcal{A}_i(x_i)$. The set of such policies is Π. The objective is to choose a feasible policy that maximizes the market value of operating the plant during the finite horizon given the initial state (x_0, F_0):

$$\max_{\pi \in \Pi} \sum_{i \in \mathcal{I}} \delta^i \mathbb{E}\left[r\left(x_i^\pi, s_i, A_i^\pi\left(x_i^\pi, F_i\right)\right) \mid x_0, F_0\right], \tag{2.2}$$

where δ is the per stage risk free discount factor; $\mathbb{E}$ is the expectation taken w.r.t. the vector of forward curves under a risk neutral measure; and x_i^π is the random operating mode reached in stage i when using policy π.

2.3 Approximate Solution Approach

We discuss a typical approximate solution approach for obtaining a feasible policy and estimating bounds on the optimal policy value of model (2.2) that relies on VFAs (see, e.g., (Powell 2007, §6.4) and (Brown et al. 2010)). We discuss how to obtain both a policy that is greedy with respect to a given VFA and its corresponding lower bound estimate in §2.3.1 and the estimation of a dual bound in §2.3.2. This section is partly based on Nadarajah and Secomandi (2020, §3.1).

2.3.1 Greedy Policy and Lower Bound

The optimality conditions of our MDP are

$$V_i(x_i, F_i) = \max_{a_i \in \mathcal{A}_i(x_i)} \left\{ r_i(x_i, s_i, a_i) + \delta \mathbb{E}\left[V_{i+1}(f(x_i, a_i), F_{i+1}) | F_i \right] \right\}, \tag{2.3}$$

for stage 0 and the initial state (x_0, F_0) and each stage $i \in \mathcal{I} \setminus \{0\}$ and state $(x_i, F_i) \in \mathcal{X} \times \mathbb{R}_+^{3(I-i)}$, where V_i is the stage i value function, with $V_I(\cdot, \cdot) := 0$. The stage i decision rule of an optimal policy is

$$\arg\max_{a_i \in \mathcal{A}_i(x_i)} \left\{ r_i(x_i, s_i, a_i) + \delta \mathbb{E}\left[V_{i+1}(f(x_i, a_i), F_{i+1}) | F_i \right] \right\}, \tag{2.4}$$

where in case of a tie we weakly prefer producing to suspending, mothballing, or abandoning; suspending to mothballing or abandoning; and mothballing to abandoning. Obtaining the value function by solving (2.3) is in general intractable, due to the high dimensionality of the forward curve and the difficulty of evaluating the expectation in this model. Thus, computing an optimal policy based on (2.4) is impractical.

To obtain a feasible policy, we replace the value function in (2.4) with a low dimensional VFA for states that include the mothballed or operational operating modes, M and O, which we include in set $\mathcal{X}'$. We specify the VFA for stage $i \in \mathcal{I} \setminus \{0, I-1\}$ and state $(x_i, F_i) \in \mathcal{X}' \times \mathbb{R}_+^{3(I-i)}$ as the linear combination

$$\sum_{b \in \mathcal{B}_i} \beta_{i,x_i,b} \phi_{i,b}(F_i), \tag{2.5}$$

where $\mathcal{B}_i := \{1, ..., B_i\}$ is the index set of B_i basis functions $\phi_{i,b}(F_i)$'s of the vector of forward curves F_i and $\beta_{i,x_i,b}$ is the weight associated with the b-th such function when the operating mode is x_i. Using the VFA (2.5) in lieu of the value function in (2.4) for states that include the operational or mothballed operating mode, that is, $x_i \in \mathcal{X}'$, gives the stage i decision rule A_i^β of the feasible policy that is greedy with respect to this VFA, that is, the greedy policy π^β (here $\mathbb{1}(\cdot)$ is the indicator function that evaluates to one when its argument is true and to zero otherwise):

$$\operatorname*{arg\,max}_{a_i \in \mathcal{A}_i(x_i)} \left\{ r(x_i, s_i, a_i) + \mathbb{1}\left(i \neq I-1, f(x_i, a_i) \in \mathcal{X}'\right) \delta \sum_{b \in \mathcal{B}_i} \beta_{i+1, f(x_i, a_i), b} \mathbb{E}[\phi_{i+1,b}(F_{i+1}) | F_i] \right\}, \tag{2.6}$$

Using the greedy policy is particularly practical if the expectation in (2.6) can be evaluated efficiently, a condition that we state as Assumption 3.

Assumption 1. *The expectation $\mathbb{E}[\phi_{j,b}(F_j)|F_i]$ can be computed in closed form for each i and $j \in \mathcal{I} \setminus \{0, I-1\}$ with $j > i$ and $F_i \in \mathbb{R}_+^{3(I-i)}$.*

This assumption holds for commonly used basis functions and stochastic models for the evolution of the vector of forward curves (see, e.g., Glasserman and Yu 2004, Nadarajah et al. 2017 and references therein), which we use in this work. If Assumption 3 is not satisfied then the expectation in (2.6) needs to be approximated, e.g., as a sample average based on Monte Carlo simulation of the next stage vector of forward curves conditional on their values in the current stage.

A lower bound on the optimal policy value can be estimated by applying the greedy policy on a set of samples of vectors of forward curves, obtained via Monte Carlo simulation, and operating modes, resulting by using this policy, in each stage starting from the initial stage and state, and averaging the sum of the resulting discounted rewards collected on each such path. Specifically, we let $\mathcal{L} := \{1, 2, ..., L\}$ be the index set of the samples of the array of forward curves from stage 0 through stage $I-1$ and estimate this bound as

$$\frac{1}{L}\sum_{l\in\mathcal{L}}\sum_{i\in\mathcal{I}}\delta^i r\left(x_i^{\pi^\beta,l}, s_i^l, A_i^\beta\left(x_i^{\pi^\beta,l}, F_i^l\right)\right), \tag{2.7}$$

where $x_i^{\pi^\beta,l}$ is the operating mode reached under policy π^β in stage i on sample l, s_i^l and F_i^l are the vectors of spot prices and forward curves, respectively, for this stage and sample, and $\left(x_0^{\pi^\beta,l}, F_0^l\right)$ equals the given initial state (x_0, F_0) for each sample l.

2.3.2 Dual Bound

The quality of a greedy policy can be assessed by comparing its estimated value against a perfect information dual bound on the optimal policy value. The idea behind this approach is to optimize the operating policy assuming perfect foresight of the future uncertainty, that is, the entire array of the vectors of forward curves in our application. Intuitively, the expected value of the resulting total discounted cash flows is an upper bound on the optimal policy value, albeit typically a loose one. The information relaxation and duality methodology aims at strengthening this bound, eliminating in theory all and otherwise some of the benefit of foreknowledge by imposing ideal and good, respectively, penalties on the amount of information that is not supposed to be known when making decisions.

Let F be the array of vectors of forward curves from the initial stage through the terminal one, $F := (F_i)_{i\in\mathcal{I}}$. The ideal dual penalty when taking feasible action a_i in state (x_i, F_i) with knowledge of each element of F is the exact additional value of also knowing the stage $i+1$ vector of forward curves F_{i+1} in stage i rather than only the vector of forward curves F_i:

$$\delta\left\{V_{i+1}\left(f(x_i,a_i),F_{i+1}\right)-\mathbb{E}\left[V_{i+1}\left(f(x_i,a_i),F_{i+1}\right)|F_i\right]\right\}. \tag{2.8}$$

For each array of vectors of forward curves $F\in\mathbb{R}^{3\sum_{i\in\mathcal{I}}(I-i)}$, the dual versions of the optimality conditions (2.3) are

$$\begin{aligned} U_i(x_i,F) \;&=\; \max_{a_i\in\mathcal{A}_i(x_i)}\left\{r_i(x_i,s_i,a_i)-\delta\left\{V_{i+1}\left(f(x_i,a_i),F_{i+1}\right)-\mathbb{E}\left[V_{i+1}\left(f(x_i,a_i),F_{i+1}\right)|F_i\right]\right\}\right.\\ &\qquad\left.+\delta U_{i+1}\left(f(x_i,a_i),F\right)\right\}, \end{aligned} \tag{2.9}$$

for stage 0 and the given operating mode x_0 and each stage $i\in\mathcal{I}\setminus\{0\}$ and operating mode $x_i\in\mathcal{X}$, where $U_i(\cdot,F)$ is the stage i dual value function, with $U_I(\cdot,F):=0$. The use of ideal dual penalties in (2.9) implies that the identity $U_i(x_i,F)\equiv V_i\left(x_i,F_i\right)$ holds for each stage i, operating mode x_i, and array of vectors of forward curves F with probability one. That is, these penalties completely remove the advantage of currently knowing the values of the future vectors of forward curves. The dual bound is $\mathbb{E}\left[U_0(x_0, F) \mid x_0, F_0\right]$, which is trivially tight because $U_0(x_0, F)$ equals $V_0(x_0, F_0)$ with probability one.

Because (2.3) is generally intractable, ideal dual penalties and the corresponding dual bound cannot be computed. However, these penalties can be approximated based on a VFA to obtain so called good dual penalties. Further, a corresponding valid dual bound can be estimated using the samples of the array of vectors of forward curves used to estimate the lower bound. Define $\Delta_i^{\mathbb{E},F}\phi_{i+1,b} := \delta\left\{\phi_{i+1,b}\left(F_{i+1}\right) - \mathbb{E}\left[\phi_{i+1,b}\left(F_{i+1}\right)|F_i\right]\right\}$. Good dual penalties result from substituting with a VFA the value function in the expression that defines them, that is, (4.3). The particular ones that ensue from using our VFA are

$$\sum_{b\in\mathcal{B}_i}\beta_{i+1,f(x_i,a_i),b}\Delta_i^{\mathbb{E},F}\phi_{i+1,b}, \tag{2.10}$$

with the provision that $f(x_i, a_i)$ is an element of set $\mathcal{X}'$. These dual penalties lead to the following model, which resembles (2.9) but is based on such penalties rather than the ideal ones, for each array of vectors of forward curves $F \in \mathbb{R}^{3\sum_{i\in\mathcal{I}}(I-i)}$:

$$\begin{aligned} U_i^{\beta}(x_i, F) \quad = \quad \max_{a_i\in\mathcal{A}_i(x_i)} \Bigg\{ & r\left(x_i, s_i, a_i\right) - \mathbb{1}\left(i \neq I-1, f(x_i,a_i) \in \mathcal{X}'\right)\sum_{b\in\mathcal{B}_i}\beta_{i+1,f(x_i,a_i),b}\Delta_i^{\mathbb{E},F}\phi_{i+1,b} \\ & +\delta U_{i+1}^{\beta}\left(f(x_i,a_i), F\right)\Bigg\}, \end{aligned} \tag{2.11}$$

for stage 0 and operating mode x_0 and each stage each $i \in \mathcal{I}\setminus\{0, I-1\}$ and operating mode $x_i \in \mathcal{X}$, where $U_i^{\beta}(\cdot, F)$ is the approximate dual value function corresponding to using the VFA associated with the weight vector β to obtain the good penalties (4.5), with $U_I^{\beta}(\cdot, F) := 0$. The expectation $\mathbb{E}\left[U_0^{\beta}(x_0, F) \mid x_0, F_0\right]$ is a valid dual bound on $V_0(x_0, F_0)$. An unbiased estimate of this bound can be obtained based on solving a version of (2.11) formulated on the samples of the array of vectors of forward curves indexed by the set of indices $\mathcal{L}$. Let F^l be this array for sample $l \in \mathcal{L}$. The sample average

$$\frac{1}{L}\sum_{l\in\mathcal{L}}U_0^{\beta}\left(x_0, F^l\right) \tag{2.12}$$

is an unbiased estimate of $\mathbb{E}\left[U_0^{\beta}(x_0, F) \mid x_0, F_0\right]$. When Assumption 3 holds the dual penalties (4.5) can be evaluated efficiently and each term $U_0^{\beta}\left(x_0, F^l\right)$ can be readily computed by backward dynamic programming.

2.4 PO

PO works with two VFAs: One is associated with the greedy policy and its corresponding lower bound, the other one with the dual bound. We first discuss the latter VFA because the method we use to obtain it yields the inputs needed to determine the former one.

The key idea behind PO is to obtain a VFA by finding values for its basis functions weights that lead to the smallest dual bound among all the ones that are supported by good penalties corresponding to such VFAs. Specifically, this approach entails solving

$$\min_{\beta} \mathbb{E}\left[U_0^{\beta}(x_0, F)|x_0, F_0\right], \tag{2.13}$$

where the decision variable vector β is an element of $\mathbb{R}^{\sum_{i\in\mathcal{I}\setminus\{0,I-1\}}\sum_{b\in\mathcal{B}_i}}$.

Exact evaluation of the objective function of model (2.13) is typically impossible. A natural approach is to solve instead its sample average approximation formulated using samples of the array of vectors of forward curves. For this purpose we employ the same such samples used by LSM, that is, the ones indexed by set $\mathcal{L}'$. The resulting model is

$$\min_{\beta} \frac{1}{L'} \sum_{l\in\mathcal{L}'} U_0^{\beta}\left(x_0, F^l\right), \tag{2.14}$$

which has a piecewise linear convex objective function (see Desai et al. 2012b). As discussed in §2.3.2, for a fixed vector β, each term $U_0^{\beta}\left(x_0, F^l\right)$ solves model (2.11) expressed with F^l in lieu of F. Observing that this version of (2.11) can be equivalently formulated as a linear program using the approach of (Manne 1960) leads to the following alternative linear programming representation of (2.14), that is, PLP:

$$\begin{aligned} \min_{\beta,u} \quad & \frac{1}{L'} \sum_{l\in\mathcal{L}'} u_{0,x_0,l} && (2.15) \\ \text{s.t.} \quad & u_{i,x_i,l} \geq r\left(x_i, s_i^l, a_i\right) - \mathbb{1}\left(i \neq I-1, f(x_i,a_i) \in \mathcal{X}'\right) \sum_{b\in\mathcal{B}_i} \beta_{i+1,f(x_i,a_i),b} \Delta_i^{\mathbb{E},l} \phi_{i+1,b} \\ & \qquad +\delta u_{i+1,f(x_i,a_i),l}, \forall (i, x_i, a_i, l) \in ((\{0\} \times \{x_0\} \times \mathcal{A}_0(x_0)) \\ & \qquad\qquad \cup (\mathcal{I} \setminus \{0\} \times \mathcal{X}_i \times \mathcal{A}_i(x_i))) \times \mathcal{L}', && (2.16) \end{aligned}$$

where u is the vector of decisions variables $u_{i,x_i,l}$'s and $\Delta_i^{\mathbb{E},l}$ is shorthand notation for $\Delta_i^{\mathbb{E},F^l}$. Similar to the PO linear program of Desai et al. (2012b) for optimal stopping, PLP can be equivalently formulated by eliminating all the $u_{i,\cdot,\cdot}$ decision variables for stages 1 through $I-1$ and replacing its constraints with ones that enforce each $u_{0,x_0,\cdot}$ choice term to be no smaller than the sum of the discounted penalized rewards along all possible sequences of feasible states and actions from the initial stage and state through the end of the horizon for each sample. In contrast to the model of these authors, in our context this version of PLP has an exponential number of constraints. Proposition 1 establishes that PLP is well posed.

Proposition 1. *PLP has a finite optimal objective function value and at least one bounded optimal solution.*

Using an argument analogous to the one that underlies the proof of Theorem 1 of Desai et al. (2012b), one could show that the optimal objective function value of model (2.14) converges almost surely to the one of (2.13) as the number of samples used to formulate it grows large.

We label as β^{PLP} the VFA weight vector obtained by solving PLP. We use it to estimate a dual bound as discussed in §2.3.2. We do so because the optimal PLP objective function value is a biased low estimator of this dual bound, as well as of the optimal objective function of model (2.13). In fact, letting $\tilde{\mathcal{L}}'$ be the index set of the random array of vectors of forward curves $\left(F^l, l \in \tilde{\mathcal{L}}'\right)$ with cardinality L' and $\mathbb{E}_{\tilde{\mathcal{L}}'}$ be expectation with respect to this random quantity, denoting as $\beta^{\mathrm{PLP}}\left(\tilde{\mathcal{L}}'\right)$ and $\beta\left(\tilde{\mathcal{L}}'\right)$, respectively, the random VFA weight vector obtained from solving PLP and vector of decision variables of model (2.14) both formulated using $\tilde{\mathcal{L}}'$, and observing that β^{PLP} is instead the vector associated with PLP expressed for $\mathcal{L}'$ taken as fixed, we have

$$\begin{aligned}
\mathbb{E}_{\tilde{\mathcal{L}}'}\left[\frac{1}{L'}\sum_{l\in\tilde{\mathcal{L}}'} U_0^{\beta^{\mathrm{PLP}}\left(\tilde{\mathcal{L}}'\right)}\left(x_0, F^l\right)|x_0, F_0\right] &= \mathbb{E}_{\tilde{\mathcal{L}}'}\left[\min_{\beta\left(\tilde{\mathcal{L}}'\right)}\frac{1}{L'}\sum_{l\in\tilde{\mathcal{L}}'} U_0^{\beta\left(\tilde{\mathcal{L}}'\right)}\left(x_0, F^l\right)|x_0, F_0\right]\\
&\leq \min_{\beta}\mathbb{E}_{\tilde{\mathcal{L}}'}\left[\frac{1}{L'}\sum_{l\in\tilde{\mathcal{L}}'} U_0^{\beta}\left(x_0, F^l\right)|x_0, F_0\right]\\
&= \min_{\beta}\mathbb{E}\left[U_0^{\beta}(x_0, F)|x_0, F_0\right]\\
&\leq \mathbb{E}\left[U_0^{\beta^{\mathrm{PLP}}}(x_0, F)|x_0, F_0\right].
\end{aligned}$$

Although the VFA weight vector β^{PLP} can be used to derive a greedy policy, its quality may be poor (Desai et al. 2012b). Indeed, consider the common assumption that the first basis function used to construct a VFA is a constant, that is, $\phi_{i,1}(\cdot) := 1$ for each $i \in \mathcal{I} \setminus \{0, I-1\}$, which implies that each term $\Delta_i^{\mathbb{E},l}\phi_{i+1,1}$ equals zero. Thus, the decision variables $\beta_{i,x_i,1}$'s have zero coefficients in PLP and in each stage the resulting VFA has an arbitrary intercept, which is undesirable from the perspective of obtaining a good greedy policy. We follow Desai et al. (2012b) to address this issue. Denote as u^{PLP} the u vector attained by solving PLP. Let $u^{\mathrm{PLP}}_{i,x_i,l}$ be the element of u^{PLP} corresponding to the triple $(i, x_i, l) \in \mathcal{I} \setminus \{0, I-1\} \times \mathcal{X}' \times \mathcal{L}'$. For each pair $(i, x_i) \in \mathcal{I} \setminus \{0, I-1\} \times \mathcal{X}'$

we solve the least squares regression model

$$\min_{\beta_{i,x_i}} \frac{1}{L'} \sum_{l \in \mathcal{L}'} \left[u^{\mathrm{PLP}}_{i,x_i,l} - \sum_{b \in \mathcal{B}_i} \beta_{i,x_i,b} \phi_{i,b} \left(F^l_i \right) \right]^2 .$$

We employ the collection of resulting optimal solutions to specify a VFA for the purposes of obtaining a greedy policy and bound pair in the manner explained in §2.3.1.

2.5 Solving the Pathwise Linear Program

This section presents our approach to solve PLP, which is both ill-conditioned and large scale in our application. We introduce our PCA preconditioning algorithm, the so pre-conditioned PLP (P2LP), and how to retrieve a PLP solution from a P2LP one in §2.5.1 and the BCD method to solve P2LP in §2.5.2.

2.5.1 Pre-conditioning Algorithm

Gurobi, a state-of-the-art commercial optimization software, either is impractical, that is, does not terminate in a reasonable amount of time, or outright fails when we embed its simplex or barrier methods within our BCD approach and attempt to solve the PLPs for the instances that we employ in the base configuration of our numerical study. This solver does so even when applied to PLPs for versions of these instances with their horizons shortened so that it can directly handle them on our high performance computer. In particular, the barrier algorithm just stops because of numerical issues, even if it is an interior point technique, a class of procedures that are well suited for large scale linear programs, especially ones that feature a block diagonal constraint matrix (Gurobi Optimization 2020, §28.5), which PLP has (the PO linear program of Desai et al. 2012b shares this aspect and they solve it using this approach without reporting any numerical issues).

Interior point methods are iterative algorithms that at each iteration solve a system of linear equations (Mikosch et al. 2006) with coefficients determined by a nonuniform scaling of the Grammatrix of an optimization model, that is, the product of the matrix of this model's constraints and itself (Nesterov and Nemirovskii 1994, Boyd et al. 2004). The condition number of a matrix is the ratio of its largest and smallest singular values. The smallest such value of a matrix that is almost rank deficient, that is, some of its columns are nearly linearly dependent, is close to zero, which leads to a large condition number. If the matrix that expresses the system of linear equations that the interior point procedures need to solve is of this type then the optimization model is said to be ill-conditioned and these techniques usually encounter numerical issues (Klotz and Newman 2013). The condition number of the Gram matrix of an optimization model is a proxy used for detecting these difficulties before attempting to solve it. For example, such instabilities are likely to arise if the Gurobi barrier algorithm, an interior point method, is applied to a model with such a condition number that exceeds 10^{12} (Gurobi Optimization 2021, Section 28.5), Desai et al. (2012b). The value of this number for one of our shortened horizon instances in which Gurobi fails to solve PLP is 1013, which is above this threshold (we obtain this value in Matlab because Gurobi does not report it when it does not terminate). This pitfall is the combined effect of the chosen basis functions and the number of available actions in our MDP. That is, it tends to disappear when we considerably simplify the basis functions that distinguish the base configuration of our numerical study, at the expense of policies and bounds of poor quality, or eliminate some decisions, which is undesirable. These findings suggest that preconditioning PLP to shrink its associated condition number is a potentially useful way to mitigate the observed difficulty of solving this model.

To ease the exposition we express the PLP constraints as

$$Du + G\beta \geq r \tag{2.17}$$

where D and G are the constraint coefficient matrices associated with the PLP decision variable vectors u and β, respectively, and r is the column vector $(r(x_i, s_i^l, a_i), (l, i, x_i, a_i) \in \mathcal{L}' \times \mathcal{I} \times \mathcal{X}_i \times \mathcal{A}(x_i))$. The PLP Gram matrix is $[D\ G]^\top [D\ G]$, where $\top$ denotes transposition.

We theoretically investigate the condition number of the PLP Gram matrix. Assumption (2) imposes structure on the G matrix, which include the dual penalty coefficients.

Assumption 2. *The G matrix has full column rank.*

This assumption is mild because it can be satisfied by deleting dependent columns from G and redefining this matrix using the remaining ones without changing the optimal PLP objective function value. Lemma (1) characterizes the column ranks of the D and PLP Gram matrices.

Lemma 1. *The D matrix has full column rank. If Assumption 2 holds then the PLP Gram matrix has full column rank.*

The properties stated in this result and Assumption (2) ensure that the singular values of the D, G, and PLP Gram matrices are strictly positive. Letting N_D, N_G, and N be the number of columns of the D, G, and $[D\ G]$ matrices, we denote the sequences of their respective singular values, each ordered from largest to smallest, as $(\kappa_n^D)_{n=1}^{N_D}$,$(\kappa_n^G)_{n=1}^{N_G}$, and $(\kappa_n)_{n=1}^{N}$.

Proposition 2 provides an upper bound on the condition number of the PLP Gram matrix in the trivial case in which there are no dual penalties, that is, there is no G

matrix and the PLP Gram matrix reduces to $D^\top D$, and a lower bound on the condition number of the PLP Gram matrix in the regular case, that is, the G matrix is present. We denote the condition number of a matrix as $\mathrm{cond}(\cdot)$. We let M_u be the maximum of all the numbers that correspond to how many constraints include each PLP variable $u_{i,x_i,l}$.

Proposition 2. *We have (i)* $\mathrm{cond}(D^\top D) \leq M_u N_D$ *and (ii) if Assumption (2) is satisfied then* $\sum_{n=1}^{N} \kappa_n^2 \geq 1$ *and*

$$\mathrm{cond}([D\ G]^\top [D\ G]) \geq \max\left\{ \frac{\sum_{n=1}^{N} \kappa_n^2 / N}{\left[\left(\Pi_{n=1}^{N_D} \kappa_n^D\right)\left(\Pi_{n=1}^{N_G} \kappa_n^G\right)\right]^{2/N}}, \mathrm{cond}(D^\top D) \right\}$$

Part (i) of this proposition implies that in the absence of dual penalties the condition number of the PLP Gram matrix is bounded above by a constant, so PLP cannot be ill-conditioned. In the presence of dual penalties, the right hand side of the lower bound on the condition number of the PLP Gram matrix in part (ii) of this result is the largest of two terms. The first one is a ratio that depends on the singular values of the PLP Gram matrix in its numerator and of the D and G matrices in its denominator. The second one is the condition number of the $D^\top D$ matrix, which a constant bounds from above by part (i). Suppose we pick basis functions that lead to dual penalties such that the columns of the G matrix are almost linearly dependent, but this matrix remains full rank, that is, Assumption 2 holds. Consequently, the smallest singular value of the G matrix is close to zero. In this case, the first term in the lower bound is large, because its numerator is bounded below by $1/N$ and its denominator is almost zero. Also assume, without loss of generality, that our choice of basis functions makes this term exceed the second one in this so it determines its value. Thus, the PLP Gram matrix is nearly rank deficient and PLP is ill-conditioned.

This discussion motivates us to obtain a version of PLP with improved conditioning by removing any close to linear dependence that may exist among the columns of the G matrix. We make these columns orthogonal by applying PCA to this matrix. Define W to be the square matrix with columns equal to the eigenvectors of the matrix $G^\top G$. Let $G^\perp$ be the product of G and W, that is, $G^\perp := GW$. It is easy to verify that the columns of $G^\perp$ are orthogonal. As discussed in Online Appendix B, we obtain W by exploiting the block diagonal structure of G for efficiency and to ensure that $G^\top$ retains this structure. P2LP results from replacing the PLP constraints (2.17) with

$$Du + G^\perp \beta \geq r \tag{2.18}$$

Proposition 3 states that solving P2LP is equivalent to solving PLP.

Proposition 3. *Every feasible PLP solution has a corresponding P2LP feasible solution with the same objective function value and vice versa.*

Given a P2LP feasible solution (β, u) we define β^W as $W\beta$. The pair (β^W, u) is a PLP feasible solution. Proposition 4 affirms that the condition number of the P2LP Gram matrix is bounded above by a constant.

Proposition 4. *If Assumption 2 holds then* $\mathrm{cond}([D\ G^{\perp}]^{\top}[D\ G^{\perp}]) \leq M_u N_D + 1$.

This result implies that P2LP is well-conditioned. P2LP is identical to PLP formulated using VFAs associated with a set of modified basis functions, each of which is a linear combination of the original basis functions with weights given by the elements of the W matrix (see Appendix A.2). Thus, our PCA preconditioning algorithm can be interpreted as a procedure that obtains new basis functions from a given pool of such functions that avoid the ill-conditioning that may otherwise affect PLP.

P2LP is well suited for interior point methods because it is well-conditioned and the $G^{\top}$ matrix is block diagonal. For example, Gurobi's barrier method readily solves P2LP for the short horizon instance mentioned above. In particular, the condition number of this P2LP Gram matrix is 10^9, which compares favorably with the Gurobi's ill-conditioning threshold reported there. More broadly, our preconditioning approach makes our BCD approach practical for the instances that we use in our numerical study.

2.5.2 BCD Optimization Algorithm

We devise a customized BCD method to deal with the large sizes of the instances that we consider in our numerical study, which make the direct solution of the resulting P2LPs an impossible task on our workstation because they exceed its available memory. Our algorithm solves a sequence of P2LPs in which the values of some of the decision variables that belong to the β vector are fixed and the values of all the others, including the ones that are part of the u vector, are optimally chosen (exact evaluation of the P2LP objective function requires an optimal selection of the values of all the elements of the u vector). In particular, it optimizes blocks of P elements of the VFA weight vector β that correspond to subsets of the index set $\bar{\mathcal{P}} := \mathcal{I} \setminus \{0, I-1\} \times \mathcal{X}'$, which we assume is sorted. We define the block corresponding to a subset $\mathcal{P}$ of $\bar{\mathcal{P}}$ as $\beta(\mathcal{P}) := (\beta_{i,x_i,b}, (i, x_i) \in \mathcal{P}, b \in \mathcal{B}_i)$.

Algorithm 1 outlines our BCD procedures. Its inputs are the initial VFA weight vector β^0, the index set $\bar{\mathcal{P}}$, the block size P, a block selection rule R, and the pair that includes the the stopping tolerance ϵ and integer valued iterationlag H. The initialization step sets the iteration counter h to zero and lets OBJ^0 be the optimal objective function value of P2LP fomulated by assigning to the variables in the β vector the values of the element of β^0. In each subsequent iteration h, Algorithm 1 (i) applies the block selection rule R to determine $\mathcal{P}$, (ii) solves the variant (2.19)-(2.21) of P2LP in which the values of the variables that belong to the u and the $\beta(\mathcal{P}^h)$ vectors are optimized, whereas the others are fixed to their iteration $h-1$ values, and (iii) makes (β^h, u^h) and OBJ^h equal to the optimal solution

```
Algorithm 1: BCD Algorithm
input          : Initial VFA weight vector $\beta^0$, index set $\bar{\mathcal{P}}$, block cardinality $P$,
                 block selection rule $R$, and stopping tolerance and iteration lag
                 pair $(\epsilon, H)$.
initialization: Set $h = 0$ and $\mathrm{OBJ}^0$ equal to the optimal objective function
                 value of P2LP solved with the values of the $\beta$ vector variables
                 fixed to the ones of the elements of $\beta_0$
do
    $h = h + 1$.
    (i) Apply block selection rule $R$ to obtain a subset $\mathcal{P}^h$ of $\bar{\mathcal{P}}$ with cardinality
     $P$.
    (ii) Let $(\beta^*, u^*)$ be the optimal solution to the LP
            $\min_{\beta,u} \frac{1}{L'} \sum_{l \in \mathcal{L}'} u_{0,x_0,l}$          (2.19)
            s.t. $Du + G^{\perp}\beta \geq r,$                                 (2.20)
                 $\beta(\bar{\mathcal{P}} \setminus \mathcal{P}^h) = \beta^{h-1}(\bar{\mathcal{P}} \setminus \mathcal{P}^h).$        (2.21)
     (iii) $\beta^h = \beta^*$, $u^h = u^*$, and $\mathrm{OBJ}^h = \frac{1}{L'} \sum_{l \in \mathcal{L}'} u^*_{0,x_0,l}$.
while $|\mathrm{OBJ}^h - \mathrm{OBJ}^{\max\{h-H,0\}}| > \epsilon$;
output         : Return $\beta^h$ and $u^h$.
```

and the optimal objective function value of this linear program, respectively. Termination occurs if and only if the quantity $\mathrm{OBJ}(\beta^h)$ and the analogous quantity $\mathrm{OBJ}^{\max\{h-H,0\}}$ differ by less than ϵ, in which case the vectors β^h and u are returned (choosing a value of the iteration lag H larger than one discourages premature BCD termination due to solutions that have similar evaluations of the objective function in consecutive iterations even if additional ones can lead to a substantial objective function value improvement).

