Carnegie Mellon
Tepper
SCHOOL OF BUSINESS

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY
INDUSTRIAL ADMINISTRATION
(OPERATIONS RESEARCH)

Titled
"APPROXIMATE DYNAMIC PROGRAMMING FOR COMMODITY AND ENERGY MERCHANT OPERATIONS"

Presented by
Selvaprabu Nadarajah

Accepted by

Nicola Secomandi — 4-29-2014

Chair: Prof. Nicola Secomandi — **Date**

Approved by The Dean

Robert M. Dammon — 5-5-14

Dean Robert M. Dammon — **Date**

Approximate Dynamic Programming for Commodity and Energy Merchant Operations

by

Selvaprabu Nadarajah

A thesis
presented to the Carnegie Mellon University
in partial fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Operations Research

Pittsburgh, PA, USA, 2014

Abstract

We study the merchant operations of commodity and energy conversion assets. Examples of such assets include natural gas pipelines systems, commodity swing options, and power plants. Merchant operations involves managing these assets as real options on commodity and energy prices with the objective of maximizing the market value of these assets.

The economic relevance of natural gas conversion assets has increased considerably since the occurrence of the oil and gas shale boom; for example, the Energy Information Agency expects natural gas to be the source of 30% of the world's electricity production by 2040 and the McKinsey Global Institute projects United States spending on energy infrastructure to be about 100 Billion dollars by 2020.

Managing commodity and energy conversion assets can be formulated as intractable Markov decision problems (MDPs), especially when using high dimensional price models commonly employed in practice. We develop approximate dynamic programming (ADP) methods for computing near optimal policies and lower and upper bounds on the market value of these assets. We focus on overcoming issues with the standard math programming and financial engineering ADP methods, that is, approximate linear programing (ALP) and least squares Monte Carlo (LSM), respectively. In particular, we develop: (i) a novel ALP relaxation framework to improve the ALP approach and use it to derive two new classes of ALP relaxations; (ii) an LSM variant in the context of popular practice-based price models to alleviate the substantial computational overhead when estimating upper bounds on the market value using existing LSM variants; and (iii) a mixed integer programming based ADP method that is exact with respect to a policy performance measure, while methods in the literature are heuristic in nature.

Computational experiments on realistic instances of natural gas storage and crude oil swing options show that both our ALP relaxations and LSM methods are efficient and deliver near optimal policies and tight lower and upper bounds. Our LSM variant is also between one and three orders of magnitude faster than existing LSM variants for estimating upper bounds. Our mixed integer programming ADP model is computationally expensive to solve but its exact nature motivates further research into its solution.

We provide theoretical support for our methods: By deriving bounds on approximation error we establish the optimality of our best ALP relaxation class in limiting regimes of practical relevance and provide a theoretical perspective on the relative performance of our LSM variant and existing LSM variants. We also unify different ADP methods in the literature using our ALP relaxation framework, including the financial engineering based LSM method.

In addition, we employ ADP to study the novel application of jointly managing storage and transport assets in a natural gas pipeline system; the literature studies these assets in isolation. We leverage our structural analysis of the optimal storage policy to extend an LSM variant for this problem. This extension computes near optimal policies and tight bounds on instances formulated in collaboration with a major natural gas trading company. We use our extension and these instances to answer questions relevant to merchants managing such assets.

Overall, our findings highlight the role of math programming for developing ADP methods. Although we focus on managing commodity and energy conversion assets, the techniques in this thesis have potential broader relevance for solving MDPs in other application contexts, such as inventory control with demand forecast updating, multiple sourcing, and optimal medical treatment design.

Acknowledgements

I would like to start by expressing my gratitude to Nicola Secomandi. His encouragement and guidance as my advisor have been invaluable in shaping my view and enthusiasm towards interdisciplinary research. Nicola has been very generous with his time and I attribute most of what I learned about research to him. This thesis would not have been possible without his foresight. I would also like to thank my co-advisor Francois Margot for always motivating me to tackle challenging problems and explore different research topics. Working with him has built my confidence and breadth of knowledge as a researcher. I feel privileged to have been co-advised by Francois and Nicola.

My PhD studies have also been considerably influenced by Egon Balas and Alan Scheller-Wolf. I thank them both for impacting my research interests through the excellent classes they taught me and providing research and career advice. Egon has served as a role model and I thank him for the opportunity to collaborate on integer programming research. I am also grateful to my dissertation committee member Duane Seppi for his feedback and time.

Words cannot express my gratitude for everything my wife Negar Soheili has done for me. She is the ideal partner and has made my life so much more enjoyable. Negar, together with Andre Cire, have been the best colleagues that I could have hoped for to join the Operations Research PhD program with me in 2009. I have learned a lot from them and appreciate all their help and support over the years.

Tepper has been a fun place to do my PhD due to a number of friends: Ishani Aggarwal, Amitabh Basu, David Bergman, Elvin Coban, Andre Cire, Sam Hoda, Abha Kapoor, Alex Kazachkov, Qihang Lin, Marco Molinaro, Andrea Qualizza, Amin Sayedi, Nandana Sengupta, Vince Slaugh, Negar Soheili, Jessie Wang, and Hellen Zhou. Outside Tepper, I would like to thank my friend Christopher Lionel from the University of Toronto for his constant encouragement. A special thanks to Lawrence Rapp for taking care of PhD students and allowing us to focus on research.

Finally, I would like to thank my parents (Padmaawathy Nadarajah and Nadarajah Angappan), sister (Sooganthy Nadarajah), and brother in law (Arun Sivakumaran) for their unwavering support. My niece and nephew, Rhia Arun and Rohan Arun, also deserve a special mention for being a great source of joy during my PhD.

Dedication

To my parents and sister.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this chapter we discuss the business problem and methodologies that are the focus of this thesis. We then describe our contributions in this context. In §1.1 we describe merchant operations of commodity and energy conversion assets and associated optimization challenges. In §1.2 we discuss the intractable Markov decision problem formulations arising in these applications. We briefly review the approximate dynamic programming literature in §1.3. We describe our contributions and provide an outline of the thesis in §1.4.

1.1 Business Background and Challenges

The energy sector in the United States is undergoing a renaissance. Energy production has grown rapidly in recent years, mostly due to the oil and gas shale boom. According to the International Energy Agency (IEA, 2012), the United States is projected to overtake Saudi Arabia to become the largest oil producer by around 2020 and could be a net exporter of oil and natural gas in about a decade. The McKinsey Global Institute identified the energy sector as a game changer for the gross domestic product growth of the United States (Lund et al., 2013), potentially adding between 360-690 Billion US dollars to the US economy by 2020. In short, it is an exciting time to engage in energy research.

The realization of these projected benefits hinges (in part) on commodity and energy value chains being able to support the processing and physical flows of shale oil and natural gas, which are relatively new and unconventional energy sources. Current infrastructure falls short (Friedman and Philbin, 2014, INGAA, 2014). In response, commodity and energy value chains, such as the ones of natural gas and electricity, are undergoing considerable change, with $180 Billion US dollars worth of projected investment in new energy infrastructure by 2020 (Lund et al., 2013).

This thesis studies *commodity and energy conversion assets* (Secomandi and Seppi, 2014), which are embedded in commodity and energy value chains and perform important economic roles, such as physically converting raw materials or modifying the availability of commodities and energy sources. Examples of commodity and energy conversion assets include natural gas pipeline systems, oil refineries, and power plants. We focus on the

operations of these assets from the perspective of managers and energy traders at chemical and petroleum companies (e.g., Dow Chemicals, Exxon); energy trading, distribution, and utility firms (e.g., Noble Energy, National Grid, Williams Partners, Sempra Energy); and investment banks (e.g., Goldman Sachs). These managers and traders are referred to as *merchants* (Secomandi and Seppi, 2014).

Merchant operations of commodity and energy conversion assets involves viewing these assets as models of operational flexibility, or real options, on uncertain commodity and energy prices (Dixit and Pindyck, 1994). The operational flexibility embedded in these assets is limited by operational constraints, a feature typically deemphasized in financial options. For example, the amount of natural gas injected into and withdrawn from a natural gas storage facility is constrained in practice by overall storage space, and injection and withdrawal capacities. The merchant's objective is to maximize the value of commodity and energy conversion assets by optimally adapting decisions endowed by their operational flexibility to the unfolding of commodity and energy prices. The maximum value is referred to as the *market value* of the commodity and energy conversion asset because this value can be replicated using a portfolio of financially traded instruments under certain market completeness assumptions (Dixit and Pindyck 1994, Seppi 2002, Secomandi and Seppi 2014 Chapter 3).

The valuation and management of commodity and energy conversion assets can be formulated as intractable Markov decision problems (MDPs). Merchants thus use heuristics to manage these assets. For example, popular practice-based heuristics for commodity storage include (i) the rolling intrinsic approach, which periodically resolves a deterministic version of the MDP; and (ii) the spread option linear program, which models the payoff from a pair of injection and withdrawal trades as a spread option on the natural gas spot prices and determines notional trading amounts over a portfolio of such spread options (Gray and Khandelwal, 2004, Secomandi, 2014). Linear programming based heuristics are also used for refinery operations (Favennec, 2001, Chapters 5 and 6). The quality of the policies computed by practice-based heuristics varies from near optimal to substantially suboptimal (de Jong et al., 2010, Lai et al., 2010).

Assessing the quality of practice-based heuristics is not straightforward because they compute an operating policy, which provides only a lower bound on the market value; that is, it does not provide an upper bound on the market value (Lai et al., 2010). Further, valuation is a building block for hedging (Nadarajah et al., 2013) and credit value adjustment computations (Thompson, 2012, 2013) which are critical for the risk management of commodity and energy conversion assets. These tasks require valuations to be performed at multiple future states (e.g., commodity and energy prices) sampled within Monte Carlo simulation. Thus being able to perform fast valuations of the asset at future states is critical. Practice based heuristics are typically not well suited for these purposes because they do not provide a valuation at future states and extending them to obtain such valuations is computationally infeasible. Thus, efficient approximate methods for computing tight lower and upper bounds on the market value are useful in practice.

1.2 MDP

Merchant operations of commodity and energy conversion assets involves solving sequential decision making problems under uncertainty. The merchant operates a commodity and energy conversion asset at a set of predefined times belonging to a stage set $\mathcal{I} := \{0, 1, \ldots, N-1\}$, where N is finite. At each stage $i \in \mathcal{I}$, the merchant has access to new information referred to as the state of the system. This state can be partitioned into an endogenous component, x_i, such as inventory or the asset operating status, and an exogenous component, F_i, which we take to be commodity and energy prices, but other possibilities include demand and interest rates. Both the endogenous and exogenous states can be vectors. Changes in the endogenous state are the result of actions taken by merchants, while the evolution of the price (exogenous) state is determined by market forces. Because the endogenous state x_i models the operational flexibility of the asset, it is reasonable to model its support by a compact set $\mathcal{X}_i$. Energy and commodity prices in F_i are continuous and belong to $\mathbb{R}^{n_{F_i}}$, where n_{F_i} is the number of components of F_i.

Given the history of states from stage 0 to stage i, the merchant takes an action at stage i. Once this action is known, the merchant receives an immediate reward, and the system transitions to a state at stage $i+1$ following a probability distribution function. This process starts from some state at the initial stage ($i = 0$) and repeats until the end of the finite horizon ($i = N-1$).

In Markov decision problems (Puterman, 1994), the action, the immediate reward, and the state transition probability distribution at stage i depend only on the current state (x_i, F_i), that is, past state information is inconsequential to the evolution of the system. Thus, at stage i, the action at each state is given by a function $a_i(x_i, F_i)$, the reward is defined by a function $r_i(x_i, F_i, a_i)$ (we suppress the dependence of the action on the state in our notation), and state transitions are given by a probability distribution function $f_i(x_{i+1}, F_{i+1}|x_i, F_i, a_i)$. Similar to the support of the endogenous state, we model the support of $a_i(x_i, F_i)$ by a compact set $\mathcal{A}_i(x_i, F_i)$.

A common assumption in real options applications, including merchant operations applications, is that the decision maker is a price taker (small player) whose decisions do not affect the evolution of F_i (Guthrie, 2009). This is a reasonable assumption in competitive commodity markets such as natural gas and crude oil. Under this assumption, the transition function can be written as $f_i(x_{i+1}, F_{i+1}|x_i, F_i, a_i) = f_i^x(x_{i+1}|x_i, F_i, a_i) f_i^F(F_{i+1}|F_i)$, where $f_i^x(x_{i+1}|x_i, F_i, a_i)$ and $f_i^F(F_{i+1}|F_i)$ are marginal transition probability distribution functions for the endogenous states and prices, respectively. The transition probability distribution for prices is typically risk-adjusted (that is, it is a so called risk-neutral distribution) and determined by stochastic differential equations governing price evolution (Seppi 2002 and Secomandi and Seppi 2014, chapter 3). This risk-adjusted measure is unique when the commodity/energy market is complete, see, e.g., Björk (2004, page 122). A common choice for the endogenous transition distribution in storage and transport applications is the following deterministic function: $\mathbb{1}(x_{i+1} = x_i - a_i)$. We will use this functional form for the endogenous transition distribution for concreteness.

Recall from §1.1 that we are ideally interested in computing the market value of a com-

modity and energy conversion asset and an optimal operating policy, which is a collection of optimal decision rules, one for each stage and state. Specifically, it is standard to define (operating) policy π as a sequence of decision rules, where the decision rule $a_i^\pi(x_i, F_i)$ returns the action taken by policy π at stage i and state (x_i, F_i). We also represent the set of all policies by Π. Denoting the market value of the asset by $V_0(x_0, F_0)$ at the initial state (x_0, F_0), our MDP formulation is

$$V_0(x_0, F_0) := \max_{\pi \in \Pi} \sum_{i \in \mathcal{I}} \delta^i \mathbb{E}\left[r(x_i^\pi, F_i, a_i^\pi(x_i^\pi, F_i))|x_0, F_0\right], \tag{1.1}$$

where δ is the risk-free discount factor from stage i back to stage $i-1$, $\forall i \in \mathcal{I} \setminus \{0\}$; $\mathbb{E}$ is expectation under the risk-neutral measure for F_i, and x_i^π the value of the endogenous state at stage i under policy π.