The performance of BCD depends most notably on the block selection rule, of which the cyclic, greedy, and random ones are examples (Nesterov 2012, Beck and Tetruashvili 2013, Saha and Tewari 2013, Richt ´arik and Tak ´a ˇc 2014, Nutini et al. 2015). The cyclic rule forms the set $\mathcal{P}^h$ by picking consecutive elements from the index set $\mathcal{P}$ with the first element shifted by P in each iteration following a repeating pattern. To illustrate, suppose the set $\bar{\mathcal{P}}$ has four elements and the value of the parameter P is three. Then the sets $\mathcal{P}^1$ and $\mathcal{P}^2$ are $\{1, 2, 3\}$ and $\{1, 2, 4\}$, respectively. The greedy and random rules construct the set $\mathcal{P}^h$ by sequentially choosing the index with largest associated total reduced cost and uniformly sampling, respectively, from the subset of yet unselected elements of $\bar{\mathcal{P}}$ in the current iteration (the total reduced cost corresponding to an index (i, x_i) is the sum of the reduced costs of all the variable $\beta_{i,x_i,b}$'s in the optimal solution of the linear program obtained in the previous iteration).

Convergence results for BCD typically assume the optimization of a function that includes a smooth non-separable component and a separable non-smooth component

(Tseng 2001). In contrast, P2LP corresponds to the minimization of a non-separable and non-smooth function. We establish convergence of an idealized version of our BCD algorithm based on assumptions analogous to the ones that are used in the literature to prove that BCD reaches an optimal limit point (see, e.g., Bertsekas 2015, §6.5). The idealized BCD procedure is analogous to the one described in Algorithm 1 but it solves the linear program (2.19)-(2.21) augmented with the constraint

$$\beta^{\mathrm{L}}_{i,x_i,b} \leq \beta^{*}_{i,x_i,b} \leq \beta^{\mathrm{H}}_{i,x_i,b} \tag{2.22}$$

for each triple $(i, x_i, b) \in \mathcal{I} \setminus \{0, I-1\} \times \mathcal{X}' \times \mathcal{B}_i$, where $\beta^{\mathsf{L}}_{i,x_i,b}$ and $\beta^{\mathsf{H}}_{i,x_i,b}$ are bounded constants that satisfy $\beta^{L}_{i,x_i,b} < \beta^{H}_{i,x_i,b}$ (the superscripted L and H abbreviate low and high, respectively), and has the stopping tolerance set equal to $-\epsilon$, that is, it never stops. Proposition 5 characterizes the behavior of this method.

Proposition 5. *Suppose the idealized BCD algorithm uses one of the cyclic, greedy, or random block selection rules. The solution sequence that it generates admits a limit. If this limit strictly satisfies the constraints* (2.22) *and its u vector component is a non-degenerate solution of P2LP then this sequence converges to the set of P2LP optimal solutions.*

In our numerical study we use the BCD technique that corresponds to Algorithm 2 rather than its idealized BCD version that this proposition characterizes. We find that in the base configuration of this investigation it always terminates and leads to PLP solutions of high quality, that is, they lead to policies for our MDP with small estimated optimality gaps. Proposition 5 provides some theoretical support for this observed behavior.

2.6 Numerical Study

This section reports the results of our numerical study. We describe the instances that form the basis of this investigation in §2.6.1, which is in part based on (Nadarajah and Secomandi 2020, §4.2), present the setup of our analysis in §2.6.2, discuss bounds and run times in §2.6.3, and examine operating policies in §2.6.4.

2.6.1 Instances

We consider an existing ethanol production asset with two years left in its lifetime. Decisions are made on a monthly basis. Each stage represents the beginning of each month in the selected horizon. Thus, we set the number of stages (I) equal to twenty four.

Our study is based on twelve instances each distinguished by a starting date corresponding to one of the months in year 2011. We refer to them using the first three letters of these months.

The monthly risk free discount factor (σ) depends on the one year United States Treasury rate observed on an instance initial date. The values of these rates are 0.29%, 0.27%, 0.25%, 0.27%, 0.22%, 0.18%, 0.20%, 0.22%, 0.10%, 0.12%, 0.13%, and 0.12% for the Jan through Dec instances, respectively.

The initial vector of forward curves (F_0), with the exclusion of the vector of spot prices (s_0), is based on the closing prices of futures for corn, ethanol, and natural gas traded on the Chicago Mercantile Exchange (CME) observed on the first trading day of the month that corresponds to each instance, whereas the spot price vector hinges on the settled futures prices for each of these months. CME trades ethanol and natural gas futures with maturities for all the months that the horizons of our instances comprise. In contrast, it deals corn futures with maturities only in March, May, July, September, and December of each year in the considered horizon. We compute the prices of corn futures for any missing maturity by interpolation as described in Guthrie (2009, §12.2.2) when this approach is applicable and extrapolation otherwise, that is, employing the observed futures price for the first or the last traded maturity as a proxy of the prices of futures with earlier or later hypothetical maturities.

We calibrate the futures price model (2.1) using CME corn, ethanol, and natural gas futures daily closing prices observed from January 2008 through December 2011, supplemented by the described interpolation and extrapolation approach for corn. As in Secomandi et al. (2015a), for each calendar month we determine a sample variance-covariance matrix of the daily log futures price returns for the next twenty three maturities for the three considered commodities and compute the principal components of this matrix to estimate the loading coefficients $\bar{\sigma}'_{c,i,j,k}s$ (the index i maps to a calendar month). We set the number of factors (K) equal to eight because it is the smallest number that explains more than 95% of the total observed variance in our data sets. Our instances share the same calibrated futures price model.

We adapt the values of the operational parameters, which are common to our instances, from Guthrie (2009, §12.2.2). Table 4.4 reports them.

Table 2.1: Values of the common instance parameters.

Parameter	Value	Parameter	Value ($ MM)
I	24 months	$\mathsf{I_M}$	0.5
γ_{C}	0.36 bushel/gallon	$\mathsf{I_R}$	2.5
γ_{N}	0.035 MMBtu/gallon	$\mathsf{C_P}$	2.25
Q	8.33 million gallons	$\mathsf{C_S}$	0.5208
S	0	$\mathsf{C_M}$	0.02917

2.6.2 Setup

The software implementation of our methods relies on C++ using the GCC 4.8.5 (Red Hat 4.8.5-11) compiler and CentOS Linux 7 operating system. We use Gurobi 7.5 to solve linear programs. We apply the dlib C++ machine learning package and LAPACKE to perform PCAs and regressions, respectively. We execute our algorithms on a server with 128 GB of RAM and 12 Intel(R) Core(TM) i7-5930K processors, of which we employ at most six when running Gurobi to reduce its memory requirement.

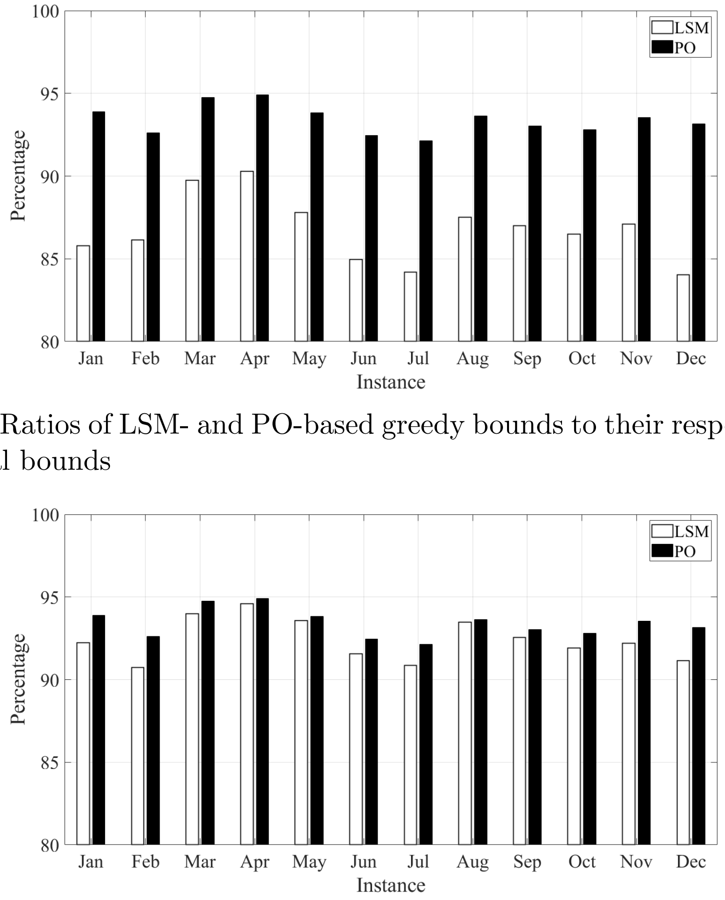
We specify the basis functions as polynomials of spot and futures prices as commonly done in the literature (Longstaff and Schwartz 2001, Boogert and De Jong 2008, Nadarajah et al. 2017). For each stage $i \in \mathcal{I}$ these functions are (i) one; (ii) $\{F^c_{i,j}, j \in \mathcal{I}_i, c \in \mathcal{C}\}$; (iii) $\{(F^c_{i,j})^2, j \in \mathcal{I}_i, c \in \mathcal{C}\}$; (iv) $\{F^c_{i,j}F^{c'}_{i,j}, j \in \mathcal{I}_i, c, c' \in \mathcal{C}, c \neq c'\}$; and (v) $\{F^c_{i,j}F^c_{i,j+1}, j \in \mathcal{I}_i \setminus \{I-1\}, c \in \mathcal{C}\}$. Their conditional expectations can be efficiently evaluated (see Nadarajah et al. 2017 for the expressions pertaining to cases (ii)-(v)), that is, Assumption 3 is satisfied. We use the BCD greedy block selection rule, setting the values of the block size (P), iteration lag for termination (H), and stopping tolerance (ϵ) to fourteen, one, and 0.01, respectively, starting with a zero β vector. The number of Monte Carlo samples that we employ to estimate bounds or obtain VFAs (L or L') is five hundred thousand or seventy thousand. The latter value is the largest value of the number of samples that allows us to apply BCD as described to solve the resulting P2LPs, which have three and ten million variables and constraints, without facing memory issues. We refer to this set of choices as the base configuration. We obtain alternative configurations by using one of ten, thirty, fifty thousand samples to obtain VFAs, changing the BCD block selection rule to be either cyclic or random (in both cases we set to four the value of the iteration lag for termination), or considering only linear basis functions, that is, types (i) and (ii). With ten or thirty thousand samples or linear basis functions we can and do directly apply Gurobi to solve the corresponding P2LPs (equivalently, we apply BCD with a block size equal to the cardinality of the index set $\mathcal{P}$).

2.6.3 Bounds and Run Times

Table 2.2 reports the estimates of the LSM- and PO-based bounds, along with their respective standard errors and the ratios of the estimated PO- and LSM-based bounds, for the base configuration. The precisions of the estimated bounds have the same order of magnitude. In particular, the standard errors are at most 0.43% of their respective estimates. PO always dominates LSM in terms of the quality of the estimated bounds, but the improvement for the greedy bounds is considerably smaller than the one for the dual bounds. Specifically, the ratios vary between 100.27% and 102.21% and average to 101.09% for the greedy bounds and they range from 94.62% to 96.81% and have an average of 95.59% for the dual bounds.

Table 2.2: LSM- and PO-based bound estimates, with standard errors reported in parenthesis, and their percentage ratios (the latter estimates divided by the former ones).

	Greedy Bound			Dual Bound		
Month	LSM	PO	Ratio (%)	LSM	PO	Ratio (%)
Jan	19.01 (0.07)	19.35 (0.06)	101.79	21.71 (0.01)	20.61 (0.05)	94.93
Feb	18.98 (0.07)	19.38 (0.07)	102.09	21.61 (0.01)	20.92 (0.04)	96.81
Mar	23.54 (0.08)	23.74 (0.08)	100.84	25.95 (0.01)	25.05 (0.07)	96.52
Apr	25.13 (0.09)	25.22 (0.09)	100.34	27.57 (0.01)	26.57 (0.07)	96.38
May	21.19 (0.08)	21.25 (0.08)	100.27	23.78 (0.01)	22.64 (0.06)	95.23
Jun	17.65 (0.08)	17.82 (0.08)	101.00	20.30 (0.01)	19.27 (0.07)	94.94
Jul	14.85 (0.07)	15.06 (0.07)	101.40	17.20 (0.01)	16.35 (0.05)	95.05
Aug	21.02 (0.08)	21.06 (0.08)	100.18	23.65 (0.01)	22.49 (0.07)	95.09
Sep	21.88 (0.09)	22.00 (0.08)	100.54	24.73 (0.01)	23.65 (0.07)	95.64
Oct	19.36 (0.07)	19.55 (0.07)	100.95	21.98 (0.01)	21.06 (0.06)	95.85
Nov	17.98 (0.07)	18.24 (0.07)	101.47	20.30 (0.01)	19.50 (0.05)	96.07
Dec	13.84 (0.06)	14.14 (0.06)	102.21	16.05 (0.01)	15.18 (0.05)	94.62



(a) Ratios of LSM- and PO-based greedy bounds to their respective dual bounds

(b) Ratios of LSM- and PO-based greedy bounds to the PO-based dual bounds

Figure 2.2: Comparison of the estimated LSM and PO gaps based on the respective PO- and LSM-based dual bounds.

Panel (a) of Figure 2.2 displays the percentage suboptimality of the LSM- and PO-based greedy policies assessed using their respective dual bound estimates. The range and average of the resulting optimality gaps are $11 - 13\%$ and 12% for LSM and $5 - 8\%$ and 7% for PO. Panel (b) of Figure 2.2 shows the stated suboptimality using the PO-based dual bound estimates as yardstick. In this case, the range and average of the optimality gaps of the LSM-based greedy policies reduce to $5 - 9\%$ and 8%, respectively. Thus, both LSM and PO lead to near optimal operating policies, even if PO marginally outperforms LSM in this respect. The estimated PO-based dual bounds play an important role in establishing this finding. Further, on average the estimates of these dual bounds are 8.95% smaller than the ones obtained by using zero dual penalties. Thus, estimating good dual bounds for the considered instances is not straightforward.

Table (4.8) reports the average CPU times required to execute LSM and PO. LSM takes fifteen minutes in total, with about the same amount of time needed to run the regression and estimate each of the two bounds. In contrast, PO necessitates a total of three hundred and twenty four minutes (roughly five and a half hours), of which BCD uses most of them (about three hundred) and PCA, regression, and the greedy and dual bound estimations employ three, eleven, four, and five, respectively. On average BCD converges in roughly fourteen iterations, thus demanding about twenty minutes per iteration. Although PO entails considerably longer CPU times than LSM, it is practical for use in our application.

Table 2.3: Average CPU times (minutes).

Method	PCA	BCD	Regression	Greedy Bound	Dual Bound	Total
LSM			6	4	5	15
PO	3	301	11	4	5	324

Table (2.4) reports the bound estimates and resulting optimality gaps for the representative Jan instance considering both the alternative configurations that correspond to varying the number of samples used to obtain the VFAs and the base configuration. The choice of this parameter value affects both the LSM performance and the PO-based greedy bound in an immaterial fashion. In contrast, the PO-based dual bound critically depends on it: At least fifty thousand samples are required to achieve an optimality gap below 10%. Employing more samples to determine the VFAs increases the CPU times, which, as Table (2.5) shows, rise by about half a time and almost three times for LSM and PO, respectively, when the number of samples changes from ten thousand to seventy thousand for the chosen illustrative instance.

The estimated PO-based bounds in the alternative configurations that correspond to changing the BCD block selection rule are minimally to moderately different compared to the ones obtained in the base configuration. In particular, with respect to this case on average the greedy bound estimates improve by 0.5% irrespective of which of the other two rules is chosen and the dual bound estimates deteriorate by 1.01% and 1.89% when using

Table 2.4: Influence of samples on LSM and PO bound estimates.

	LSM			PO		
Sample	Greedy Bound	Dual Bound	Gap (%)	Greedy Bound	Dual Bound	Gap (%)
10000	19.06	21.64	14	19.35	24.68	28
30000	19.04	21.64	14	19.35	21.33	10
50000	19.04	21.63	14	19.32	20.82	8
70000	19.05	21.71	14	19.35	20.61	7

the cyclic and random rules, respectively. The average BCD run times corresponding to these rules increase substantially, specifically they triplicate, compared to the baseline.

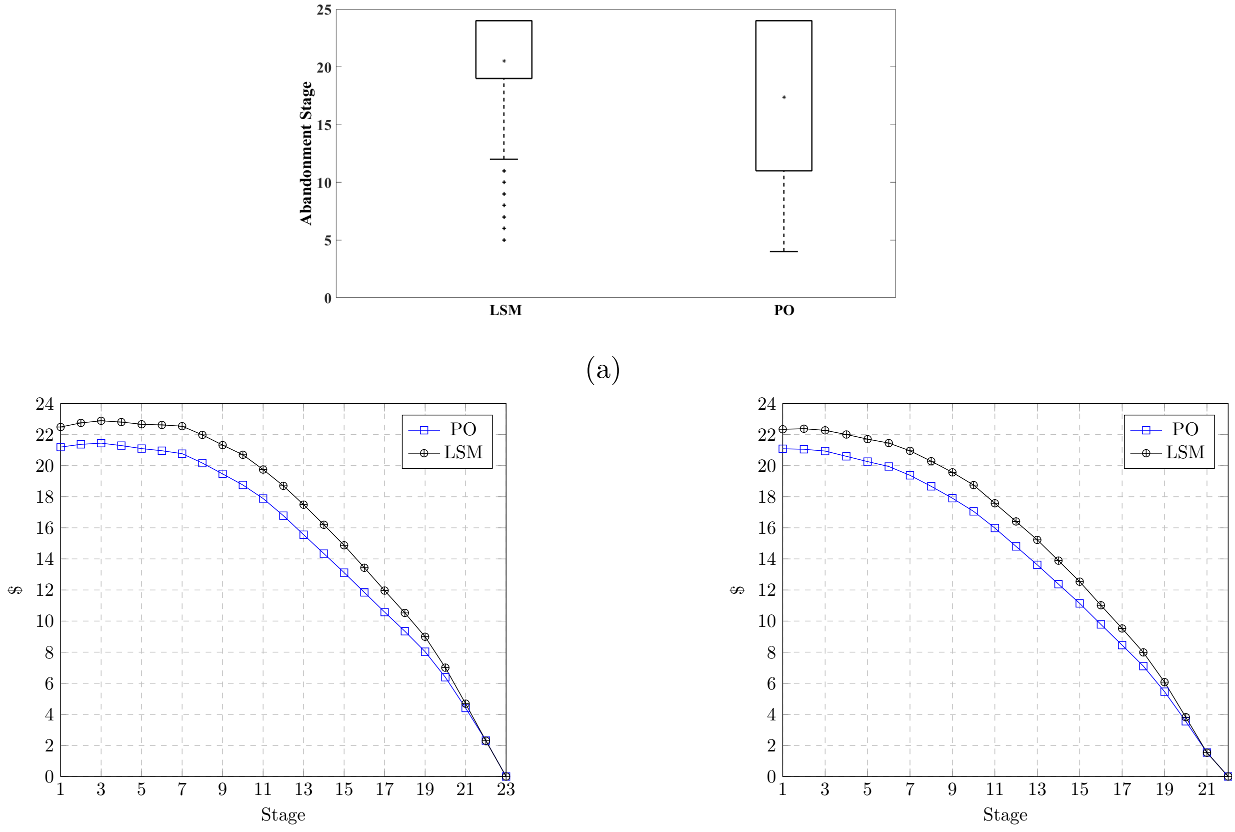
Table 2.5: CPU times (minutes) corresponding to varying the number of samples used to estimate the VFAs for the Jan instance.

	Number of Samples			
Method	10,000	30,000	50,000	70,000
LSM	10	12	13	16
PO	103	130	227	289

Relative to the base configuration, the absence of nonlinear basis functions has a considerable, yet distinct effect on the performance of LSM and PO. Specifically, in the alternative configuration associated with linear basis functions the estimated greedy and dual bounds, respectively, worsen by 3% and 15% for LSM and 15% and 1% for PO. This finding reflects the differential natures of LSM and PO, which are rooted on obtaining greedy and dual bounds, respectively. It suggests that these methods can be used in a complementary fashion when employing PO based on simple basis functions: LSM yields a good greedy bound and operating policy pair (with nonlinear basis functions) and PO generates a fine dual bound in this case. The average total run times when only linear basis functions are considered are two minutes and about three hours for LSM and PO, respectively, which are close to one tenth and half of what they are in the base case.

2.6.4 Operating Policy

We investigate the behaviors of the LSM- and PO-based greedy policies for the representative Jan instance in the base configuration. Panel (a) of Figure (2.3) displays the box plots of the stage in which these policies abandon the plant. The PO-based policy tends to do so sooner than the LSM-based one. To gain some intuition on this difference, panels (b) and (c) Figure (2.3) depict the average VFAs in each stage for both LSM and PO (the averaging is with respect to the sampled forward curves used to estimate the bounds). LSM obtains average VFAs that always exceed the ones that PO determines. This finding suggests that the merchant has more incentives to defer abandonment when using the greedy policy corresponding to the LSM-based VFAs rather than the ones associated with PO. Figure (2.4) presents the box plots of the number of stages in which the two greedy policies take the other available decisions. The LSM-based policy has a tendency to employ all these choices more often than the PO-based one, as suggested by the observation that it is prone to abandon the plant later compared to the other policy. This discussion indicates that the two considered policies differ in how they manage the plant even if their values are similar. We exemplify how the actions prescribed by the greedy decision rules relate to the shapes of the forward curves. We consider a sample for the representative Jan instance in which the two greedy policies make the same choices. Panel (a) of Figure (2.5) shows these decisions and the corresponding operational mode transitions. Production is suspended and occurs for the first and next five stages, respectively. At this point the plant is mothballed and remains so for eight more stages. It is then reactivated. Production happens in the subsequent stage, after which abandonment takes place. Panel (b) of Figure (2.5) illustrates the discounted total forward conversion spread curve for the residual maturities in stages zero, five, ten, nineteen, twenty, and twenty one, that is, the vector with elements $\delta^{j-i}\big[\big(F^{\mathrm{E}}_{i,j} - \gamma_{\mathrm{C}}F^{\mathrm{C}}_{i,j} - \gamma_{\mathrm{N}}F^{\mathrm{N}}_{i,j}\big)Q - \mathsf{C}_{\mathsf{P}}\big]$ for each maturity j that equals or exceeds each considered stage i. The negative and somewhat at discounted total forward conversion spreads in the initial stage bode well with the



(a)

(b)

(c)

Figure 2.3: (a) Box plots of the stage when the greedy policies abandon the plant (the 75-th percentiles and the maxima coincide) and (b)-(c) average VFAs corresponding to the (b) operational and (c) mothballed modes for the Jan instance.

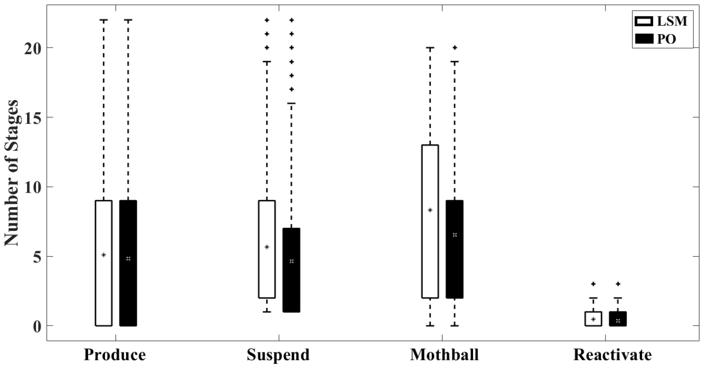


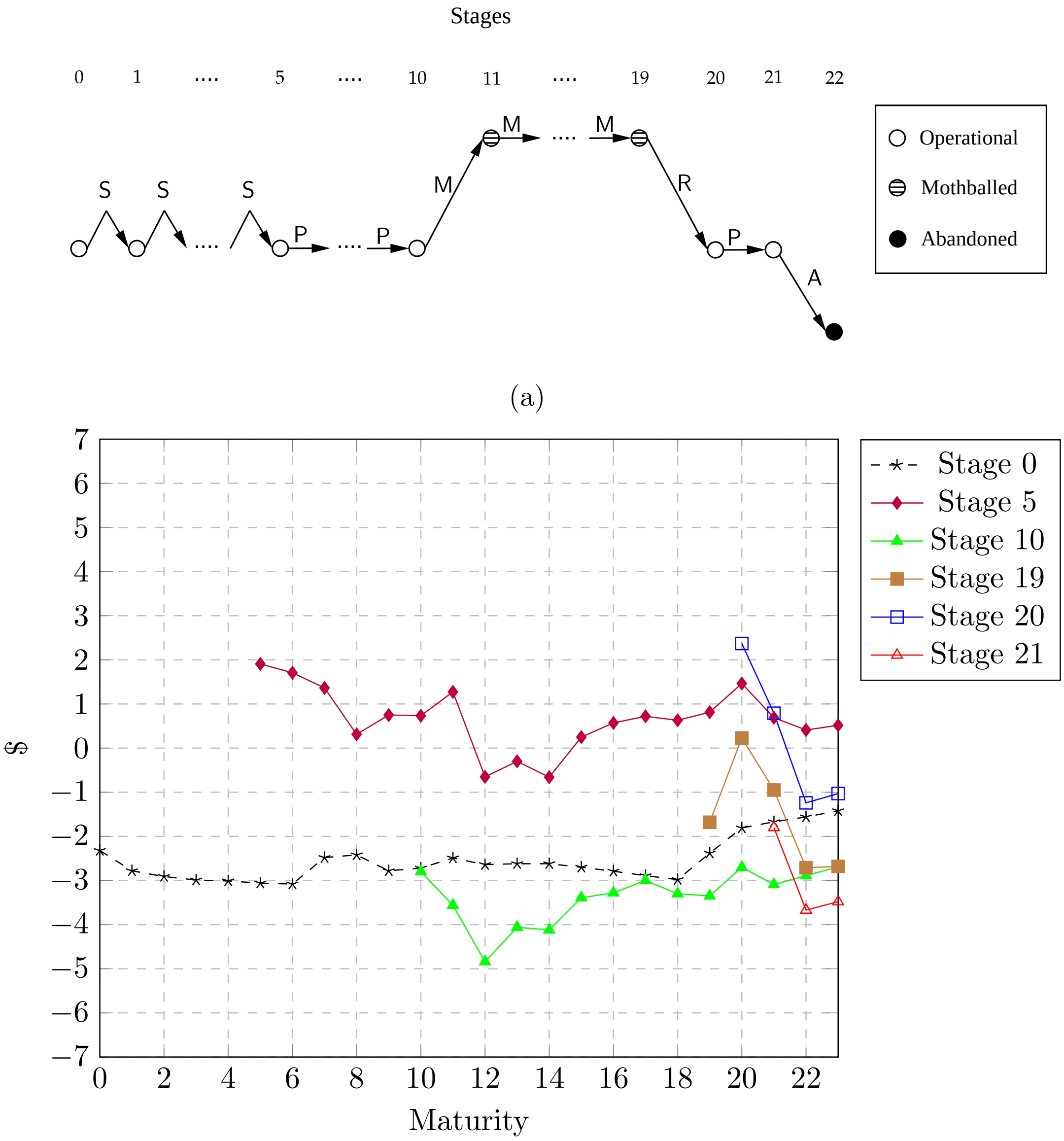
Figure 2.4: Box plots of the number of stages in which the greedy policies take the available decisions excluding abandonment for the Jan instance (the minima and the 25-th percentile coincide except for suspension with respect to the LSM-based greedy policy).

decision to suspend production. The positive total spot conversion spread in stage five calls for producing. In stage ten the discounted total forward conversion spreads are all negative and even smaller for the next two maturities compared to the current one, a situation that suggests mothballing the plant is beneficial. In stage nineteen this spread for the next maturity is positive, which makes reactivation appealing. Producing in stage twenty is analogous to taking this action in stage five. Abandonment in stage twenty one coincides with negative residual discounted total forward conversion spreads. These behaviors are reasonable.

To assess the usefulness of managing the plant using a dynamic approach, we consider the optimal static policy, which solves the version of our MDP in which uncertainty is suppressed and the spot prices in each stage are replaced by their corresponding forward prices available in the initial stage; that is, $F^c_{0,i}$ is used in lieu of s^c_i for each stage $i \neq 0$ and commodity c ($F^c_{0,0}$ and s^c_0 match). This policy exhibits dismal performance: It has essentially zero value in every instance because it abandons the plant early on. Thus, using a good dynamic policy is critical in our instances.

2.7 Conclusions

We investigate a compound switching and timing option model of merchant energy production that gives rise to an intractable MDP. We compare on realistic instances the performance of LSM, a state-of-the-art RL approach for option models, and PO, an alternative methodology so far developed for optimal stopping models, extending the literature that benchmarks these methods. Whereas using LSM involves following standard steps, applying PO to our considered context demands algorithmic development. We devise novel PCA and BCD methods to deal with the illconditioned and large scale nature



(a)

(b)

Figure 2.5: (a) Greedy actions and operational mode transitions (the abandon decisions in stages twenty two and twenty three are ignored for simplicity) and (b) total forward conversion spread curve in a subset of stages of the Jan instance for the considered sample.

of the resulting PLP, which is out of direct reach even for a modern commercial solver such as Gurobi. Both LSM and PO lead to near optimal policies, but PO yields substantially stronger dual bounds compared to LSM, at the expense of larger computational requirements, in terms of both run times and memory use. PO provides good quality dual bounds also when it is specified in a simple manner, in which case it complements LSM. Our findings on the relative size of the LSM-and PO-based bounds differ from known ones. Our research may be relevant in other commodity merchant operations settings and stimulates further methodological contributions in the realm of PO.

Chapter 3

Constraint Generation for Pathwise Reinforcement Learning

3.1 Introduction

In this chapter, we provide an alternative method, a constraint generation approach, to solve (2.15)-(2.16). The idea of the proposed method is based on an observation that the number of tight constraints at optimality is substantially less than the total number of constraints in (2.15)-(2.16). This is true because the tight constraints on every sample path $l \in \mathcal{L}$ correspond to the optimal action sequence on that sample path. In other words, the number of tight constraints on each l is at most I, if there is no degeneration in (2.15)-(2.16). The total number of tight constraints should be less than or equal to IL. Based on this observation, we can directly obtain the optimal solution if we solve a linear system with those IL constraints, which is significantly smaller than the original PLP. However, the idea is impractical because picking the correct constraints requires the optimal dual solution. So we propose a constraint generation approach that can iteratively generate a subset of constraints from PLP. By solving a new LP with this subset of constraints, we obtain a near optimal solution that is sufficiently good for generating competing bounds in our application.

Specifically, the proposed method iterates between a subproblem and a master problem. The master problem has the same objective function as PLP but with a few selected constraints. The subproblem is the PLP dual with a fixed current VFA value, which decouples according to samples by construction. In each iteration, we solve the decoupled subproblem to select potential constraints and then insert them into the master problem. We use the master problem to update the VFA. The objective function values of the sub and master problems provide upper and lower bounds on the optimal objective function value, respectively. The former value decreases during the iteration because the solution is approaching optimality; the latter increases as there are more and more constraints in the master problem. So the algorithm terminates when the gap between these two problems is sufficiently small.