In theory, an optimal policy of MDP (1.1) can be computed by solving a stochastic dynamic program (SDP). We denote by $V_i(x_i, F_i)$ the value function of this SDP, which in our setting represents the market value of the asset starting at stage i from state (x_i, F_i) and operating until the end of the horizon. The SDP formulation is

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

$\forall (i, x_i, F_i) \in \mathcal{I} \times \mathcal{X}_i \times \mathbb{R}^{n_{F_i}}$, with boundary conditions $V_N(x_N, F_N) := 0, \forall x_N \in \mathcal{X}_N$. The structure we imposed on the state transition distribution functions is apparent in how the stage i state (x_i, F_i) transitions to the stage $i+1$ state $(x_i - a_i, F_{i+1})$ in this SDP. The presence of a vector of random variables, F_i, in the state is an important feature of this SDP (these random variables need not appear in the state if their future distributions do not depend on their prior realized values). The presence of these random variables in the state is common in real and financial options. Examples of applications that fit formulation (1.2) include chooser flexible caps (Meinshausen and Hambly, 2004), portfolio liquidation (Gyurko et al., 2011), swing options (Barbieri and Garman, 1996, Jaillet et al., 2004, Chandramouli and Haugh, 2012), switching options (Cortazar et al., 2008), commodity processing and storage (Maragos, 2002, Boogert and De Jong, 2008, 2011/12, Secomandi, 2010, Lai et al., 2010, Arvesen et al., 2013, Boogert and Mazières, 2011, Devalkar et al., 2011, Thompson, 2012, Wu et al., 2012), and power plants (Tseng and Barz, 2002, Thompson, 2013).

In general, solving SDP (1.2) is challenging. First, the value function $V_i(x_i, F_i)$ needs to be computed at infinitely many states. Some discretization is typically possible to avoid an infinite number of states but the number of states in the discretization still grows exponentially with the number of components in the state. This state space dimensionality issue is referred to as the first curse of dimensionality (Powell, 2011, §4.1). Second, evaluating the right hand side of (1.2) requires computing an expectation that is typically not available in closed form (see Secomandi 2014, Proposition 4 for an exception). Discrete approximations of transition probability distributions using grids or lattices are popular in the literature (see, e.g., Levy 2004 and references therein) to overcome this issue when the number of stochastic factors driving the evolution of prices is small, typically less than three (see,

e.g., Schwartz and Smith 2000 and Jaillet et al. 2004). However, capturing the volatility in energy and commodity prices typically requires a larger number of factors and, hence, higher dimensional models for the evolution of prices are commonly employed in practice and academia (Ho and Lee, 1986, Cortazar and Schwartz, 1994, Clewlow and Strickland, 2000, Maragos, 2002, Eydeland and Wolyniec, 2003, Veronesi, 2010). Thus, computing expectations also has an associated curse of dimensionality referred to as the second curse of dimensionality. Finally, even if this expectation were computable, the optimization over actions in set $\mathcal{A}(x_i, F_i)$ on the right hand side of (1.2) is in general a challenging nonlinear, possibly, stochastic mixed integer optimization problem. However, there are applications where this optimization can be performed efficiently such as when the value function is concave in the endogenous state (see for example Lai et al. 2010, Powell 2011, ch. 13, and Nascimento and Powell 2013a).

In summary, solving SDP (1.2) in the context of merchant operations of commodity and energy conversion assets poses multiple challenges. Methods to approximately solve such SDPs are thus justified.

1.3 Approximate Dynamic Programming

As discussed in §1.1, several challenges make the exact computation of an optimal operating policy and the associated market value $V_0(x_0, F_0)$ intractable. Approximate dynamic programming (ADP; Bertsekas 2007 and Powell 2011) is an area at the intersection of machine learning, financial engineering, and optimization that takes the practical approach of computing suboptimal policies and lower and upper bounds on their market values. A common ADP approach computes low-dimensional approximations to the value function $V_i(x_i, F_i)$ of SDP (1.2), or its continuation function $C_i(x_{i+1}, F_i) := \delta\mathbb{E}\left[V_{i+1}\left(x_{i+1}, F_{i+1}\right)|F_i\right]$. We use the labels $\hat{V}_i(x_i, F_i)$ and $\hat{C}_i(x_{i+1}, F_i)$ to denote a value function approximation and continuation function approximation, respectively. We briefly discuss the use of these approximations to compute suboptimal policies and lower bounds in §1.3.1 and upper bounds in §1.3.2. We review methods to compute value function and continuation function approximations in §1.3.3.

In this thesis, we will largely apply the lower and upper bounding approaches available from the literature and discussed in §§1.3.1-1.3.2, although we will exploit application-specific structure in certain cases to make these methods efficient. Our contributions add to the literature on methods for computing value function approximations discussed in §1.3.3.

1.3.1 Greedy Policies and Lower Bounds

At a given stage i and state (x_i, F_i), value function and continuation function approximations can be combined with the Bellman operator of SDP (1.2) to compute a heuristic

feasible action, referred to as a greedy action (see §6.1.1 in Bertsekas 2007):

$$\underset{a_i \in \mathcal{A}_i(x_i, F_i)}{\operatorname{argmax}} \; r_i(x_i, F_i, a_i) + \delta \mathbb{E}\left[\hat{V}_i(x_i - a_i, F_{i+1}) | F_i\right], \tag{1.3}$$

$$\underset{a_i \in \mathcal{A}_i(x_i, F_i)}{\operatorname{argmax}} \; r_i(x_i, F_i, a_i) + \hat{C}_i(x_i - a_i, F_i). \tag{1.4}$$

Notice that (1.3) is a stochastic optimization problem while (1.4) is a deterministic optimization problem; the expectation in (1.3) is typically replaced by a sample approximation. Finding optimal actions in both (1.3) and (1.4) can be done easily using enumeration if $\mathcal{A}_i(x_i, F_i)$ is a finite set of small cardinality. Otherwise, special structure on the reward function and the value function approximation, such as concavity in the actions a_i, is needed for the efficient computation of greedy actions. The (possibly infinite) collection of greedy actions at all stages and states defines a greedy policy and the value of the commodity and energy conversion asset operated under this policy is a "greedy" lower bound on the market value. Computing this lower bound exactly may not be possible because of the state space curse of dimensionality. Instead, an estimate of the greedy lower bound can be obtained in Monte Carlo simulation by evaluating the greedy policy over a fixed set of sample paths of prices (the exogenous state).

1.3.2 Upper Bounds

Upper bounds can be estimated via Monte Carlo simulation of prices (the exogenous state) using the so called information relaxation and duality approach. This approach has its roots in the financial engineering literature (see Rogers 2002, Andersen and Broadie 2004, Chapter 8 in Glasserman 2004, Haugh and Kogan 2004, Detemple 2006, Haugh and Kogan 2007, Brown et al. 2010, and references therein). The intuition behind a dual upper bound is to allow the decision maker to access future state information when making decisions, that is, relax the "nonanticipativity" constraints related to state information implicit in the SDP formulation. Knowledge of future information results in an upper bound on the market value that is further tightened by penalizing this knowledge using well-constructed dual penalties.

Brown et al. (2010) identify a hierarchy of information relaxations based on how much future information the decision maker is allowed to access. Perfect information relaxations correspond to the extreme case where the decision maker has access to all possible future state information. Estimating perfect information upper bounds in Monte Carlo simulation requires the solution of a deterministic version of SDP (1.2) with rewards modified by the dual penalty along each sample path of prices (the exogenous state). Therefore, estimating these upper bounds is easier when the endogenous state space is low dimensional and optimization over actions in this deterministic dynamic program can be performed efficiently, or the deterministic version of dynamic program (1.2) can be formulated as a tractable math program, e.g., a convex program.

The construction of dual penalties is the key component for estimating dual upper bounds. These penalties can be instantiated using value function and continuation func-

tion approximations (Haugh and Kogan, 2004, Lai et al., 2010). The definition of dual penalties involves an expectation that has to be typically approximated by a sample approximation. Brown et al. (2010) and Lai et al. (2010) instantiate dual penalties using a value function approximation and such sample approximations. Other approaches to instantiate dual penalties include using the continuation function of a heuristic policy (Andersen and Broadie, 2004), gradient based penalties (Brown and Smith, 2011) or functions related to the value functions of easier-to-solve versions of an SDP (Brown and Smith, 2013, Secomandi, 2014). It is known that computing dual upper bounds using a continuation function approximation is computationally challenging (see Chapter 8 in Glasserman 2004).

1.3.3 Approximations of the Value and Continuation Functions

A key ingredient for computing greedy policies and upper and lower bounds is a value (continuation) function approximation. A value function approximation is an approximation architecture $\hat{V}_i(\cdot, \cdot; \beta_i)$ parameterized by the vector $\beta_i \in \mathbb{R}^{B_i}$. The parameterization in $\hat{V}_i(\cdot, \cdot; \beta_i)$ could be nonlinear, for example a neural network (Vapnik, 2000). Although nonlinear parameterizations are possible, the most common choice is a linear parameterization of a set of potentially nonlinear *basis* functions, where each basis function is a mapping from the stage i state space to the real line (Tsitsiklis and Van Roy, 1996). Examples of basis functions include polynomials of the components of the state, and call and put options on commodity and energy prices. Choice of basis functions are typically application specific, but a few recent papers have focused on automatic basis function generation (Klabjan and Adelman, 2007, Bhat et al., 2012). Continuation function approximations $\hat{C}_i(\cdot, \cdot; \theta_i)$ can be defined in an analogous manner with parameters $\theta_i \in \mathbb{R}^{B_i}$. The problem of determining a value function and continuation function approximation thus reduces to the problem of determining β_i and θ_i, respectively.

Developing methods to compute good linearly parametrized value (continuation) function approximations is an active area of ADP research. The applicability of methods in the literature can be broadly classified based on the size of the endogenous and exogenous state spaces of the SDP. We will refer to a endogenous/exogenous state space as small if it is a vector with less than three components, and as large otherwise. We note that the action space size, not included in this classification, is also important in determining the difficulty of an SDP.

Class (i). The easiest SDP class has both small endogenous and exogenous state spaces. Specialized methods are not critical here because the state space can typically be discretized and a version of SDP (1.2) defined on this discretized state space can be solved using backward recursion to obtain a good approximation. Thus, the interesting SDP classes have a large endogenous state space or a large exogenous state space.

Class (ii). SDPs with a large endogenous state space and a small exogenous state space are arguably the most well studied problem class in operations research. A popular math programming approach applied to problems in this class is approximate linear programming (Schweitzer and Seidmann 1985, de Farias and Van Roy 2003). This approach solves a linear

program, known as an approximate linear program, with the value function approximation weights β_i as variables and a large number of constraints, one for every stage-state-action triple. Because ALP has a manageable number of variables, row-generation schemes, or column generation in its dual, have been used to solve it exactly in applications (Adelman, 2003, 2004, 2007). Alternatively, sampled approximate versions of ALP have also been considered to reduce the number of constraints (de Farias and Van Roy, 2003, 2004). Although constraint reduction is needed, adding constraints to impose structure on the value function approximation has been shown to substantially speed up ALP solve times (Adelman, 2007). On the theoretical side, an appealing feature of the ALP approach is its strong performance guarantees, in terms of the difference between the value function of SDP (1.2) and the value function approximation, evaluated using an ∞-norm or a weighted 1-norm.

Recent research has focused on alternate math programs to ALP for computing a value function approximation. Petrik and Zilberstein (2009) and Desai et al. (2012a) develop a linear program, known as the smoothed ALP, by relaxing the constraints of ALP. Desai et al. (2012a) show that the performance guarantee of smoothed ALP is superior to ALP. Solving this ALP relaxation is more difficult than ALP because it has both a large number of variables and constraints. Row generation for its exact solution is thus not viable. However, Desai et al. (2012a) discuss how a sampled version of the smoothed ALP can be solved instead. Wang and Boyd (2010) propose a different ALP relaxation scheme that improves upon the ALP performance guarantee as well but leads to a more controlled increase in the number of variables compared to the smoothed ALP. Motivated by improving greedy policy performance, Petrik (2012) develops a mixed integer programming ADP model that maximizes a lower bound on the value of a greedy policy. This mixed integer program is challenging to solve.

Another ADP approach for SDP class (iii) is based on stochastic gradient methods (see Chapters 9 and 10 in Powell 2011 and Chapter 6 in Bertsekas 2007). These methods start with an initial value (continuation) function approximation and apply an iterative scheme to improve this approximation. An important component in designing such schemes is a mechanism for updating the value (continuation) function approximation. Under certain conditions, convergence to the exact value (continuation) function can be guaranteed (Nascimento and Powell, 2009), but performance guarantees after a finite number of iterations are typically absent. Methods of this type have been successfully applied to compute policies for applications with very large endogenous state spaces (Simo et al., 2008).

Upper bound estimation based on the information relaxation and duality idea (see §1.3.2) may be challenging for SDPs with a large endogenous state space, unless the action space, rewards, and dual penalties have special structure. An alternate upper bound on the optimal policy that has been used in this literature is from the ALP value function approximation (Adelman, 2003, 2004, 2007). This value function approximation is an upper bound on the exact value function at every stage and state, and the exact value function at the initial stage and state coincides with the value of an optimal policy. However, care must be taken when solving a sampled version of ALP because the upper bound from this sampled version may not be an upper bound on the optimal policy value of the original problem.

As noted in §1.2, realistic models of commodity and energy price evolution are typically high dimensional. Thus, the most relevant problem classes for merchant operations applications are the two with large exogenous state spaces.

Class (iii). A number of important merchant operations applications have small endogenous states and large exogenous states, for example, multiple exercise options such as commodity storage and swing options. For this SDP class, the standard approach both in practice and the financial engineering literature computes a continuation function approximation by approximating the SDP (1.2) using regression and Monte Carlo simulation (see Appendix B in Eydeland and Wolyniec 2003, Glasserman and Yu 2004, Meinshausen and Hambly 2004, Detemple 2006, Boogert and De Jong 2008, 2011/12, Bender 2011, Gyurko et al. 2011). This approach was pioneered by Carriere (1996), Longstaff and Schwartz (2001), and Tsitsiklis and Van Roy (2001).

It is known than the standard LSM approach is not well suited for estimating information relaxation and duality based upper bounds (see §1.3.2) because it computes a continuation function approximation (see Chapter 8 in Glasserman 2004). Recent LSM variants by Desai et al. (2012b) and Gyurko et al. (2011) focus on overcoming this issue, but these variants rely on sample average approximations that are computationally expensive.

We note that math programming based ADP methods that have been extensively applied for solving SDPs in class (ii) can in principle also be applied for solving SDPs in class (iii). However, such applications have been limited (Nascimento and Powell, 2013b); in particular, we are not aware of an application of ALP.

Class (iv). The most challenging SDPs have both large endogenous and exogenous states. The development of ADP methods for these problems is scant (Powell et al., 2012a, Scott et al., 2014). However, operations of important commodity and energy conversion assets, such as refineries, belong to this class. The endogenous state in a refinery application could include conversion decisions of multiple input commodities into multiple output commodities and storage decisions of these inputs and outputs, and the exogenous state could include input and output commodity prices.

1.4 Thesis Contributions and Outline

This thesis focuses on computing near optimal policies and lower and upper bounds on the market value for SDPs arising in the merchant operations of commodity and energy conversion assets, that is, SDPs with large endogenous and/or exogenous state spaces (see classes (iii) and (iv) discussed in §1.3). We develop effective math programming and LSM based ADP methods to overcome deficiencies of existing ALP and LSM approaches. We provide theoretical support for these methods using bounds on the value function approximation error and also provide some limiting optimality results. We also use a math programming perspective to unify existing ADP methods and the ones we develop. These results connect seemingly different financial engineering and math programming ADP approaches. We also

study a novel commodity and energy conversion asset application. Overall, our findings highlight the role of a math programming perspective when developing ADP methods.

Although the focus of this thesis is on managing commodity and energy conversion assets, the ideas and methods we develop have potential broader relevance for solving MDPs in applications such as inventory control with demand forecast updates (Iida and Zipkin, 2006), multiple sourcing (Veeraraghavan and Scheller-Wolf, 2008), and optimal medical treatment design (Schaefer et al., 2005, Mason, 2012).

We provide an outline of the thesis below, also elaborating on the contributions of each chapter.

Chapter 2. ALP has been successfully applied for solving SDPs arising in applications with small exogenous state spaces but has not been explored for solving SDPs with large ones, which are typical in merchant operations applications. We explore the ALP approach for these SDPs. We analyze the optimal solution sets of both the ALP dual and the dual of the linear program associated with an MDP, which we refer to as the exact dual. The optimal solutions of the exact dual are in one to one correspondence with the state-action probability distributions induced by optimal policies (Puterman, 1994). In contrast, we find that all the optimal solutions of the ALP dual may correspond to state-action probabilities induced by infeasible policies. This inconsistency may lead to poor ALP value function approximations and greedy policies, especially when the state space of the SDP includes large exogenous information. We develop a novel ALP relaxation framework that overcomes this issue by restricting the ALP dual using constraints. We use this framework to develop two new classes of ALP relaxations. These ALP relaxations are different from the existing ALP relaxations in the literature (Petrik and Zilberstein, 2009, Desai et al., 2012a, Wang and Boyd, 2010). Moreover, the existing ALP relaxations are not derived by constraining the ALP dual.

We provide theoretical support for our best ALP relaxation class by deriving bounds on the value function approximation error and using them to establish the optimality of ALP relaxations in limiting regimes of practical relevance. Our best ALP relaxation improves on the best known bounds on realistic commodity storage instances and is competitive with two state of the art approaches: least squares Monte Carlo (Longstaff and Schwartz, 2001) and the rolling intrinsic (see, e.g., Lai et al. 2010 and references therein).

Chapter 3. As discussed in §1.3.3, LSM methods are the standard approach in financial engineering for approximating SDPs with small endogenous states and large exogenous states. However, existing LSM variants require substantial computational overhead for upper bound estimation either due to the use of a continuation function approximation (Longstaff and Schwartz, 2001) or sample average approximations (Desai et al., 2012b, Gyurko et al., 2011). We develop an LSM variant for estimating greedy lower bounds and dual upper bounds on the value of multiple exercise options in conjunction with common term structure models for commodity and energy price evolution (Ho and Lee, 1986, Cortazar and Schwartz, 1994, Clewlow and Strickland, 2000, Maragos, 2002, Eydeland and Wolyniec, 2003, Veronesi, 2010). Our approach computes a value function approximation and eliminates the need for sample average approximations during bound estimation. Interestingly, a number of papers that employ the standard LSM approach in the academic

literature use price models that are special cases of term structure models and would benefit from using our variant.

We numerically benchmark our LSM variant against the standard LSM method (Longstaff and Schwartz, 2001) and a recent LSM variant (Desai et al., 2012b, Gyurko et al., 2011) on new realistic energy swing and storage option instances. We find that our LSM technique requires significantly fewer regression samples than the two other LSM methods to deliver near optimal bound estimates with about the same accuracy and precision. For the same number of evaluation samples, our LSM approach is between one and three orders of magnitude faster than the two existing LSM approaches when estimating dual upper bounds, while all the three LSM methods exhibit comparable computational effort when estimating greedy lower bounds. We also conduct a worst case error bounding analysis that provides a theoretical perspective on the relative quality of the bounds estimated by these methods on our instances.

Chapter 4. We consider the novel application of the merchant management of natural gas pipeline systems, which give merchants the ability to trade natural gas across time and geographical markets. That is, these systems embed two types of assets that merchants manage as real options: storage and transport. The current literature has studied the management of these assets in isolation, while we consider their joint management.

We formulate this problem as an SDP with large endogenous and exogenous state spaces, that is, this SDP belongs to class (iv) of §1.3.3. The curse of dimensionality in the endogenous state space makes it difficult to apply LSM methods. We thus use an equivalent reformulation of this SDP that has a one dimensional endogenous state space but requires solving a linear program for evaluating its reward function, that is, we move the SDP from class (iv) to class (iii) without loss of generality. Our LSM variant of Chapter 3 is now applicable but takes substantial computational time to estimate lower and upper bounds because of the expensive reward function evaluations. To overcome this issue, we characterize the structure of an optimal storage policy and leverage it to efficiently reduce the number of reward function evaluations. We find that this extension of our LSM method is efficient and performs near optimally on realistic instances formulated in collaboration with a natural gas trading company in the United States.

Using this policy and instances, we find that (i) the joint, rather than decoupled, merchant management of storage and transport assets has substantial value; (ii) this management can be nearly optimally simplified by prioritizing storage relative to transport, despite the considerable substitutability between these activities; (iii) the value due to price uncertainty is large but can be almost entirely captured by sequentially reoptimizing a deterministic version of our MDP, an approach included in existing commercial software; and (iv) the value of transport trading across different pipelines is substantial.

Chapter 5. As discussed in §1.3.3, LSM is the standard financial engineering approach for solving SDPs with a small endogenous states space and a large exogenous state space. We show in Chapter 2 that our constraint-based ALP relaxations are also effective for solving SDPs in this class. Thus, one wonders if the LSM or ALP relaxation approaches are related in anyway. We settle this question by showing that our LSM variant and the constraint-based ALP relaxations are special cases of the same family of ALP relaxations.

We derive this family using our ALP relaxation framework of Chapter 2, which is based on constraining the ALP dual. This unification result provides a new math programming perspective on LSM methods.

A related question is whether our ALP relaxation framework can also be used to derive the existing ALP relaxations of Desai et al. (2012a) and O'Donoghue et al. (2013), which are not based on the idea of constraining the ALP dual. We answer this question in the affirmative. We interpret the dual constraints that we use in deriving these ALP relaxations as different ways of overcoming a potential under exploration of the state space by the set of optimal solutions to the ALP dual. These derivations suggest that our framework for generating ALP relaxations via restrictions of the ALP dual is a useful way of thinking about ALP relaxations.

Finally, we study the mixed integer program of Petrik (2012) that computes a value function approximation by maximizing a lower bound on the value of a greedy policy (see §1.3.1 for definition of a greedy policy). This mixed integer program is challenging to solve. Nevertheless, it is a heuristic approach that attempts to get closer to the objective of maximizing the value of the greedy policy associated with a value function approximation when determining such approximations, which is a natural objective to consider. Moreover, it has strong worst case guarantees for greedy policy performance. Motivated to overcome the heuristic nature of this approach, we develop a mixed integer program for computing a value function approximation that maximizes the greed policy value for a class of structured SDPs. In other words, our mixed integer program is exact for this class of SDPs (modulo sampling error) which encompass many commodity and energy conversion assets. Our mixed integer program is also challenging to solve but its exact nature motivates further research into its solution.

Chapter 6 and Appendices A-C. We provide a summary of insights and a discussion of future research directions in Chapter 6. Appendices A-C contain supporting material and proofs for Chapters 2-4, respectively.

Chapter 2

Relaxations of Approximate Linear Programs for the Real Option Management of Commodity Storage

(Joint work with François Margot and Nicola Secomandi)

2.1 Introduction

Real options are models of projects that exhibit managerial flexibility (Dixit and Pindyck, 1994). In commodity settings, this flexibility arises from the ability to adapt the operating policy of commodity conversion assets to the uncertain evolution of commodity prices. For example, consider a merchant that manages a natural gas storage asset (Maragos, 2002). This merchant can purchase natural gas from the wholesale market at a given price, and store it for future resale into this market at a higher price. Other examples of commodity conversion assets include assets that produce, transport, ship, and procure energy sources, agricultural products, and metals.

Managing commodity conversion assets as real options (Smith and McCardle, 1999, Geman, 2005) gives rise to, generally, intractable Markov Decision Processes (MDPs). In a given stage, the state of such an MDP includes both endogenous and exogenous information. The endogenous information describes the current operating conditions of the conversion asset, while the exogenous information represents current market conditions. Changes in the endogenous information are caused by managerial decisions. The exogenous information evolves as a result of market dynamics. The MDP intractability is due primarily to the common use in practice of high dimensional models of the evolution of the exogenous information (Eydeland and Wolyniec, 2003). To illustrate, consider the MDP for the real options management of a commodity storage asset formulated by Lai et al. (2010; LMS hereafter for short) using a multi-maturity version of the Black (1976) model of futures price evolution. The endogenous information is the asset available inventory at a given date, a one dimensional variable; the exogenous information is the commodity

forward curve at a given time, an object with much higher dimensionality than inventory. Approximations are thus typically needed to solve such MDPs.

Approximate linear programing (ALP; Schweitzer and Seidmann 1985, de Farias and Van Roy 2003) is an approach that approximates the primal linear program associated with an MDP (Manne, 1960, Puterman, 1994) by applying a lower dimensional representation of its variables. Solving an approximate linear program (which we also abbreviate as ALP for convenience) provides a value function approximation that can be used to obtain a heuristic control policy and estimate lower and upper bounds on the value of an optimal policy (see Bertsekas 2007, Brown et al. 2010, Powell 2011, and references therein). Applications of this approach include Trick and Zin (1997) in economics; Adelman (2004) and Adelman and Klabjan (2012) in inventory routing and control; Adelman (2007), Farias and Van Roy (2007), Zhang and Adelman (2009), and Adelman and Mersereau (2013) in revenue management; and Morrison and Kumar (1999), de Farias and Van Roy (2004, 2003), Moallemi et al. (2008), and Veatch (2010) in queuing control. To the best of our knowledge, ALP has not yet been applied to approximately solve MDPs that arise in the real option management of commodity conversion assets.

We focus on the use of ALP for the real option management of commodity storage. We analyze the optimal solution sets of both the ALP dual and the dual of the linear program associated with an MDP, which we refer to as the exact dual. The optimal solutions of the exact dual are in one to one correspondence with the state-action probability distributions induced by optimal policies (Puterman, 1994). In contrast, we find that all the optimal solutions of the ALP dual may correspond to state-action probabilities induced by infeasible policies. ALP can thus yield low quality value function approximations that lead to poor control policies and bounds. Motivated by this insight, we develop a novel approximate dynamic programming approach that (i) addresses this deficiency of the ALP dual by adding constraints to this dual to approximate a key property of the exact dual, and (ii) obtains a value function approximation by solving the ALP relaxation obtained as the primal linear program corresponding to this restricted ALP dual.

We apply our approach using look-up table value function approximations that, in the spirit of LMS, are discrete grids depending on at most two prices in the forward curve (the spot price and the prompt month futures price). We propose two classes of ALP relaxations: *constraint-based* relaxations and *multiplier-based* relaxations. We derive three constraint-based ALP relaxations and one multiplier-based ALP relaxation. Our constraint-based ALP relaxations can be equivalently reformulated as recursive optimization models that we refer to as approximate dynamic programs (ADPs). Two of these ADPs are new. Interestingly, we show that our third constraint-based ALP relaxation yields the LMS ADP, which we label as storage ADP (SADP). We provide a bound on the difference between each ADP value function approximation and the exact value function. We show that this bound tends to zero in limiting regimes of practical relevance. Overall, our analysis provides theoretical support for the use of these ADPs rather than their respective ALPs, as well as the use of our ADP based on the spot and prompt month futures prices instead of the other ADPs.

We numerically evaluate our approach on the LMS natural gas instances. Our results

are encouraging. Our ALP relaxations significantly outperform their corresponding ALPs in terms of both the estimated lower and upper bounds. Our best model is the ADP that uses both the spot and prompt futures prices in its value function approximation. Compared to the other ADPs, this ADP yields better upper bounds and substantially better lower bounds, most of which are near optimal. In addition, it relies less on periodic re-optimizations to obtain near optimal bounds, and is thus a better approximation of the commodity storage MDP than these other models. Our ADPs, that is, constrained-based ALP relaxations, outperform our multiplier-based ALP relaxation in terms of both upper and lower bounds. Moreover, our best ADP is competitive with two state-of-the-art techniques for computing a heuristic operating policy for commodity storage and a lower bound on the value of commodity storage: The practice based rolling intrinsic method (see, e.g., LMS and references therein) and the least squares Monte Carlo approach (Longstaff and Schwartz 2001; see Boogert and De Jong 2008, 2011/12, for commodity storage applications). However, our approach is more directly applicable for dual upper bound estimation because it gives explicit value function approximations while these other methods do not. Our research thus adds to the literature on commodity storage real option valuation and management (see, e.g., Chen and Forsyth 2007, Boogert and De Jong 2008, Thompson et al. 2009, Carmona and Ludkovski 2010, LMS, Secomandi 2010, Birge 2011, Boogert and De Jong 2011/12, Felix and Weber 2012, Secomandi et al. 2012, and Wu et al. 2012).