We apply the constraint generation approach to merchant energy production. Numerical results show that our approach can generate a near optimal solution with an order of magnitude fewer constraints (10%) than directly solving PLP. The qualities of the bounds and the feasible operating policy are comparable to the results in Chapter 2. The constraint generation approach uses less memory (70%) and longer CPU times (100%), compared to the BCD approach.

3.2 Literature Review

Constraint generation, a.k.a. Benders decomposition, was initially proposed to solve mixed integer linear programming (MILP; Benders 1962). Since then, many extensions have been made to apply the algorithm to other problems, such as linear, nonlinear, integer, stochastic, multi stage, and other optimization problems (Geoffrion 1972, Hooker and Ottosson 2003, Adulyasak et al. 2015, Cordeau et al. 2001, Cai et al. 2001, Li et al. 2021, Côté et al. 2014). Typical applications include planning and scheduling (Canto 2008, Hooker 2007, Fischetti et al. 2017), health care (Luong 2015), transportation and telecommunications (Costa 2005, Maheo et al. 2019, Crainic et al. 2021), and energy and resource management (Cai et al. 2001, Zhang and Ponnambalam 2006). Our work mainly contributes to this stream of literature.

Many enhancement strategies that can accelerate the algorithm have also been developed since it's well known that classical Benders decomposition converges slowly (Rahmaniani et al. 2017). Those strategies can be categorized into four classes: decomposition strategy, cut generation, solution procedure, and solution generation (Rahmaniani et al. 2017).

Our paper employs standard solution generation and decomposition strategies but customized (sub) solution procedure and cut generation strategies. Our subproblem is very easy to solve because it decouples according to samples. Besides, we only need to solve deterministic dynamic programming via the standard backward induction on each sample path. In literature, solving the subproblem is not always a simple task because it is a large scale LP. The simplex method is the most commonly used algorithm for the subproblem (Rahmaniani et al. 2017). Specialized methods are much more powerful than the commercial solver when there are special structures for the subproblems. Cordeau et al. (2001) and Mercier et al. (2005) use the column generation approach to solve their subproblems. Fischetti et al. (2016) and Mahey et al. (2001) reduce the subproblem as a knapsack and a network flow problem with closed form solutions, respectively.

The way we generate constraints (cuts) is also different from the literature. Our subproblem is always feasible, so all generated constraints in our context are optimality constraints. Also, our subproblem rarely becomes degenerated, so we do not need to choose the so-called Pareto optimal cut among multiple candidates (Magnanti et al. 1978, Magnanti and Wong 1981), which requires solving a secondary problem for the subproblem. Although many techniques can tackle the secondary problem efficiently, generating a Pareto optimal cut may not yield a net computational advantage (Mercier and Soumis 2007). Besides, the constraints we use are the tight constraints in the subproblem, which are the most dominated constraints under the current solution. It is different from the commonly used "most violated" constraints in literature (Rahmaniani et al. 2017).

3.3 Constraint Generation

Our approach follows the general framework of the constraint generation approach in the literature (Bertsimas and Tsitsiklis 1997). We construct sub and master problems based on PLP, respectively. The subproblem is the dual of PLP with a fixed β variable values because the β variables are the complicating variables that couples the constraints across all samples. Once we fix its value, the subproblem decouples by samples. The master problem shares the objective function from PLP. However, it contains partial constraints of PLP. In each iteration, we solve the master problem to update the current solution and the subproblem to generate the constraints based on the obtained solution. We then insert the new constraints into the master problem and repeat these steps.

Supposed we use the quadruplet (l, i, x_i, a_i) to refer to the constraint of taking action a_i at x_i on sample path l. The set of constraints contained in the master problem in the k-th iteration is defined as $\mathcal{C}^k$. We have $\mathcal{C}^k \subset \mathcal{L} \times \mathcal{I} \times \mathcal{X}_i \times \mathcal{A}_i(x_i), \forall k$. So the master problem is simply a LP that contains partial constraints in (2.15)-(2.16):

$$\min_{U,\beta} \frac{1}{L} \sum_{l\in\mathcal{L}} U_0^{l,\beta}(x_0) \tag{3.1}$$

$$s.t.\ U_i^l(x_i) \geq r(x_i, F_i^l, a_i) - \sum_{b\in\mathcal{B}_i} \beta_{i+1,f(x_i,a_i),b} \Delta_i^{\mathbb{E},l} \phi_{i+1,b} + \delta U_{i+1}^{l,\beta}(f(x_i, a_i))$$

$$\forall (i, x_i, a_i, l) \in \mathcal{C}^k \tag{3.2}$$

$$U_0^{l,\beta}(x_0) \geq r(x_0, F_0, \mathsf{A}), \forall l \in \mathcal{L}. \tag{3.3}$$

The last constraint guarantees that the master problem is well-defined in every iteration. By following the same procedure of Proposition 1 in Chapter 2, we immediately obtain the following proposition.

Proposition 6. *(3.1)-(3.2) has a finite optimal objective function value and at least one bounded optimal solution.*

To define the subproblem, we first consider the following LP:

$$\min_U \frac{1}{L}\sum_{l\in\mathcal{L}} U_0^l(x_0) \tag{3.4}$$

$$s.t.\ U_i^l(x_i) \geq r(x_i, F_i^l, a_i) - \sum_{b\in\mathcal{B}_i} \hat{\beta}_{i+1,f(x_i,a_i),b}\Delta_i^{\mathbb{E},l}\phi_{i+1,b} + \delta U_{i+1}^l(f(x_i,a_i))$$

$$\forall(i, x_i, a_i, l) \in \mathcal{I}\setminus\{I-1\}\times\mathcal{X}_i\times\mathcal{A}(x_i)\times\mathcal{L} \tag{3.5}$$

$$U_{I-1}^{l,\beta}(x_{I-1}) \geq r(x_{I-1}, F_{I-1}^l, a_{I-1}), \forall(x_{I-1}, a_{I-1}, l) \in \mathcal{X}_{I-1}\times\mathcal{A}(x_{I-1})\times\mathcal{L} \tag{3.6}$$

where $\hat{\beta}^k_{i+1,f(x_i,a_i),b}$ is the solution from the master problem in k-th iteration. This LP is generated by fixing $\beta^k_{i+1,f(x_i,a_i),b} = \hat{\beta}^k_{i+1,f(x_i,a_i),b}$ in (2.15)-(2.16). Denote as $\hat{r}(x_i, F_i^l, a_i)$ the penalized reward $r(x_i, F_i^l, a_i) - \sum_{b\in\mathcal{B}_i}\hat{\beta}_{i+1,f(x_i,a_i),b}\Delta_i^{\mathbb{E},l}\phi_{i+1,b}$. We define the dual of (3.4)-(3.6) as our subproblem:

$$\max_\mu \sum_{l\in\mathcal{L}}\sum_{i\in\mathcal{I}}\sum_{x_i\in\mathcal{X}_i}\sum_{a_i\in\mathcal{A}(x_i)} \mu^l_{x_i,a_i}\hat{r}(x_i, s_i^l, a_i) \tag{3.7}$$

$$s.t. \sum_{a_0\in\mathcal{A}(x_0)} \mu^l_{x_0,a_0} = \frac{1}{L}, \forall l\in\mathcal{L} \tag{3.8}$$

$$\sum_{x_i\in\mathcal{X}_i}\sum_{a_i\in\mathcal{A}(x_i)} \mathbb{1}_{\{f(x_i,a_i)=x_{i+1}\}}\mu^l_{x_i,a_i} = \sum_{a_{i+1}\in\mathcal{A}(x_{i+1})}\mu^l_{x_{i+1},a_{i+1}},$$

$$\forall(i+1, x_{i+1}, l) \in \mathcal{I}\setminus\{0\}\times\mathcal{X}_{i+1}\times\mathcal{L} \tag{3.9}$$

$$\mu^l_{x_i,a_i} \geq 0, \forall(l, x_i, a_i) \in \mathcal{L}\times\mathcal{X}_i\times\mathcal{A}(x_i) \tag{3.10}$$

where $\mu^l_{x_i,a_i}$, $\forall(l, i, x_i, a_i) \in \mathcal{L}\times\mathcal{I}\times\mathcal{X}_i\times\mathcal{A}(x_i)$ are the dual variables corresponding corresponds to the primal constraints. The objective function is the total rewards ensued by all $\mu^l_{x_i,a_i}, \forall(l, i, x_i, a_i) \in \mathcal{L}\times\mathcal{I}\times\mathcal{X}_i\times\mathcal{A}_i$.

This subproblem has two kinds of constraints: the balance constraints (3.8)-(3.9) and the bound constraints (3.10). Balance constraints (3.8)-(3.9) indicate that the sum of the dual variables reaching a state pair $(i+1, x_{i+1}) \in \mathcal{I}\setminus\{0\}\times\mathcal{X}_{i+1}$ equals the sum of dual variables leaving that state pair on every sample path. For $(0, x_0)$, this sum is equal to $1/L$. Intuitively, balance constraints are analogous to the flow constraints in a network problem. They restrict the incoming flow equals the outgoing flow for each node. The bound constraints require that every feasible dual solution should be nonnegative. Since s, the subproblem (3.7)-(3.10) is well-defined, i.e., it can never become infeasible or unbounded.

Proposition 7. *(3.7)-(3.10) has a finite optimal objective function value and bounded optimal solution.*

In each iteration, we first solve the subproblem. The complementary slackness conditions suggest that if $\mu^l_{x_i,a_i} > 0$, the corresponding constraint is tight; otherwise, the constraint is redundant at the optimality. Then we select every constraint in P2LP with its dual variable value strictly positive and insert those constraints into the master problem. The complete algorithm can be summarized below:

```
Algorithm 2: Constraint Sampling and Generation Algorithm
input         : Initial vector $\beta^0$, OBJ($\beta^0$), $\mathcal{C}^0$, and stopping tolerance $\epsilon > 0$.
initialization: Set $k = 0$, $\beta^0 = 0$, $\mathcal{C}^0 = \emptyset$
do
    $k = k + 1$.
    (i) Compute $\hat{r}^k(x_i, s^l_i, a_i)$ for each $(l, i, x_i, s_i, a_i) \in \mathcal{L} \times \mathcal{I} \times \mathcal{X}_i \times \mathcal{A}_i(x_i)$ with $\beta^k$
    (ii) Solve the subproblem and obtain $\mu^k$

      $\max_{\mu} \sum_{l\in\mathcal{L}} \sum_{i\in\mathcal{I}} \sum_{x_i\in\mathcal{X}_i} \sum_{a_i\in\mathcal{A}(x_i)} \mu^l_{x_i,a_i} \hat{r}^k(x_i, s^l_i, a_i)$
      $s.t. \sum_{a_0\in\mathcal{A}(x_0)} \mu^l_{x_0,a_0} = \frac{1}{L}, \forall l \in \mathcal{L}$
          $\sum_{x_i\in\mathcal{X}_i} \sum_{a_i\in\mathcal{A}(x_i)} \mathbb{1}_{\{f(x_i,a_i)=x_{i+1}\}} \mu^l_{x_i,a_i} = \sum_{a_{i+1}\in\mathcal{A}(x_{i+1})} \mu^l_{x_{i+1},a_{i+1}},$
                                    $\forall (i+1, x_{i+1}, l) \in \mathcal{I} \setminus \{0\} \times \mathcal{X}_{i+1} \times \mathcal{L}$
          $\mu^l_{x_i,a_i} \geq 0, \forall (l, x_i, a_i) \in \mathcal{L} \times \mathcal{X}_i \times \mathcal{A}(x_i)$
    (iii) Update $\mathcal{C}^k$ based on the dual solution $\mu^k$.
    (iv) Solve the master problem and obtain $\beta^k$

      $\min_{U,\beta} \frac{1}{L} \sum_{l\in\mathcal{L}} U^{l,\beta}_0(x_0)$
      $s.t.\ U^l_i(x_i) \geq r(x_i, F^l_i, a_i) - \sum_{b\in\mathcal{B}_i} \beta_{i+1,f(x_i,a_i),b} \Delta^{\mathbb{E},l}_i \phi_{i+1,b} + \delta U^{l,\beta}_{i+1}(f(x_i, a_i))$
                                              $\forall (i, x_i, a_i, l) \in \mathcal{C}^k$
while $|\text{Sub\_OBJ}^k - \text{Master\_OBJ}^k| > \epsilon$;
output        : Return $\beta^k$
```

Also, we can show that Algorithm (2) is convergent.

Proposition 8. *Algorithm* (2) *converges to the optimal solution of the LP* (4.11)-(4.13) *in finite iterations.*

3.4 Numerical Study

In this section, we show the effectiveness of the constraint generation approach via merchant ethanol production. The basic settings of this application is from Chapter 2.

We compare our CG approach with the BCD method proposed in Chapter 2. Table 3.1 shows the comparison of dual bounds between CG and BCD in these instances. The

CG dual bounds are slightly worse than BCD in all instances. The percentage ratio, which is 1 minus the ratio of CG dual bound and BCD dual bound, ranges between -1.00% and -0.20% with an average of -0.65%.

Table 3.1: Comparison of dual bounds between CG and BCD approaches on benchmark instances.

Instance	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
UB-CG	19.95	19.46	24.20	25.58	21.97	18.77	16.04	22.34	23.34	20.78	19.12	14.72
UB-BCD	19.90	19.32	23.96	25.45	21.82	18.59	15.92	22.22	23.18	20.62	18.98	14.69
Ratio (%)	-0.25	-0.72	-1.00	-0.51	-0.69	-0.97	-0.75	-0.54	-0.69	-0.78	-0.74	-0.20

Table 4.6 reports the lower bounds for these two approaches. The lower bounds from CG are slightly better than the BCD bounds in all instances except in Mar, Oct, and Dec. The average improvement on the lower bound is 0.16%. In the worst case (March), the CG lower bound is 0.66% worse than BCD. In the best case (May), the improvement is 0.74%. The optimality gap, which is one minus the ratio of lower bound and the best known upper bound, for CG is slightly worse than the BCD with respective average optimality gaps of 7.56% and 7.08% for each approach. CG generates comparable bounds as the BCD approach since the difference between the results of CG and BCD are always within 1%. However, CG generates those results with 70% BCD memory. Compared to the 11 hours for the BCD approach, it takes 22 hours for CG to solve the problem. Table 3.3 reports the average optimality gaps, memories, and CPU times.

Table 3.2: Comparison of lower bounds between CG and BCD approaches on benchmark instances.

Instance	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
LB-CG	18.64	18.01	22.66	23.87	20.28	17.20	14.62	20.55	21.54	19.06	17.68	13.47
LB-BCD	18.58	17.97	22.76	23.86	20.25	17.18	14.63	20.55	21.50	19.12	17.64	13.49
Ratio (%)	0.32	0.22	-0.44	0.04	0.14	0.11	-0.01	0.00	0.19	-0.31	0.22	-0.15

Table 3.3: Comparison of the average optimality gaps, CPU times, and memories between CG and BCD.

	Gap (%)	CPU Time (hours)	Memory (%)
PO-BCD	7.08	11	100
PO-CG	7.67	22	70

3.5 Conclusion

PO has been used to obtain high-quality bounds and control policies for intractable Markov decision processes, e.g., financial and real option models. PO solves a linear program which has a large number of constraints. To improve the efficiency of solving such LP, we propose a constraint generation approach. Our solution methodology constructs a master and a subproblem to generate an iterative solution and find the violation constraints, respectively. The master problem contains a small portion of the total constraints. The subproblem decouples according to samples. So both of them are tractable compared to the original LP. We also show that the proposed approach is provably convergent. We demonstrate the use of our approach in merchant energy production problems modeled as real options. The numerical results show that our approach generates comparable lower and dual bounds to the state-of-the-art methods. Our research has potential relevance beyond option related models, e.g., for inventory/production and capacity-investment management with demand forecast updates.

Chapter 4

Modeling and Algorithmic Generalizations

4.1 Introduction

PO has been employed to obtain high quality bounds and control policies for intractable Markov decision processes (MDPs) such as optimal stopping and merchant energy production (Desai et al. 2012b, Yang et al. 2021). The state space of these MDPs typically consists of a rich information component and a finite controllable component. The former state component tracks the evolution of market information, e.g., prices and demands, while the latter one describes the assets' operational status. We further require the action space to be finite and relatively small. Many applications share this common feature when modeled as MDPs. Typical examples include financial and real option valuations (Adkins and Paxson 2011, Boogert and De Jong 2011, Boomsma et al. 2012, Carmona and Ludkovski 2010, Carriere 1996, Chandramouli and Haugh 2012, Cortazar et al. 2008, Denault et al. 2013, Devalkar et al. 2011, Enders et al. 2010, Gyurkó et al. 2015, Jaillet et al. 2004, Lai et al. 2010, Muñoz et al. 2011, Tsitsiklis and Van Roy 2001) and production and inventory management (Heath and Jackson 1994, Iida and Zipkin 2006). This paper focuses on extending PO to MDPs with such structures.

PO is rooted on the information relaxation and duality technique (Rogers 2002, Haugh and Kogan 2004, Brown et al. 2010) and value function approximation approach (VFA; see, e.g., Powell 2007, Bertsekas 2019). PO constructs a dual version of the MDP by relaxing the hindsight information while simultaneously introducing a penalty specified in linear VFAs to subtract the benefit of such information. It then solves a sample average approximation to the dual MDP and obtains VFAs that can be used to obtain lower and dual (upper) bounds, as well as a feasible operating policy. A salient feature of PO is that it generates a dual bound that is tighter than any other RL method that employs the same VFAs. In practice, PO typically outperforms other benchmark methods such as least squares Monte Carlo (LSM; Desai et al. 2012b, Yang et al. 2021).

Despite its appeal, practical use of PO requires solving a pathwise linear program (PLP) based on a Monte Carlo simulation for the informationally rich state component. However, a well defined PLP requires feasible terminating decisions at the initial controllable state; otherwise, it becomes unbounded. This requirement is naturally satisfied by optimal stoppings and merchant energy production (e.g., "Stop" and "Abandon") but is not met by many other applications such as swing options and chooser caps and inventory control. Besides, solving PLP typically necessitates developing specialized algorithms (Chandramouli 2019, Yang et al. 2021) because commercial solvers confront severe computational issues in complicated settings, e.g., chooser caps and swing options (Chandramouli 2019) and merchant energy production (Yang et al. 2021). The state of the art approach of solving PLP is the block coordinate descent (BCD) and constraint generation methods proposed in Chapter 2 and 3. These techniques have limited scalability due to their high per iteration time and space complexities.

We put forth a pseudo action scheme and a coordinate decomposition and regression method to address the above two issues, respectively. Our pseudo action scheme adds (lower) bound constraints to PLP by leveraging nonanticipative policies, i.e., policies that only rely on current information. With such constraints, PLP becomes well defined for applications that do not have terminating decisions at the controllable state. Our coordinate decomposition and regression approach involves two steps: (i) Solving the PLP dual and (ii) Recovering a primal solution based on a near optimal dual solution from (i). Specifically, we craft algorithms that rely on coordinated decomposition, i.e., the alternating direction method of multipliers (ADMM; Boyd et al. 2011), to solve the PLP dual. ADMM exploits the MDP's structures to decompose the ensuing math programming model into subproblems with closed form solutions. We use least squares regression in step (ii) to approximately enforce the complementary slackness (CS) conditions. Although not having closed form solutions, step (ii) generates near optimal primal solutions with a small portion of constraints in PLP. We provide an error bound analysis for the proposed approach and show its asymptotic convergence property. Our technique exhibits better per iteration computational complexity than both the known BCD method and directly applying ADMM to PLP.

We demonstrate the effectiveness of our pseudo action scheme with the natural gas storage problem (Secomandi et al. 2015a) where a direct implementation of PO is impossible. The numerical results show that our technique bounds PLP in this context. The obtained lower and dual bounds are comparable to LSM, the state of the art approach for storage problems. We test the performance of ADMM and regression in merchant ethanol production modeled as a compound switching and timing option (Yang et al. 2021, Guthrie 2009). For realistic instances in literature, our approach generates almost the same results as the state of the art approach for PLP, i.e., the BCD approach, but with substantially less (10%) memory and (50%) run time. Our algorithm can also deal with larger instances that are out of the reach of BCD, achieving near optimal performance and dominating LSM, a standard competitor in this case, in terms of solution quality. These results suggest that the insights in Yang et al. (2021) also hold for larger instances.

Though we focus on natural gas storage and ethanol production in this paper, our research is potentially relevant for other merchant operations contexts and related real option models (Secomandi and Seppi 2014, 2016), including oil and natural gas extraction fields, liquefied natural gas facilities, copper mines, and renewable energy plants (Brennan and Schwartz 1985, Smith and McCardle 1998, 1999, Cortazar et al. 2008, Rømo et al. 2009, Enders et al. 2010, Lai et al. 2011, Arvesen et al. 2013, Denault et al. 2013, Hinz and Yee 2018, Zhou et al. 2019).

We present the considered MDP and PO in §4.3. We introduce the pseudo action scheme and our solution approach in §4.4 and §4.5, respectively. We provide a convergence and error bound analyses in §4.6. The numerical results are reported in §4.7. We conclude in §4.8.

4.2 Literature Review

Our work first contributes to PO literature where PO has been applied to optimal stopping (Desai et al. 2012b) and merchant energy production (Yang et al. 2021). These two applications have terminating decisions ("stop" and "abandon") in their feasible action sets and thus can be naturally modeled as PLPs. Our paper extends PO to new applications that do not share the same feature in the action set, e.g., the energy storage and multiple stopping models. This extension significantly broadens the applicability of PO.

The solution approach in this paper substantially reduces the computational complexities, particularly the space complexity, of solving PLP with the extant approaches in the literature. Desai et al. (2012b) apply a commercial linear programming solver (CPLEX) to optimize their PO model readily. Chandramouli (2019) approximately solves his PO linear program as a sequence of single stopping PO models formulated and optimized as in Desai et al. (2012b). Yang et al. (2021) uses a block coordinate descent method (BCD) to solve the resulting PLP in merchant energy production. Aside from optimal stopping (Desai et al. 2012b), both Yang et al. (2021) and Chandramouli and Haugh (2012) report the difficulty of solving large scale PLP in their applications. The main bottleneck of solving PLP is the excessive memory requirement. Our work alleviates this difficulty by providing a more efficient algorithm than their approaches.

The literature on ADMM is vast but combining ADMM with PO is new. The ADMM we use can be viewed as a multi-block ADMM, which does not converge in general (Chen et al. 2016). Nevertheless, the structure of our formulation satisfies one of the sufficient conditions in Chen et al. (2016). Under such a condition, our ADMM is essentially an extension of the 2-block ADMM. Thus all existing convergence results still hold, i.e., the linear global convergence of ADMM stands in our context (Chen et al. 2016, Tao and Yuan 2012). Besides, our work shows that ADMM can employ the MDP structure embedded in the state space in the PO context. The utilization of the structure leads to a decoupling of the ADMM formulation and closed-form updating formulas in each step.

We are not the only ones who apply ADMM to an LP. Wang and Shroff (2017) also do that. Our approach differs from their work in two ways. First, our ADMM formulation is different from theirs. We treat the LP constraints as feasible sets for the variables, whereas Wang and Shroff (2017) dualize the equality constraints. Second, we decouple the subproblems in each updating step according to the MDP structure and solve the decoupled subproblems with closed-form expressions. In contrast, Wang and Shroff (2017) tried to find an approximate solution to the subproblems via the first order method. Applied to our setting, the approach of Wang and Shroff (2017) would yield a methodology with inferior computational complexity compared to ours.

As far as we know, our primal solution recovery approach is novel in the literature. Existing primal solution recovery methods mainly focus on recovering primal solutions for the dual subgradient method via an averaging scheme (Sherali and Choi 1996, Larsson et al. 1999, Barahona and Anbil 2000, Nedić and Ozdaglar 2009, Nesterov and Shikhman 2018, Gustavsson et al. 2015). On the other hand, our approach provides a way to recover a primal solution with a near optimal dual solution from ADMM. Though we directly solve the regression in this paper, our work motivates additional research to reduce the computational burden of the regression with advanced algorithms.

4.3 Preliminaries

We formulate the considered MDP and its dual version in §4.3.1 and §4.3.2, respectively. We introduce PO in §4.3.3. The regression and the greedy policy are discussed in §4.3.4.

4.3.1 Markov Decision Process

We focus on a discrete time MDP with finite horizons. The sequential decision process is over I stages. The stage number indexed by i is an element of the stage index set $\mathcal{I} := \{0, ..., I-1\}$. The state of the considered MDP consists of both the rich information component and the controllable component. For simplicity, we refer to the former one as the exogenous state and the latter as the endogenous state. The endogenous state component in stage i is denoted as x_i. It belongs to a small and finite set $\mathcal{X}_i$. The corresponding exogenous state component is F_i. It contains high dimensional information that describes the evolution of dynamic information. We denote $\mathcal{F}_i$ as the feasible set of F_i. Typical examples of applications that can be formulated as this MDP are the valuation of energy assets. In those applications, the endogenous state describes the asset's status, such as the discretized inventory level of the energy storage facility and the operational mode of the plant in energy production. The exogenous state component tracks the evolution of the energy commodity's futures prices in the wholesale market.

At each stage, the decision maker chooses an action a_i based on x_i. We denote as $\mathcal{A}_i(x_i)$ the feasible action set for $x_i \in \mathcal{X}_i$, i.e., $a_i \in \mathcal{A}_i(x_i)$. Each action a_i incurs an immediate payoff defined by the function $r(x_i, F_i, a_i) : \mathcal{X}_i \times \mathcal{F}_i \times \mathcal{A}_i(x_i) \to \mathbb{R}$.

The dynamics of the endogenous state component is governed by the transition function $f(x_i, a_i) : \mathcal{X}_i \times \mathcal{A}(x_i) \to \mathcal{X}_{i+1}$. The evolution of F_i is governed by a predetermined stochastic process such as the geometric Brownian motion. A fundamental assumption for the transitions of $F_i \in \mathcal{F}_i$ is that it is independent of the decision a_i. This assumption is also known as the small player assumption, i.e., the decision maker does not impact the entire market.

The decision rule $D_i : \mathcal{X}_i \times \mathcal{F}_i \to \mathcal{A}(x_i)$ is a mapping from the state space to the action set for each stage i. A policy π is a collection of these decision rules, i.e., $\pi = \{D_0, D_1, ..., D_{I-1}\}$. We let Π be the set of all feasible policies. We use a constant risk-free discount factor $\delta \in (0, 1)$ to calculate the current value of cash flows in the future. Suppose the initial state is (x_0, F_0), the optimal policy π^* that maximizes the expected total cash flows can be obtained by solving:

$$\max_{\pi\in\Pi} \mathbb{E}\left[\sum_{i=0}^{I-1} \delta^i r(x_i, F_i, a_i^\pi)\middle| x_0, F_0\right], (x_i, F_i, a_i^\pi) \in \mathcal{X}_i \times \mathcal{F}_i \times \mathcal{A}(x_i) \tag{4.1}$$

where $\mathbb{E}$ is the expectation w.r.t. F_i. The expectation and stochastic process that governs the transition of F_i are typically under the risk neutral measure (Shreve 2004) in financial and real option valuations. The stochastic dynamic programming (SDP) formulation of (4.1) for each $(i, x_i, F_i) \in \mathcal{I} \times \mathcal{X}_i \times \mathcal{F}_i$ is

$$V_i(x_i, F_i) = \max_{a_i \in \mathcal{A}(x_i)} \left\{ r(x_i, F_i, a_i) + \delta\mathbb{E}\left[V_{i+1}(f(x_i, a_i), F_{i+1})\middle| F_i\right]\right\} \tag{4.2}$$

where the terminal conditions are $V_I(x_I, F_I) = 0,\ \forall (x_I, F_I) \in \mathcal{X}_I \times \mathcal{F}_I$. (4.2) is computationally intractable due to the high dimensional $\mathcal{F}_i$, which is known as the "curse of dimentionality" (Powell 2007).

4.3.2 Information Relaxation and Duality Techniques

There is also a dual version of (4.1) in which decisions are made with information on the realized random variables in the future, but the benefit of this foresight is entirely eliminated by penalties (Brown et al. 2010). The dual MDP generates a dual (upper) bound on the optimal policy value of (4.1).

Let $\bar{F}$ be a sample path that includes exogenous states from stages 0 through $I-1$ starting with F_0 (we suppress this dependence from our notation for ease of exposition). The set $\bar{\mathcal{F}}$ is the collection of all such paths. We denote by $F_i(\bar{F})$ the stage i exogenous state corresponding to sample path $\bar{F}$. The dual policy $\bar{\pi}$ is the collection of decision rules $\{\bar{D}_i^{\bar{\pi}}, i \in \mathcal{I}\}$, where $\bar{D}_i^{\bar{\pi}} : \mathcal{X}_i \times \bar{\mathcal{F}} \to \mathcal{A}_i(x_i)$ prescribes a feasible action for stage i, endogenous state x_i, and sample path $\bar{F}$. The set of such policies is $\bar{\Pi}$.

Ideal penalties depend on the value function associated with (4.2). Consider stage $i \neq I-1$ and suppose we take feasible action a_i for endogenous state x_i and sample path $\bar{F}$. The ideal penalty is the additional value of knowing the stage $i+1$ information $F_{i+1}(\bar{F})$ at stage i relative to only having knowledge of the information $F_i(\bar{F})$ at this stage, which corresponds to the discounted difference

$$\delta \left(V_{i+1}\left(f(x_i, a_i), F_{i+1}(\bar{F})\right) - \mathbb{E}[V_{i+1}\left(f(x_i, a_i), F_{i+1}\right) | F_i(\bar{F})] \right). \tag{4.3}$$

We use (4.3) to reduce the cash flow that ensues in the stage $i \neq I-1$ from applying the decision rule $\bar{D}_i^{\bar{\pi}}$ to the pair $(x_i, \bar{F})$. The resulting dual MDP is

$$\begin{aligned}\mathbb{E}\Bigg[\max_{\bar{\pi}\in\bar{\Pi}} \Bigg\{ \sum_{i\in\mathcal{I}\setminus\{I-1\}} \delta^i \Big[r(x_i^{\bar{\pi}}, s_i(\bar{F}), \bar{A}_i^{\bar{\pi}}) - \delta\big(V_{i+1}(f(x_i^{\bar{\pi}}, \bar{A}_i^{\bar{\pi}}), F_{i+1}(\bar{F}))\\ - \mathbb{E}[V_{i+1}(f(x_i^{\bar{\pi}}, \bar{A}_i^{\bar{\pi}}), F_{i+1}) | F_i(\bar{F})]\big)\Big] + \delta^{I-1} r(x_{I-1}^{\bar{\pi}}, s_{I-1}(\bar{F}), \bar{A}_{I-1}^{\bar{\pi}}) \Bigg\} \Bigg| x_0, F_0 \Bigg],\end{aligned} \tag{4.4}$$

where we use the shorthand notation $\bar{D}_i^{\bar{\pi}}$ instead of $\bar{D}_i^{\bar{\pi}}(x_i^{\bar{\pi}}, \bar{F}_i)$. This model differs from (4.1) in two key ways: (i) The maximization is inside the expectation because dual policies depend on sample paths and (ii) its objective function is the sum of the discounted ideally penalized rewards and the last stage reward. Let $V_0(x_0, F_0)$ be the value function for stage 0 and the given state (x_0, F_0), which is obtained in a manner analogous to (4.2) for this stage and state. At optimality the objective function (4.4) equals $V_0(x_0, F_0)$ for each sample path (Brown et al. 2010). It follows that (4.1) and (4.4) are equivalent at the optimality.