The use of relaxations in ALP is relatively new and the literature is scant: Petrik and Zilberstein (2009) propose a relaxation method for ALPs that penalizes violated constraints in the objective function; Desai et al. (2012a) relax an ALP by allowing budgeted violation of constraints. In contrast to these authors, we introduce a general approach for deriving ALP relaxations from ALP dual restrictions. Further, the two classes of ALP relaxations that we obtain differ from the ALP relaxations proposed by these authors because they are not based on the idea of budgeted constraint violations.

de Farias and Van Roy (2003) and Desai et al. (2012a) derive error bounds for ALP and ALP relaxations, respectively. In contrast to de Farias and Van Roy (2003) but similar to Desai et al. (2012a), our error bounds are for ALP relaxations. Different from the error bounds of Desai et al. (2012a), ours rely on the recursive structure of the ADPs that correspond to our constraint-based ALP relaxations.

Although our focus is on commodity storage, our proposed methodology is potentially relevant for the approximate solution of intractable MDPs that arise in the real option management of other commodity conversion assets, as well as the valuation of real and financial options that depend on forward curve dynamics; that is, MDPs whose states include both endogenous and exogenous information. Examples include commodity processing assets, energy swing options, put-call Bermudan options, and mortgages and interest rate caps and floors (see, e.g., Longstaff and Schwartz 2001, Jaillet et al. 2004, Cortazar et al. 2008, Devalkar et al. 2011).

We provide background material in §2.2. We discuss the ALP associated with the storage MDP in §2.3 and analyze it in §2.4. We describe our ALP relaxation approach and apply it using look-up table value function approximations in §2.5. We discuss our performance bounds and conduct a computational complexity analysis in §2.6 and §2.7,

respectively. We present our numerical results in §2.8. We conclude in §2.9. Appendix A.1 discusses our multiplier-based ALP relaxation and its numerical performance. Appendix A.2 includes proofs. Appendix A.3 reports the greedy lower bound estimates from the LMS ADP.

2.2 Background Material

In §§2.2.1-2.2.2 we present the commodity storage MDP and the bounding approach that we use. These subsections are in part based on §2 and §4.2 in LMS.

2.2.1 Commodity Storage MDP

A commodity storage asset provides a merchant with the option to purchase-and-inject, store (do-nothing), and withdraw-and-sell a commodity during a predetermined finite time horizon, while respecting injection and withdrawal capacity limits, as well as inventory constraints. The merchant's goal is to maximize the market value of the storage asset. We model this valuation problem as an MDP. Purchases-and-injections and withdrawals-and-sales give rise to cash flows. The storage asset has N possible dates with cash flows. The i-th cash flow occurs at time T_i, $i \in \mathcal{I} := \{0, \ldots, N-1\}$. Each such time is also the maturity of a futures contract. Since the trading times in our model coincide with monthly futures maturity dates, we discretize time into monthly intervals. We denote by $F_{i,j}$ the price at time T_i of a futures contract maturing at time T_j, $j \geq i$. The forward curve is the collection of futures prices $F_i := \{F_{i,i}, F_{i,i+1}, \ldots, F_{i,N-1}\}$. We adopt the convention $F_N \equiv 0$.

Set $\mathcal{I}$ is the stage set. The inventory level at the initial stage 0 is the given singleton x_0. The set of inventory levels at every other stage $i \in \mathcal{I}\backslash\{0\}$ is $\mathcal{X}' := [0, \bar{x}]$, where 0 and $\bar{x} \in \mathbb{R}_+$ represent the minimum and maximum inventory levels, respectively. The (absolute value of) the injection capacity C^I (< 0) and the withdrawal capacity C^W (> 0) represent the maximum amounts that can be injected and withdrawn in between two successive futures contract maturities, respectively. An action a corresponds to an inventory change during this time period. A positive action represents a withdraw-and-sell decision, a negative action a purchase-and-inject decision, and the zero action is the do-nothing decision. Define $\cdot \wedge \cdot \equiv \min\{\cdot, \cdot\}$ and $\cdot \vee \cdot \equiv \max\{\cdot, \cdot\}$. The set of feasible injections, withdrawals, and overall actions are $\mathcal{A}^I(x) := \left[C^I \vee (x - \bar{x}), 0\right]$, $\mathcal{A}^W(x) := \left[0, x \wedge C^W\right]$, and $\mathcal{A}'(x) := \mathcal{A}^I(x) \cup \mathcal{A}^W(x)$, respectively.

The immediate reward from taking action a at time T_i is the function $r(a, s_i)$, where $s_i \equiv F_{i,i}$ is the spot price at this time. The coefficients $\alpha^W \in (0, 1]$ and $\alpha^I \geq 1$ model commodity losses associated with withdrawals and injections, respectively. The coefficients c^W and c^I represent withdrawal and injection marginal costs, respectively. The immediate

reward function is defined as

$$r(a,s) := \begin{cases} (\alpha^I s + c^I)a, & \text{if } a \in \mathbb{R}_-, \\ 0, & \text{if } a = 0, \\ (\alpha^W s - c^W)a, & \text{if } a \in \mathbb{R}_+, \end{cases} \quad \forall s \in \mathbb{R}_+. \tag{2.1}$$

Let Π denote the set of all the feasible storage policies. Given the initial state (x_0, F_0), valuing a storage asset entails finding a feasible policy that achieves the maximum time T_0 $(:= 0)$ market value of this asset in this state, $V_0(x_0, F_0)$. Thus, we are interested in solving the following problem:

$$V_0(x_0, F_0) := \max_{\pi \in \Pi} \sum_{i \in \mathcal{I}} \delta^i \mathbb{E}\left[r(A_i^\pi(x_i^\pi, F_i), s_i) | x_0, F_0\right], \tag{2.2}$$

where δ is the risk free discount factor from time T_i back to time T_{i-1}, $\forall i \in \mathcal{I} \setminus \{0\}$; $\mathbb{E}$ is expectation under the risk neutral measure for the forward curve evolution (this measure is unique when the commodity market is complete, see, e.g., Björk 2004, page 122, which we assume to be the case in this chapter); x_i^π is the inventory level at stage i when using policy π; and $A_i^\pi(\cdot,\cdot)$ is the decision rule of policy π for stage i.

In our MDP formulation, committing on date T_i to perform a physical trade on date $T_j > T_i$ does not add any value, because the payoff from purchasing-and-injecting or withdrawing-and-selling the commodity is linear in the transacted price, given the size of a trade, and we use risk neutral valuation. To illustrate, without loss of generality suppose that $\alpha^W = \alpha^I = 1$ and $c^W = c^I = 0$. Consider the trade that commits at stage i to withdraw-and-sell 1 unit of commodity at stage $j > i$ using a futures contract with price $F_{i,j}$. The stage i value of the resulting stage j payoff is $\delta^{j-i}F_{i,j}$, which is also the stage i value of withdrawing-and-selling 1 unit of commodity at the stage j spot price s_j, because $\delta^{j-i}\mathbb{E}[s_j|F_{i,j}] = \delta^{j-i}F_{i,j}$, which follows from Shreve (2004, page 244).

When C^I, C^W, and $\bar{x}$ are integer multiples of a maximal number $Q \in \mathbb{R}_+$, Lemma 1 in Secomandi et al. (2012) establishes that we can optimally discretize the continuous inventory set $\mathcal{X}'$ into the finite set $\mathcal{X} := \{0, Q, 2Q, \ldots, \bar{x}\}$, and the feasible action set $\mathcal{A}'(x)$ for inventory level $x \in \mathcal{X}$ into the finite set $\mathcal{A}(x) := \{\left[C^I \vee (x - \bar{x})\right], \left[C^I \vee (x - \bar{x})\right] + Q, \ldots, \left[x \wedge C^W\right]\}$. In other words, under this assumption, we can replace $\mathcal{X}'$ and $\mathcal{A}'(x)$ in (2.2) by $\mathcal{X}$ and $\mathcal{A}(x)$, respectively, without sacrificing optimality. Moreover, an optimal policy for problem (2.2) can be obtained by solving a stochastic dynamic program that uses the sets $\mathcal{X}$ and $\mathcal{A}(\cdot)$. Letting $V_i(x_i, F_i)$ be the optimal value function in stage i and state (x_i, F_i), this stochastic dynamic program is

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

$\forall (i, x_i, F_i) \in \mathcal{I} \times \mathcal{X} \times \mathbb{R}_+^{N-i}$, with boundary conditions $V_N(x_N, F_N) := 0, \forall x_N \in \mathcal{X}$. We refer to the stochastic dynamic program (2.3) as the exact dynamic program (EDP).

Consistent with the practice-based literature (Eydeland and Wolyniec 2003, Chapter 5, Gray and Khandelwal 2004, and the discussion in LMS), we assume that EDP is formulated

using a full dimensional model of the risk neutral evolution of the forward curve. An example is the multi-maturity version of the Black (1976) model of futures price evolution used by LMS, which we also use for our computational experiments. In this continuous time model, the time t futures price with maturity at time T_i is denoted $F(t, T_i)$ ($F(T_{i'}, T_i) \equiv F_{i',i}$ for $i', i \in \mathcal{I}$, $i' \leq i$). This price evolves during the interval $[0, T_i]$ as a driftless geometric Brownian motion with maturity specific and constant volatility $\sigma_i > 0$ and standard Brownian motion increment $dZ_i(t)$. The instantaneous correlation between the standard Brownian motion increments $dZ_i(t)$ and $dZ_j(t)$ corresponding to the futures prices with maturities T_i and T_j, $i \neq j$, is $\rho_{ij} \in [-1, 1]$ ($\rho_{ii} = 1$). This model is

$$\frac{dF(t, T_i)}{F(t, T_i)} = \sigma_i dZ_i(t), \ \forall i \in \mathcal{I}, \tag{2.4}$$

$$dZ_i(t)dZ_j(t) = \rho_{ij}dt, \ \forall i, j \in \mathcal{I}, \ i \neq j. \tag{2.5}$$

Property 1, which is easy to verify, states that under this price model *each* futures price in the forward curve evolves in a Markovian fashion.

Property 1. *At a given stage $i \in \mathcal{I}$ and for a given maturity T_j with $j \in \{i+1, \ldots, N-1\}$, the futures price $F_{i,j}$ is sufficient to obtain the probability distribution of the random futures price $F_{i+1,j}$.*

We use Property 1 in §2.2.2 and §2.5 to simplify the computation of expectations. This property also holds for common futures price evolution models used in real option applications (Cortazar and Schwartz, 1994). Model (2.4)-(2.5) can be extended by making time dependent the constant volatilities and instantaneous correlations without affecting Property 1 or our analysis in this chapter.

2.2.2 Bounding Approach

In general, computing an optimal policy for EDP under a price model such as (2.4)-(2.5) is intractable. We now describe Monte Carlo simulation procedures for estimating lower and upper bounds on the EDP optimal value function in the initial stage and state given an approximation of the EDP value function. The lower bound estimation relies on the Monte Carlo simulation of a greedy heuristic policy given this value function approximation (see §6.1.1 in Bertsekas 2007 and Powell 2011). The upper bound estimation applies the information relaxation and duality approach (see Brown et al. 2010 and references therein) based on this value function approximation. We illustrate these procedures using the value function approximation $\hat{V}_i(x_i, s_i)$, which we assume is available. This function only uses the spot price s_i from the forward curve F_i. Nevertheless, the same approach extends in a straightforward manner to value function approximations that depend on a larger subset of prices in this forward curve.

Consider the lower bound estimation. Given an inventory level x_i and a forward curve F_i in stage i, we use $\hat{V}_i(x_i, s_i)$ to compute a greedy action by solving the greedy optimization

problem

$$\max_{a_i \in \mathcal{A}(x_i)} r(a_i, s_i) + \delta \mathbb{E}\left[\hat{V}_{i+1}\left(x_i - a_i, s_{i+1}\right) | F_{i,i+1}\right], \tag{2.6}$$

where $F_{i,i+1}$ is sufficient for computing the expectation by Property 1. We obtain (2.6) from (2.3) by replacing $V_{i+1}(\cdot,\cdot)$ with $\hat{V}_{i+1}(\cdot,\cdot)$ and F_i with $F_{i,i+1}$. In computations, we numerically approximate this expectation using Rubinstein (1994) lattices, as discussed in Appendix 2.7. We apply the action $a_i(x_i, s_i)$ computed in (2.6) (breaking ties by picking $a_i(x_i, s_i)$ such that the inventory change $|a_i(x_i, s_i)|$ is minimized), and sample the forward curve F_{i+1} to obtain the new state $(x_i - a_i(x_i, s_i), F_{i+1})$ in stage $i+1$. Starting from the initial stage and state, we continue in this fashion until we reach time T_{N-1}. We then discount back to time 0 the cash flows generated by this process, and add them up. We repeat this process for multiple forward curve samples and average the sample discounted total cash flows to estimate the value of the greedy policy, that is, the policy defined by the greedy action in each stage and state. This provides us with an estimate of a (greedy) lower bound on the value of storage, that is, $V_0(x_0, F_0)$.

When a value function approximation is computed by an approximate dynamic programming model it is typically possible to generate an improved greedy lower bound estimate by sequentially reoptimizing this model to update its value function approximations within the Monte Carlo simulation used for lower bound estimation (Secomandi, 2008). Specifically, solving such a model at time T_i yields value function approximations for stages i through $N-1$. However, we only implement the greedy action induced by the stage i value function approximation. At time T_{i+1}, we reoptimize the "residual" model, that is, the one defined over the remaining stages $i+1$ through $N-1$, given the inventory level resulting from performing this action and the newly available forward curve. We repeat this procedure until time T_{N-1}. Repeating this process over multiple forward curve samples allows us to estimate a reoptimized greedy lower bound.

For upper bound estimation, we sample a sequence of spot price and prompt month futures price pairs $P_0 := ((s_i, F_{i,i+1}))_{i=0}^{N-1}$ starting from the forward curve F_0 at time 0. We use our value function approximation $\hat{V}_{i+1}(x_{i+1}, s_{i+1})$ to define the following dual penalty for executing the feasible action a_i in stage i and state (x_i, F_i) given knowledge of the prompt month futures price $F_{i,i+1}$ and the stage $i+1$ spot price s_{i+1}:

$$\hat{p}_i(x_i, a_i, s_{i+1}, F_{i,i+1}) := \hat{V}_{i+1}(x_i - a_i, s_{i+1}) - \mathbb{E}\left[\hat{V}_{i+1}(x_i - a_i, s_{i+1}) | F_{i,i+1}\right], \tag{2.7}$$

where $F_{i,i+1}$ is sufficient for computing the expectation by Property 1. For computational purposes, we numerically approximate the expectation in (2.7) using Rubinstein (1994) lattices (see Appendix 2.7). This penalty approximates the value of knowing the next stage spot price when performing this action. Then, we solve the following deterministic dynamic program given the sequence P_0 (see Brown et al. 2010 and LMS):

$$U_i(x_i; P_0) \quad = \quad \max_{a_i \in \mathcal{A}(x)} r(a_i, s_i) - \hat{p}_i(x_i, a_i, s_{i+1}, F_{i,i+1}) + \delta U_{i+1}(x_i - a_i; P_0), \tag{2.8}$$

$\forall(i, x_i) \in \mathcal{I} \times \mathcal{X}$, with boundary conditions $U_N(x_N; P_0) := 0$, $\forall x_N \in \mathcal{X}$. In (2.8), the immediate reward $r(a_i, s_i)$ is modified by the penalty $\hat{p}_i(x_i, a_i, s_{i+1}, F_{i,i+1})$ for using the future information available in P_0. We solve a collection of deterministic dynamic programs (2.8), each one corresponding to a sample sequence P_0. We estimate a dual upper bound on the value of storage as the average of the value functions of these deterministic dynamic programs in the initial stage and state; that is, we estimate $\mathbb{E}\left[U_0(x_0; P_0)|F_0\right]$, where the expectation is taken with respect to the risk neutral distribution of the random sequence P_0 conditional on F_0.