4.3.3 PO

The dual model (4.4) is intractable because (i) the ideal penalties are unknown, and (ii) the outer expectation is impossible to evaluate exactly in general. We formulate PO and its associated linear program to deal with these two issues.

PO addresses (i) by replacing the value function $V_{i+1}(f(x_i, a_i), F_{i+1})$ with a linear VFA $\sum_{b=1}^{B} \phi_{i+1,b}(F_{i+1}(\bar{F}))\beta_{i+1,f(x_i,a_i),b}$, where $\phi_{i+1,b}(\cdot)$ is the basis function in set $\Phi_{i+1} =$

$\{\phi_{i+1,1}, \phi_{i+1,2}, ..., \phi_{i+1,B}\}$ for stage $i+1$, and $\beta_{i+1,f(x_i,a_i),b} \in \mathbb{R}$ is the weight associated with the b-th basis function for the stage and state pair $(i+1, f(x_i,a_i))$. Let $\Delta_i^{\mathbb{E}}\phi_{i+1,b} := \delta\{\phi_{i+1,b}(F_{i+1}(\bar{F})) - \mathbb{E}[\phi_{i+1,b}(F_{i+1})|F_i(\bar{F})]\}$, a good dual penalty based on the linear VFA is

$$\sum_{b\in\mathcal{B}_i} \beta_{i+1,f(x_i,a_i),b}\Delta_i^{\mathbb{E}}\phi_{i+1,b}. \tag{4.5}$$

The conditional expectation in (4.5) needs to be evaluated. Approximating them by sample average approximations is a possibility (Desai et al. 2012b) but introduces an error in the dual bound estimate. We thus choose basis functions and stochastic models for the evolution of the vector of forward curves that satisfy Assumption 3, which is common in the literature (see, e.g., Glasserman and Yu 2004, Nadarajah et al. 2017 and references therein)

Assumption 3. *The expectation $\mathbb{E}[\phi_{i+1,b}(F_{i+1})|F_i(\bar{F})]$ is available in an efficiently computable closed form for each i and $i+1 \in \mathcal{I}\setminus\{I-1\}$ and $F_i \in \mathcal{F}_i$.*

We obtain PO by replacing the ideal dual penalty with (4.5), and minimizing the objective value over β, :

$$\min_{\beta}\left\{\mathbb{E}\left[\max_{\pi\in\Pi}\sum_{i=0}^{N-1}\delta^i\left(r(x_i,F_i(\bar{F}),a_i^\pi) - \sum_{b\in\mathcal{B}_i}\beta_{i+1,f(x_i,a_i),b}\Delta_i^{\mathbb{E}}\phi_{i+1,b}\right)\middle| x_0,F_0\right]\right\},$$
$$(x_i,F_i(\bar{F}))\in\mathcal{X}_i\times\mathcal{F}_i, a_i^\pi\in\mathcal{A}(x_i) \tag{4.6}$$

The minimization in (4.6) suggests that PO generates the *tightest* dual bound than any other dual approaches using VFA in the span of the set of basis functions by construction. Besides (4.6) is essentially an unconstrained convex optimization over β because operators $\min\{\cdot\}$ and $\mathbb{E}[\cdot]$ preserve the convexity of the piecewise linear function (Desai et al. 2012b).

Computing the expectation in (4.6) is challenging in general. However the value of (4.6) can be approximated by Monte Carlo simulation conveniently. Consider generating L random sample paths from the underlying stochastic process. For any fixed β, the sample average of the maximal policy value on each sample is a good proxy of the value of (4.6) with a sufficiently large L, so the sampled PO used to approximate (4.6) is defined as

$$\min_{\beta}\left\{\frac{1}{L}\sum_{l\in\mathcal{L}}\left[\max_{\pi\in\Pi}\sum_{i=0}^{I-1}\delta^i\left(r(x_i^l,F_i^l,a_i^{\pi,l}) - \sum_{b\in\mathcal{B}_i}\beta_{i+1,f(x_i,a_i),b}\Delta_i^{\mathbb{E},l}\phi_{i+1,b}\right)\right]\right\},$$
$$(x_i^l,F_i^l,l)\in\mathcal{X}_i\times\mathcal{F}_i\times\mathcal{L}, a_i^{\pi,l}\in\mathcal{A}(x_i^l) \tag{4.7}$$

where $\mathcal{L} = \{1,2,...,L\}$ is the set of sample paths. We initialize every sample path as (x_0,F_0), i.e., $(x_0^l,F_0^l) := (x_0,F_0), \forall l \in \mathcal{L}$. Note that the maximization in (4.7) for each sample path is deterministic. For a fixed β and sample path l, we can denote the policy value for state x_i as $U_0^{l,\beta}(x_0)$, i.e., for $(i, x_i, F_i^l, l) \in \mathcal{I} \setminus \{I-1\} \times \mathcal{X}_i \times \mathcal{F}_i \times \mathcal{L}$

$$U_0^{l,\beta}(x_0) = \max_{\pi \in \Pi} \sum_{i=0}^{I-1} \delta^i \left(r(x_i^l, F_i^l, a_i^{\pi,l}) - \sum_{b \in \mathcal{B}_i} \beta_{i+1, f(x_i, a_i), b} \Delta_i^{\mathbb{E},l} \phi_{i+1,b} \right) \tag{4.8}$$

By substituting the maximization in (4.7) with $U_0^{l,\beta}(x_0)$ on each sample path, (4.7) can be equivalently expressed as solving

$$\min_{U,\beta} \frac{1}{L} \sum_{l \in \mathcal{L}} U_0^{l,\beta}(x_0) \tag{4.9}$$

with the dual value variable $U_i^l(x_i)$ defined by

$$U_i^{l,\beta}(x_i) = \max_{a_i \in \mathcal{A}(x_i)} \left\{ r(x_i, F_i^l, a_i) - \sum_{b \in \mathcal{B}_i} \beta_{i+1, f(x_i, a_i), b} \Delta_i^{\mathbb{E},l} \phi_{i+1,b} + \delta U_{i+1}^{l,\beta}(f(x_i, a_i)) \right\},$$
$$(i, x_i, F_i^l, l) \in \mathcal{I} \times \mathcal{X}_i \times \mathcal{F}_i \times \mathcal{L} \tag{4.10}$$

where $U_i^{l,\beta}(x_i)$ is the dual value function for each $(i, x_i, l) \in \mathcal{I} \times \mathcal{X}_i \times \mathcal{L}$. The boundary conditions for (4.10) is $U_I^{l,\beta}(x_I) = 0, (x_I, l) \in \mathcal{X}_I \times \mathcal{L}$. We can reformulate (4.9)-(4.10) as the following linear program (Manne 1960):

$$\min_{U,\beta} \frac{1}{L} \sum_{l \in \mathcal{L}} U_0^{l,\beta}(x_0) \tag{4.11}$$

$$s.t.\ U_i^l(x_i) \geq r(x_i, F_i^l, a_i) - \sum_{b \in \mathcal{B}_i} \beta_{i+1, f(x_i,a_i), b} \Delta_i^{\mathbb{E},l} \phi_{i+1,b}, + \delta U_{i+1}^{l,\beta}(f(x_i, a_i))$$
$$\forall (i, x_i, a_i, l) \in \mathcal{I} \setminus \{I-1\} \times \mathcal{X}_i \times \mathcal{A}(x_i) \times \mathcal{L} \tag{4.12}$$

$$U_{I-1}^{l,\beta}(x_{I-1}) \geq r(x_{I-1}, F_{I-1}^l, a_{I-1}), \forall (x_{I-1}, a_{I-1}, l) \in \mathcal{X}_{I-1} \times \mathcal{A}(x_{I-1}) \times \mathcal{L} \tag{4.13}$$

(4.11)-(4.13), known as PLP, is well defined when there is a nonpenalized action in $\mathcal{A}(x_0)$ (see Proposition 1 in Yang et al. 2021). This condition is naturally satisfied in optimal stopping ("Stop") and merchant energy production ("Abandon"). Nevertheless, many applications, e.g., multiple stoppings and inventory control, do not have such action at x_0. So (4.11)-(4.13) becomes unbounded in those cases. We will discuss the reason for this issue and propose a pseudo action scheme to fix it in §4.4.

4.3.4 Unbiased Upper Bound and Greedy Lower Bound

The optimal objective function value of (4.11) is a biased estimate of (4.6) due to the finite sample paths of the vectors of forward curves used in practice. An unbiased dual bound can be obtained with an independent set of Monte Carlo simulation sample paths and the solution of the VFA coefficient vector β^{PLP} from (4.11)-(4.13). This unbiased estimation can be performed by solving an analogy of (4.11)-(4.13) with the new sample paths but fixing the value of β as β^{PLP}.

We use the common greedy approach in the literature (see, e.g., Powell 2007, §6.4) to obtain a feasible operating policy and a lower bound. Fixing a VFA weight vector β, the greedy decision rule for the pair $(i, x_i) \in \mathcal{I} \times \mathcal{X}_i$ is

$$\arg\max_{a_i \in \mathcal{A}_i(x_i)} \left\{ r(x_i, F_i, a_i) + \delta \sum_{b \in \mathcal{B}_i} \beta_{i+1, f(x_i, a_i), b} \mathbb{E}[\phi_{i+1,b}(F_{i+1}) | F_i] \right\},$$

with ties broken in some prespecified way. To estimate its associated lower bound, we employ the same set of sample paths of the vectors of forward curves used to obtain an unbiased dual bound estimate and apply the greedy policy to the states visited along each such path starting from the initial stage and state. The average of the sum of the resulting discounted rewards is an unbiased lower bound estimate.

We determine VFAs for the lower bound with the vector U^{PLP} obtained from PLP because β^{PLP} may lead to a weak lower bound estimate (Desai et al. 2012b). Following Desai et al. (2012b), we obtain the vector $\beta_{i,x_i} := (\beta_{i,x_i,b}, b \in \{0, 1, 2, ..., B_i\})$ via solving the regression

$$\min_{\beta_{i,x_i}} \frac{1}{L} \sum_{l \in \mathcal{L}} \left(U_i^{l, \beta^{\text{PLP}}}(x_i) - \sum_{b \in \mathcal{B}_i} \beta_{i,x_i,b} \phi_{i,b}(F_i^l) \right)^2 .$$

where $\phi_{i,b} \in \{1\} \cup \Phi_i$. We employ these resulting optimal solutions to specify VFAs and consequently obtain a greedy policy, from which we estimate a lower bound.

4.4 Pseudo Action Scheme

As discussed in §4.3.3, the boundedness of (4.11)-(4.13) relies on the feasibility of terminating actions that do not incur dual penalties at x_0; otherwise, the optimal value of (4.11) goes to negative infinity. The unboundedness issue is mainly due to the sample average approximation to the expectation in (4.6). With finite sample paths, the average of the coefficient for each $\beta_{1,x_1,b}, b \in \mathcal{B}$, i.e., $1/L \sum_{l=1}^{L} \Delta_0^{\mathbb{E},l} \phi_{1,b}$, is not strictly 0. Since $\beta_{1,x_1,b}, \forall b \in \mathcal{B}$ is a free variable, there exists a $\beta_{1,x_1} := \{\beta_{1,x_1,b}, b \in \mathcal{B}\}$ vector for each $x_1 \in \mathcal{X}_1$ that makes the corresponding averaged dual penalty goes to positive infinity. So the averaged penalized payoff for every action becomes negative infinity. If $U_0^{l,\beta}(x_0), \forall l \in \mathcal{L}$ does not have any lower bound, the sample average, $1/L \sum_{l=1}^{L} U_0^{l,\beta}(x_0)$, becomes unbounded. Appendix B.2 illustrates this issue with a simple example.

We propose a pseudo action scheme to fix the issue. The main idea is to add a pseudo terminating action to $\mathcal{A}(x_0)$ with its payoff determined by nonanticipative policies. Since the nonanticipative policy does not benefit from any foresight information, the payoff of the action following such policy incurs 0 dual penalties. By setting its payoff equal to the payoff of a nonanticipative action sequence from stage 0 to $I-1$, the pseudo terminating action provides a lower bound for each $U_0^l(x_0), l \in \mathcal{L}$.

Specifically, we label as A both the pseudo action and the associated pseudo terminal state. The extended endogenous state set is defined as $\bar{\mathcal{X}}_i := \mathcal{X}_i \bigcup \{\mathsf{A}\}$ for $i \in \mathcal{I}$. The new feasible action set is denoted as $\bar{\mathcal{A}}_i(x_i)$. We have $\bar{\mathcal{A}}_0(x_0) := \mathcal{A}_0(x_0) \bigcup \{\mathsf{A}\}$; $\bar{\mathcal{A}}_i(x_i) := \mathcal{A}_i(x_i)$, if $(i, x_i) \in \mathcal{I} \setminus \{0\} \times \mathcal{X}_i$; $\bar{\mathcal{A}}_i(x_i) := \{\mathsf{A}\}$, if $(i, x_i) \in \mathcal{I} \setminus \{0\} \times \{\mathsf{A}\}$. The endogenous state transition function for $(i, x_i, a_i) \in \mathcal{I} \times \bar{\mathcal{X}}_i \times \bar{\mathcal{A}}_i$ is

$$\bar{f}(x_i, a_i) := \begin{cases} f(x_i, a_i), & \text{if } (x_i, a_i) \in \mathcal{X}_i \times \mathcal{A}_i(x_i), \\ \mathsf{A}, & \text{if } (x_0, a_0) \in \mathcal{X}_0 \times \{\mathsf{A}\}, \\ \mathsf{A}, & \text{if } (x_i, a_i) \in \{(\mathsf{A}, \mathsf{A})\}. \end{cases} \tag{4.14}$$

(4.14) suggests that the pseudo state A can only be reached by taking A at x_0. Once the state is reached, there is no other way to transit to states in $\mathcal{X}_i$.

Let π_s be a nonanticipative policy and $\sum_{i=0}^{I-1} \delta^i r(x_i^{\pi_s}, F_i, a_i^{\pi_s})$ the corresponding policy value at (x_0, F_0). Then we define the payoff for the pseudo action A at state (x_0, F_0) and (A, F_i) as $\sum_{i=0}^{I-1} \delta^i r(x_i^{\pi_s}, F_i, a_i^{\pi_s})$ and 0, respectively. So the payoff function, denoted as $\bar{r}(x_i, F_i, a_i)$, for each $(x_i, F_i, a_i) \in \bar{\mathcal{X}}_i \times \mathcal{F}_i \times \bar{\mathcal{A}}_i(x_i)$ is

$$\bar{r}(x_i, F_i, a_i) := \begin{cases} r(x_i, F_i, a_i), & \text{if } (x_i, F_i, a_i) \in \mathcal{X}_i \times \mathcal{F}_i \times \mathcal{A}_i(x_i), \\ \sum_{j=0}^{I-1} \delta^j r(x_j^{\pi_s}, F_j, a_j^{\pi_s}), & \text{if } (x_i, F_i, a_i) \in \mathcal{X}_0 \times \mathcal{F}_0 \times \{\mathsf{A}\}, \\ 0, & \text{if } (x_i, F_i, a_i) \in \{\mathsf{A}\} \times \mathcal{F}_i \times \{\mathsf{A}\}. \end{cases} \tag{4.15}$$

Denote as $\bar{V}_i(x_i, F_i)$ the value function for state (x_i, F_i) in the extended model. The corresponding SDP is

$$\bar{V}_i(x_i, F_i) = \max_{a_i \in \bar{\mathcal{A}}_i(x_i)} \left\{ \bar{r}(x_i, F_i, a_i) + \delta \mathbb{E}\left[\bar{V}_{i+1}(\bar{f}(x_i, a_i), F_{i+1}) \middle| F_i \right] \right\}, \tag{4.16}$$

with the boundary condition $\bar{V}_I(x_I, F_I) = 0$. The dual SDP becomes

$$\min_{\beta} \mathbb{E}[u_0^{F,\beta}(x_0) | x_0, F_0] \tag{4.17}$$

where the dual value variables $u_i^{l,\beta}(x_i)$ are defined by the following equations

$$u_i^{l,\beta}(x_i) = \max_{a_i \in \bar{\mathcal{A}}_i(x_i)} \Big\{ \bar{r}(x_i, s_i^l, a_i) - \mathbb{1}(\bar{f}(x_i, a_i) \in \mathcal{X}_{i+1}) \sum_{b \in \mathcal{B}_i} \beta_{i+1, \bar{f}(x_i,a_i), b} \Delta_i^{\mathbb{E},l} \phi_{i+1,b} + \delta u_{i+1}^{l,\beta}(\bar{f}(x_i, a_i)) \Big\}, \tag{4.18}$$

where $\forall (i, x_i) \in \mathcal{I} \times \bar{\mathcal{X}}_i$. The boundary conditions are $u_I^{l,\beta}(x_I) = 0$. The PLP associated with PO (4.17) is

$$\min_{u,\beta} \frac{1}{L} \sum_{l \in \mathcal{L}} u_0^{l,\beta}(x_0) \tag{4.19}$$

$$s.t.\ u_i^{l,\beta}(x_i) \geq \bar{r}(x_i, F_i^l, a_i) - \mathbb{1}(\bar{f}(x_i, a_i) \in \mathcal{X}_i) \sum_{b \in \mathcal{B}_i} \beta_{i+1, \bar{f}(x_i,a_i), b} \Delta_i^{\mathbb{E},l} \phi_{i+1,b}, + \delta u_{i+1}^{l,\beta}(f(x_i, a_i)) \quad \forall (i, x_i, a_i, l) \in \mathcal{I} \setminus \{I-1\} \times \bar{\mathcal{X}}_i \times \bar{\mathcal{A}}(x_i) \times \mathcal{L} \tag{4.20}$$

$$u_{I-1}^{l,\beta}(x_{I-1}) \geq \bar{r}(x_{I-1}, F_{I-1}^l, a_{I-1}), \forall (x_{I-1}, a_{I-1}, l) \in \bar{\mathcal{X}}_{I-1} \times \bar{\mathcal{A}}(x_{I-1}) \times \mathcal{L} \tag{4.21}$$

A major difference between (4.11)-(4.13) and (4.19)-(4.21) is that there is an indicator function in front of the dual penalty in (4.20). This indicator function puts a null penalty on the payoff when the action is A. The constraints associated with the pseudo actions provide valid lower bounds on $u_0^{l,\beta}(x_0)$. Furthermore, (4.17) and (4.19) have the same optimal objective function value since the nonanticipative policy is dominated by the optimal policy. These properties are formally stated in Proposition 9.

Proposition 9. *(4.17) is equivalent to (4.6). Moreover, the associated PLP (4.19)-(4.21) is bounded from below and converges to the optimal objective function value of (4.17) almost surely.*

Proposition 9 also suggests that the optimal objective function value of (4.19) and (4.21) is independent of the nonanticipative policy used. That is, we can use either a static policy or a dynamic policy to bound PLP. Although the values of these two policies are different, the two resulting PLPs have the same optimal objective function value.

4.5 Alternating Direction Method of Multipliers and Primal Solution Recovery

It is challenging to solve (4.19)-(4.21) because: (i) PLP is potentially ill conditioned due to the colinearity among coefficient columns of β variables; (ii) the size of PLP is large in practical instances. The former issue is well solved by applying principal component analysis (PCA) to the coefficient matrix of β (please see Yang et al. 2021 for a detailed discussion on the ill conditioning and the PCA approach). We focus on dealing with the later issue by developing an algorithm that features low computational complexities in this section. The proposed method first solves PLP dual and then recovers a primal solution.

In §4.5.1, we introduce an ADMM reformulation of PLP dual. We then discuss the primal solution recovery approach in §4.5.2.

4.5.1 ADMM Reformulation

Despite the difficulty of solving (4.19)-(4.21) directly, its dual problem exhibits a decomposition structure. We exploit this structure via the ADMM approach. To illustrate, we start from PLP dual:

$$\max_{\mu} \sum_{l\in\mathcal{L}} \sum_{i\in\mathcal{I}} \sum_{x_i\in\bar{\mathcal{X}}_i} \sum_{a_i\in\bar{\mathcal{A}}(x_i)} \mu^l_{x_i,a_i} \bar{r}(x_i, F^l_i, a_i) \tag{4.22}$$

$$s.t. \sum_{a_0\in\bar{\mathcal{A}}(x_0)} \mu^l_{x_0,a_0} = \frac{1}{L}, \forall l\in\mathcal{L} \tag{4.23}$$

$$\sum_{x_i\in\bar{\mathcal{X}}_i} \sum_{a_i\in\bar{\mathcal{A}}(x_i)} \mathbb{1}_{\bar{f}(x_i,a_i)=x_{i+1}} \mu^l_{x_i,a_i} = \delta \sum_{a_{i+1}\in\bar{\mathcal{A}}(x_{i+1})} \mu^l_{x_{i+1},a_{i+1}}, \quad \forall (i+1, x_{i+1}, l) \in \mathcal{I}\times\bar{\mathcal{X}}_{i+1}\times\mathcal{L} \tag{4.24}$$

$$\sum_{l\in\mathcal{L}} \sum_{i\in\mathcal{I}} \sum_{x_i\in\bar{\mathcal{X}}_i} \sum_{a_i\in\bar{\mathcal{A}}(x_i)} \mathbb{1}_{\bar{f}(x_i,a_i)=x_{i+1}} \Delta^{\mathbb{E},l}_i \phi_{i+1,b} \mu^l_{x_i,a_i} = 0, \quad \forall (i+1, x_{i+1}, b) \in \mathcal{I}\setminus\{I-1\}\times\mathcal{X}_{i+1}\times\mathcal{B}_{i+1} \tag{4.25}$$

$$\mu^l_{x_i,a_i} \geq 0, \forall (l, x_i, a_i) \in \mathcal{L}\times\bar{\mathcal{X}}_i\times\bar{\mathcal{A}}(x_i). \tag{4.26}$$

The decision variable $\mu^l_{x_i,a_i}$ in (4.22)-(4.26) corresponds to the endogenous state-action pair $(x_i, a_i) \in \bar{\mathcal{X}}_i \times \bar{\mathcal{A}}(x_i)$ on each sample path $l \in \mathcal{L}$. The objective function is the total payoffs weighted by $\mu^l_{x_i,a_i}, \forall (l, i, x_i, a_i) \in \mathcal{L}\times\mathcal{I}\times\bar{\mathcal{X}}_i\times\bar{\mathcal{A}}(x_i)$.

The PLP dual contains three kinds of constraints. We refer them as balance constraints (4.23)-(4.24), strong duality constraints (4.25), and nonnegativity constraints (4.26). The balance constraints suggest that the sum of the dual variables reaching the pair $(i+1, x_{i+1}) \in \mathcal{I}\setminus\{0\}\times\bar{\mathcal{X}}_{i+1}$ equals the sum of dual variables leaving that pair on every sample path. For $(0, x_0)$, the summation is equal to $1/L$. Intuitively, the balance constraints are analogous to flow constraints in a network problem, which restrict the incoming flow equals the outgoing flow for each node. The strong duality constraints (4.25) suggest that the sum of coefficients weighted by $\mu^l_{x_i,a_i}$ for each primal variable

$\beta_{i+1,x_{i+1},b}$ over all samples is 0. (4.25) enforces the strong duality property (Brown et al. 2010, Theorem 2.1 and 2.2) of the dual bound in a finite sample space $\mathcal{L}$. The constraint ensures that the weighted sum of penalties is 0, so the optimal dual bound equals the optimal policy value of the MDP in $\mathcal{L}$.

To simplify the exposition, (4.22)-(4.26) can be written in a compact matrix form

$$\begin{aligned} \max_{\mu} \quad & \mu^\top \bar{r} && (4.27)\\ s.t. \quad & A\mu = d; && (4.28)\\ & C\mu = 0; && (4.29)\\ & \mu \geq 0; && (4.30) \end{aligned}$$

where μ is the variable vector defined as $\mu := (\mu^l_{x_i,a_i}, (l, x_i, a_i) \in \mathcal{L} \times \bar{\mathcal{X}}_i \times \bar{\mathcal{A}}(x_i))$. $\bar{r}$ is the payoff vector $\bar{r} := (r(x_i, F^l_i, a_i), (l, i, x_i, a_i) \in \mathcal{L} \times \mathcal{I} \times \bar{\mathcal{X}}_i \times \bar{\mathcal{A}}(x_i))$. A and d are the coefficient matrix for μ and RHS of (4.23)-(4.24), respectively. C is the coefficient matrix in (4.25).

To derive an ADMM reformulation for (4.27)-(4.30), we first duplicate the dual variable μ as μ_1, μ_2 and μ_3, and set each of these duplicates satisfying one of the three constraints, respectively. So the respective domains of μ_1, μ_2 and μ_3 become the affine spaces defined by (4.28)-(4.30). To make the new formulation equivalent to (4.27)-(4.30), we add constraints to force μ_1, μ_2, and μ_3 to equal each other. Specifically, suppose we use μ_1 in (4.28), μ_2 in (4.29), and μ_3 in (4.30), the ADMM reformulation of (4.27)-(4.30) is

$$\begin{aligned} \min_{\mu_1,\mu_2,\mu_3} \quad & f(\mu_1) + g(\mu_2) + h(\mu_3) && (4.31)\\ s.t. \quad & A_1\mu_1 + A_2\mu_2 + A_3\mu_3 = 0 && (4.32)\\ & \mu_1 \in \mathcal{P};\;\; \mu_2 \in \mathcal{Q};\;\; \mu_3 \in \mathcal{R} && (4.33) \end{aligned}$$

where $A_1 := [I, 0]^\top$, $A_2 := [0, I]^\top$, $A_3 := [-I, -I]^\top$, $\mathcal{P} := \{\mu_1 : A\mu_1 = d\}$, $\mathcal{Q} := \{\mu_2 : C\mu_2 = 0\}$, and $\mathcal{R} := \{\mu_3 : \mu_3 \geq 0\}$. $h(\mu_3)$ is defined as $h(\mu_3) := -\mu_3^\top r$. $f(\mu_1)$ and $g(\mu_2)$ are the respective indicator functions of $\mathcal{P}$ and $\mathcal{Q}$:

$$f(\mu_1) = \begin{cases} 0 & \mu_1 \in \mathcal{P} \\ +\infty & \text{else} \end{cases} \qquad g(\mu_2) = \begin{cases} 0 & \mu_2 \in \mathcal{Q} \\ +\infty & \text{else} \end{cases} \qquad (4.34)$$

(4.27)-(4.30) and (4.31)-(4.33) are equivalent because the optimal solution to the former one must also optimally solves the latter one. (4.31)-(4.33) share the same spirit of the consensus ADMM (Boyd et al. 2011, §7). The main difference between our reformulation and the consensus ADMM is that each duplicate has a different domain in our setting .

By dualizing (4.32) with Lagrangian multipliers $y := [y_1, y_2]$ and adding a regularization term amplified by a penalty parameter $\rho > 0$, the augmented Lagrangian function for (4.31)-(4.33) is

$$L_\rho(\mu_1, \mu_2, \mu_3, y) := f(\mu_1)+g(\mu_2)+h(\mu_3)+y^\top(A_1\mu_1+A_2\mu_2+A_3\mu_3)+\rho \left\|A_1\mu_1 + A_2\mu_2 + A_3\mu_3\right\|_2^2 \tag{4.35}$$

ADMM consists of iterations:

$$\mu_1^{k+1} = \arg\min_{\mu_1 \in \mathcal{P}} L_\rho(\mu_1, \mu_2^k, \mu_3^k, y^k) \tag{4.36}$$

$$\mu_2^{k+1} = \arg\min_{\mu_2 \in \mathcal{Q}} L_\rho(\mu_1^{k+1}, \mu_2, \mu_3^k, y^k) \tag{4.37}$$

$$\mu_3^{k+1} = \arg\min_{\mu_3 \in \mathcal{R}} L_\rho(\mu_1^{k+1}, \mu_2^{k+1}, \mu_3, y^k) \tag{4.38}$$

$$y^{k+1} = y^k + \rho(A_1\mu_1^{k+1} + A_2\mu_2^{k+1} + A_3\mu_3^{k+1}) \tag{4.39}$$

where $(\mu_1^k, \mu_2^k, \mu_3^k, y^k)$ is the value of (μ_1, μ_2, μ_3, y) in the k-th iteration. (4.36)-(4.38) minimize over μ_1, μ_2 and μ_3 respectively while fixing other variables' values. The Lagrangian multipler is updated in (4.39).

The affine spaces $\mathcal{P}$, $\mathcal{Q}$, and $\mathcal{R}$ in (4.36)-(4.39) have decomposition structures. Specifically, $\mathcal{P}$ decouples by samples because each balance constraint only involves decision variables on the same sample path. In other words, (4.28) essentially describes L independent flow networks. $\mathcal{Q}$ decouples according to the pair $(i, x_i) \in \mathcal{I} \times \bar{\mathcal{X}}_i$. To see this, we first partition (4.29) into $\sum_{i\in\mathcal{I}} |\mathcal{X}_i|$ groups according to (i, x_i). We can do this because there is no common variables in any two different groups of constraints. That is, for each pair $(i, x_i) \in \mathcal{I} \times \mathcal{X}_i$, the (i, x_i)-th group strong duality constraints exclusively contain dual variables $\mu^l_{x_{i-1}, a_{i-1}}$ satisfying $f(x_{i-1}, a_{i-1}) = x_i$ for all $l \in \mathcal{L}$. Finally, $\mathcal{R}$, i.e., the nonnegative constraint (4.26), decouples by both samples and endogenous states because it holds for every dual variable.

(4.36), (4.37) and (4.38), which are essentially orthogonal projections onto $\mathcal{P}$, $\mathcal{Q}$ and $\mathcal{R}$, also decouple according to samples, endogenous states, and both samples and endogenous states, respectively. Furthermore, these updates have closed form expressions. Specifically, (4.36) requires solving the quadratic programming (QP):

$$\min_{\mu_1} \quad \left\|\mu_1 - \mu_3^k + y_1^k\right\|_2^2 \tag{4.40}$$

$$s.t. \quad A\mu_1 = d. \tag{4.41}$$

(4.40) is a squared 2-norm so it decouples by samples. (4.41) also decouples by samples as discussed above. Thus, we can solve (4.40)-(4.41) optimally by solving L sample-wise QPs separately. Suppose the sample-wise component in A is A^l, i.e., A^l is the coefficient matrix of the dual variable vector $\mu_1^l := (\mu_{1,x_i,a_i}^l, (x_i, a_i) \in \bar{\mathcal{X}}_i \times \bar{\mathcal{A}}_i)$ for a fixed $l \in \mathcal{L}$. The sample-wise QP is

$$\min_{\mu_1^l} \quad \left\| \mu_1^l - \mu_3^{k,l} + y_1^{k,l} \right\|_2^2 \tag{4.42}$$

$$s.t. \quad A^l \mu_1^l = d^l; \tag{4.43}$$

where d^l, $\mu_2^{k,l}$ and $y_1^{k,l}$ are the corresponding sample-wise components in d, μ_2^k and y_1^k, respectively. The optimal solutions to (4.42)-(4.43) for all $l \in \mathcal{L}$ consist an optimal solution to (4.40)-(4.41).

(4.42)-(4.43) indeed have a closed form solution because (i) A^l is identical on different sample paths, i.e., $A^l = A^{l'}$ $\forall l, l' \in \mathcal{L}$, and (ii) A^l is of full row rank. (i) is true because A^l describes the transition of x_i, which is independent of the sample paths. We show (ii) in the following lemma.

Lemma 2. *$A^l, \forall l \in \mathcal{L}$ has full row rank*

Based on the decomposable structure of (4.40)-(4.41) and the features of A^l, (4.40)-(4.41) have the following closed form solution.

$$\mu_1^l = z_1^{l,k} - A^{l,\top}(A^l A^{l,\top})^{-1}(A^l z_1^{l,k} - d^l), \forall l \in \mathcal{L} \tag{4.44}$$

where $z_1^{l,k} := \mu_3^{l,k} - y_1^{l,k}$. Since the inverse $(A^l A^{l,\top})^{-1}$ does not change during iterations, it can be calculated in advance and stored in cache.

The projection (4.37) requires solving

$$\min_{\mu_2} \quad \left\| \mu_2 - \mu_3^k + y_2^k \right\|_2^2 \tag{4.45}$$

$$s.t. \quad C\mu_2 = 0. \tag{4.46}$$

which decouples by endogenous states due to the decomposition of the objective function and (4.46). Analogous to the previous projection, (4.45)-(4.46) also have closed form solutions for the decoupled QPs. Let C_{i,x_i} denote the coefficient matrix in (4.46) for the variable vector $\mu_{2,i,x_i} := (\mu_{2,x_{i-1},a_{i-1}}^l, s.t.\ f(x_{i-1}, a_{i-1}) = x_i,\ \forall (i-1, x_{i-1}, a_{i-1}) \in \mathcal{I} \setminus \{0\} \times \mathcal{X}_i \times \mathcal{A}(x_i))$. The decoupled Q for (4.45)-(4.46) is

$$\min_{\mu_{2,i,x_i}} \quad \left\| \mu_{2,i,x_i} - \mu_{3,i,x_i}^k + y_{2,i,x_i}^k \right\|_2^2 \tag{4.47}$$

$$s.t. \quad C_{i,x_i} \mu_{2,i,x_i} = 0; \tag{4.48}$$

where μ_{3,i,x_i}^k and y_{2,i,x_i}^k are the corresponding state-wise components in μ_3^k and y_2^k, respectively. Assumption 4 imposes a property on C_{i,x_i}.

Assumption 4. *C_{i,x_i} has full row rank for every $(i, x_i) \in \mathcal{I} \setminus \{0\} \times \mathcal{X}_i$.*

This assumption is mild because we can always remove correlated rows in C_{i,x_i} without changing the optimal objective value of PLP, which is formally stated in the following lemma:

Lemma 3. *Suppose C_{i,x_i} is rank deficient and C'_{i,x_i} is obtained by deleting linear correlated rows from C_{i,x_i}. Then the optimal objective value of PLP dual stays the same if C_{i,x_i} is replaced by C'_{i,x_i}.*

With Assumption 4 and its decomposition structures, (4.45)-(4.46) can be solved in closed forms

$$\mu_{2,i,x_i} = C_{i,x_i}^\top (C_{i,x_i} C_{i,x_i}^\top)^{-1} C_{i,x_i} z_{2,i,x_i}^k, \forall (i, x_i) \in \mathcal{I} \setminus \{0\} \times \mathcal{X}_i \tag{4.49}$$

where $z_{2,i,x_i}^k := \mu_{3,i,x_i}^k - y_{2,i,x_i}^k$. In practice, the ill conditioning of C_{i,x_i} may influence the accuracy of the matrix inversion in (4.49). To deal with this issue, we apply PCA to each $C_{i,x_i}^\top$ to orthogonalize the columns. It can be shown that this preconditioning procedure will not change the optimal objective function value (see more details in Yang et al. 2021).

The last projection is

$$\begin{aligned} \min_{\mu_3} \quad & -\mu_3^\top r + \frac{\rho}{2} \left\| \mu_1^{k+1} - \mu_2 + y_1^k \right\|^2 + \left\| \mu_3^{k+1} - \mu_2 + y_2^k \right\|^2 && (4.50) \\ s.t. \quad & \mu_3 \geq 0; && (4.51) \end{aligned}$$

Since the feasible region is a nonnegative orthant, the closed-form solution is simply a nonnegative truncation of the optimal solution to unconstrained optimization (4.50), i.e.,

$$\mu_3 = \frac{1}{2}(\frac{r}{\rho} + \mu_1^{k+1} + \mu_2^{k+1} + y_1^k + y_2^k)^+. \tag{4.52}$$

which clearly can be computed elementwisely. We summarize the closed-form updates (4.44), (4.49) and (4.52) in Proposition 10. The complete algorithm is summarized in Algorithm 3.

Proposition 10. *The ADMM iterations (4.36)-(4.38) have respective closed-form solutions (i)-(iii):*

$$(i)\ \mu_1^{l,k+1} = z_1^{l,k} - A^{l,\top}(A^l A^{l,\top})^{-1}(A^l z_1^{l,k} - d^l), \forall l \in \mathcal{L}$$

$$(ii)\ \mu_{2,i,x_i}^{k+1} = C_{i,x_i}^\top (C_{i,x_i} C_{i,x_i}^\top)^{-1} C_{i,x_i} z_{2,i,x_i}^k, \forall (i, x_i) \in \mathcal{I} \setminus \{0\} \times \mathcal{X}_i$$

$$(iii)\ \mu_{3,x_i,a_i}^{l,k+1} = \frac{1}{2}(\frac{r(x_i, a_i, s_i)}{\rho} + \mu_1^{k+1} + \mu_2^{k+1} + y_1^k + y_2^k)^+, \forall (l, i, x_i, a_i) \in \mathcal{L} \times \mathcal{I} \times \bar{\mathcal{X}}_i \times \bar{\mathcal{A}}(x_i)$$

The inputs to Algorithm 3 are the coefficient matrices A^l and C_{i,x_i}, the coefficient vector $\bar{r}$ in the objective function, the RHS of constraints d, the stopping tolerance ϵ,

```
Algorithm 3: Decoupled ADMM
input          : $C_{i,x_i}, \forall (i,x_i) \in \mathcal{I} \setminus \{0\} \times \mathcal{X}_i$; $A^l, \forall l \in \mathcal{L}$;
                 $(C_{i,x_i} C_{i,x_i}^\top)^{-1}, \forall (i,x_i) \in \mathcal{I} \setminus \{0\} \times \mathcal{X}_i$; $(A^l A^{l,\top})^{-1}, \forall l \in \mathcal{L}$; stopping
                 tolerance $\epsilon > 0$; $\rho > 0$.
initialization: Set $k = 0$, $(\mu_1^1, \mu_2^1, \mu_3^1, y_1^1, y_2^1) = (0,0,0,0,0)$
do
    $k = k + 1.$
    $(i)\ \mu_1^{l,k+1} = z_1^{l,k} - A^{l,\top}(A^l A^{l,\top})^{-1}(A^l z_1^{l,k} - d^l), \forall l \in \mathcal{L}$
    $(ii)\ \mu_{2,i,x_i}^{k+1} = C_{i,x_i}^\top (C_{i,x_i} C_{i,x_i}^\top)^{-1} C_{i,x_i} z_{2,i,x_i}^k, \forall (i,x_i) \in \mathcal{I} \times \mathcal{X}_i$
    $(iii)\ \mu_{3,x_i,a_i}^{l,k+1} = \frac{1}{2}(\frac{\bar{r}(x_i,a_i,s_i)}{\rho} + \mu_1^{k+1} + \mu_2^{k+1} + y_1^k + y_2^k)^+,$
            $\forall (l,i,x_i,a_i) \in \mathcal{L} \times \mathcal{I} \times \bar{\mathcal{X}}_i \times \bar{\mathcal{A}}(x_i)$
    $(iv)\ y_{1,x_i,a_i}^{l,k+1} = y_{1,x_i,a_i}^{l,k} + \mu_{1,x_i,a_i}^{l,k+1} - \mu_{3,x_i,a_i}^{l,k+1}, \forall (l,i,x_i,a_i) \in \mathcal{L} \times \mathcal{I} \times \bar{\mathcal{X}}_i \times \bar{\mathcal{A}}(x_i)$
    $(v)\ y_{2,x_i,a_i}^{l,k+1} = y_{2,x_i,a_i}^{l,k} + \mu_{2,x_i,a_i}^{l,k+1} - \mu_{3,x_i,a_i}^{l,k+1}, \forall (l,i,x_i,a_i) \in \mathcal{L} \times \mathcal{I} \times \bar{\mathcal{X}}_i \times \bar{\mathcal{A}}(x_i)$
    $(vi)\ \epsilon^{k+1} = \left\|\mu_1^{k+1} - \mu_3^{k+1}\right\|_2 + \left\|\mu_2^{k+1} - \mu_3^{k+1}\right\|_2$
    $(vii)\ s^{k+1} = \left\|\mu_3^{k+1} - \mu_3^k\right\|_2$
while $\epsilon^{k+1} > \epsilon$ and $s^{k+1} > \epsilon$;
output         : Return
                 $(\bar{\mu}_1, \bar{\mu}_2, \bar{\mu}_3, \bar{y}_1, \bar{y}_2, \bar{\epsilon}, \bar{s}) = (\mu_1^{k+1}, \mu_2^{k+1}, \mu_3^{k+1}, y_1^{k+1}, y_2^{k+1}, \epsilon^{k+1}, s^{k+1})$
```

and the penalty parameter ρ. In addition, we compute the $(C_{i,x_i} C_{i,x_i}^\top)^{-1}$ and $(A^l A^{l,\top})^{-1}$ in advance and store the results in cache. The algorithm starts from an initial solution $(0,0,0,0,0)$. It updates μ_1, μ_2, μ_3 and y in a coordinated way through (i) to (v). In steps (vi) and (vii), Algorithm 3 computes the primal residual (ϵ^{k+1}) and dual residual (s^{k+1}) for the $(k+1)$-th iteration. These two values reflect the difference among the duplicates and the change of μ_3 in two consecutive iterations, respectively. At optimality, both the primal and dual residuals are 0. In practice, we set a small tolerance ϵ for these two residuals. The algorithm terminates once $\epsilon^{k+1} < \epsilon$ and $s^{k+1} < \epsilon$.

The stopping criteria indeed provides a theoretical guarantee for the absolute gap between the current objective value and the optimal objective value. Suppose the gap for a solution $\bar{\mu}_3$ and the optimal solution μ^* is denoted as $AG(\bar{\mu}_3)$, i.e., $AG(\bar{\mu}_3) := (\bar{\mu}_3)^\top \bar{r} - (\mu^*)^\top \bar{r}$ and the suboptimality of the current solution is upper bounded by $\mathcal{C}$, i.e., $\|\bar{\mu}_3 - \mu^*\|_2 \leq \mathcal{C}$. Using the results in Boyd et al. (2011), we have

$$AG(\bar{\mu}_3) \leq (\|\bar{y}\|_2 + \mathcal{C})\epsilon \tag{4.53}$$

(4.53) means that $AG(\bar{\mu}_3)$ is bounded by a value related to the Lagrangian dual, the current solution and the primal and dual residual. Since $\|\bar{y}\|_2 + \mathcal{C}$ is finite, if ϵ is sufficiently small, the absolute gap approaches 0.

4.5.2 Primal Solution Recovery

As discussed in §4.3, we need to recover the VFA weight vector β in PLP to compute a feasible operating policy and lower bound. A common way for linear programming is to use complementary slackness. If a dual variable is nonzero, the constraint associated with the dual variable is tight; otherwise, the constraint is redundant at the optimality. The optimal primal solution can be obtained by solving a recovery LP with the same objective function as the original LP and tight constraints corresponding to nonzero dual solutions.

Using CS conditions to recover a primal solution requires a highly accurate optimal dual solution. If a dual solution is suboptimal, a slight deviation from its optimal value (e.g., 0) largely influences whether the associated constraint should be kept in the CS-based recovery LP. Consequently, the LP may produce primal solution with poor quality or become infeasible due to contradictory equality constraints.

Meanwhile, it is well known that ADMM converges to modest accuracy (e.g., 10^{-3}) in the first thousands of iterations but requires substantially more iterations for solutions with high accuracy. This feature of ADMM makes it difficult to recover a primal solution directly through the CS condition. Besides, a highly accurate solution is not essential in our context and most ADP applications as it brings little to no improvements on the accuracy of VFA (Boyd et al. 2011).

To overcome this issue, we develop a primal solution recovery method via simple linear regression with a near-optimal dual solution from Algorithm 3. We define differences between the LHS and RHS of (4.20)-(4.21) for $\forall(l, i.x_i, a_i) \in \mathcal{L} \times \mathcal{I} \times \bar{\mathcal{X}}_i \times \bar{\mathcal{A}}(x_i)$ as

$$\mathcal{D}^l_{i,x_i,a_i} := u^{l,\beta}_i(x_i) - \bar{r}(x_i, F^l_i, a_i) + \sum_{b \in \mathcal{B}_i} \beta_{i+1,f(x_i,a_i),b} \Delta^{\mathbb{E},l}_i \phi_{i+1,b} - \delta u^{l,\beta}_{i+1}(f(x_i, a_i))$$

The value of the VFA weight vector, denoted as $\bar{\beta}$, is recovered by the following regression

$$\min_{u,\beta} \sum_{\substack{(l,i,x_i,a_i) \in \\ \mathcal{L} \times \mathcal{I} \times \bar{\mathcal{X}}_i \times \bar{\mathcal{A}}(x_i)}} \mathbb{1}(\bar{\mu}^l_{1,i,x_i,a_i} \neq 0) \left(\mathcal{D}^l_{i,x_i,a_i}\right)^2 \tag{4.54}$$

The objective function minimizes the overall differences between the LHS and RHS of (4.20)-(4.21). The indicator function in (4.54) means we only consider constraints corresponding to nonzero dual variables, i.e., $\bar{\mu}^l_{1,i,x_i,a_i} \neq 0$. We can also use $\bar{\mu}^l_{2,i,x_i,a_i}$ and $\bar{\mu}^l_{3,i,x_i,a_i}$ to select constraints. This choice doesn't influence the bounds' quality because there are only minor differences among those variables when the algorithm terminates. (4.54) approximately enforces the complementary slackness but does not strictly require the equality holds. Clearly, if the dual solution is optimal, the optimal objective function value of (4.54) is 0.

```
Algorithm 4: Primal Solution Recovery
  input  : Solution $\bar{\mu}_1$ from Algorithm 3; The coefficient matrix $C$ and $A$;
           The coefficient vectors $\bar{r}$, $d$; Sample set $\mathcal{L}'$
  solve  :
           $(\bar{u}, \bar{\beta}) = \arg\min_{u,\beta} \sum_{\substack{(l,i,x_i,a_i)\in \\ \mathcal{L}'\times\mathcal{I}\times\bar{\mathcal{X}}_i\times\bar{\mathcal{A}}(x_i)}} \mathbb{1}(\bar{\mu}^l_{1,i,x_i,a_i} \neq 0)\left(\mathcal{D}^l_{i,x_i,a_i}\right)^2 \qquad (4.55)$
  output : return $\bar{\beta}$
```

We can recover a near optimal primal solution $\bar{\beta}$ with a subset of $\mathcal{L}$. Intuitively this is due to the "similarity" among the i.i.d. samples in $\mathcal{L}$. That is, if we select a subset of "representative" samples ($\mathcal{L}'$) from $\mathcal{L}$ in the regression, the recovered solution is close to the one recovered by $\mathcal{L}$, because samples in $\mathcal{L}'$ "represent" those in $\mathcal{L} \setminus \mathcal{L}'$. In the ideal case where the dual solution is optimal, it can be shown that the number of samples required only depends on the number of β variables. We'll further discuss this point in §4.6.2. We summarize the regression method in Algorithm 4.

4.5.3 Complexities

PLP dual is solved with less time and space complexities by Algorithm 3 compared to the state-of-the-art BCD method in Yang et al. (2021). We denote as M and D the cardinalities of the largest endogenous state and feasible action sets, respectively. That is, $M := \max\{|\mathcal{X}_1|, ...|\mathcal{X}_{I-1}|\}$ and $D := \max\{|\bar{\mathcal{A}}(x_i)|, (i, x_i) \in \mathcal{I} \times \bar{\mathcal{X}}_i\}$. We let $H := \sum_{i\in\mathcal{I}} \sum_{x_i\in\mathcal{X}_i} |\mathcal{A}(x_i)|$ be the number of feasible actions, and $B := \max\{|\mathcal{B}_i|, i \in \mathcal{I}\}$ be the largest number of basis functions. Then the per iteration time and space complexities of executing steps (ii)-(iv), which are essentially matrix operations, can be summarized as below:

Table 4.1: Computational Complexities for the ADMM Updates

	Time	Space
Step (ii)	$\mathcal{O}(I^2M^2HL)$	$\mathcal{O}(IMH)$
Step (iii)	$\mathcal{O}(BDL)$	$\mathcal{O}(BDL)$
Step (iv)	$\mathcal{O}(HL)$	$\mathcal{O}(H)$

Since most applications have $L \gg H$, the per iteration space complexity of Algorithm 3 is $\mathcal{O}(BDL)$. The per iteration time complexity depends on the problem structure. It is $\mathcal{O}(I^2M^2HL)$, if $I^2M^2HL > BD$; $\mathcal{O}(BDL)$, otherwise.

Algorithm 4 solves a QP. The time complexity of the interior point method for a QP is $\mathcal{O}((I^3M^3L^3 + I^3M^3B^3)/K^3)$, where $K := |\mathcal{L}|/|\mathcal{L}'|$. Since $L \gg B$, the time complexity is determined by $\mathcal{O}(I^3M^3L^3/K^3)$. The space complexity is $\mathcal{O}(IMHL^2/K^2)$. Compared to Algorithm 3, the complexities of Algorithm 4 are much higher, which makes solving (4.55) the bottleneck of our proposed method. However, these complexities are still much smaller than the BCD approach in the literature, which has the respective time and space complexities $\mathcal{O}(I^3M^3L^3)$ and $\mathcal{O}(IMHL^2)$.

It is worth mentioning that Algorithm 4 provides a framework to recover a primal solution to PLP. Using advanced algorithms to solve the regression may further reduce the complexities. For example, the two-block coordinate descent (CD) approach can decompose the regression: if we fix the u variable, the problem decouples according to endogenous states; if we fix the β variable, it decouples by samples. Besides, the variables can be updated with closed-form solutions in every iteration. The resulting per-iteration time and space complexities are respective $\mathcal{O}(IML)$ and $\mathcal{O}(IL)$. However, implementing CD approach may require sophisticated coding and tuning skills. Thus, finding a reliable algorithm that can solve the regression with less computing efforts is an interesting direction for future research.

4.6 Guarantees

In this section, we present theoretical guarantees on the performance of Algorithm 3 and 4. In §4.6.1, we discuss the convergence results for Algorithm 3. In §4.6.2, we show the asymptotic convergence property of Algorithm 4 and provide an error bound analysis for it.

4.6.1 Convergence of ADMM

Our ADMM reformulation for (4.27)-(4.30) is a multi-block ADMM which does not have convergence property in general (Chen et al. 2016). However, our reformulation satisfies $A_1^\top A_2 = [I, 0] \times [0, I]^\top = 0$ which is one of the sufficient conditions for a convergent multi-block ADMM (Chen et al. 2016). Under such condition, (4.31)-(4.33) are equivalent to a 2 block ADMM by viewing (μ_1, μ_2) as a variable, $[A_1, A_2]$ its coefficient matrix and $f(\mu_1) + g(\mu_2)$ its objective function. So the convergence analysis for the 2 block ADMM in literature (Tao and Yuan 2012) holds for Algorithm 3. For completeness, we repeat the result in Chen et al. (2016) and Tao and Yuan (2012).

Proposition 11. (Theorem 4.1 in Tao and Yuan 2012 and Theorem 2.5 in Chen et al. 2016) *Algorithm 3 converges to an optimal solution of (4.27)-(4.30) at a rate of* $\mathcal{O}(1/k)$.

4.6.2 Analysis on the Primal Solution Recovery

A major advantage of our primal solution recovery approach is that for any given $\bar{\mu}_1$, (4.54) is always feasible. In addition, the recovered $\bar{\beta}$ is also feasible to (4.19)-(4.21) because β is free in PLP. In fact, this recovery is exact in the ideal case. The following proposition shows that if $\bar{\mu}_1$ is optimal, the recovered $\bar{\beta}$ is also optimal.

Proposition 12. *Given an optimal dual solution, (4.54) recovers an optimal $\bar{\beta}$ to (4.19)-(4.21).*

Proposition (12) seems trivially true: if the dual solution is optimal, a solution recovered by (4.54) satisfies the CS conditions, and therefore should be optimal. However, this may not be the case in general because the least squares regression may generate multiple solutions. It is possible that one of those recovered solutions is optimal to PLP, but the others are not. Proposition (12) and its proof indicate that in our case, we can rule out this possibility because the solution set of the regression is indeed a singleton if $\bar{\mu}_1$ is optimal. So the recovered $(\bar{u}, \bar{\beta})$ must be optimal. Furthermore the following proposition shows that we can recover the optimal primal solution with $\sum_{i \in \mathcal{I}} \sum_{x_i \in \mathcal{X}_i} B_i$ samples.

Proposition 13. *Given an optimal dual solution, Algorithm 4 can recover an optimal primal solution with at least $\sum_{i \in \mathcal{I}} \sum_{x_i \in \mathcal{X}_i} B_i$ samples.*

The intuition behind this proposition can be interpreted as follows: the primal solution recovery can be performed in a low dimensional space whose dimension only depends on the number of β variables, i.e., $\sum_{i \in \mathcal{I}} \sum_{x_i \in \mathcal{X}_i} B_i$. In other words, once the value of β variables is fixed in the regression in Algorithm 4, the value of U variables is uniquely determined.

In practice, since the dual solution is suboptimal, the recovered $\bar{\beta}$ is not optimal in general. We provide a theoretical bound on the difference between the objective function value associated with $\bar{\beta}$ and the optimal objective function value. It guarantees that if the dual solution is near optimal, so is the recovered primal solution.

Proposition 14. *Suppose that the optimal dual solution is μ^* and that the near optimal dual solution $\bar{\mu}_1$ from Algorithm 4 satisfies $\|\bar{\mu}_1 - \mu^*\|_2 \leq \mathcal{C}$. The suboptimality of the objective function value associated with $\bar{\beta}$ from Algorithm 4 is bounded by*

$$OBJ(\bar{\beta}) - OPT \leq \|\bar{y}\|_2 \|\epsilon\|_2 + \mathcal{C} \|\epsilon\|_2 + \|\bar{\mu}_1\|_2 \|\tilde{\epsilon}\|_2 \tag{4.56}$$

where OPT, ϵ and $\tilde{\epsilon}$ represent the optimal objective function value of PLP, the primal residual in Algorithm 3 and the regression residual in Algorithm 4, respectively.

Proposition 14 indicates that the obtained solution from Algorithm 3 and 4 has an objective function value that deviates at most $\|\bar{y}\|_2 \|\epsilon\|_2 + D \|\epsilon\|_2 + \|\bar{\mu}_1\|_2 \|\tilde{\epsilon}\|_2$ from the optimal objective function value of PLP. The error bound relates to the difference among variable duplicates, i.e., ϵ, the Lagrangian multiplier $\bar{y}$, the solution $\bar{\mu}_1$, and the residual $\tilde{\epsilon}$.

Proposition 14, in conjunction with Proposition 11 and 12, shows the asymptotic convergence property of the recovered primal solution: When $\epsilon \to 0$ and $\tilde{\epsilon} \to 0$ as $k \to \infty$, $OBJ(\bar{\beta})$ approaches OPT gradually. The deviation of the primal objective function value consists of two parts: the error from solving PLP dual and the one from recovering a primal solution. If these two errors are sufficiently small, the $OBJ(\bar{\beta})$ is close to OPT.

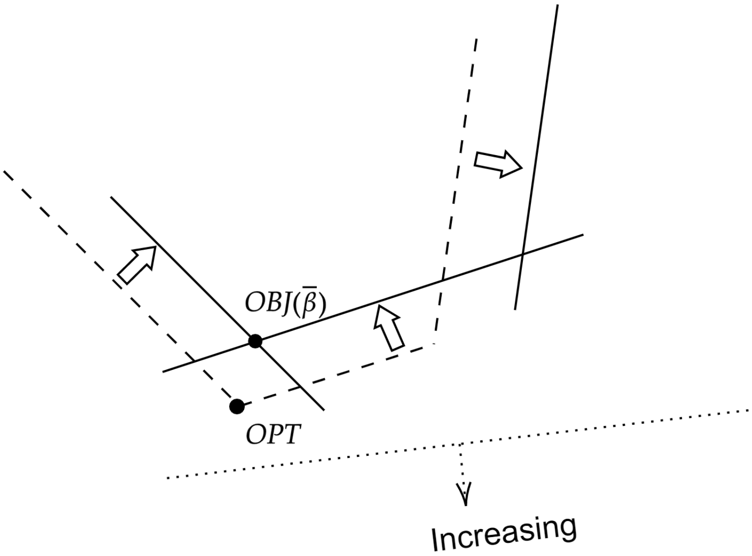


Figure 4.1: Suboptimality of OBJ($\bar{\beta}$)

The intuition for Proposition 14 is illustrated in Figure 4.1. The dashed lines and the area surrounded by them represent the original constraints and feasible region. OPT is the optimal objective function value. The area enclosed by solid lines corresponds to the new feasible region based on the near-optimal dual solutions from Algorithm 3. $OBJ(\bar{\beta})$ is the objective function value associated with the recovered solution. As shown in Figure 4.1, $OBJ(\bar{\beta})$ deviates slightly from OPT due to the small shift (arrows) of the constraints. In fact, we can show that $(\bar{\mu}_1, \bar{\mu}_2, \bar{\mu}_3)$ is an optimal solution to an adjust LP.

Proposition 15. *Suppose $(\bar{\mu}_1, \bar{\mu}_2, \bar{\mu}_3) = (\mu_1^{k+1}, \mu_2^{k+1}, \mu_3^{k+1})$ is a solution from Algorithm 3, then it is the optimal solution to the following LP, where $\epsilon_1 = \bar{\mu}_1 - \mu_3^k$, $\epsilon_2 = \bar{\mu}_2 - \mu_3^k$ and $\delta = \bar{\mu}_3 - \mu_3^k$*

$$\max_{\mu} \quad \mu_1^\top (r - 2\rho\delta) \tag{4.57}$$

$$s.t. \quad \mu_1 - \mu_3 = \epsilon_1 - \delta \tag{4.58}$$

$$\mu_2 - \mu_3 = \epsilon_2 - \delta \tag{4.59}$$

$$\mu_1 \in \mathcal{P};\ \ \mu_2 \in \mathcal{Q};\ \ \mu_3 \in \mathcal{R} \tag{4.60}$$

Proposition (15) suggests that Algorithm 4 indeed recovers the optimal primal solution to (4.57)-(4.60). Compared to (4.27)-(4.30), the above LP has different RHS and coefficients in the objective function. Since these differences approaches 0 as Algorithm 3 converges to the optimality, the recovered solution from Algorithm 4 should also converges to the optimal primal solution.

4.7 Numerical Study

In this section, we apply the pseudo action scheme to natural gas storage in Secomandi et al. (2015a) and the ADMM and regression approach to merchant ethanol production in Chapter 2, respectively. PO can not be directly applied to natural gas storage because the underlying PLP is unbounded. We apply the pseudo action scheme to fix the issue and therefore show its effectiveness. We demonstrate the use of the ADMM and regression approach in merchant energy production. We show that our approach can solve both existing instances with substantially less computational efforts and larger instances that are unsolvable by the state of the art method in the literature.

We implemented all proposed approaches in CentOS Linux 7 and C++ with the GCC 4.8.5 (Red Hat 4.8.5-11) compiler. Gurobi 7.5 is used to solve the QP in Algorithm 4. We apply the LAPACKE package to perform PCAs and matrix operations, respectively. All simulations are performed on a server with 128 GB of RAM and 12 Intel(R) Core(TM) i7-5930K processors.

The section is organized as follows: in §4.7.1 and §4.7.2, we introduce the natural gas storage and ethanol production settings, respectively. We specify the MDP discussed in §4.3 as the instances in each of these subsections and then present the results.

4.7.1 Example: Managing Natural Gas Storage Asset

Consider a natural gas commodity merchant who makes intertemporal commodity tradings by renting storage capacity from the owner of the storage facilities. The merchant is interested in maximizing the total cash flows brought by the lease contract. At each stage, the merchant can: (i) buy natural gas from the wholesale market at the spot price and inject it into the rented storage facility (if there is still capacity), (ii) withdraw the natural gas from the facility (if the inventory is nonempty) and sell it to the market, and (iii) do nothing. The merchant's decision is complicated by the commodity market's highly volatile natural gas price.

Instance

We consider a 2 year instance with monthly stages. Suppose the endogenous state x_i is the inventory level in storage at stage i. We denote the storage capacity as $\bar{C}$. So the inventory level is between 0 and $\bar{C}$ at each stage. An action $a \in \mathbb{R}$ represents the inventory change between stages i and $i+1$. A positive action is a withdrawal-and-sell decision, a negative action is an energy purchase-and-inject decision, and zero is the do-nothing decision.

The marginal payoff of a withdraw-and-sell decision is $s^W := \phi^W s - c^W$, where $\phi^W \in (0,1]$ is a fraction factor that models the inventory withdraw loss, s is the spot price of natural gas, and c^W is the fixed marginal cost of withdrawing. Similarly, the marginal payoff of the inject-and-buy decision is $s^I := \phi^I s + c^I$ where ϕ^I and c^I are the respective fraction of inventory injection loss and the marginal cost of injection. If the action is do-nothing, we assume it incurs 0 costs for simplicity. So the intermediate payoff is equal to $s^I a$ if $a < 0$, $s^W a$ if $a > 0$, and 0 if $a = 0$.

The inventory transition function $f(x_i, a)$ is $x_i - a$. The per-stage capacities for the withdrawal and injection are $C^W > 0$ and $C^I < 0$, respectively. So the feasible action sets for the withdraw and injection at the inventory level x are respective $\big[0, \min\{x_i, C^W\}\big]$ and $\big[\max\{C^I, (x_i - \bar{C})\}, 0\big]$. The entire feasible action set is the union of these two sets, i.e., $\big[0, \min\{x_i, C^W\}\big] \bigcup \big[\max\{C^I, (x_i - \bar{C})\}, 0\big]$. The optimal policy to this problem is known to have a double basestock structure (Secomandi 2010, Secomandi et al. 2015b). If the capacities C^I, C^W, and $\bar{C}$ are integer multiples of some positive real number Q, the inventory level can be optimally discretized as $\{0, Q, 2Q..., \bar{C}\}$. The initial inventory level x_0 is 0. Table 4.2 summarizes the specific parameter values used in our simulation.

We denote the stage i price of the maturity $j \geq i$ natural gas futures as $F_{i,j} \in \mathbb{R}$. The forward curve at stage i is $F_i := (F_{i,j}, j \in \mathcal{I}, i \leq j)$. The stage i spot price is $F_{i,i}$. The transition of the forward curves is governed by a multi-factor term structure model which is commonly used in both practice and literature (Clewlow and Strickland 2000, Lai et al. 2010, Secomandi et al. 2015a, Nadarajah et al. 2017):

$$\frac{dF(i,j)}{F(i,j)} = \sum_{k=1}^{K} \sigma_{i,j,k} dW_k$$

where $F(i,j)$ is the time i futures price with maturity j ($F(i,j) = F_{i,j}$), K is the number of factors, $\sigma_{i,j,k}$ and dW_k are the k-th loading coefficient and standard Brownian motion increment, respectively. The K standard Brownian motions $dW_k, k \in \{1, 2, ..., K\}$ are mutually independent, i.e., $dW_k dW_{k'} = 0, k' \neq k$, and $k, k' \in \{1, 2, .., K\}$.