2.3 ALP

In this section we apply the ALP approach for heuristically solving MDPs with finite state and action spaces (Schweitzer and Seidmann, 1985, de Farias and Van Roy, 2003). EDP has a finite action space but its state space is in part continuous. To be able to apply the ALP approach, we discretize the forward curve part of the EDP state to obtain a discretized version of EDP (DDP). We let $\mathcal{F}_i \subset \mathbb{R}_+^{N-i}$ represent a finite set of forward curves at time T_i. We denote by $\mathcal{F}_{i,j} \subset \mathbb{R}_+$ the finite set of values of the futures price $F_{i,j}$ in the forward curve $F_i \in \mathcal{F}_i$. We denote by $\{\Pr(F_{i+1}|F_i), \forall F_{i+1} \in \mathcal{F}_{i+1}\}$ the probability mass function of the random vector F_{i+1} on the set $\mathcal{F}_{i+1}$ conditional on the forward curve $F_i \in \mathcal{F}_i$. We make Assumption 1 to ensure that all the forward curves in our discretized sets have positive probability.

Assumption 1. $\Pr(F_{i+1}|F_i) > 0$ $\forall(F_i, F_{i+1}) \in \mathcal{F}_i \times \mathcal{F}_{i+1}$.

To simplify the notational burden, in the rest of this chapter we omit the sets indexing a tuple. For example, we write (i, x_i, F_i, a_i) in lieu of $(i, x_i, F_i, a_i) \in \mathcal{I} \times \mathcal{X} \times \mathcal{F}_i \times \mathcal{A}(x_i)$. We write $(\cdot)_{-(i)}$ to indicate that i is excluded from $\mathcal{I}$ in the tuple ground set. Replacing the continuous forward curve sets that define EDP with the discretized sets discussed in this section yields DDP. Letting $V_i^D(x_i, F_i)$ be the DDP optimal value function in stage i and state (x_i, F_i), DDP is

$$V_i^D(x_i, F_i) = \max_{a_i} r(a_i, s_i) + \delta\mathbb{E}\left[V_{i+1}^D\left(x_i - a_i, F_{i+1}\right)|F_i\right], \tag{2.9}$$

$\forall(i, x_i, F_i)$, with boundary conditions $V_N^D(x_N, F_N) := 0$, $\forall x_N$. The expectation in (2.9) is expressed with respect to the probability mass function $\{\Pr(F_{i+1}|F_i), \forall F_{i+1}\}$, even though the notation does not make it explicit.

It is well known that DDP can be reformulated as a linear program, which we refer to as the exact primal linear program (PLP; Manne 1960, Puterman 1994, §6.9; also see §5.2 of the thesis). PLP has a variable for every stage and state and a constraint for every stage, state, and action. We refer to the PLP dual as DLP (see Puterman 1994, page 223; also see §5.2 of the thesis).

Solving PLP or DLP is typically intractable due to the exponential number of variables and constraints as a function of the number of futures prices in the forward curve. Computational tractability dictates approximating these models.

Following the ADP literature (Schweitzer and Seidmann, 1985, de Farias and Van Roy, 2003), PLP can be approximated by replacing its variables by lower dimensional approximations defined as linear combinations of a manageable number of basis functions. Let $\psi_{i,x_i,b} : \mathbb{R}^{N-i} \mapsto \mathbb{R}$ be the b-th basis function corresponding to the pair (i, x_i). There are B_i basis functions for each stage i, that is, $b \in \{1, \ldots, B_i\}$. The weight associated with the b-th basis function for each pair (i, x_i) is $\beta_{i,x_i,b} \in \mathbb{R}$. The value function approximation is $\sum_b \psi_{i,x_i,b}(F_i)\beta_{i,x_i,b}$. Since the stage 0 state space is the singleton (x_0, F_0), we choose $B_0 = 1$ and $\psi_{0,x_0,1}(F_0) = 1$ without loss of generality. The value function approximation weights are computed by solving the following ALP:

$$\min_{\beta} \beta_{0,x_0,1} \tag{2.10}$$

$$\text{s.t.} \quad \sum_b \psi_{N-1,x_{N-1},b}(F_{N-1})\beta_{N-1,x_{N-1},b} \geq r(a_{N-1}, s_{N-1}), \forall (x_{N-1}, F_{N-1}, a_{N-1}), \tag{2.11}$$

$$\sum_b \psi_{i,x_i,b}(F_i)\beta_{i,x_i,b} \geq r(a_i, s_i) + \delta \mathbb{E}\left[\sum_b \psi_{i+1,x_i-a_i,b}(F_{i+1})\beta_{i+1,x_i-a_i,b} \mid F_i\right], \quad \forall (i, x_i, F_i, a_i)_{-(N-1)}. \tag{2.12}$$

The objective function (2.10) minimizes the approximate value function corresponding to the initial stage and state. The ALP constraints can be obtained from DDP as follows: For each triple (i, x_i, F_i) express the maximization over the set $\mathcal{A}(x_i)$ in (2.9) as $|\mathcal{A}(x_i)|$ inequalities, one for each a_i, and then replace $V_i(x_i, F_i)$ by $\sum_b \psi_{i,x_i,b}(F_i)\beta_{i,x_i,b}$. Constraints (2.12) ensure that the ALP value function approximation is an upper bound on the DDP value function approximation at every stage and state (de Farias and Van Roy, 2003).

Let $\mathbb{1}(\cdot)$ represent the indicator function that evaluates to 1 when the expression inside its parentheses is true and zero otherwise. Denoting by $w_i(x_i, F_i, a_i)$ the dual variable of the ALP constraint corresponding to (i, x_i, F_i, a_i), the dual of this ALP (DALP) is

$$\max_{w} \sum_{(i,x_i,F_i,a_i)} r(a_i, s_i) w_i(x_i, F_i, a_i) \tag{2.13}$$

$$\text{s.t.} \quad \sum_{a_0} w_0(x_0, F_0, a_0) = 1, \tag{2.14}$$

$$\sum_{F_i} \psi_{i,x_i,b}(F_i) \left[\sum_{a_i} w_i(x_i, F_i, a_i)\right] =$$

$$\sum_{F_i} \psi_{i,x_i,b}(F_i) \left[\delta \sum_{F_{i-1}} \Pr(F_i|F_{i-1}) \sum_{(x_{i-1},a_{i-1})} \mathbb{1}(x_{i-1} - a_{i-1} = x_i) w_{i-1}(x_{i-1}, F_{i-1}, a_{i-1})\right], \quad \forall (i, x_i, b)_{-(0)}, \tag{2.15}$$

$$w_i(x_i, F_i, a_i) \geq 0, \forall (i, x_i, F_i, a_i). \tag{2.16}$$

It can be verified that the DALP objective function (2.13) and the constraints (2.14) and (2.16) are identical to the corresponding DLP objective function and constraints (the DLP

formulation is not given here for brevity). In contrast, the flow conservation constraints (2.15) differ from the DLP flow conservation constraints. Specifically, for each pair (i, x_i) DALP has one constraint (2.15) for each basis function, and the DALP constraint corresponding to the triple (i, x_i, b) is a linear combination of the DLP flow conservation constraints corresponding to the triples in the set $\{(i, x_i, F_i), \forall F_i\}$ taken using the coefficients $\psi_{i,x_i,b}(F_i)$. Therefore, DALP is a relaxation of DLP.

2.4 ALP Analysis

In this section we analyze the ALP and DALP models introduced in §2.3. We begin this analysis by discussing the relationship between feasible DLP solutions and feasible DDP policies. Following a DDP feasible policy starting from the initial stage and state induces a collection of probability mass functions defined over the feasible state and action spaces in each stage. Given a feasible DDP policy π and such a probability mass function, we denote by $\Pr^{\pi}(x_i, F_i, a_i)$ the probability of visiting state (x_i, F_i) in stage i and taking action a_i under policy π (this probability depends on the initial stage and state but we suppress this dependence from our notation for expositional convenience). Therefore, a feasible DDP policy π can be equivalently specified by the set of probabilities $\{\Pr^{\pi}(x_i, F_i, a_i), \forall (i, x_i, F_i, a_i)\}$. It follows from Theorem 6.9.1 in Puterman (1994, page 224) that the set of feasible DLP solutions encodes the set of feasible DDP policies: There is a one to one correspondence between feasible DDP policies and feasible DLP solutions. In particular, for every feasible DDP policy π there exists a feasible DLP solution u such that

$$u_i(x_i, F_i, a_i) = \delta^i \Pr^{\pi}(x_i, F_i, a_i), \quad \forall (i, x_i, F_i, a_i). \tag{2.17}$$

It follows from the equalities (2.17) that every optimal DDP policy is related to an optimal DLP solution in this manner.

Let $\Pr^{*}(x_i, F_i, a_i)$ denote the probability of visiting state (x_i, F_i) in stage i and taking action a_i under an optimal DDP policy. For this optimal policy, we now investigate whether there exists an optimal DALP solution w^* that satisfies a condition analogous to (2.17), that is,

$$w_i^*(x_i, F_i, a_i) = \delta^i \Pr^{*}(x_i, F_i, a_i), \quad \forall (i, x_i, F_i, a_i). \tag{2.18}$$

We make Assumption 2 to ensure feasibility of ALP (de Farias and Van Roy 2003):

Assumption 2. $\psi_{i,x_i,1} = 1 \ \forall (i, x_i)$.

We denote by $\mathcal{F}_i^{=}(\beta^*)$ the set of stage i forward curves for which at least one ALP constraint corresponding to stage i holds as an equality at an ALP optimal solution β^*. Proposition 1 is useful to identify possible violations of (2.18) by the set of optimal DALP solutions.

Proposition 1. *Suppose Assumption 2 holds. For every feasible DALP solution w it holds that*

$$\sum_{(x_i, F_i, a_i)} w_i(x_i, F_i, a_i) = \delta^i, \forall i. \tag{2.19}$$

Moreover, for every optimal DALP solution w^* it holds that

$$\sum_{(x_i,a_i)} w_i^*(x_i, F_i, a_i) = 0, \forall (i, F_i) \in \mathcal{I} \times \{\mathcal{F}_i \setminus \mathcal{F}_i^{=}(\beta^*)\}. \tag{2.20}$$

Condition (2.19) states that a feasible DALP solution specifies a collection of discounted probability mass functions defined over the DDP state and action spaces in each stage. Suppressing its dependence on F_0 for notational convenience, let $\Pr(F_i)$ denote the probability of observing the forward curve F_i. Condition (2.20) implies that the collection of probability mass functions corresponding to an optimal DALP solution violates (2.18) when the set $\mathcal{F}_i^{=}(\beta^*)$ is a proper subset of $\mathcal{F}_i$, because the conditions

$$\sum_{(x_i,a_i)} w_i^*(x_i, F_i, a_i) = \delta^i \Pr(F_i), \quad \forall (i, F_i), \tag{2.21}$$

obtained by summing both sides of (2.18) over (x_i, a_i), are necessary for the validity of (2.18) and $\Pr(F_i) > 0$ by Assumption 1. In other words, comparing (2.20) and (2.21) shows that the collection of discounted probability mass functions associated with an arbitrary optimal DALP solution can be distorted relative to the analogous collection associated with an optimal DDP policy.

These distortions can lead to pathological cases. To elaborate, let $\mathbb{P}$ be the probability mass function defined by the probabilities $\Pr(F_i)$. Suppose that the forward curves in $\mathcal{F}_i^{=}(\beta^*)$ lie in the right tail of this probability mass function and $\sum_{F_i \in \mathcal{F}_i^{=}(\beta^*)} \Pr(F_i) = \epsilon$ for some positive ϵ much smaller than one. In this case, the probability distortion implied by (2.20) is large and the value function approximation is determined by extreme forward curves under $\mathbb{P}$. Such a situation is evidently undesirable for bounding purposes.

In contrast to such pathological cases, Proposition 2 states that when at least one optimal DDP policy and one optimal DALP solution satisfy (2.18) every ALP optimal solution (β^*) enjoys a desirable property. (The equality $\mathcal{F}^{=}(\beta^*) \equiv \mathcal{F}_i$ is necessary for (2.18) to hold, which follows form the proof of Proposition 2 and complementary slackness.) We denote by $\Pi^g(\beta^*)$ the set of greedy policies induced by the value function approximation specified by β_i^*.

Proposition 2. *If an optimal DDP policy and an optimal DALP solution satisfy (2.18), then for every ALP optimal solution β^* there exists a deterministic optimal DDP policy π^* that is greedy with respect to the value function approximation defined by β^*; that is, $\pi^* \in \Pi^g(\beta^*)$.*

Proposition 2 suggests that, if possible, it may be useful to require an optimal DALP solution to be consistent, in the sense of (2.18), with a deterministic optimal DDP policy.

2.5 ALP Relaxations

In §2.5.1 we present our approach to derive ALP relaxations. In §2.5.2 we formulate and analyze an ALP based on a look-up table value function approximation. In §2.5.3 we apply our relaxation approach to this ALP, and a variant thereof, to derive our constraint-based ALP relaxations. Multiplier-based ALP relaxations are discussed in Appendix A.1.

2.5.1 Approach for Deriving ALP Relaxations

Motivated by our analysis in §2.4, we would like to add constraints to DALP requiring its feasible solutions to match the discounted probability mass function induced by an optimal policy for DDP. The specific constraints that we would like to add to DALP are

$$w_i(x_i, F_i, a_i) = \delta^i \Pr^*(x_i, F_i, a_i), \quad \forall (i, x_i, F_i, a_i). \tag{2.22}$$

Although the probability on the right hand side of (2.22) is unknown in applications, we proceed temporarily ignoring this important fact.