Table 4.2: Values of the common parameters.

Parameter	Value
c^I	0.02\$
c^W	0.01\$
ϕ^I	1.01
ϕ^W	0.99
$\bar{C}$	1
Q	1

We calibrated our term structure model to the daily data of natural gas futures prices observed between January 2016 and December 2017 in the New York Mercantile Exchange (NYMEX). We consider four instances with the initial date corresponding to the first day of March, June, September, and December in 2016, respectively. We denote these four instances as Mar, Jun, Sep, and Dec. We use the one month US Treasury rate as the constant risk free interest rate on each initial date. They are respective 0.18%, 0.10%, 0.27%, and 0.25% for the Mar, Jun, Sep, and Dec instances.

We use the linear basis functions. For each stage $i \in \mathcal{I}$ these functions are (i) one; (ii) $\{F_{i,j}, j \in \mathcal{I}, j \geq i\}$. The selected basis functions satisfy Assupmtion 3. The conditional expectations of the basis functions are $\mathbb{E}[F_{i+1,j}|F_{i,j}] = F_{i,j}, \forall i, j \in \mathcal{I}, j \geq i$.

Results

We choose least squares Monte Carlo method (LSM) as the benchmark method as it is widely used in the literature and practice (Carriere 1996, Longstaff and Schwartz 2001, Smith 2005, Cortazar et al. 2008, Nadarajah and Secomandi 2018b). We also use the corresponding LSM policy as the nonanticipative policy to bound the PLP. The PO and LSM are executed with 10,000 Monte Carlo sample paths of the vector of forward curves starting from the initial date of each instance. We generate an independent set of 10,000 vectors of forward curve sample paths to estimate the lower and dual bounds for these two approaches.

Table 4.3 reports the bound estimates with the standard error of each estimate displayed in parentheses. PO and LSM yield almost the same lower and upper bounds in all instances. The PO lower bounds are slightly worse than LSM in the Mar, Jun, and Dec instances. However, the PO dual bound is better than LSM in the Mar instance. The suboptimality of the PO and LSM greedy policies with respect to their corresponding upper bound estimates are less than 2.2% and 1.4%, respectively. The standard errors of the PO and LSM bound estimates are at most 0.6% of their corresponding estimates. These results show that our pseudo action scheme effectively provides a lower bound for PLP. The resulting PLP generates a high quality operating policy and bounds.

Table 4.3: The estimated lower and upper bounds of natural gas storage (standard errors in parenthesis)

	PO		LSM	
	LB	UB	LB	UB
Mar	1.49 (0.001)	1.51 (0.010)	1.50 (0.006)	1.52 (0.001)
Jun	1.47 (0.010)	1.48 (0.009)	1.48 (0.010)	1.48 (0.002)
Sep	1.20 (0.007)	1.21 (0.005)	1.20 (0.010)	1.21 (0.002)
Dec	0.90 (0.010)	0.92 (0.006)	0.91 (0.010)	0.92 (0.002)

4.7.2 Example: Merchant Energy Production

The basic settings are largely from Chapter 2. The only difference here is that the mothballing and reactivation processes may last multiple stages, which is a typical case in practice. To capture this change, we redefine the MDP in an analogous way as we did in Chapter 2.

Instance

The status of the production facility in stage i is tracked by x_i. The feasible state set for x_i in stage i is denoted as $\mathcal{X}_i := \{\mathsf{A}\}\bigcup\{\mathsf{O}\}\bigcup\mathcal{M}\bigcup\mathcal{R}$ where A represents the abandoned state; O is the operational state; $\mathcal{M}$ is a set containing the K^M-stage duration of mothballing processes, i.e., $\mathcal{M} = \{\mathsf{M}_1, ..., \mathsf{M}_{K^M}\}$; $\mathcal{R} = \{\mathsf{R}_1, ..., \mathsf{R}_{K^R-1}\}$ describes the K^R-stage reactivation processes. For $i, j \in \mathcal{I}$, we use $F^c_{i,j}$ to represent the futures price in stage i for corn with maturity j. When $i = j$, $F^c_{i,i}$ is the spot price s^c_i. The price vector F^c_i is $F^c_i = (F^c_{i,j}, j \in \{i, ..., I-1\})$. We let each $F^c_{i,j} \in \mathbb{R}$. The spot and futures prices for ethanol, corn and natural gas in stage i are denoted by $F_i := (F^c_i, c \in \{\mathrm{C}, \mathrm{NG}, \mathrm{E}\})$ where C, NG and E are abbreviation of corn, natural gas and ethanol.

We denote action produce as P; suspend as S; mothball as M; reactivate as R and abandon as A. The feasible action sets for state O and M_{K^M} are $\mathcal{A}(\mathsf{O}) = \{\mathsf{A}, \mathsf{P}, \mathsf{S}, \mathsf{M}\}$ and $\mathcal{A}(\mathsf{M}_{K^M}) = \{\mathsf{A}, \mathsf{M}, \mathsf{R}\}$ respectively. We assume the mothballing and reactivation processes can not be interrupted so the feasible action set is $\mathcal{A}(x_i) = \{\mathsf{M}\}$ for $x_i \in \{\mathsf{M}_1, ..., \mathsf{M}_{K^M-1}\}$ and $\mathcal{A}(x_i) = \{\mathsf{R}\}$ for $x_i \in \{\mathsf{R}_1, ..., \mathsf{R}_{K^R-1}\}$. Once the plant is abandoned, the decision process is ended so $A(x_i) = \emptyset$ for $x_i \in \{\mathsf{A}\}$. In the last stage, we set the feasible action set as a singleton $\{\mathsf{A}\}$ for $\forall x_{I-1} \in \mathcal{X}_{I-1}$.

The production margin is $s^{\mathrm{E}}_i - \gamma_{\mathrm{C}} s^{\mathrm{C}}_i - \gamma_{\mathrm{NG}} s^{\mathrm{NG}}_i$ which depends on spot prices of both inputs and outputs. γ_{C} and γ_{NG} are the respective consumptions of corn and natural gas for one unit of ethanol. The production quantity is a constant Q, i.e., we assume the plant produces at full capacity. A fixed cost for production is C_P. The maintenance cost for suspension is C_S. The mothballing will trigger an one time cost I_M and a maintenance cost $\mathsf{C}_\mathsf{M}(< \mathsf{C}_\mathsf{S})$. The reactivation has a fixed cost I_R. The plant has a salvage value S_A if abandoned. The reward function for every $(x_i, F_i, a_i) \in \mathcal{X}_i \times \mathcal{F}_i \times \mathcal{A}(x_i)$ is

$$r(x_i, s_i, a_i) := \begin{cases} (s^{\mathrm{E}}_i - \gamma_{\mathrm{C}} s^{\mathrm{C}}_i - \gamma_{\mathrm{N}} s^{\mathrm{N}}_i)Q - \mathsf{C}_\mathsf{P}, & \text{if } (x_i, a_i) \in (\mathsf{O}, \mathsf{P}), \\ -\mathsf{C}_\mathsf{S}, & \text{if } (x_i, a_i) = (\mathsf{O}, \mathsf{S}), \\ -\mathsf{I}_\mathsf{M}, & \text{if } (x_i, a_i) = (\mathsf{O}, \mathsf{M}_1), \\ -\mathsf{C}_\mathsf{M}, & \text{if } (x_i, a_i) = (\mathsf{M}_{K^\mathsf{M}}, \mathsf{M}_{K^\mathsf{M}}), \\ -\mathsf{I}_\mathsf{R}, & \text{if } (x_i, a_i) = (\mathsf{M}_{K^\mathsf{M}}, \mathsf{R}_1), \\ S_\mathsf{A}, & \text{if } (x_i, a_i) \in \{(\mathsf{O}, \mathsf{A}), (\mathsf{M}_{K^\mathsf{M}}, \mathsf{A}), (\mathsf{R}_{K^\mathsf{R}-1}, \mathsf{O})\} \end{cases}$$

We denote $f(x_i, a_i) : \mathcal{X}_i \times \mathcal{A}_i \to \mathcal{X}_{i+1}$ as the state transition function governing the transitions among endogenous states. At the state O, the plant's next state is still O if the action is P and S. If $(x_i, a_i) = (\mathsf{O}, \mathsf{M})$, the state will transit from O to M_1. The mothballing process lasts K^M stages so the next state is M_{n+1} for $(x_i, a_i) \in \{(\mathsf{M}_n, \mathsf{M}) | 1 \leq n < K^M\}$. When the plant is fully mothballed, the merchant can keep it mothballed or reactivate it to operational. If $(x_i, a_i) = (\mathsf{M}_{K^M}, \mathsf{M})$, the next state is still M_{K^M}; if $(x_i, a_i) = (\mathsf{M}_{K^M}, \mathsf{R})$, the next state is R_1. Analogous to the mothballing process, the next state during the reactivation process is R_{n+1} for $(x_i, a_i) \in \{(\mathsf{R}_n, \mathsf{R}) | 1 \leq n < K^R - 1\}$. The state transition function therefore can be defined as

$$f(x_i, a_i) := \begin{cases} \mathsf{O}, & \text{if } (x_i, a_i) \in \{(\mathsf{O}, \mathsf{P}), (\mathsf{O}, \mathsf{S}), (\mathsf{R}_{N^R - 1}, \mathsf{R})\} \\ \mathsf{M}_1, & \text{if } (x_i, a_i) = (\mathsf{O}, \mathsf{M}) \\ \mathsf{M}_{n+1}, & \text{if } (x_i, a_i) \in \{(\mathsf{M}_n, \mathsf{M}) | n \neq N^M\} \\ \mathsf{M}_{N^M}, & \text{if } (x_i, a_i) = (\mathsf{M}_{N^M}, \mathsf{M}) \\ \mathsf{R}_1, & \text{if } (x_i, a_i) = (\mathsf{M}_{N^M}, \mathsf{R}) \\ \mathsf{R}_{n+1}, & \text{if } (x_i, a_i) \in \{(\mathsf{R}_n, \mathsf{R}) | n \neq N^R - 1\} \\ \mathsf{A}, & \text{if } (x_i, a_i) \in \{(\mathsf{O}, \mathsf{A}), (\mathsf{M}_{N^M}, \mathsf{A})\} \end{cases}$$

The transitions among exogenous state are determined by a similar term structure model in §4.7.1. The k-th loading coefficient for period $[i, j)$ and commodity c is denoted as $\sigma^c_{i,j,k}$ where $k \in \{1, 2, ..., K\}$. The randomness of prices is simulated with K mutually independent standard Brownian motions $dW_k, k \in \{1, 2, ..., K\}$, i.e., $dW_k dW_{k'} = 0, k' \neq k$, and $k, k' \in \{1, 2, .., K\}$. So the stochastic process governing the evolution of $F^c(i, j)$ is

$$\frac{dF^c(i,j)}{F^c(i,j)} = \sum_{k=1}^{K} \sigma^c_{i,j,k} dW_k$$

We use the same loading coefficients as Yang et al. (2021), which are calibrated with data from NYMEX. We also specify parameters as Yang et al. (2021) did in their paper. All parameters and their values are shown in the Table 4.4.

Table 4.4: Values of the common parameters.

Parameter	Value	Parameter	Value ($ MM)
I	24 months	$\mathsf{I_M}$	0.5
γ_{C}	0.36 bushel/gallon	$\mathsf{I_R}$	2.5
γ_{N}	0.035 MMBtu/gallon	$\mathsf{C_P}$	2.25
N^{M}	1 month	$\mathsf{C_S}$	0.5208
N^{R}	3 months	$\mathsf{C_M}$	0.02917
Q	8.33 million gallon	S	0.0

We implement the ADMM algorithm using the same basis functions employed by Yang et al. (2021). For each stage $i \in \mathcal{I}$ these functions are (i) one; (ii) $\{F^c_{i,j}, j \in \mathcal{I}_i, c \in \mathcal{C}\}$; (iii) $\{(F^c_{i,j})^2, j \in \mathcal{I}_i, c \in \mathcal{C}\}$; (iv) $\{F^c_{i,j} F^{c'}_{i,j}, j \in \mathcal{I}_i, c, c' \in \mathcal{C}, c \neq c'\}$; and (v)

$\{F^c_{i,j}F^c_{i,j+1}, j \in \mathcal{I}_i \setminus \{I-1\}, c \in \mathcal{C}\}$. These basis functions also satisfy Assumption 3. The conditional expectations of the basis functions of the vector of forward curves are available in Nadarajah and Secomandi (2018b).

Comparisons between ADMM-based approach and BCD method

We compare our approach with the BCD method in Yang et al. (2021). We use the standard cyclic block selection rule for the BCD approach, with the block size and stopping tolerance equal to four and 0.01, respectively. We employ 70,000 samples in PLP, which is the largest number of samples that allows us to apply BCD without facing any memory issue. In this case, PLP has three and ten million variables and constraints.

Table 4.5 shows the comparisons of dual bounds between ADMM and BCD in these instances. The negative sign in front of the ratio suggests that ADMM dual bounds are worse than BCD. Specifically, the ADMM dual bound is, on average, 0.68% worse than the BCD dual bound, with the percentage ranging between 0.41% and 1%.

Table 4.5: Comparison of dual bounds between ADMM and BCD approaches on benchmark instances.

Instance	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
ADMM	19.97	19.45	24.21	25.57	21.97	18.77	16.04	22.40	23.32	20.74	19.12	14.75
BCD	19.90	19.32	23.96	25.45	21.82	18.59	15.92	22.22	23.18	20.62	18.98	14.69
Ratio (%)	-0.50	-0.67	-1.00	-0.47	-0.69	-0.97	-0.75	-0.81	-0.60	-0.58	-0.74	-0.41

Table 4.6: Comparison of lower bounds between ADMM and BCD approaches on benchmark instances.

Instance	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
ADMM	18.58	17.98	22.61	23.99	20.40	17.20	14.62	20.70	21.55	19.15	17.67	13.51
BCD	18.58	17.97	22.76	23.86	20.25	17.18	14.63	20.55	21.50	19.12	17.64	13.49
Ratio (%)	0.00	0.06	-0.66	0.54	0.74	0.12	-0.01	0.73	0.23	0.16	-0.17	0.15

The ADMM lower bounds are slightly better than BCD in all instances except Mar, Jul, and Nov, as reported in Table 4.6. The average improvement on the lower bound is 0.16%. In the worst case (Mar), the ADMM lower bound is 0.66% worse than BCD. In the best case (May), the improvement is 0.74%.

Table 4.7: Optimality Gaps of ADMM and BCD

Instance	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Average
ADMM (%)	7.10	7.56	6.57	6.18	7.15	8.36	8.85	7.59	7.59	7.67	7.74	8.41	7.56
BCD (%)	6.63	6.99	5.01	6.25	7.20	7.61	8.11	7.53	7.26	7.29	6.93	8.19	7.08

Table 4.7 reports the optimality gap, which is one minus the ratio of lower bound to the best known upper bound from these two approaches. The optimality gap for the proposed approach is slightly worse than BCD. The average optimality gap is 7.56% for ADMM and 7.08% for BCD. Based on the results above, we conclude that our approach generates almost the same bounds as the BCD approach, because the difference between the results of ADMM and BCD are always within 1%.

However, our approach uses one order of magnitude less memory and 50% CPU times to generate these results. Table 4.8 reports the average CPU times and the relative memory requirement to BCD. ADMM uses 4% of the memory used by BCD to solve the PLP dual and 15% memory to recover the primal solution. Compared to the 11 hours for the BCD approach, it takes 5 hours for our approach to solving the problem.

Table 4.8: Comparison of the averaged CPU Times and Memory

		PO-ADMM	
	PO-BCD	ADMM	Recovery
CPU Times (hours)	11	1	4
Memory Relative to BCD (%)	100%	4%	15%

New instances

Our approach could solve larger instances that can not be solved by BCD. We increase the stage number (I in Table 4.4) and sample paths (L) to 36 and $150,000$, respectively while keeping all other parameter values the same. The resulting PLP has 30 million constraints and 10 million variables.

Our benchmark is the LSM approach because the problem size exceeds the maximal problem that can be solved by BCD. We use the same basis functions and sample paths for these two approaches.

Figure 4.2: Estimated Lower Bounds as the Percentage of the Estimated Dual Bounds.

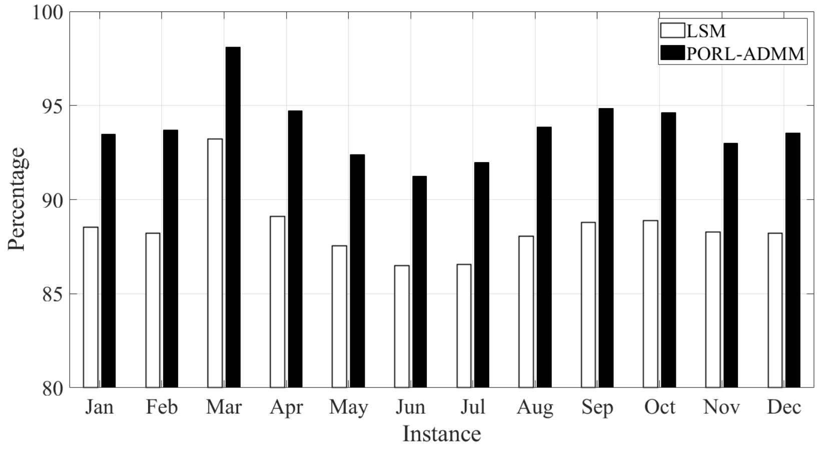


Table 4.9: The estimated lower and upper bounds of 36-stage instances (standard errors in parenthesis)

	PO-ADMM		LSM	
	UB	LB	UB	LB
Jan	54.32 (0.11)	50.78 (0.17)	57.27 (0.03)	50.70 (0.17)
Feb	49.75 (0.11)	46.61 (0.16)	52.76 (0.03)	46.54 (0.17)
Mar	80.28 (0.11)	78.76 (0.22)	84.47 (0.03)	78.75 (0.22)
Apr	60.23 (0.11)	57.05 (0.18)	64.03 (0.03)	57.05 (0.19)
May	51.99 (0.11)	48.04 (0.18)	54.85 (0.03)	48.02 (0.18)
Jun	49.41 (0.11)	45.08 (0.18)	52.09 (0.03)	45.06 (0.18)
Jul	39.96 (0.11)	36.76 (0.18)	42.00 (0.03)	36.36 (0.15)
Aug	55.41 (0.11)	52.01 (0.18)	58.55 (0.03)	51.56 (0.19)
Sep	58.21 (0.11)	55.21 (0.18)	61.70 (0.03)	54.79 (0.20)
Oct	50.1 (0.11)	47.41 (0.18)	53.28 (0.03)	47.35 (0.17)
Nov	47.44 (0.11)	44.12 (0.18)	49.93 (0.03)	44.08 (0.16)
Dec	45.11 (0.11)	42.2 (0.18)	47.83 (0.03)	42.19 (0.16)

Table 4.9 shows the lower and upper bounds of the 36-stage instances with corresponding standard errors in the parenthesis. For all instances, PO-ADMM tighter lower and upper bounds than LSM does. Although the PO-ADMM has larger standard errors for upper bounds than LSM, the absolute values of standard errors relative to the upper bounds never exceed 0.3% across all instances. PO-ADMM improves upper bounds by 5.69% on average with a minimal value 5.10% and a maximal value 6.34%. The lower bounds are also enhanced by 0.01% to 0.87% with an average value of 0.27%. Figure 4.2 shows the percentage gap between the lower and dual bounds for these two approaches. We can see that the PO-ADMM substantially reduces the LSM gap through all instances.

Figure 4.3: Comparison of the estimated LSM- and PO-ADMM-based optimality gaps.

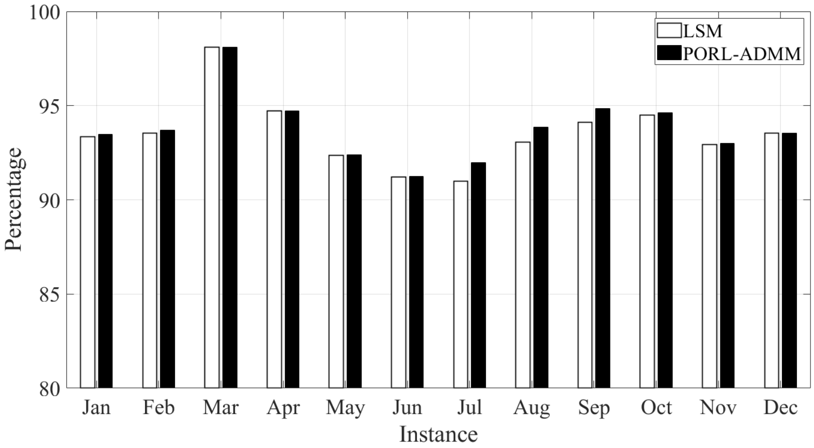


Figure 4.3 reports the optimality gaps for the LSM- and ADMM-based approaches. That is, we use the ADMM upper bounds as the yardstick to evaluate the lower bounds

from these two approaches. In this case, the optimality gaps of LSM reduces from 11-14% to 2-9% with an average value 6%. Thus, both LSM and PO lead to near optimal operating policies, even if PO marginally outperforms LSM in this respect.

4.8 Conclusion

PO has been used to obtain lower and dual bounds for intractable MDPs arising in optimal stopping and merchant energy production. Many financial and real option applications share a similar structure if modeled as MDPs. However, the PO formulation in those context is well defined only when there are terminating decisions in the controllable states. The extant solution method of the LP also exhibits high per-iteration complexities in complicated settings. These two limits restrict the applicability of PO. In this paper, we develop a pseudo action scheme and a new coordinate decomposition and regression approach to deal with these two issues, respectively. The pseudo action scheme extends PO to new applications without the required decisions. Our new solution approach, which employs ADMM to solve the dual of PLP and recovers the primal solution via a least squares regression, substantially reduces the required computational complexities. We also establish the convergence results of the new solution approach. We test the performance of the proposed techniques in the context of natural gas storage and merchant energy production. The numerical results suggest that both techniques are effective. Our technique is potentially relevant to other financial and real options applications.

Chapter 5

Conclusion

Merchant energy operation is an important business application that consists of both the physical operation to maximize the expected total cash flows and financial operations to hedge the risks in the cash flows. This thesis focuses on managing the physical operation of energy conversion assets, which typically gives rise to intractable MDPs due to the intertemporal linkages between decision and the high dimensional state space. We extend PO, a state of the art RL approach of generating lower and upper bounds on the MDPs, from optimal stopping to those applications, and solve the resulting model with novel algorithms based on math programming techniques. In contrast to most RL approaches in the literature, which aim at improving the lower bound, our approach features optimizing the upper bound. We provide theoretical guarantees to support the proposed approaches. We examine the effectiveness of our approaches with realistic energy production and storage application. Our approach generates substantially and slightly better upper and lower bounds than the state of the art approach in the literature. We further provide numerical evidence to show that the extant algorithms generate near optimal lower bounds and loose upper bounds. The techniques in this dissertation have potential relevance to broader application contexts, such as inventory control, portfolio optimization, network revenue management, and assortment problem (Brown and Smith 2021). In §5.1, we summarize the main contributions from each chapter. In §5.2, we discuss directions for future research.

5.1 Summary

This thesis extends PO, an RL approach of optimizing the upper bound, from optimal stopping to merchant energy operations and real option models. The extant methods in the RL and financial engineering literature typically solve this kind of MDPs from its primal side, i.e., improving the lower bound, and then generate upper bounds by the information relaxation techniques. However, those methods do not have any mechanism to guarantee the quality of the upper bound. The performance of those methods relies on trial and error in finding good *heuristic* penalties. In applications such as optimal stopping, multiple stopping, and merchant energy production, there are large gaps between the two bounds. A natural conjecture is that the sizable gap is mainly due to the looseness of the upper bound, but investigations into this question require tightening the bounds. Meanwhile, PO is an emerging RL method that provides an *optimal* dual penalty in the context of optimal stopping. However, the applicability of PO is severely limited due to the difficulty of solving the underlying LP in complicated settings such as energy production and multiple stoppings. It is unclear whether people can prescribe PO as the required benchmark method to the aforementioned conjecture. This thesis provides a positive answer to this question.

We extend PO on both modeling and solution aspects. We first introduce the dual value variables in PO to avoid the exponential increase of constraints in the considered MDP settings. We also identified that PLP is unbounded when there are no terminating decisions in the feasible action set of the MDP. We proposed a novel pseudo action scheme based on nonanticipated policies to fix this issue. The proposed method, though simple, substantially extends PO to broader applications other than optimal stopping and energy production. The lower and dual bounds generated by PO and the pseudo action scheme are competitive to the state of the art approach.

Solving the underlying PLP in PO turns out to be an extremely difficult task. The difficulties are twofolds: (i) PLP is ill conditioned, and (ii) PLP is large scale. We first provided analysis to show that the source of the ill conditioning is the highly correlated coefficient columns in the dual penalty. Although it is impossible to precondition the entire coefficient matrix of the dual penalty, we leveraged the block diagonal structure of the matrix and applied PCA to each of those blocks to orthogonalize the columns. The numerical studies show that PCA effectively remove the ill conditioning and substantially improves the efficiency of the solution methods. Different from the common situation where PCA is used, our use of PCA is to precondition the linear programming. Besides, PCA is very reachable because it is embedded in many off the shelf packages as a standard statistic tool.

Our next contribution was to overcome the size issue in PLP. We first propose the BCD approach to take advantage of the block diagonal structure of the dual penalty. The BCD approach iteratively solves PLP and reduces the computational burden of solving the problem as a whole. The combination of PO with BCD is new. We also use commercial solvers in each BCD iteration to avoid turning parameters such as the step size in the algorithm. Consequently, the algorithm generates solutions with high accuracy in our context. With this approach and the PCA-based preconditioning, we solved PLP for 24-stage ethanol production instances. Compared to standard methods, our approach leads to substantially tighter dual bounds and smaller optimality gaps at the expense of considerably larger computational efforts. Specifically, we provide numerical evidence for the near optimality of the policies based on least squares Monte Carlo and compute slightly better policies on a set of existing benchmark ethanol production instances.

We developed a second approach, i.e., a coordinated decomposition and regression approach, to further reduce the computational complexities of solving PLP. This approach features lower per iteration computational complexity than the BCD approach because it exploits the problem's decomposition structure. The proposed method generates roughly the same solution as the BCD approach to the 24-stage ethanol production instance with much less memory and CPU times. It can also solve 36-stage instances whose PLP is almost one order of magnitude larger than the previous instances. The results for the new instances suggest that the insights for the 24-stage instance still hold for larger instances.

Overall, our findings suggest that PO outperforms LSM in the context of energy production and that the ADMM and regression approach is the most efficient method for solving PLP.

5.2 Future Research

The results in this thesis suggest several methodological and theoretical directions for future research and one application extension. We discuss them below.

First of all, while progress on solving PO has been made, it is still hard to solve PLP for the current application with practical horizons, e.g., 10 years or even 30 years. Much stronger methods are required in those contexts because the resulting PLPs have billions of variables and constraints. Although the issue can somehow be avoided by using stages with different time scales in the MDP, it would be interesting and important to further increase the efficiency of the solution method.

The algorithms proposed in this thesis (BCD in Chapter 2 and ADMM in Chapter 4) belong to high order methods as they utilize more information than first order methods. It is known that the high order method typically converges faster but at a higher computational cost than the first order method. Thus, exploring first order updates that use less information may provide a way to get more scalable PO methods.

The current PO approach requires an additional step to generate a second group of VFAs for the lower bound. It would be interesting to automate this process and develop methods that systematically generate feasible policies from PO. Intuitively, such policies should outperform the feasible policies obtained from solving the primal MDP because they are "corrected" from the policy that utilizes all future information. However, a rigorous explanation based on mathematical analysis is lacking in this aspect.

Exploring PO in a more general MDP context is certainly another promising direction. This thesis only considered MDPs with small and large endogenous and exogenous state spaces and assumes that the decision does not impact the transitions of the exogenous states. It is unclear how to generate high quality upper bounds with PO for MDPs with large endogenous and exogenous state spaces. Besides, it is also interesting to study extending PO to MDPs where the decisions can influence the transitions of the exogenous states.

Broadly speaking, it would be interesting to connect PO to other reinforcement learning approaches. The recent work of Min et al. (2019), and Jiang et al. (2020) provide some ideas along these lines.

Bibliography

Daniel Adelman. Price-directed replenishment of subsets: Methodology and its application to inventory routing. *Manufacturing & Service Operations Management*, 5(4):348–371, 2003.

Daniel Adelman. A price-directed approach to stochastic inventory/routing. *Operations Research*, 52(4):499–514, 2004.

Daniel Adelman. Dynamic bid prices in revenue management. *Operations Research*, 55(4): 647–661, 2007.

Roger Adkins and Dean Paxson. Reciprocal energy-switching options. *Journal of Energy Markets*, 4(1):91–120, 2011.

Yossiri Adulyasak, Jean-François Cordeau, and Raf Jans. Benders decomposition for production routing under demand uncertainty. *Operations Research*, 63(4):851–867, 2015.

Dennis Amelunxen and Peter Burgisser. A coordinate-free condition number for convex programming. *SIAM Journal on Optimization*, 22(3):1029–1041, 2012.

Bancha Ariyajunya, Ying Chen, Victoria CP Chen, Seoung Bum Kim, and Jay Rosenberger. Addressing state space multicollinearity in solving an ozone pollution dynamic control problem. *European Journal of Operational Research*, 289(2):683–695, 2021.

Øystein Arvesen, Vegard Medbø, S-E Fleten, Asgeir Tomasgard, and Sjur Westgaard. Linepack storage valuation under price uncertainty. *Energy*, 52(1):155–164, 2013.

Santiago R Balseiro and David B Brown. Approximations to stochastic dynamic programs via information relaxation duality. *Operations Research*, 67(2):577–597, 2019.

Francisco Barahona and Ranga Anbil. The volume algorithm: producing primal solutions with a subgradient method. *Mathematical Programming*, 87(3):385–399, 2000.

Alexandre Belloni and Robert M Freund. A geometric analysis of Renegar's condition number, and its interplay with conic curvature. *Mathematical Programming*, 119(1):95–107, 2009.

Jacques F Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik*, 4(1):238–252, 1962.

D. P. Bertsekas. *Convex Optimization Algorithms.* Athena Scientific, Belmont, MA, USA, 2015.

Dimitri Bertsekas. *Reinforcement and Optimal Control.* Athena Scientific, 2019.

Dimitris Bertsimas and John N Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena Scientific Belmont, MA, 1997.

Alexander Boogert and Cyriel De Jong. Gas storage valuation using a monte carlo method. *Journal of Derivatives*, 15(3):81, 2008.

Alexander Boogert and Cyriel De Jong. Gas storage valuation using a multifactor price process. *Journal of Energy Markets*, 4(4):29–52, 2011.

Trine Krogh Boomsma, Nigel Meade, and Stein-Erik Fleten. Renewable energy investments under different support schemes: A real options approach. *European Journal of Operational Research*, 220(1):225–237, 2012.

O. Boyabatli, J. Nguyen, and T. Wang. Capacity management in agricultural commodity processing and application in the palm industry. *Manufacturing & Service Operations Management*, 19(4):551–567, 2017.

Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Stephen Boyd, Neal Parikh, and Eric Chu. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

Robert Brelsford. Calumet advances montana refinery's renewable diesel project. *Oil & Gas Journal*, 2021a.

Robert Brelsford. Imperial weighs renewable diesel complex at strathcona refinery. *Oil & Gas Journal*, 2021b.

M. J. Brennan and E. S. Schwartz. Evaluating natural resource investments. *Journal of Business*, 58(2):135–157, 1985.

David B Brown and Martin B Haugh. Information relaxation bounds for infinite horizon markov decision processes. *Operations Research*, 65(5):1355–1379, 2017.

David B Brown and James E Smith. Dynamic portfolio optimization with transaction costs: Heuristics and dual bounds. *Management Science*, 57(10):1752–1770, 2011.

David B Brown and James E Smith. Information relaxations, duality, and convex stochastic dynamic programs. *Operations Research*, 62(6):1394–1415, 2014.

David B Brown and James E Smith. Information relaxations and duality in stochastic dynamic programs: A review and tutorial. 2021.

David B Brown, James E Smith, and Peng Sun. Information relaxations and duality in stochastic dynamic programs. *Operations Research*, 58(4-part-1):785–801, 2010.

Ximing Cai, Daene C McKinney, Leon S Lasdon, and David W Watkins Jr. Solving large nonconvex water resources management models using generalized benders decomposition. *Operations Research*, 49(2):235–245, 2001.

Salvador Perez Canto. Application of benders' decomposition to power plant preventive maintenance scheduling. *European journal of operational research*, 184(2):759–777, 2008.

René Carmona and Michael Ludkovski. Valuation of energy storage: An optimal switching approach. *Quantitative finance*, 10(4):359–374, 2010.

Jacques F Carriere. Valuation of the early-exercise price for options using simulations and nonparametric regression. *Insurance: Mathematics and Economics*, 19(1):19–30, 1996.

Shyam S Chandramouli and Martin B Haugh. A unified approach to multiple stopping and duality. *Operations Research Letters*, 40(4):258–264, 2012.

Shyam Sundar Chandramouli. A convex optimization approach to multiple stopping: Pricing chooser caps and swing options. 2019. Working Paper, IEOR Department, Columbia University, New York, NY, USA.

Caihua Chen, Bingsheng He, Yinyu Ye, and Xiaoming Yuan. The direct extension of admm for multi-block convex minimization problems is not necessarily convergent. *Mathematical Programming*, 155(1-2):57–79, 2016.

Nan Chen, Xiang Ma, Yanchu Liu, and Wei Yu. Information relaxation and a duality-driven algorithm for stochastic dynamic programs. *arXiv preprint arXiv:2007.14295*, 2020.

Dennis Cheung and Felipe Cucker. A new condition number for linear programming. *Mathematical Programming*, 91(1):163–174, 2001.

Les Clewlow and Chris Strickland. *Energy Derivatives: Pricing and Risk Management.* Lacima Publications, London, England, UK, 2000.

Jean-François Cordeau, Goran Stojković, François Soumis, and Jacques Desrosiers. Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transportation science*, 35(4):375–388, 2001.

G. Cortazar and E. S. Schwartz. The valuation of commodity contingent claims. *Journal of Derivatives*, 1(4):27–39, 1994.

Gonzalo Cortazar, Miguel Gravet, and Jorge Urzua. The valuation of multidimensional American real options using the LSM simulation method. *Computers & Operations Research*, 35(1):113–129, 2008.

Alysson M Costa. A survey on benders decomposition applied to fixed-charge network design problems. *Computers & operations research*, 32(6):1429–1450, 2005.

Jean-François Côté, Mauro Dell'Amico, and Manuel Iori. Combinatorial benders' cuts for the strip packing problem. *Operations Research*, 62(3):643–661, 2014.

Teodor Gabriel Crainic, Mike Hewitt, Francesca Maggioni, and Walter Rei. Partial benders decomposition: general methodology and application to stochastic network design. *Transportation Science*, 55(2):414–435, 2021.

Daniela Pucci De Farias and Benjamin Van Roy. The linear programming approach to approximate dynamic programming. *Operations research*, 51(6):850–865, 2003.

Daniela Pucci De Farias and Benjamin Van Roy. On constraint sampling in the linear programming approach to approximate dynamic programming. *Mathematics of operations research*, 29(3):462–478, 2004.

Michel Denault, Jean-Guy Simonato, and Lars Stentoft. A simulation-and-regression approach for stochastic dynamic programs with endogenous state variables. *Computers & Operations Research*, 40(11):2760–2769, 2013.

Vijay V Desai, Vivek F Farias, and Ciamac C Moallemi. Approximate dynamic programming via a smoothed linear program. *Operations Research*, 60(3):655–674, 2012a.

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

Sripad K Devalkar, Ravi Anupindi, and Amitabh Sinha. Integrated optimization of procurement, processing, and trade of commodities. *Operations Research*, 59(6):1369–1381, 2011.

A. K. Dixit and R. S. Pindyck. *Investment under Uncertainty.* Princeton University Press, Princeton, NJ, USA, 1994.

Lingxiu Dong, Panos Kouvelis, and Xiaole Wu. The value of operational flexibility in the presence of input and output price uncertainties with oil refining applications. *Management Science*, 60(12):2908–2926, 2014.

Ibrahim El Shar and Daniel Jiang. Lookahead-bounded q-learning. In *International Conference on Machine Learning*, pages 8665–8675. PMLR, 2020.

Paul Enders, Alan Scheller-Wolf, and Nicola Secomandi. Interaction between technology and extraction scaling real options in natural gas production. *IIE Transactions*, 42(9):643–655, 2010.

Marina Epelman and Robert M Freund. A new condition measure, preconditioners, and relations between different measures of conditioning for conic linear systems. *SIAM Journal on Optimization*, 12(3):627–655, 2002.

A. Eydeland and K. Wolyniec. *Energy and Power Risk Management: New Developments in Modeling, Pricing, and Hedging.* John Wiley & Sons, Inc., Hoboken, NJ, USA, 2003.

Matteo Fischetti, Ivana Ljubić, and Markus Sinnl. Benders decomposition without separability: A computational study for capacitated facility location problems. *European Journal of Operational Research*, 253(3):557–569, 2016.

Matteo Fischetti, Ivana Ljubić, and Markus Sinnl. Redesigning benders decomposition for large-scale facility location. *Management Science*, 63(7):2146–2162, 2017.

H. Geman. *Commodities and Commodity Derivatives: Modeling and Pricing for Agriculturals, Metals and Energy.* John Wiley & Sons Ltd, Chichester, England, UK, 2005.

Arthur M Geoffrion. Generalized benders decomposition. *Journal of optimization theory and applications*, 10(4):237–260, 1972.

Paul Glasserman and Bin Yu. Simulation for american options: Regression now or regression later? In *Monte Carlo and Quasi-Monte Carlo Methods 2002*, pages 213–226. Springer, 2004.

Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

Luigi Grippo and Marco Sciandrone. On the convergence of the block nonlinear Gauss–Seidel method under convex constraints. *Operations Research Letters*, 26(3):127–136, 2000.

LLC Gurobi Optimization. Gurobi optimizer reference manual, 2021. URL http://www.gurobi.com.

Emil Gustavsson, Michael Patriksson, and Ann-Brith Strömberg. Primal convergence from dual subgradient methods for convex optimization. *Mathematical Programming*, 150(2): 365–390, 2015.

Graeme Guthrie. *Real Options in Theory and Practice.* Oxford University Press, New York, NY, USA, 2009.

Lajos Gergely Gyurkó, Ben M Hambly, and Jan Hendrik Witte. Monte carlo methods via a dual approach for some discrete time stochastic control problems. *Mathematical Methods of Operations Research*, 81(1):109–135, 2015.

Martin B Haugh and Leonid Kogan. Pricing american options: a duality approach. *Operations Research*, 52(2):258–270, 2004.

David C Heath and Peter L Jackson. Modeling the evolution of demand forecasts ith application to safety stock analysis in production/distribution systems. *IIE transactions*, 26(3):17–30, 1994.

Juri Hinz and Jeremy Yee. Optimal forward trading and battery control under renewable electricity generation. *Journal of Banking & Finance*, 95(October):244–254, 2018.

John N Hooker. Planning and scheduling by logic-based benders decomposition. *Operations research*, 55(3):588–602, 2007.

John N Hooker and Greger Ottosson. Logic-based benders decomposition. *Mathematical Programming*, 96(1):33–60, 2003.

Tetsuo Iida and Paul H Zipkin. Approximate solutions of a dynamic forecast-inventory model. *Manufacturing & Service Operations Management*, 8(4):407–425, 2006.

Patrick Jaillet, Ehud I Ronn, and Stathis Tompaidis. Valuation of commodity-based swing options. *Management science*, 50(7):909–921, 2004.

Daniel R Jiang, Lina Al-Kanj, and Warren B Powell. Optimistic monte carlo tree search with sampled information relaxation dual bounds. *Operations Research*, 68(6):1678–1697, 2020.

Ian T. Jolliffe. *Principal Component Analysis.* Springer-Verlag, New York, NY, USA, second edition, 2002.

Ed Klotz and Alexandra M Newman. Practical guidelines for solving difficult mixed integer linear programs. *Surveys in Operations Research and Management Science*, 18(1-2):18–32, 2013.

G. Lai, M. X. Wang, S. Kekre, A. Scheller-Wolf, and N. Secomandi. Valuation of storage at a liquefied natural gas terminal. *Operations Research*, 59(3):602–616, 2011.

Guoming Lai, François Margot, and Nicola Secomandi. An approximate dynamic programming approach to benchmark practice-based heuristics for natural gas storage valuation. *Operations research*, 58(3):564–582, 2010.

Torbjörn Larsson, Michael Patriksson, and Ann-Brith Strömberg. Ergodic, primal convergence in dual subgradient schemes for convex programming. *Mathematical programming*, 86(2): 283–312, 1999.

Yantong Li, Jean-François Côté, Leandro Callegari-Coelho, and Peng Wu. Novel formulations and logic-based benders decomposition for the integrated parallel machine scheduling and location problem. *INFORMS Journal on Computing*, 2021.

Qihang Lin, Selvaprabu Nadarajah, and Negar Soheili. Revisiting approximate linear programming: Constraint-violation learning with applications to inventory control and energy storage. *Management Science*, 66(4):1544–1562, 2020.

Francis A Longstaff and Eduardo S Schwartz. Valuing American options by simulation: A simple least-squares approach. *Review of Financial studies*, 14(1):113–147, 2001.

Curtiss Luong. *An examination of Benders' decomposition approaches in large-scale healthcare optimization problems*. University of Toronto (Canada), 2015.

Thomas L Magnanti and Richard T Wong. Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. *Operations research*, 29(3):464–484, 1981.

Thomas L Magnanti, Robert W Simpson, et al. Transportation network analysis and decomposition methods. Technical report, United States. Dept. of Transportation. Research and Special Programs . . . , 1978.

Arthur Maheo, Philip Kilby, and Pascal Van Hentenryck. Benders decomposition for the design of a hub and shuttle public transit system. *Transportation Science*, 53(1):77–88, 2019.

P Mahey, Henrique Pacca Loureiro Luna, and CD Randazzo. Benders decomposition for local access network design with two technologies. *Discrete Mathematics & Theoretical Computer Science*, 4, 2001.

Alan S Manne. Linear programming and sequential decisions. *Management Science*, 6(3): 259–267, 1960.

Anne Mercier and François Soumis. An integrated aircraft routing, crew scheduling and flight retiming model. *Computers & Operations Research*, 34(8):2251–2265, 2007.

Anne Mercier, Jean-François Cordeau, and François Soumis. A computational study of benders decomposition for the integrated aircraft routing and crew scheduling problem. *Computers & Operations Research*, 32(6):1451–1476, 2005.

Thomas V Mikosch, J Wright Stephen, and Nocedal Jorge. *Numerical Optimization*. Springer New York, 2006.

Seungki Min, Costis Maglaras, and Ciamac C Moallemi. Thompson sampling with information relaxation penalties. *Advances in Neural Information Processing Systems*, 32, 2019.

José Ignacio Muñoz, Javier Contreras, J Caamaño, and PF Correia. A decision-making tool for project investments based on real options: the case of wind power generation. *Annals of Operations Research*, 186(1):465, 2011.

Selvaprabu Nadarajah. Approximate dynamic programming for commodity and energy merchant operations. 2014.

Selvaprabu Nadarajah and Nicola Secomandi. Regress-later least squares monte carlo: Duality perspective and energy real option application. 01 2015.

Selvaprabu Nadarajah and Nicola Secomandi. Relationship between least squares monte carlo and approximate linear programming. *Operations Research Letters*, 45(5):409–414, 2017.

Selvaprabu Nadarajah and Nicola Secomandi. Merchant energy trading in a network. *Operations Research*, 66(5):1304–1320, 2018a.

Selvaprabu Nadarajah and Nicola Secomandi. Least squares monte carlo and approximate linear programming: Error bounds and energy real option application. *Available at SSRN 3232687*, 2018b.

Selvaprabu Nadarajah and Nicola Secomandi. Least squares monte carlo and approximate linear programming with an energy real option application. *Foundations and Trends® in Technology, Information and Operations Management*, 14(1–2):178–202, 2020.

Selvaprabu Nadarajah, François Margot, and Nicola Secomandi. Relaxations of approximate linear programs for the real option management of commodity storage. *Management Science*, 61(12):3054–3076, 2015.

Selvaprabu Nadarajah, François Margot, and Nicola Secomandi. Comparison of least squares Monte Carlo methods with applications to energy real options. *European Journal of Operational Research*, 256(1):196–204, 2017.

Angelia Nedić and Asuman Ozdaglar. Approximate primal solutions and rate analysis for dual subgradient methods. *SIAM Journal on Optimization*, 19(4):1757–1780, 2009.

Yu Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

Yu Nesterov and Vladimir Shikhman. Dual subgradient method with averaging for optimal resource allocation. *European Journal of Operational Research*, 270(3):907–916, 2018.

Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

Oil & Gas Journal. Kmi, neste partner on louisiana renewables storage. *Oil & Gas Journal*, 2021.

Javier Peña, Vera Roshchina, and Negar Soheili. Some preconditioners for systems of linear inequalities. *Optimization Letters*, 8(7):2145–2152, 2014.

Warren B Powell. *Approximate Dynamic Programming: Solving the curses of dimensionality*, volume 703. John Wiley & Sons, 2007.

Ragheb Rahmaniani, Teodor Gabriel Crainic, Michel Gendreau, and Walter Rei. The benders decomposition algorithm: A literature review. *European Journal of Operational Research*, 259(3):801–817, 2017.

James Renegar. Incorporating condition measures into the complexity theory of linear programming. *SIAM Journal on Optimization*, 5(3):506–524, 1995a.

James Renegar. Linear programming, complexity theory and elementary functional analysis. *Mathematical Programming*, 70(1-3):279–351, 1995b.

Renewable Fuels Association. 2021 Ethanol Industry Outlook. Technical report, 07 2021. URL https://ethanolrfa.org/wp-content/uploads/2021/02/RFA_Outlook_2021_fin_low.pdf.

Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38, 2014.

Leonard CG Rogers. Monte carlo valuation of american options. *Mathematical Finance*, 12(3): 271–286, 2002.

Frode Rømo, Asgeir Tomasgard, Lars Hellemo, Marte Fodstad, Bjørgulf Haukelidsæter Eidesen, and Birger Pedersen. Optimizing the Norwegian natural gas production and transport. *INFORMS Journal on Applied Analytics*, 39(1):46–56, 2009.

RWH Sargent and DJ Sebastian. On the convergence of sequential minimization algorithms. *Journal of Optimization Theory and Applications*, 12(6):567–575, 1973.

N. Secomandi. Approximations for high dimensional commodity and energy merchant operations models. *Foundations and Trends in Technology, Information and Operations Management*, 11(1-3):144–164, 2017.

Nicola Secomandi. Optimal commodity trading with a capacitated storage asset. *Management Science*, 56(3):449–467, 2010.

Nicola Secomandi and Duane J Seppi. Real options and merchant operations of energy and other commodities. *Foundations and Trends in Technology, Information and Operations Management*, 6(3–4):161–331, 2014.

Nicola Secomandi and Duane J. Seppi. Energy real options: Valuation and operations. In V. Kaminski, editor, *Managing Energy Price Risk*, pages 449–477. Risk Books, London, England, UK, fourth edition, 2016.

Nicola Secomandi, Guoming Lai, François Margot, Alan Scheller-Wolf, and Duane J Seppi. Merchant commodity storage and term-structure model error. *Manufacturing & Service Operations Management*, 17(3):302–320, 2015a.

Nicola Secomandi, Guoming Lai, François Margot, Alan Scheller-Wolf, and Duane J Seppi. Merchant commodity storage and term-structure model error. *Manufacturing & Service Operations Management*, 17(3):302–320, 2015b.

Hanif D Sherali and Gyunghyun Choi. Recovery of primal solutions when using subgradient optimization methods to solve lagrangian duals of linear programs. *Operations Research Letters*, 19(3):105–113, 1996.

S. E. Shreve. *Stochastic Calculus for Finance II: Continuous-Time Models*. Springer, New York, NY, USA, 2004.

James E. Smith. Alternative approaches for solving real-options problems: (comment on brandão et al. 2005). *Decision Analysis*, 2(2):89–102, 2005.

James E. Smith and K. F. McCardle. Valuing oil properties: Integrating option pricing and decision analysis approaches. *Operations Research*, 46(2):198–217, 1998.

James E Smith and Kevin F McCardle. Options in the real world: Lessons learned in evaluating oil and gas investments. *Operations Research*, 47(1):1–15, 1999.

Min Tao and Xiaoming Yuan. On the o(1/t) convergence rate of alternating direction method with logarithmic-quadratic proximal regularization. *SIAM Journal on optimization*, 22(4): 1431–1448, 2012.

Matt Thompson. Optimal economic dispatch and risk management of thermal power plants in deregulated markets. *Operations Research*, 61(4):791–809, 2013.

Lenos Trigeorgis et al. *Real options: Managerial flexibility and strategy in resource allocation.* MIT press, 1996.

Alessio Trivella, Selvaprabu Nadarajah, Stein-Erik Fleten, Denis Mazieres, and David Pisinger. Managing shutdown decisions in merchant commodity and energy production: A social commerce perspective. *Manufacturing & Service Operations Management, Forthcoming*, 2019.

Chung Li Tseng and Graydon Barz. Short-term generation asset valuation: A real options approach. *Operations Research*, 50(2):297–310, 2002.

Chung Li Tseng and Kyle Y Lin. A framework using two-factor price lattices for generation asset valuation. *Operations Research*, 55(2):234–251, 2007.

Paul Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109(3):475–494, 2001.

John N Tsitsiklis and Benjamin Van Roy. Regression methods for pricing complex american-style options. *IEEE Transactions on Neural Networks*, 12(4):694–703, 2001.

Sinong Wang and Ness Shroff. A new alternating direction method for linear programming. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1479–1487, 2017.

Yang Wang, Brendan O'Donoghue, and Stephen Boyd. Approximate dynamic programming via iterated bellman inequalities. *International Journal of Robust and Nonlinear Control*, 25(10):1472–1496, 2015.

Bo Yang, Selvaprabu Nadarajah, and Nicola Secomandi. Least squares monte carlo and pathwise optimization for merchant energy production. *Available at SSRN 3900797*, 2021.

Dan Zhang and Daniel Adelman. An approximate dynamic programming approach to network revenue management with customer choice. *Transportation Science*, 43(3):381–394, 2009.

Joyce Li Zhang and K Ponnambalam. Hydro energy management optimization in a deregulated electricity market. *Optimization and Engineering*, 7(1):47–61, 2006.

Y. Zhou, A. Scheller-Wolf, N. Secomandi, and S. J. Smith. Managing wind-based electricity generation in the presence of storage and transmission capacity. *Production and Operations Management*, 28(4):970–989, 2019.

Appendix A

Supplement for Chapter 2

A.1 Block-diagonal Structure of G and PCA

$$G := \begin{bmatrix} G_{1,\mathsf{O}} & 0 & \cdots & 0 \\ 0 & G_{1,\mathsf{M}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & G_{I-2,\mathsf{M}} \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \qquad G_{i,\mathsf{M}} := \begin{bmatrix} C_i \\ C_i \end{bmatrix} \qquad G_{i,\mathsf{O}} := \begin{bmatrix} C_i \\ C_i \\ C_i \end{bmatrix}$$

$$C_i := \begin{bmatrix} \Delta^{\mathbb{E},1}_{i-1}\phi_{i,1} & \Delta^{\mathbb{E},1}_{i-1}\phi_{i,2} & \ldots & \Delta^{\mathbb{E},1}_{i-1}\phi_{i,B_i} \\ \Delta^{\mathbb{E},2}_{i-1}\phi_{i,1} & \Delta^{\mathbb{E},2}_{i-1}\phi_{i,2} & \ldots & \Delta^{\mathbb{E},2}_{i-1}\phi_{i,B_i} \\ \vdots & \vdots & \ddots & \vdots \\ \Delta^{\mathbb{E},L'}_{i-1}\phi_{i,1} & \Delta^{\mathbb{E},L'}_{i-1}\phi_{i,2} & \ldots & \Delta^{\mathbb{E},L'}_{i-1}\phi_{i,B_i} \end{bmatrix}$$

Figure A.1: Structure of the G matrix.

The block diagonal structure of G holds because MDP (2.2) has a finite number of endogenous states and an associated deterministic transition function $f(x_i, a_i)$. Figure A.1 illustrates this structure. For each pair $(i, x_i) \in \mathcal{I} \setminus \{0, I-1\} \times \mathcal{X}'$, we define the set of tuples $\mathcal{T}_i(x_i) := \{(l, x_{i-1}, a_{i-1}) | (l, x_{i-1}, a_{i-1}) \in \mathcal{L}' \times \mathcal{X}' \times \mathcal{A}(x_{i-1}), f(x_{i-1}, a_{i-1}) = x_i\}$ to describe it. The (i, x_i)-th block of G, G_{i,x_i}, includes the columns associated with the triples (i, x_i, b)'s for each $b \in \mathcal{B}_i$ and the rows corresponding to the tuples (l, x_{i-1}, a_{i-1})'s in set $\mathcal{T}_i(x_i)$. Since the basis functions do not depend on a_{i-1} and x_{i-1}, the matrix G_{i,x_i} duplicates a matrix C_i that has B_i columns and L' rows, with its entry in column $b \in \mathcal{B}_i$ and row $l \in \mathcal{L}'$ is equal to $\Delta^{\mathbb{E},l}_{i-1}\phi_{i,b}$. The number of copies of C_i in G_{i,x_i} equals the number of the pairs $(x_{i-1}, a_{i-1}) \in \mathcal{X}' \times \mathcal{A}(x_{i-1})$ that satisfy $f(x_{i-1}, a_{i-1}) = x_i$. The 0 entries below $G_{I-2,\mathrm{M}}$ in G correspond to constraints that have no penalties.

Exploiting this structure, at each stage $i \in \mathcal{I} \setminus \{0, I-1\}$, we perform PCA on the matrix C_i and use the resulting pre-conditioned matrix to construct as follows a matrix analogous to G that also has block-diagonal structure but differs because it has orthogonal columns. Define W_i to be the square $B_i \times B_i$ matrix with columns equal to the eigenvectors of $C_i^\top C_i$. Let $G^\perp_{i,x_i}$ be the analogue of G_{i,x_i} with C_i replaced by $C_i^\perp := C_i W_i$, which has orthogonal columns by definition. Similarly, $G^\perp$ shares the block-diagonal structure of G with each block G_{i,x_i} replaced by $G^\perp_{i,x_i}$, which has orthogonal columns as it is composed of row copies of $C_i^\perp$.

A.2 Interpretation of P2LP using Modified Basis Functions

We establish the equivalence of P2LP and a PLP formulated with modified basis functions. These basis functions are defined as $\phi^W_{i,b}(F_i) := \sum_{b' \in \mathcal{B}_i} W_{i,b',b}\phi_{i,b}(F_i)$ for all $(i,b) \in \mathcal{I} \setminus \{i, ..., I-1\} \times \mathcal{B}_i$. That is, each basis function $\phi^W_{i,b}$ is a linear combination of basis functions $(\phi_{i,b}, b \in \mathcal{B}_i)$ using the weights in the b-th column of matrix W_i. Evaluating the VFA constructed using basis functions $(\phi_{i,b}, b \in \mathcal{B}_i)$ and VFA weights β^W at stage i and state (x_i, F_i) is equivalent to evaluating a modified VFA that is based on $(\phi^W_{i,b}, b \in \mathcal{B}_i)$ and β:

$$\begin{aligned}\sum_{b \in \mathcal{B}_i} \phi_{i,b}(F_i)\beta^W_{i,x_i,b} &= \sum_{b \in \mathcal{B}_i} \phi_{i,b}(F_i) \sum_{b' \in \mathcal{B}_i} W_{i,b,b'}\beta_{i,x_i,b'} \\ &= \sum_{b' \in \mathcal{B}_i} \left(\sum_{b \in \mathcal{B}_i} \phi_{i,b}(F_i) W_{i,b,b'} \right) \beta_{i,x_i,b'} \\ &= \sum_{b' \in \mathcal{B}_i} \phi^W_{i,b'}(F_i)\beta_{i,x_i,b'}.\end{aligned}$$

The first equality uses the definition of β^W, the second re-arranges the two summations and groups terms, and the third uses the definition of $\phi^W_{i,b}(F_i)$.

A.3 Proofs

Proof of Proposition 1. Let $U_i^0(x_i, F^l)$ be the value function for stage i and state x_i of the dynamic program (2.11) for sample path l formulated with the β vector set equal to zero. Denote by $U_0^0(x_0, F^l)$ the corresponding value of $U_0^\beta(x_0, F^l)$. This term is finite because it is the discounted sum of bounded rewards from the initial stage through the final one along the given sample path. The pair (β, U) associated with this particular choice of β vector and the resulting U vector is a feasible PLP solution with finite objective function value. Thus, the optimal value of the PLP objective function is bounded from above. The PLP constraint for each tuple (l, i, x_i, A) is $u_{i,x_i,l} \geq r(x_i, s_i^l, \mathsf{A})$. The right hand side of this inequality evaluates to 0 or S when x_i equals A or it belongs to $\mathcal{X}'$. It follows that the optimal PLP objective function value is bounded from below.

Suppose that all optimal PLP solutions have at least one infinite element of their corresponding β vector. That is, $\beta_{i,x_i,b} = \infty$ for some $(i, x_i, b) \in \mathcal{I} \setminus \{0\} \times \mathcal{X}'_i \times \mathcal{B}_i$. Pick an arbitrary optimal PLP solution (β^*, u^*). Let $(0, u^{(0)})$ be the PLP solution used to establish that the optimal PLP objective function value is bounded from above. In particular, it is a basic feasible solution for PLP. Consider a sequence of such solutions that starts from $(0, u^{(0)})$, ends at (β^*, u^*), and includes as additional elements, if any, points that belong to the boundary of the PLP feasible set. The points generated by pivots in the simplex method starting from the former solution and ending at the latter one is an example of such a sequence. The penultimate item of this sequence is a vertex (β', u') that has an extreme ray connecting it to (β^*, u^*). Such an extreme ray exists because the former solution is finite while the latter one (by assumption) has at least one infinite element. If (β', u') was not finite, the sequence would end at this solution and not reach (β^*, u^*). In addition, since the optimal PLP objective value is bounded from both below and above, the objective function gradient along the extreme ray must be zero. It follows that (β', u') is a finite optimal PLP solution, which contradicts the assumption that all optimal PLP solutions have at least one infinite element. Thus, PLP has at least one finite optimal solution.□

Proof of Proposition 1. Matrix D has a column for each variable $u_{i,x_i,l}$. Consider the PLP constraints (2.16) indexed by tuples of the form (i, x_i, A, l), that is, those that correspond to an abandon action. For each triple (i, x_i, l), the variable $u_{i,x_i,l}$ appears in the left hand side of exactly one such constraint and the right hand side of this constraint has no variables from the vector u. Therefore, the submatrix of D defined by the aforementioned tuples (with some possible row re-arrangement) is a square identity matrix I, which allows us to write

$$D = \begin{bmatrix} D_1 \\ I \end{bmatrix}, \tag{A.1}$$

where D_1 corresponds to the constraints (2.16) associated with non-abandonment decisions. It follows immediately that D is full column rank.

Next, suppose G is full column rank. Based on the representation of D in (A.1), we have

$$\begin{bmatrix} D & G \end{bmatrix} = \begin{bmatrix} D_1 & G \\ I & 0 \end{bmatrix}$$

because the abandonment decision has no associated dual penalty. By performing row operations on this matrix using the submatrix $[I \;\; 0]$, which preserve rank, we obtain

$$\operatorname{rank}\left(\begin{bmatrix} D_1 & G \\ I & 0 \end{bmatrix}\right) = \operatorname{rank}\left(\begin{bmatrix} 0 & G \\ I & 0 \end{bmatrix}\right).$$

All columns in the matrix obtained from the row operations are linearly independent because I and G are both full column rank. Hence, $[D \;\; G]$ is full column rank. The matrix $[D \;\; G]^\top [D \;\; G]$ is also full rank because the rank of any matrix Y and its Gram matrix $Y^\top Y$ are equal, which follows from the equivalence between the null spaces of the former and latter matrices. □

We use Lemma 4 in the proofs of propositions 2 and 4.

Lemma 4. *$k_1^D \leq \sqrt{M_u N_D}$ and $k_{N_D}^D \geq 1$.*

Proof of Lemma 4. Let M_D denote the number of rows of D, $d_{m,n}$ the entry in row m and column n of this matrix, and d_n its n-th column. From equivalent definitions of the Frobenius norm $\|D\|_F$ of D we have

$$\|D\|_F = \sqrt{\sum_{n=1}^{N_D} (\kappa_n^D)^2} = \sqrt{\sum_{n=1}^{N_D} \sum_{m=1}^{M_D} d_{m,n}^2} = \sqrt{\sum_{n=1}^{N_D} \|d_n\|_2^2},$$

where $\|\cdot\|_2$ represents the Euclidean norm. We can then bound the largest singular value of D as follows:

$$\kappa_1^D \leq \sqrt{\sum_{n=1}^{N_D} (\kappa_n^D)^2} = \sqrt{\sum_{n=1}^{N_D} \|d_n\|_2^2} \leq \sqrt{N_D \max_{n \in \{1,2,\ldots,N_D\}} \|d_n\|_2^2}. \tag{A.2}$$

Since column d_n corresponds to a particular variable in the vector u and its entries are either 0, $-\delta$ (> -1), or 1, inequality $\|d_n\|_2^2 \leq M_u$ holds for any n. Combining this inequality with (A.2) establishes $\kappa_1^D \leq \sqrt{N_D M_u}$.