Let $d_i(x_i, F_i, a_i)$ be the dual variable associated with the constraint in (2.22) corresponding to (i, x_i, F_i, a_i). The dual of the DALP restricted by constraints (2.22) is the ALP relaxation

$$\min_{\beta, d} \beta_{0,x_0,1} + \sum_{(i,x_i,F_i,a_i)} \delta^i \Pr^*(x_i, F_i, a_i) d_i(x_i, F_i, a_i) \tag{2.23}$$

$$\text{s.t. } \beta_N = 0, \tag{2.24}$$

$$\sum_b \psi_{i,x_i,b}(F_i)\beta_{i,x_i,b} + d_i(x_i, F_i, a_i) \geq r(a_i, s_i) + \delta \mathbb{E}\left[\sum_b \psi_{i+1,x_i-a_i,b}(F_{i+1})\beta_{i+1,x_i-a_i,b} \mid F_i\right], \quad \forall (i, x_i, F_i, a_i). \tag{2.25}$$

Compared to ALP, that is, (2.10)-(2.12), the linear program (2.23)-(2.25) includes the variables $d_i(x_i, F_i, a_i)$ (i) on the left hand side of its constraints (2.25) and (ii) in a term in its objective function that penalizes relaxations of constraints (2.25) when $d_i(x_i, F_i, a_i)$ is strictly positive and rewards the tightening of these constraints when $d_i(x_i, F_i, a_i)$ is strictly negative.

The ALP relaxation (2.23)-(2.25) is impractical because it depends on the unknown terms $\Pr^*(\cdot, \cdot, \cdot)$, in addition to having exponentially many variables $d_i(x_i, F_i, a_i)$ and constraints (2.25). We thus focus on deriving practical ALP relaxations by adding constraints to DALP that approximate (2.22) and avoid this exponential growth in the number of variables and constraints in the resulting ALP relaxation. Our approach is summarized in Figure 2.1. The solid arrows in this figure show the process of constructing an ALP relaxation: (i) Starting from ALP, (ii) we formulate DALP, (iii) restrict it in the stated manner, and (iv) take the dual of this restriction to obtain an ALP relaxation.

Our approach relaxes the ALP constraints (2.12), which ensure that the ALP value function is an upper bound on the DDP value function at each stage and state (de Farias

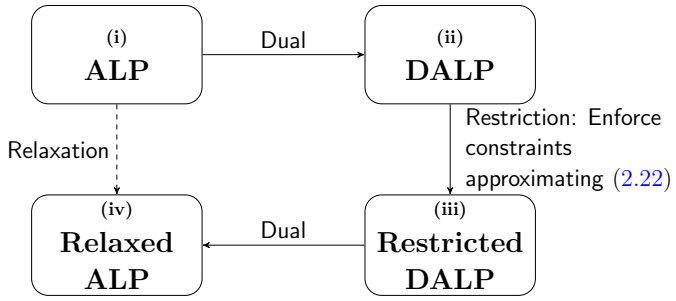


Figure 2.1: Schematic illustration of the ALP relaxation framework.

and Van Roy, 2003). Therefore, unlike ALP, the value function approximation obtained by solving an ALP relaxation may not provide an upper bound on the DDP value function at every stage and state. Nevertheless, if the constraints used to restrict DALP are implied by (2.22), it can be verified that (i) the optimal objective function value of the ALP relaxation provides an upper bound on the DDP optimal value function at the initial stage and state, $V_0^D(x_0, F_0)$, and (ii) this upper bound is no worse than the corresponding ALP upper bound.

2.5.2 An ALP Based on a Look-up Table Value Function Approximation

In the rest of this chapter, we focus on using low-dimensional look-up table value function approximations: *Discrete grids* that in each stage depend on the inventory level and at most the first two futures prices in the forward curve. In light of Property 1, these look-up table value function approximations are appealing because they result in a dimensionality reduction that makes tractable (i) computing the expectation in (2.25) and (ii) solving the resulting linear program.

Our starting point is an ALP formulated using the look-up table value function approximation $\phi_i(x_i, s_i)$, which in stage i depends on the inventory x_i and the spot price s_i, as in LMS. This look-up table contains the weights associated with indicator basis functions, that is, it defines a value function approximation for the pair (i, x_i) as $\sum_{\bar{s}_i} \mathbb{1}(s_i = \bar{s}_i)\phi_i(x_i, s_i)$. By Property 1, the expectation $\mathbb{E}[\phi_{i+1}(\cdot, s_{i+1})|F_i]$ can be simplified to $\mathbb{E}[\phi_{i+1}(\cdot, s_{i+1})|F_{i,i+1}]$. The corresponding ALP, which has a much smaller number of constraints than the ALP (2.10)-(2.12), is

$$\min_{\phi} \phi_0(x_0, s_0) \tag{2.26}$$

$$\text{s.t. } \phi_{N-1}(x_{N-1}, s_{N-1}) \geq r(a_{N-1}, s_{N-1}), \quad \forall (x_{N-1}, s_{N-1}, a_{N-1}), \tag{2.27}$$

$$\phi_i(x_i, s_i) \geq r(a_i, s_i) + \delta \mathbb{E}\left[\phi_{i+1}(x_i - a_i, s_{i+1})|F_{i,i+1}\right], \quad \forall (i, x_i, s_i, F_{i,i+1}, a_i)_{-(N-1)}. \tag{2.28}$$

Proposition 3 states that the optimal value function of the following ADP, labeled

ADP0, is an optimal solution to (2.26)-(2.28):

$$\phi_i^{ADP0}(x_i, s_i) = \max_{F_{i,i+1}} \left\{ \max_{a_i} r(a_i, s_i) + \delta \mathbb{E}\left[\phi_{i+1}^{ADP0}(x_i - a_i, s_{i+1}) | F_{i,i+1}\right\}\right], \tag{2.29}$$

$\forall (i, x_i, s_i)$, with $\phi_N^{ADP0}(x_N, s_N) := 0$, $\forall x_N$.

Proposition 3. *The terms $\phi_i^{ADP0}(x_i, s_i)$, $\forall (i, x_i, s_i)$, optimally solve (2.26)-(2.28).*

ADP0 has two maximizations: The first over the price $F_{i,i+1}$ and the second over the action a_i. The second maximization is analogous to the maximization in DDP (see (2.9)). By Proposition 3, the first maximization implies that the ALP (2.26)-(2.28) treats the exogenous futures price $F_{i,i+1}$ as a choice, which is unrealistic. Moreover, given a pair (x_i, s_i), we have verified numerically on the instances discussed in §2.8 that the maximizer in this optimization is typically the largest price in the set $\mathcal{F}_{i,i+1}$, which has a low probability of occurring given s_i according to $\mathbb{P}$. In other words, on our instances the ALP (2.26)-(2.28) yields value function approximations that are determined by unlikely prompt-month futures prices. This situation illustrates the pathological case discussed at the end of §2.4. Therefore, the ALP (2.26)-(2.28) seems a particularly poor model. Our ALP relaxation approach addresses this issue.

2.5.3 Constraint-based ALP Relaxations

We now discuss ALP relaxations that use constraints to control the extent to which an ALP is relaxed. We thus label the relaxations in this class as *constraint-based* ALP relaxations. We derive two relaxations of the ALP (2.26)-(2.28) and a relaxation of an ALP analogous to (2.26)-(2.28) but formulated using a look-up table value function approximation that in every stage also depends on the prompt-month futures price. These ALP relaxations have equivalent ADP reformulations that: (i) allow us to interpret how these relaxations overcome the ADP0 pathology pointed out at the end of §2.5.2; (ii) have optimal policies that share the structure of DDP optimal policies (this is easy to verify; see Lemma 1 of LMS for the DDP optimal policy structure); and (iii) are easier to solve using backward recursion than their corresponding linear programming formulations, because of the low dimensionality of the endogenous state and action spaces – this solution approach is thus suitable for problems with this feature.

We denote $\{F_{i,i+2}, \ldots, F_{i,N-1}\}$ as $F_i \backslash \{s_i, F_{i,i+1}\}$ and the sum $\sum_{F_i \backslash \{s_i, F_{i,i+1}\}} \Pr^*(x_i, F_i, a_i)$ as $\Pr^*(x_i, s_i, F_{i,i+1}, a_i)$. We use $w_i(x_i, s_i, F_{i,i+1}, a_i)$ to indicate the DALP variables, which is consistent with how the constraints of the ALP (2.26)-(2.28) are expressed. The analogue of constraints (2.22) is

$$w_i(x_i, s_i, F_{i,i+1}, a_i) = \delta^i \Pr^*(x_i, s_i, F_{i,i+1}, a_i), \quad \forall (i, x_i, s_i, F_{i,i+1}, a_i). \tag{2.30}$$

Constraints (2.30) are derived from constraints (2.22) by (i) summing the latter constraints over the futures prices in set $F_i \backslash \{s_i, F_{i,i+1}\}$ and (ii) replacing the sum $\sum_{F_i \backslash \{s_i, F_{i,i+1}\}} w_i(x_i, F_i, a_i)$ by $w_i(x_i, s_i, F_{i,i+1}, a_i)$. We obtain the constraints that we add to DALP by approximating

constraints (2.30) in three steps. In the first step we sum both sides of constraints (2.30) over the feasible actions:

$$\sum_{a_i} w_i(x_i, s_i, F_{i,i+1}, a_i) = \delta^i \Pr^*(x_i, s_i, F_{i,i+1}), \quad \forall (i, x_i, s_i, F_{i,i+1}). \tag{2.31}$$

In the second step we express the discounted probabilities $\delta^i \Pr^*(x_i, s_i, F_{i,i+1})$ on the right hand side of (2.31) as $\delta^i \Pr^*(F_{i,i+1}|x_i, s_i) \cdot \Pr^*(x_i, s_i)$ and replace the term $\delta^i \Pr^*(x_i, s_i)$ by the new variable $\theta_i(x_i, s_i)$ to get the constraints

$$\sum_{a_i} w_i(x_i, s_i, F_{i,i+1}, a_i) = \Pr^*(F_{i,i+1}|x_i, s_i)\theta_i(x_i, s_i), \quad \forall (i, x_i, s_i, F_{i,i+1}). \tag{2.32}$$

In the third step we approximate the unknown probability $\Pr^*(F_{i,i+1}|x_i, s_i)$ on the right hand side of (2.32) by a known probability $p(F_{i,i+1}|s_i, F_0)$, which we discuss below. The specific constraints that we add to DALP are

$$\sum_{a_i} w_i(x_i, s_i, F_{i,i+1}, a_i) = p(F_{i,i+1}|s_i, F_0)\theta_i(x_i, s_i), \ \forall (i, x_i, s_i, F_{i,i+1}). \tag{2.33}$$

Constraints (2.32) are implied by (2.22), and are thus satisfied by at least one optimal solution of the exact dual, DLP. In contrast, constraints (2.33) may not be implied by (2.22), because the probability $p(F_{i,i+1}|s_i, F_0)$ does not depend on the stage i inventory level obtained by an optimal policy. More specifically, $\Pr^*(F_{i,i+1}|x_i, s_i)$ may differ from $\Pr(F_{i,i+1}|s_i)$ in general because the (random) inventory level reached in stage i by following an optimal policy may be correlated with the prompt-month futures price $F_{i,i+1}$. As a consequence, the optimal objective function of the resulting ALP relaxation may not be an upper bound on the corresponding DDP optimal value function at the initial stage and state, $V_0^D(x_0, F_0)$.

The ALP relaxation obtained by adding constraints (2.33) to DALP is

$$\min_{\beta, d} \phi_0(x_0, s_0) \tag{2.34}$$

$$\text{s.t. } \phi_N(x_N, s_N) = 0, \forall x_N, \tag{2.35}$$

$$\phi_i(x_i, s_i) + d_i(x_i, s_i, F_{i,i+1}) \geq r(a_i, s_i) + \delta \mathbb{E}\left[\phi_{i+1}(x_i - a_i, s_{i+1})|F_{i,i+1}\right], \quad \forall (i, x_i, s_i, F_{i,i+1}, a_i), \tag{2.36}$$

$$\sum_{F_{i,i+1}} p(F_{i,i+1}|s_i, F_0) d_i(x_i, s_i, F_{i,i+1}) = 0, \forall (i, x_i, s_i). \tag{2.37}$$

This relaxed ALP differs from the ALP relaxation (2.23)-(2.25) in two aspects. First, compared to the decision variables $d_i(x_i, F_i, a_i)$, the variables $d_i(x_i, s_i, F_{i,i+1})$ do not depend on the action a_i and the set of futures prices $F_i \setminus \{s_i, F_{i,i+1}\}$. Second, the amount of relaxation in (2.34)-(2.37) is controlled by the constraints (2.37), whereas in (2.23)-(2.25) it is regulated by the second term in the objective function (2.23). Specifically, the constraints (2.37) set to zero the weighted average of the variables in set $\{d_i(x_i, s_i, F_{i,i+1}), \forall F_{i,i+1}\}$ where

the weights are the probabilities in set $\{p(F_{i,i+1}|s_i, F_0), \forall F_{i,i+1}\}$ for each pair (i, x_i, s_i). Hence, large relaxations of a subset of constraints (2.37) corresponding to forward curves that occur with low probability (under $\mathbb{P}$) can be balanced by small restrictions of a subset of constraints (2.37) corresponding to forward curves that happen with high probability.

The ALP relaxation of Desai et al. (2012a) also includes variables that relax the ALP constraints, but these variables are nonnegative and hence only capture violations of ALP constraints. In contrast to model (2.34)-(2.37), the model of these authors uses a constraint to impose a budget on an average of the constraint violations.

Proposition 4 states that an optimal solution to the constraint-based ALP relaxation (2.34)-(2.37) can be computed by solving the following ADP, which depends on the conditional probability mass function $\{p(F_{i,i+1}|s_i, F_0), \forall F_{i,i+1}\}$:

$$\phi_i^p(x_i, s_i) = \sum_{F_{i,i+1}} p(F_{i,i+1}|s_i, F_0) \left[\max_{a_i} r(a_i, s_i) + \delta \mathbb{E}\left[\phi_{i+1}^p(x_i - a_i, s_{i+1})\middle|F_{i,i+1}\right]\right], \quad (2.38)$$

$\forall (i, x_i, s_i)$, with $\phi_N^p(x_N, s_N) := 0$, $\forall x_N$. We define $d_i^p(x_i, s_i, F_{i,i+1}), \forall (i, x_i, s_i, F_{i,i+1})$, as

$$\max_{a_i} \left\{r(a_i, s_i) + \delta \mathbb{E}\left[\phi_{i+1}^p(x_i - a_i, s_{i+1})|F_{i,i+1}\right]\right\} - \phi_i^p(x_i, s_i).$$

Proposition 4. *The terms $\phi_i^p(x_i, s_i)$ and $d_i^p(x_i, s_i, F_{i,i+1})$ optimally solve (2.34)-(2.37).*

In light of Proposition 4, comparing (2.29) and (2.38) reveals that the constraint-based ALP relaxation (2.34)-(2.37) effectively replaces the maximization over the set $\mathcal{F}_{i,i+1}$ in (2.29) with an expectation taken with respect to the probability mass function $\{p(F_{i,i+1}|s_i, F_0), \forall F_{i,i+1}\}$.

Different constraint-based ALP relaxations can be obtained from (2.38) by varying the choice of the conditional probability mass function $\{p(F_{i,i+1}|s_i, F_0), \forall F_{i,i+1}\}$. We consider the following choices for $p(F_{i,i+1}|s_i, F_0)$:

$$\Pr(F_{i,i+1}|s_i, F_{0,i+1}), \quad (2.39)$$

$$\mathbb{1}(F_{i,i+1} = \mathbb{E}[F_{i,i+1}|s_i, F_{0,i+1}]); \quad (2.40)$$

The term $\Pr(F_{i,i+1}|s_i, F_{0,i+1})$ is the conditional probability of $F_{i,i+1}$ given s_i and $F_{0,i+1}$ under the forward curve probability mass function discussed at the beginning of §2.3; $\mathbb{1}(F_{i,i+1} = \mathbb{E}[F_{i,i+1}|s_i, F_{0,i+1}])$ is a degenerate conditional probability mass function on the set $\mathcal{F}_{i,i+1}$ that places all its mass on the value $\mathbb{E}[F_{i,i+1}|s_i, F_{0,i+1}]$ (this expectation is also under the forward curve probability mass function discussed in §2.3 and is assumed to be in set $\mathcal{F}_{i,i+1}$).

Using (2.39), that is, letting $p(F_{i,i+1}|s_i, F_0) = \Pr(F_{i,i+1}|s_i, F_{0,i+1})$ in (2.38), gives the following constraint-based ALP relaxation:

$$\phi_i^{SADP}(x_i, s_i) = \mathbb{E}\left[\max_{a_i} r(a_i, s_i) + \delta \mathbb{E}\left[\phi_{i+1}^{SADP}(x_i - a_i, s_{i+1})|F_{i,i+1}\right] \mid s_i, F_{0,i+1}\right], \quad (2.41)$$

$\forall (i, x_i, s_i)$, with $\phi_N^{SADP}(x_N, s_N) := 0$, $\forall x_N$. This is the ADP of LMS, that is, SADP

(hence the superscript on ϕ_i and ϕ_{i+1} in (2.41); LMS do not show that SADP is an ALP relaxation).

Using (2.40), that is, letting $p(F_{i,i+1}|s_i, F_0) = \mathbb{1}(F_{i,i+1} = \mathbb{E}[F_{i,i+1}|s_i, F_{0,i+1}])$ in (2.38), gives ADP1:

$$\phi_i^{ADP1}(x_i, s_i) = \max_{a_i} r(a_i, s_i) + \delta \mathbb{E}\left[\phi_{i+1}^{ADP1}(x_i - a_i, s_{i+1})|\mathbb{E}[F_{i,i+1}|s_i, F_{0,i+1}]\right], \tag{2.42}$$

$\forall (i, x_i, s_i)$, with $\phi_N^{ADP1}(x_N, s_N) := 0$, $\forall x_N$. ADP1 is a new model.

Our two choices of $p(F_{i,i+1}|s_i, F_{0,i+1})$ can be interpreted as restricting the amount of information revealed by the stochastic process (2.4)-(2.5) that an ALP relaxation uses to obtain a value function approximation. In both cases, at time T_i the joint probability mass function of the price pair $(s_i, F_{i,i+1})$ conditional on $(F_{0,i}, F_{0,i+1})$ is replaced by the marginal probability mass function of s_i given $F_{0,i}$ and a conditional probability mass function for $F_{i,i+1}$ given $(s_i, F_{0,i+1})$: The one based on (2.39) for SADP and (2.40) for ADP1.

Proposition 5 provides some support for these choices of $p(F_{i,i+1}|s_i, F_{0,i+1})$: They imply conditions analogous to properties satisfied by optimal DLP solutions. (These properties may not characterize optimal DALP solutions.)

Proposition 5. *Let w be a feasible solution to the restricted DALP.*

(a) If $p(F_{i,i+1}|s_i, F_0) = \Pr(F_{i,i+1}|s_i, F_{0,i+1})$, then w matches the discounted probability mass function of the price pair $(s_i, F_{i,i+1})$:

$$\sum_{(x_i, a_i)} w_i(x_i, s_i, F_{i,i+1}, a_i) = \delta^i \Pr(s_i, F_{i,i+1}), \forall (i, s_i, F_{i,i+1}). \tag{2.43}$$

(b) If $p(F_{i,i+1}|s_i, F_0) = \mathbb{1}(F_{i,i+1} = \mathbb{E}[F_{i,i+1}|s_i, F_{0,i+1}])$, then, assuming $\mathbb{E}[F_{i,i+1}|s_i, F_{0,i+1}] \in \mathcal{F}_{i,i+1}$, w matches the first moment $\mathbb{E}[F_{i,i+1}|s_i, F_{0,i+1}]$ of the prompt-month futures price $F_{i,i+1}$ given the spot price s_i and the time zero prompt-month futures price $F_{0,i+1}$:

$$\sum_{F_{i,i+1}} F_{i,i+1} \left(\frac{\sum_{(x_i, a_i)} w_i(x_i, s_i, F_{i,i+1}, a_i)}{\sum_{(x_i, F_{i,i+1}, a_i)} w_i(x_i, s_i, F_{i,i+1}, a_i)} \right) = \mathbb{E}[F_{i,i+1}|s_i, F_{0,i+1}], \forall (i, s_i). \tag{2.44}$$

It is easy to verify that optimal DLP solutions satisfy conditions analogous to (2.32) and thus correspondingly implied conditions analogous to (2.43)-(2.44). Moreover, it can be verified that DALP feasible solutions that satisfy constraints (2.33) also satisfy conditions (2.43)-(2.44). Thus, our choices of $p(F_{i,i+1}|s_i, F_{0,i+1})$ limit the extent to which constraints (2.33) approximate (2.32). As discussed in Appendix 2.7, in our numerical implementation of ADP1 we force the assumption in part (b) of Proposition 5 by computing $\mathbb{E}[F_{i,i+1}|s_i, F_{0,i+1}]$ in closed form under price model (2.4)-(2.5) for each given s_i, which is

equivalent to having a grid with the single value $\mathbb{E}[F_{i,i+1}|s_i, F_{0,i+1}]$ for each considered value of s_i.

Because conditions (2.43) capture more properties of optimal DLP solutions than conditions (2.44), it seems that SADP is a better ADP model than ADP1. However, ADP1 has computational advantages over SADP: It requires computing fewer expectations than SADP and discretizing only the spot price when the term $\mathbb{E}[F_{i,i+1}|s_i, F_{0,i+1}]$ is available in closed form, which is the case for the price model (2.4)-(2.5), as just pointed out.

The computational advantage of ADP1 over SADP motivates us to extend this model to an ADP with value function $\phi_i(x_i, s_i, F_{i,i+1})$, which is a look-up table that in every stage also depends on the prompt-month futures price. We briefly discuss the derivation of this ADP1 extension, ADP2, without presenting formulations for brevity. ADP2 is derived in a manner analogous to the derivation of ADP1 starting from an ALP formulated using the value function approximation $\phi_i(x_i, s_i, F_{i,i+1})$. An optimal solution to this ALP can be computed by solving an ADP that we label ADP0′ and is analogous to ADP0. We then add to the dual of this ALP the following constraints that are analogous to the constraints (2.33) used in the derivation of ADP1:

$$\sum_{a_i} w_i(x_i, s_i, F_{i,i+1}, F_{i,i+2}, a_i) = \mathbb{1}(F_{i,i+2} = \mathbb{E}[F_{i,i+2}|s_i, F_{i,i+1}, F_0])\theta_i(x_i, s_i, F_{i,i+1}),$$
$$\forall (i, x_i, s_i, F_{i,i+1}, F_{i,i+2}).$$

The primal linear program corresponding to this ALP dual restriction is an ALP relaxation that can be reformulated as ADP2. The ADP2 model is

$$\begin{aligned}\phi_i^{ADP2}(x_i, s_i, F_{i,i+1}) = &\max_{a_i} r(a_i, s_i) + \delta\mathbb{E}\left[\phi_{i+1}^{ADP2}(x_i - a_i, s_{i+1})|F_{i,i+1}\right], \\ &\forall i \in \{N-2, N-1\}, (x_i, s_i), \qquad (2.45)\\ \phi_i^{ADP2}(x_i, s_i, F_{i,i+1}) = &\max_{a_i} r(a_i, s_i) \\ &+ \delta\mathbb{E}\left[\phi_{i+1}^{ADP2}(x_i - a_i, s_{i+1}, F_{i+1,i+2})|F_{i,i+1}, \mathbb{E}[F_{i,i+2}|s_i, F_{i,i+1}, F_{0,i+2}]\right], \\ &\forall i \in \mathcal{I} \setminus \{N-2, N-1\}, (x_i, s_i, F_{i,i+1}), \qquad (2.46)\end{aligned}$$

with $\phi_N^{ADP2}(x_N, s_N) := 0$, $\forall x_N$.

2.6 Error Bound Analysis for Constraint-based ALP Relaxations

In this section we analyze the value function approximations obtained by versions of the ADPs discussed in §2.5.2-2.5.3: ADP0, SADP, ADP1, ADP0′, and ADP2. Our analysis provides insights into the relative performance of these ADPs, in particular the benefit of the constraint-based ALP relaxations proposed in §2.5.3 relative to their corresponding ALPs, and sheds light on when SADP, ADP1, and ADP2 can be expected to perform well.

Let ℓ represent an ADP in the set $\mathcal{L} := \{\text{SADP}, \text{ADP1}, \text{ADP2}\}$. Consistent with how EDP is formulated, we analyze versions of ADP0, ADP0′, and the ADPs in set $\mathcal{L}$ reformulated assuming that the forward curve F_i at each stage i belongs to $\mathbb{R}_+^{N-i}$ instead of the finite set $\mathcal{F}_i$ (and hence the first "max" in (2.29) is assumed to be replaced with "sup"; an analogous substitution is assumed for ADP0′). For simplicity, we continue to use the same labels for these reformulated models. Under a mild assumption satisfied by price model (2.4)-(2.5), Proposition 6 compares the value function approximations of ADP0 and ADP0′ against the value function of EDP and the value function approximations of the ADPs in set $\mathcal{L}$. The mild assumption in Proposition 6 is that the distributions of the random variables $s_{i+1}|F_{i,i+1}$ and $s_{i+2}|F_{i,i+2}$ are stochastically increasing in $F_{i,i+1}$ and $F_{i,i+2}$, respectively (see, e.g., Topkis 1998, Lemma 3.9.1 (b)).

Proposition 6. *(i) If the distribution of $s_{i+1}|F_{i,i+1}$ is stochastically increasing in $F_{i,i+1} \in \mathbb{R}_+$, $\forall i \in \mathcal{I}_{-(N-1)}$, then the ADP0 value function is unbounded in every state in stages 0 through $N-2$. (ii) If the distribution of $s_{i+2}|F_{i,i+2}$ is stochastically increasing in $F_{i,i+2} \in \mathbb{R}_+$, $\forall i \in \mathcal{I} \setminus \{N-1, N-2\}$, then the ADP0′ value function is unbounded in every state in stages 0 through $N-3$. (iii) The value functions of EDP and the ADPs in set $\mathcal{L}$ are bounded at every stage and state.*

Parts (i) and (ii) of Proposition 6 are consistent with the discussion given after Proposition 3. Together with part (iii) of this proposition they suggest that there is potential benefit in using constraint-based relaxations of an ALP rather than an ALP. We thus focus on providing approximation guarantees for the value functions of the ADPs in set $\mathcal{L}$. Our approximation guarantees are based on the norm $\|g\|_{\mathbb{E},\infty}$, which we define as $\max_x \mathbb{E}[g(x, F_i)|F_0]$, where $g(x, F_i)$ is a generic function with support on the stage i EDP state space and the expectation is with respect to the distribution of F_i conditional on F_0. The expectation over the stage i futures prices in this norm is consistent with EDP, while the maximum over inventory is for analytical tractability. Our choice of norm is based on analytical tractability. Specifically, we analyze the following errors between the stage i EDP value function V_i and each ADP ℓ value function ϕ_i^ℓ:

$$\|V_i - \phi_i^\ell\|_{\mathbb{E},\infty} := \begin{cases} \max_{x_i} \mathbb{E}\left[|V_i(x_i, F_i) - \phi_i^\ell(x_i, s_i)| \middle| F_0\right], & \forall \ell \in \{\text{ADP1}, \text{SADP}\}, \\ \max_{x_i} \mathbb{E}\left[|V_i(x_i, F_i) - \phi_i^{ADP2}(x_i, s_i, F_{i,i+1})| \middle| F_0\right], & \end{cases} \quad \forall i.$$

We refer to $\|V_i - \phi_i^\ell\|_{\mathbb{E},\infty}$ as the ℓ-error at stage i.