Next we move to lower bounding the smallest singular of D. From the representation of D in (A.1) we have

$$z^T D^T D z = z^\top \begin{bmatrix} D_1 \\ I \end{bmatrix}^\top \begin{bmatrix} D_1 \\ I \end{bmatrix} z = z^\top D_1^\top D_1 z + z^\top z = \|Dz\|_2^2 + \|z\|_2^2 \geq \|z\|_2^2.$$

We then apply this inequality with the definition of the smallest singular to obtain

$$(\kappa_{N_D}^D)^2 = \min_{\|z\|_2 = 1} z^T D^T D z \geq \|z\|_2^2 = 1,$$

which shows that $\kappa_{N_D}^D \geq 1$. □

Proof of Proposition 2 (i) Since the singular values of $D^\top D$ are squares of the singular values of D, we have $\mathrm{cond}(D^\top D) = (\mathrm{cond}(D))^2 = (\kappa_1^D/\kappa_{N_D}^D)^2$. In addition, using the bounds from Lemma 4, it follows that $(\kappa_1^D/\kappa_{N_D}^D)^2 \leq M_u N_D$.

(ii) The inequality $\sum_{n=1}^N \kappa_n^2 \geq 1$ can be established using the following sequence of relationships:

$$\sum_{n=1}^N \kappa_n^2 \geq \kappa_1^2 \geq (\kappa_1^D)^2 \geq (\kappa_{N_D}^D)^2 \geq 1,$$

where the second inequality is due to the largest singular value of a matrix (weakly) increasing as columns are added (Golub and Van Loan 2013, Page 78), and the fourth from Lemma 4. The lower bound on the condition number of $[D\ G]^\top[D\ G]$ is the maximum of two terms. We proceed by showing that each term is a valid lower bound.

The first term we consider is $\mathrm{cond}(D^\top D)$. The validity of the first lower bound term holds because

$$\mathrm{cond}([D\ G]^\top[D\ G]) = (\mathrm{cond}([D\ G]))^2 \geq (\mathrm{cond}(D))^2 = \mathrm{cond}(D^\top D),$$

where the inequality is a consequence of the condition number of a matrix always increasing as more columns are added (Golub and Van Loan 2013, Page 78).

Now we show the validity of the remaining term in the lower bound as follows:

$$\begin{aligned}
\frac{1/N \sum_{n=1}^N \kappa_n^2}{(\Pi_{n=1}^{N_D}(\kappa_n^D)^2 \Pi_{n=1}^{N_G}(\kappa_n^G)^2)^{1/N}} &= \frac{1/N \sum_{n=1}^N \kappa_n^2}{|\det(D^T D)\det(G^T G)|^{1/N}} \\
&\leq \frac{1/N \sum_{n=1}^N \kappa_n^2}{|\det([D\ G]^\top[D\ G])|^{1/N}} \\
&= \frac{1/N \sum_{i=n}^N \kappa_n^2}{(\Pi_{n=1}^N \kappa_n^2)^{1/N}} \\
&\leq \frac{1/N \sum_{n=1}^N \kappa_1^2}{(\Pi_{n=1}^N \kappa_N^2)^{1/N}} \\
&= \frac{\kappa_1^2}{\kappa_N^2} \\
&= \mathrm{cond}([D\ G]^\top[D\ G]).
\end{aligned}$$

The first and second equalities follow from the absolute value of the determinant of a matrix being equal to the product of the squares of its singular values, and the first and second inequalities hold because of Fischer's inequality (Zhang 2011, Theorem 7.11) and $\kappa_N \leq \kappa_n$ along with $\kappa_1 \geq \kappa_n$, respectively. □

Proof of Proposition 3 As outlined in §2.5.1 and discussed in more detail in A.1, the PCA matrix W can be constructed using submatrices W_i of size $B_i \times B_i$ for each

$i \in \mathcal{I} \setminus \{0, I-1\}$. Pick a feasible PLP solution (β, u). Define β' as the vector with (i, x_i) component β'_{i,x_i} equal to $W_i^{-1}\beta_{i,x_i}$. Evaluating the left hand sides of the PLP constraints and the P2LP ones at (β, u) and (β', u), respectively, yields the same values. An analogous result holds for the feasible P2LP solution (β, u) and the PLP one (β', u) for which the (i, x_i) part β'_{i,x_1} of the vector β' is $W_i\beta_{i,x_i}$. That is, there is a one to one mapping between the respective sets of PLP and P2LP feasible solutions. PLP and P2LP have the same objective function. Thus, their optimal solution sets coincide. □

Proof of Proposition 4 Since $\text{cond}([D\ G^\perp]^\top[D\ G^\perp]) = (\kappa_1/\kappa_N)^2$, we obtain the required upper bound on this quantity by finding upper and lower bounds on κ_1 and κ_N, respectively, in this order.

Let $Y = [D\, G^\perp]^\top$. Suppose $\|\cdot\|$ is the two norm. We have

$$\begin{aligned}
\|Yz\|^2 &= \left\| \begin{bmatrix} D^\top \\ (G^\perp)^\top \end{bmatrix} z \right\|^2 \\
&= \left\|D^\top z\right\|^2 + \left\|(G^\perp)^\top z\right\|^2 \\
&\leq \left((\kappa_1^D)^2 + (\kappa_1^{G^\perp})^2 \right) \|z\|^2 ,
\end{aligned}$$

where $\kappa_1^{G^\perp}$ is the largest singular value of $G^\perp$. The last inequality is implied by the following definitions of maximum singular values:

$$\kappa_1^D = \max_{z \neq 0} \frac{\left\|D^\top z\right\|}{\|z\|}, \text{ and } \kappa_1^{G^\perp} = \max_{z \neq 0} \frac{\left\|(G^\perp)^\top z\right\|}{\|z\|}.$$

Applying an analogous definition to matrix Y gives

$$\kappa_1^Y = \max_{z \neq 0} \frac{\|Yz\|}{\|z\|} \leq \sqrt{(\kappa_1^D)^2 + (\kappa_1^{G^\perp})^2}.$$

The largest singular values of Y and $Y^\top = [D\ \ G^\perp]$ are the same since the transpose operator does not alter singular values. Therefore, we have $\kappa_1^2 = (\kappa_1^Y)^2 \leq (\kappa_1^D)^2 + (\kappa_1^{G^\perp})^2$.

For lower bounding κ_N, consider

$$Y^1 = \begin{bmatrix} D^\top D & 0 \\ 0 & (G^\perp)^\top(G^\perp) \end{bmatrix}, \; Y^2 = \begin{bmatrix} 0 & D^\top(G^\perp) \\ (G^\perp)^\top D & 0 \end{bmatrix}.$$

Clearly, $[D\ (G^\perp)]^\top[D\ (G^\perp)] = Y^1 + Y^2$. Then by Theorem 8.13 in Zhang (2011), we have

$$\kappa_N \geq \kappa_N^{Y^1} + \kappa_N^{Y^2} \geq \kappa_N^{Y^1}$$

and because Y^1 is block diagonal that $\kappa_N^{Y^1} = \min\{(\kappa_{N_D}^D)^2, (\kappa_{N_G}^{G^\perp})^2\}$, which together imply that $\kappa_N \geq \min\{(\kappa^D_{N_D})^2, (\kappa^{G^\perp}_{N_G})^2\}$.

Combining these bounds, we have the desired bound on $\mathrm{cond}([D\ G^\perp]^\top [D\ G^\perp])$, that is,

$$(\kappa_1/\kappa_N)^2 \leq \frac{(\kappa^D_1)^2 + (\kappa^{G^\perp}_1)^2}{\min\{(\kappa^D_{N_D})^2, (\kappa^{G^\perp}_{N_G})^2\}} = \frac{(\kappa^D_1)^2 + 1}{\min\{(\kappa^D_{N_D})^2, 1\}} = (\kappa^D_1)^2 + 1$$

The second equality holds because all singular values of the orthogonal matrix $G^\perp$ are one. The last equality follows from Lemma 4, in particular, $\kappa^D_{N_D} \geq 1$. □

Proof of Proposition 5. Suppose the sequence of solutions generated by Algorithm 1 is $(\beta^h, u^h)_{h\in\mathbb{N}}$, where $\mathbb{N}$ denotes the set of (non-negative) natural numbers. Let's consider a subsequence $(\beta^{h_k}, u^{h_k})_{k\in\mathbb{N}}$, where $(h_k)_{k\in\mathbb{N}}$ is a strictly increasing sequence of positive numbers, such that this subsequence converges to a point $(\overline{\beta}, \overline{u})$. Such a subsequence always exists because of the Bolzano-Weierstrass theorem. Because the objective function is continuous, it follows that the sequence $(\mathrm{OBJ}(\beta^{h_k}, u^{h_k}))_{k\in\mathbb{N}}$ converges to $\mathrm{OBJ}(\overline{\beta}, \overline{u})$. This sequence is also non-increasing and bounded from below. The former owing to the optimization performed by BCD at each iteration, which implies

$$\mathrm{OBJ}(\beta^{h+1}, u^{h+1}) \leq \mathrm{OBJ}(\beta^h, u^h), \quad \forall h \in \mathbb{N}.$$

The latter follows from the proof of Proposition 1 and the equivalence between PLP and P2LP established in Proposition 3. Then the monotone convergence theorem ensures that $\mathrm{OBJ}(\overline{\beta}, \overline{u})$ is the infimum of the sequence $(\mathrm{OBJ}(\beta^{h_k}, u^{h_k}))_{k\in\mathbb{N}}$ and thus once BCD reaches $\mathrm{OBJ}(\overline{\beta}, \overline{u})$, its objective function value will remain at this value in later iterations.

By assumption, $\bar{u}$ is a non-degenerate optimal solution to the variant of P2LP with β equal to $\bar{\beta}$, which implies that its dual model has a unique optimal solution. We denote the latter solution as $\mu^*(\overline{\beta})$. Consider the idealized P2LP dual model. It has (i) vectors of decision variables μ and θ that are associated with primal constraints and bound constraints and (ii) two sets of constraints that are related to the β and u vectors of the variables of the idealized P2LP. The β-related constraints and the θ variables can be subdivided according to the elements of the given partition $\overline{\mathcal{P}}$. At each iteration, the idealized BCD method solves the linear program in Algorithm 1, which we denote as LP^h, for a set $\mathcal{P}^h \subseteq \overline{\mathcal{P}}$ chosen by the block selection rule. The dual of this model features the μ and $\theta(\mathcal{P}^h) := (\theta_{i,x_i,b}, (i, x_i, b) \in \mathcal{P}^h \times \mathcal{B}_i)$ variable vectors and the u- and $\beta(\mathcal{P}^h)$-related constraints. The pair $(\overline{\beta}(\mathcal{P}^h), \overline{u})$ is an optimal LP^h solution. Consider the complementary slackness conditions for this model and its dual expressed with respect to it. Solving the ones associated with $\overline{u}$ amounts to finding a solution to the system of u-related constraints that define $\mu^*(\overline{\beta})$, which we know uniquely solves it. Moreover, because we assume that $\overline{\beta}(\mathcal{P}^h)$ strictly satisfies the bound inequalities, the elements of the corresponding optimal vector $\theta^*(\mathcal{P}^h; \overline{\beta}(\mathcal{P}^h))$ equal zero. Thus, $(\overline{\beta}, \overline{u})$ and $(\mu^*(\overline{\beta}), \theta^*(\mathcal{P}^h; \overline{\beta}(\mathcal{P}^h))) \equiv (\mu^*(\overline{\beta}), 0)$

comply with both the complementary slackness equations for the idealized P2LP and its dual and the ones for P2LP and its dual. Moreover, the pair $(\mu^*(\overline{\beta}), 0)$ fulfills the set of $\beta(\mathcal{P}^h)$-related constraints of the idealized P2LP dual, because it is the only solution for the dual of LP^h that satisfies complementary slackness with respect to its primal optimal solution $(\overline{\beta}(\mathcal{P}^h), \overline{u})$.

For the greedy selection rule, since the chosen block $\mathcal{P}^h$ has the largest reduced cost, it indicates that $\mu^*(\overline{\beta})$ also satisfies the $\beta(\overline{\mathcal{P}} \setminus \mathcal{P}^h)$-related constraints of the idealized P2LP dual. So we conclude that $\mu^*(\overline{\beta})$ is feasible for the P2LP dual. When using the cyclic rule the blocks $\overline{\beta}(\mathcal{P}^h)$ optimized across iterations after reaching $(\overline{\beta}, \overline{u})$ cover $\overline{\mathcal{P}}$. Under the random rule, the same holds with probability 1 since we sample uniformly. Therefore, we can repeat the argument described for the chosen $\mathcal{P}^h$ to show that $\mu^*(\overline{\beta})$ is feasible for the P2LP dual. Consequently, $(\overline{\beta}, \overline{u})$ and $\mu^*(\overline{\beta})$ optimally solve P2LP and its dual, respectively, i.e., $(\overline{\beta}, \overline{u})$ belongs to the set of P2LP optimal solutions.

Since the objective function value is nonincreasing and the limit point $(\overline{\beta}, \overline{u})$ is optimal, the solutions generated after $(\overline{\beta}, \overline{u})$ must also be optimal. Thus BCD converges to the set of P2LP optimal solutions. □

A.4 Numerical Results for Linear Basis Functions

Table A.1: LSM- and PO-based lower and dual bound estimates with linear basis

	Upper Bounds		Lower Bounds		CPU Time (mins)	
Month	PO	LSM	PO	LSM	PO	LSM
Jan	20.90 (0.06)	25.00 (0.03)	16.83 (0.07)	18.59 (0.07)	166	2
Feb	20.75 (0.06)	24.75 (0.03)	16.10 (0.07)	18.48 (0.07)	153	2
Mar	25.28 (0.07)	29.37 (0.03)	20.37 (0.08)	23.15 (0.08)	200	2
Apr	26.86 (0.07)	30.98 (0.03)	21.52 (0.08)	24.72 (0.09)	202	2
May	23.16 (0.07)	27.58 (0.03)	17.76 (0.08)	20.63 (0.08)	175	2
Jun	19.69 (0.07)	24.31 (0.03)	15.59 (0.08)	16.90 (0.08)	179	2
Jul	16.66 (0.06)	20.97 (0.03)	12.29 (0.06)	14.01 (0.07)	194	2
Aug	23.10 (0.07)	27.80 (0.03)	18.61 (0.08)	20.31 (0.08)	187	2
Sep	24.17 (0.07)	28.93 (0.04)	19.64 (0.08)	21.16 (0.09)	163	2
Oct	21.40 (0.06)	25.61 (0.03)	17.15 (0.07)	18.68 (0.07)	162	2
Nov	19.85 (0.06)	24.06 (0.03)	15.53 (0.07)	17.30 (0.07)	165	2
Dec	15.43 (0.06)	19.43 (0.03)	11.69 (0.06)	13.20 (0.06)	174	2

A.5 Numerical Results Under Different Block Selection Rules

Table A.2: LSM- and PO-based lower and dual bound estimates with different block selection rules

	Upper Bounds			Lower Bounds		
Month	Cyclic	Greedy	Random	Cyclic	Greedy	Random
Jan	20.81 (0.08)	20.61 (0.05)	21.03 (0.03)	19.35 (0.06)	19.35 (0.06)	19.38 (0.07)
Feb	20.61 (0.04)	20.92 (0.04)	21.82 (0.04)	19.21 (0.07)	19.37 (0.07)	19.30 (0.07)
Mar	25.19 (0.05)	25.05 (0.07)	25.25 (0.03)	23.76 (0.08)	23.74 (0.08)	23.79 (0.08)
Apr	26.72 (0.05)	26.57 (0.07)	26.91 (0.03)	25.22 (0.09)	25.21 (0.09)	25.31 (0.09)
May	22.98 (0.05)	22.64 (0.06)	23.10 (0.03)	21.26 (0.08)	21.24 (0.08)	21.32 (0.08)
Jun	19.53 (0.05)	19.27 (0.07)	19.78 (0.03)	17.89 (0.08)	17.82 (0.08)	17.91 (0.08)
Jul	16.56 (0.04)	16.34 (0.05)	16.80 (0.03)	15.11 (0.07)	15.05 (0.07)	15.25 (0.07)
Aug	22.97 (0.05)	22.49 (0.07)	22.25 (0.03)	21.13 (0.08)	21.06 (0.08)	21.15 (0.08)
Sep	24.03 (0.05)	23.64 (0.07)	24.13 (0.03)	22.06 (0.08)	22.00 (0.08)	22.09 (0.09)
Oct	21.33 (0.06)	21.06 (0.06)	21.38 (0.03)	19.61 (0.07)	19.54 (0.07)	19.67 (0.07)
Nov	19.75 (0.04)	19.49 (0.05)	19.90 (0.04)	18.29 (0.07)	18.23 (0.07)	18.31 (0.07)
Dec	15.34 (0.04)	15.18 (0.05)	15.54 (0.03)	14.18 (0.06)	14.14 (0.06)	14.20 (0.06)

Table A.3: CPU times for the bound estimates

	CPU Times (mins)		
Month	Cyclic	Greedy	Random
Jan	933	289	966
Feb	951	313	985
Mar	975	394	1010
Apr	1059	332	1099
May	1051	390	1090
Jun	1067	369	1105
Jul	1031	321	1071
Aug	1116	323	1155
Sep	1137	302	1178
Oct	1021	286	1058
Nov	947	301	981
Dec	1136	300	1178
Avg.	1035	327	1073

Bibliography

Golub GH, Van Loan CF (2013) *Matrix Computations*, volume 3 (JHU press).

Zhang F (2011) *Matrix Theory: Basic Results and Techniques* (Springer Science & Business Media).

Appendix B

Supplement for Chapter 4

B.1 Proofs

Proof of Lemma 2. For every pair $(i, x_i) \in \times\mathcal{I} \times \mathcal{X}_i$ on the sample path l, since there is a feasible action which terminates the decision process and has a deterministic reward, the column in A^l corresponding to this action contains only 0 entries except the one associated to the $U_i^l(x_i)$ row. Therefore we can eliminate all entries in the $U_i^l(x_i)$ row with this column by column operations. We can repeat this step for every pair (i, x_i) on sample path l to obtain a matrix containing only one nonzeor element in each row. These nonzero elements are also in different columns since they represent different constraints. Thus A^l is full row rank. □

Proof of Lemma 3 We consider two LPs: PLP dual and adjusted PLP dual. We'll show that these two LPs have equivalent feasible sets. A solution feasible to PLP dual must also be feasible to the adjust PLP dual and vice versa. To see this, we only need to consider the strong duality constraints since the other two kinds constraints are the same in two LPs.

We first show if $C_{i,x_i}\mu_{i,x_i} = 0$, then $C'_{i,x_i}\mu_{i,x_i} = 0$. It's easy to see that if μ_{i,x_i} is a solution to the first linear system, it must be a solution to the second one because the constraints in $C'_{i,x_i}\mu_{i,x_i} = 0$ are also in $C_{i,x_i}\mu_{i,x_i} = 0$. Reversely, suppose μ_{i,x_i} is a solution to the second linear system, then it's also a solution to the first linear system because each constraint in $C_{i,x_i}\mu_{i,x_i} = 0$ can be expressed as a linear combination of constraints in $C'_{i,x_i}\mu_{i,x_i} = 0$. □

Proof of Proposition 9 The first statement is true because the constraint generated by the pseudo action scheme is a lower bound on the optimal policy value of the MDP almost everywhere. Thus, (4.17) has the same optimal objective function value as the (4.6) because the optimal objective value of both (4.17) and (4.6) are upper bounds on the MDP. In other words, the generated constraints do not influence the optimal objective value at all. The second statement holds by a direct extension of Theorem 1 in Desai et al. (2012b).□

Proof of Proposition 12 The proof is based on using the block coordinate descent method to solve the regression. We divide the variables into two blocks, U variable and β variable blocks. In every iteration of BCD, we iteratively fix the value of U variable block or the β variable block while optimizing over the other variable block. The algorithm will converge to an optimal solution because the objective function is convex and (twice) differentiable. We'll further show that the optimal solution is in fact unique.

Step 1: Fixing the value of β variables and optimizing over U variables. In step 1, the optimization is still a least squares regression. The difference between the regression in this step and the original regression is that every β variable has a fixed value. So the variables contained in this regression are U variables only. Note that the least squares regression decouples by samples because the U variable and its coefficient matrix A can be decoupled into L groups based on samples. Following the same spirit of Lemma 2, it can be shown, on each sample path, the coefficient matrix of U variables is full row rank. So the regression on each sample path has a unique solution. Step 1 generates unique solutions in BCD iterations.

Step 2: Fixing the values of U variables and optimizing over β variables. In the second step, the subproblem is still a linear regression. But the variables become β variables and all values of U variables are fixed. Also note that this regression decouples according to endogenous states since the β variables can be decoupled according to endogenous states. Since $\bar{\mu}$ is a solution to the linear system $C_{i,x_i}\mu_{i,x_i} = 0, \forall (i, x_i) \in \mathcal{I} \times \mathcal{X}_i$, the coefficient matrix of the β_{i,x_i} in this step must contain a set of bases in the solution space and therefore is full row rank. So the regression for each endogenous state also has a unique solution in step 2.

Since both step 1 and 2 generate unique solutions, the optimal solution to the regression is also unique. Due to the uniqueness of the optimal solution to the regression, (U^*, β^*) must also be the optimal primal solution to PLP. □

Proof of Proposition 13 The proof is also based on using the block coordinate descent method to solve the regression in Algorithm 4.55. We still divide all variables in the regression into two blocks, one block is U variables and the other is β variables. In every iteration of BCD, we either fix the value of β variables and update U, denoted as step 1 or fix the value of U and update β, denoted as step 2.

In step 1, there are only U variables in the regression because β is fixed. The regression will decouple according to samples as discussed in the proof for proposition 1. Furthermore by Lemma 2, the coefficient matrix of $U_i^l(x_i), \forall (i, x_i) \in \mathcal{I}^l \times \mathcal{X}_i^l$ is always full row rank on every sample path l given the optimal dual variable μ. Therefore, as long as we pick up all constraints in the regression by the unit of sample path, that is, we either include all constraints related to a sample path l or none of them in the regression, the step 1 will always generate a unique solution when updating U variables by BCD.

In step 2, there are only β variables in the regression which decouples by endogenous states. As discussed in proposition 12, the coefficient matrix of β_{i,x_i}, denoted as $\hat{C}_{i,x_i}$, is full row rank for each $(i, x_i) \in \mathcal{I} \times \mathcal{X}_i$. Since each $\hat{C}_{i,x_i}$ only has B_i rows, we can find at most B_i linearly independent columns from $\hat{C}_{i,x_i}$, i.e., we have B_i constraints for each $(i, x_i) \in \mathcal{I} \times \mathcal{X}_i$. In the worst case, these B_i constraints come from B_i different samples. So the total number of samples we need is $\sum_{i\in\mathcal{I}} \sum_{x_i\in\mathcal{X}_i} B_i$.□

Proof of Proposition 14 By the Hölder's inequality, we have

$$\bar{\mu}_1^\top (A\bar{U} + C\bar{\beta} - r) \leq \|\bar{\mu}_1\|_2 \left\|A\bar{U} + C\bar{\beta} - r\right\|_2 \tag{B.1}$$

Note that

$$\bar{\mu}_1^\top (A\bar{U} + C\bar{\beta} - r) = d^\top \bar{U} + \hat{c}\bar{\beta} - \bar{\mu}_1^\top r \tag{B.2}$$

If defining $OBJ(\beta) = d^\top \bar{U} + \hat{c}\bar{\beta}$, the RHS of (B.2) becomes

$$d^\top \bar{U} + \hat{c}\bar{\beta} - \bar{\mu}_1^\top r = OBJ(\bar{\beta}) - \bar{\mu}_1^\top r \tag{B.3}$$

Based on the results in Boyd et al. (2011), we have

$$\bar{\mu}_1^\top r - OPT \leq \bar{y}^\top \bar{\epsilon} + (\mu_1 - \mu^*)^\top \bar{\epsilon} \tag{B.4}$$

Therefore, we have the following inequality by combining (B.3) and (B.4)

$$OBJ(\bar{\beta}) - OPT \leq OBJ(\bar{\beta}) - \bar{\mu}_1^\top r + \bar{y}^\top \bar{\epsilon} + (\mu_1 - \mu^*)^\top \bar{\epsilon} \tag{B.5}$$

With (B.2), the above inequality becomes

$$OBJ(\bar{\beta}) - OPT \leq \bar{\mu}_1^\top (A\bar{U} + C\bar{\beta} - r) + \bar{y}^\top \bar{\epsilon} + (\bar{\mu}_1 - \mu^*)^\top \bar{\epsilon} \tag{B.6}$$

So the suboptimality of $OBJ(\bar{\beta})$ is bounded by the following inequality

$$\begin{aligned} OBJ(\bar{\beta}) - OPT &\leq \|\bar{y}\|_2 \|\bar{\epsilon}\|_2 + \|\bar{\mu}_1 - \mu^*\|_2 \|\bar{\epsilon}\|_2 + \|\bar{\mu}_1\|_2 \left\|A\bar{U} + C\bar{\beta} - r\right\|_2 && \text{(B.7)} \\ &\leq \|\bar{y}\|_2 \|\bar{\epsilon}\|_2 + D \|\bar{\epsilon}\|_2 + \|\bar{\mu}_1\|_2 \|\tilde{\epsilon}\|_2 && \text{(B.8)} \end{aligned}$$

where $\tilde{\epsilon} := A\bar{U} + C\bar{\beta} - r$. The first inequality holds because of the Holder's inequality. The second inequality is due to $\|\bar{\mu}_1 - \mu^*\|_2 \leq D$. □

Proof of Proposition 15 First it's easy to check $(\bar{\mu}_1, \bar{\mu}_2, \bar{\mu}_3)$ is feasible. Then analogous to the PLP dual (4.27)-(4.30), we apply ADMM to solve LP (4.57)-(4.60) and set $(\bar{\mu}_1, \bar{\mu}_2, \bar{\mu}_3)$ as the initial point. If the solution satisfies the optimality conditions of the Lagrangian function (4.35), it is optimal. The ADMM steps in the first iteration are as follows

$$\mu_1^1 = \underset{\mu_1 \in \mathcal{P}}{\arg\min} \left\{ \|\mu_1 - \bar{\mu}_3 + \delta - \epsilon_1 + \bar{y}_1\|_2^2 \right\} \tag{B.9}$$

$$\mu_2^1 = \underset{\mu_2 \in \mathcal{Q}}{\arg\min} \left\{ \|\mu_2 - \bar{\mu}_3 + \delta - \epsilon_2 + \bar{y}_2\|_2^2 \right\} \tag{B.10}$$

$$\begin{aligned} \mu_3^1 = \underset{\mu_3 \in \mathcal{R}}{\arg\min} \Big\{ & -\mu_3^\top (r - 2\rho\delta) + \rho/2 \, \|\bar{\mu}_1 - \mu_3 + \bar{y}_1 + \delta - \epsilon_1\|_2^2 \\ & + \rho/2 \, \|\bar{\mu}_2 - \mu_3 + \bar{y}_2 + \delta - \epsilon_2\|_2^2 \Big\} \end{aligned} \tag{B.11}$$

$$y_1^1 = \bar{y}_1 + \mu_1^1 - \mu_3^1 - \epsilon_1 + \delta \tag{B.12}$$

$$y_2^1 = \bar{y}_2 + \mu_2^1 - \mu_3^1 - \epsilon_2 + \delta \tag{B.13}$$

For (B.9), note that $\bar{y}_1^{k-1} = \bar{y}_1 - \epsilon_1$ and $\bar{\mu}_3^{k-1} = \bar{\mu}_3 - \delta$. By plugging in these values, (B.9) becomes

$$\mu_1^1 = \underset{\mu_1 \in \mathcal{P}}{\arg\min} \left\{ \|\mu_1 - \bar{\mu}_3^{k-1} + \bar{y}_1^{k-1}\|_2^2 \right\} \tag{B.14}$$

so $\mu_1^1 = \bar{\mu}_1^k = \bar{\mu}$ and similarly $\mu_2^1 = \bar{\mu}_2$. For (B.11), by replacing $\bar{y}_1 - \epsilon_1$ with $\bar{y}_1^{k-1}$ and $\bar{\mu}_3 - \delta$ with $\bar{\mu}_3^{k-1}$ and rearranging, it can be shown that the optimal solution is $\bar{\mu}_3$. So the optimal solution to (B.11) maintains the same during the iteration as well.

This update means that if we start from $(\bar{\mu}_1, \bar{\mu}_2, \bar{\mu}_3)$, it will still be the optimal solution in the following rounds of iterations. Thus, both the primal and dual residuals are 0, which satisfies the optimality conditions for the Lagrangian function (4.35).□

B.2 Illustration of the Unbounded issue in PLP

We provide a simple example to illustrate the unboundedness issue of PLP. We start from PLP (4.11)-(4.12):

$$\min_{U^l,\beta} \quad \frac{1}{L}\sum_{l=1}^{L} U_0^l(x_0) \tag{B.15}$$

$$\text{s.t.} \quad U_i^l(x_i) \geq r(x_i, s_i^l, a_i) - \sum_{b\in\mathcal{B}_i} \beta_{i+1,f(x_i,a_i),b}\Delta_i^{\mathbb{E},l}\phi_{i+1,b}$$

$$+ \delta U_{i+1}^l(f(x_i,a_i)), \forall (i,x_i,a_i) \in \mathcal{I}\setminus\{I-1\}\times\mathcal{X}_i\times\mathcal{A}_i(x_i), \tag{B.16}$$

$$U_{I-1}^l(x_{I-1}) \geq r(x_{I-1}, s_{I-1}^l, a_{I-1}), \quad \forall (x_{I-1}, a_{I-1}) \in \times\mathcal{X}_{I-1}\times\mathcal{A}_{I-1}(x_{I-1}), \tag{B.17}$$

for each $x_{I-1} \in \mathcal{X}_{I-1}$. For simplicity, we use two sample paths and set three as the time horizon in the example, i.e., $L = 2$ and $I = 3$. The endogenous state component set $\mathcal{X}_i$ for each stage is set to be $\{\mathsf{O}\}$. The corresponding dual value variables are $U_i^l(x_i)$ with $i \in \{0,1,2\}$, $x_i \in \{\mathsf{O}\}$, and $l \in \{1,2\}$. We restrict P the only feasible action for stage 0 and 1. The decision process stops with an action A in the last stage. We further assume there is only one β in the VFA, i.e., $B = 1$. The discount factor is also set to be 1. Figure B.1 shows the endogenous state transition for this illustrating example.

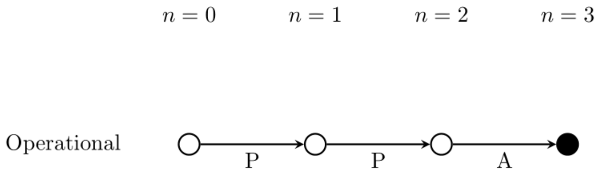


Figure B.1: Decision Tree for the Illustrating Example

Suppose the value of the coefficients corresponding to β on these two samples are 1.5 and -1.3, respectively. Then PLP for this example is (after simplification):

$$\min_{U,\beta} \quad \frac{1}{2}\sum_{l=1}^{2} U_0^l(x_0) \tag{B.18}$$

$$U_0^1(x_0) \geq r_0^1 + 1.5\beta + r_1^1 \tag{B.19}$$

$$U_0^2(x_0) \geq r_0^2 - 1.3\beta + r_1^2 \tag{B.20}$$

The above PLP is unbounded as β goes to the negative infinity. Even if we increase the number of samples, the issue will always exist as long as the sample average of the coefficient column of β is not 0 (in the specified example, the value is $1.5 - 1.3 = 0.2$).