Our analysis is based on the concept of an *ideal* value function approximation. This function is defined by replacing with V_{i+1}, the EDP stage $i+1$ value function, the function ϕ_{i+1}^ℓ on the right hand side of each ADP ℓ recursion, that is, (2.41), (2.42), and (2.45)-(2.46) reformulated as discussed above, and modifying the conditional expectations accordingly. Recall that $F_i' \equiv \{F_{i,i+1}, F_{i,i+2}, \ldots, F_{i,N-1}\}$. Let $F_i'' := \{F_{i,i+2}, F_{i,i+3}, \ldots, F_{i,N-1}\}$. To ease the exposition we define $\bar{F}_i'(s_i, F_0)$ as $\mathbb{E}[F_i'|s_i, F_0]$ and $\bar{F}_i''(s_i, F_{i,i+1}, F_0)$ as $\mathbb{E}[F_i''|s_i, F_{i,i+1}, F_0]$. The ideal value function approximations for the ADPs in set $\mathcal{L}$ are defined as

$$\phi_i^{SADP,V}(x_i, s_i) \;\; := \;\; \mathbb{E}\left[\max_{a_i \in \mathcal{A}(x_i)} r(a_i, s_i) + \delta \mathbb{E}\left[V_{i+1}\left(x_i - a_i, F_{i+1}\right)|F_i\right]|s_i, F_0\right],$$

$$\begin{aligned}
\phi_i^{ADP1,V}(x_i, s_i) &:= \max_{a_i \in \mathcal{A}(x_i)} r(a_i, s_i) + \delta \mathbb{E}\left[V_{i+1}\left(x_i - a_i, F_{i+1}\right) | \bar{F}_i'(s_i, F_0)\right], \\
\phi_i^{ADP2,V}(x_i, s_i, F_{i,i+1}) &:= \max_{a_i \in \mathcal{A}(x_i)} r(a_i, s_i) \\
&\qquad + \delta \mathbb{E}\left[V_{i+1}\left(x_i - a_i, F_{i+1}\right) | F_{i,i+1}, \bar{F}_i''(s_i, F_{i,i+1}, F_0)\right],
\end{aligned}$$

where the superscript in the notation for an ideal value function approximation indicates that it is defined using the exact value function and an ADP recursion.

We now bound the various ℓ-errors using recursive functions that depend on the absolute value of the differences between the EDP value function and the ℓ-ideal value function approximations. These recursive functions for SADP and ADP1 are defined, $\forall (i, x_i, F_i)_{-(N-1)}$, as

$$\begin{aligned}
\gamma_i^{SADP}(x_i, F_i) &:= |V_i(x_i, F_i) - \phi_i^{SADP,V}(x_i, s_i)| + \delta \mathbb{E}\left[\max_{x_{i+1}} \mathbb{E}\left[\gamma_{i+1}^{SADP}(x_{i+1}, F_{i+1}) \middle| F_i'\right] \middle| s_i, F_0\right], \\
\gamma_i^{ADP1}(x_i, F_i) &:= |V_i(x_i, F_i) - \phi_i^{ADP1,V}(x_i, s_i)| + \delta \max_{x_{i+1}} \mathbb{E}\left[\gamma_{i+1}^{ADP1}(x_{i+1}, F_{i+1}) \middle| \bar{F}_i'(s_i, F_0)\right],
\end{aligned}$$

with boundary conditions $\gamma_{N-1}^{\ell}(\cdot) \equiv 0$, $\forall \ell \in \{SADP, ADP1\}$. For ADP2, this recursive function is defined, $\forall (i, x_i, F_i)_{-(N-1,N-2)}$, as

$$\begin{aligned}
\gamma_i^{ADP2}(x_i, F_i) := |V_i(x_i, F_i) - \phi_i^{ADP2,V}(x_i, s_i, F_{i,i+1})| \\
+ \delta \max_{x_{i+1}} \mathbb{E}\left[\gamma_{i+1}^{ADP2}(x_{i+1}, F_{i+1}) \middle| F_{i,i+1}, \bar{F}_i''(s_i, F_{i,i+1}, F_0)\right],
\end{aligned}$$

with boundary conditions $\gamma_i^{ADP2}(\cdot) \equiv 0$, $\forall i \in \{N-2, N-1\}$. As shown in page 141 of Appendix A.2, our bound on the stage i ℓ-error is

$$\|V_i - \phi_i^{\ell}\|_{\mathbb{E},\infty} \leq \|\gamma_i^{\ell}\|_{\mathbb{E},\infty}, \tag{2.47}$$

$\forall (\ell, i) \in \mathcal{L} \times \mathcal{I}$. The bound (2.47) formalizes the intuition that the ADPs in set $\mathcal{L}$ incur an error when the exact value function differs from its corresponding ℓ-ideal value function approximation at a given stage and state. This bound is finite (by part (iii) of Proposition 6).

The bound (2.47) is zero in the last stage $(N-1)$ and also in the penultimate stage (N-2) for ADP2. Proposition 7 identifies limiting regimes under price model (2.4)-(2.5) for which the bound (2.47) tends to zero in all other stages. We denote by ρ the matrix of the correlations between the standard Brownian motion increments of price model (2.4)-(2.5). We let $\overline{\rho}$ be a rank 2 matrix ρ such that each of its elements $\bar{\rho}_{i,i+1}$ satisfies $|\bar{\rho}_{i,i+1}| < 1$. We also denote by 1 a matrix of ones that is compatible with ρ.

Proposition 7. *Under price model (2.4)-(2.5) it holds that : (i)* $\lim_{\rho \to 1} \|\gamma_i^{\ell}\|_{\mathbb{E},\infty} = 0$ *for all* $\ell \in \mathcal{L}$ *and* $i \in \mathcal{I} \setminus \{N-1\}$*; and (ii)* $\lim_{\rho \to \overline{\rho}} \|\gamma_i^{ADP2}\|_{\mathbb{E},\infty} = 0$ *for all* $i \in \mathcal{I} \setminus \{N-1, N-2\}$.

Part (i) of Proposition 7 suggests that the ADPs in $\mathcal{L}$ should perform near optimally under price model (2.4)-(2.5) when the correlations in this model are sufficiently large and positive. The intuition for this conclusion is that at the stated limit there is a single

source of uncertainty in price model (2.4)-(2.5), and hence the current spot price is a sufficient statistic for the future evolution of the entire forward curve. Part (ii) of this proposition suggests that ADP2 is also near optimal because at the limit there are two sources of uncertainty in price model (2.4)-(2.5) and the spot price and the prompt month futures price are not sufficient statistics for each other ($|\rho_{i,i+1}| < 1$). Because the ADP2 value function approximation depends on these two prices but the SADP and ADP1 value function approximations are only based on the spot price, this result provides theoretical support for the intuition that ADP2 should outperform both SADP and ADP1. Moreover, this result also encompasses a weakening of the limiting condition in part (i), that is, the case when $\bar{\rho}$ corresponds to a rank 2 matrix where all correlations except $\rho_{i,i+1}$ are equal to 1.

Our use of an ideal value function approximation to bound the ℓ-error is similar in spirit to the approach taken in de Farias and Van Roy (2003) and Desai et al. (2012a) to bound the error incurred by the value function approximation determined by their models. However, the error bounds of these authors do not apply to the ADPs in set $\mathcal{L}$ because (i) de Farias and Van Roy (2003) provide bounds for an ALP while we analyze ADPs corresponding to ALP relaxations, and (ii) Desai et al. (2012a) study an ALP relaxation that is different from the ones that we consider, as discussed in §2.5.3. Likewise, our bound (2.47) is specific to the ADPs considered here.

2.7 Computational Complexity Analysis

In this section we discuss the computational complexity of solving the ALP relaxations presented in §2.5.3 and estimating greedy lower and dual upper bounds. This complexity depends on the specific technique used for discretizing the relevant price sets. Our computational study in §2.8 is based on the multi-maturity Black (1976) price model (2.4)-(2.5) discretized via Rubinstein (1994) binomial lattices when discretizations are needed. We thus focus on this discretization approach. Our analysis uses the (easy to establish) property that the policies associated with SADP, ADP1, and ADP2 share the basestock target structure of an optimal DDP policy (see Proposition 4 and Lemma 2 in Secomandi et al. 2012 for details).

Consider ADP1. Figure 2.2 illustrates our discretization approach. We obtain the set $\mathcal{F}_{i,i}$, that is, we discretize $\mathbb{R}_+$, by evolving the time 0 futures price $F_{0,i}$ using a two-dimensional Rubinstein binomial tree based on the volatility σ_i (see the top part of Figure 2.2). Let m_i be the number of time steps used to discretize the time interval $[0, T_i]$. Building this lattice results in a set $\mathcal{F}_{i,i}$ with $m_i + 1$ values. This requires $O(m_i)$ operations.

At each stage i, solving ADP1 entails executing the following steps:
Step 1: Determine a probability mass function with support $\mathcal{F}_{i+1,i+1}$ for the random variable s_{i+1} given $\mathbb{E}[F_{i,i+1}|s_i, F_{0,i+1}]$ for all s_i;
Step 2: Compute the optimal ADP1 basestock targets for all s_i;
Step 3: Evaluate $\phi_i^{ADP1}(x_i, s_i)$ for all (x_i, s_i).
In step 1, we evolve a two-dimensional Rubinstein lattice, starting from each price $\mathbb{E}[F_{i,i+1}|s_i, F_{0,i+1}]$,

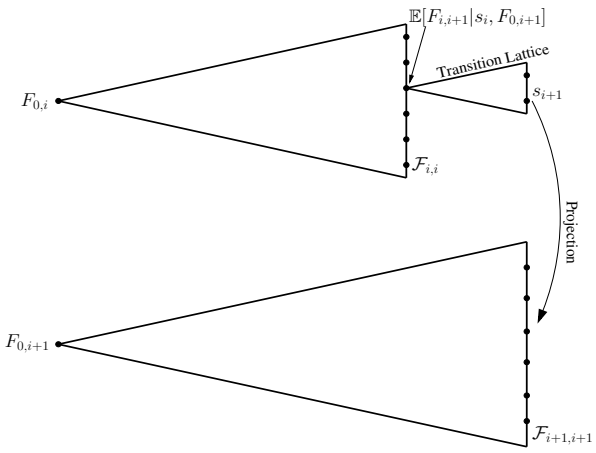


Figure 2.2: Illustration of our discretization approach for ADP1.

referred to as the transition lattice, by using m time steps to discretize the interval $[T_i, T_{i+1}]$ (see the top part of Figure 2.2). In particular, this price depends on the correlation coefficient $\rho_{i,i+1}$. Each price $\mathbb{E}[F_{i,i+1}|s_i, F_{0,i+1}]$ can be computed in closed-form in $O(1)$ operations under the price model (2.4)-(2.5). Each transition lattice yields a discretization of s_{i+1} with $m+1$ values. Building all the m_i transition lattices thus takes $O(m_i \cdot m)$ operations. To obtain the distribution of s_{i+1} given $\mathbb{E}[F_{i,i+1}|s_i, F_{0,i+1}]$ with support on $\mathcal{F}_{i+1,i+1}$, we project each price s_{i+1} in each transition lattice onto the set $\mathcal{F}_{i+1,i+1}$ by rounding each price s_{i+1} to the closest spot price in $\mathcal{F}_{i+1,i+1}$ (see Figure 2.2). The set $\mathcal{F}_{i+1,i+1}$ is constructed in a manner analogous to how we generate the set $\mathcal{F}_{i,i}$, but using the parameters m_{i+1}, T_{i+1}, $F_{0,i+1}$, and σ_{i+1} (see the bottom part of Figure 2.2). Since the s_{i+1} values in each transition lattice and the set $\mathcal{F}_{i+1,i+1}$ are sorted, this projection takes a total of $O(m_{i+1} \cdot m)$ operations at stage i. Therefore, the time complexity for step 1 at stage i is $O(m_i \cdot m + m_{i+1} \cdot m)$.

Executing step 2 requires performing the maximization in (2.42) at inventory levels 0 and $\bar{x}$ with the injection and withdrawal capacities relaxed to $-\bar{x}$ and $\bar{x}$, respectively, which requires $O(m_i \cdot |\mathcal{X}| \cdot m)$ operations. Executing step 3 also takes $O(m_i \cdot |\mathcal{X}| \cdot m)$ operations. Therefore, computing $\phi_i^{ADP1}(x_i, s_i)$ for all (x_i, s_i) in stage i involves $O(m \cdot (m_i + m_{i+1} + 2 \cdot m_i \cdot |\mathcal{X}|))$ operations. Using $m' := \max_{i \in \mathcal{I}} m_i$, this number of operations simplifies to $O(m' \cdot |\mathcal{X}| \cdot m)$, since $|\mathcal{X}| \geq 2$. Thus, for an N-stage problem, solving ADP1 entails $O(N \cdot m' \cdot |\mathcal{X}| \cdot m)$ operations.

For SADP and ADP2, we determine the set $\mathcal{F}_{i,i} \times \mathcal{F}_{i,i+1}$ for each stage i using a three dimensional Rubinstein lattice. For SADP, we use two dimensional binomial lattices and projections to obtain the probability mass function of s_{i+1} conditional on each of the m_i^2 values of $F_{i,i+1}$. In contrast, for ADP2 we use three dimensional lattices and projections to obtain the joint probability mass function of each random pair $(s_{i+1}, F_{i+1,i+2})$ on the support $\mathcal{F}_{i+1,i+1} \times \mathcal{F}_{i+1,i+2}$ conditional on the pair $(F_{i,i+1}, \mathbb{E}[F_{i,i+2}|s_i, F_{i,i+1}, F_{0,i+2}])$. An analysis similar to the one performed for ADP1 shows that we can solve SADP and ADP2 in $O(N \cdot (m')^2 \cdot |\mathcal{X}|^2 \cdot m)$ and $O(N \cdot (m')^2 \cdot |\mathcal{X}|^2 \cdot m^2)$ operations, respectively.

Table 2.1 summarizes the computational complexity of solving each of SADP, ADP1, and ADP2. This table indicates the following ordering of these models in terms of increas-

Table 2.1: Computational complexity of solving SADP, ADP1, and ADP2.

Method	Computational Complexity
ADP1	$O(N \cdot m' \cdot \lvert\mathcal{X}\rvert \cdot m)$
SADP	$O(N \cdot m'^2 \cdot \lvert\mathcal{X}\rvert \cdot m)$
ADP2	$O(N \cdot m'^2 \cdot \lvert\mathcal{X}\rvert \cdot m^2)$

ing computational complexity: ADP1, SADP, and ADP2.

The operations count for estimating upper and lower bounds depends on the number of prices included in a look-up table value function approximation. Let n_s denote the number of price sample paths used in a Monte Carlo simulation used to estimate a greedy lower bound and a dual upper bound (see §2.2.2). Different from how we obtain each discretization $\mathcal{F}_{i,i}$, this simulation is based on evolving the entire forward curve. A simple analysis shows that estimating lower and upper bounds, respectively, when using the look-up table value function approximation $\phi_i(x_i, s_i)$ requires $O(n_s \cdot N \cdot \log m' + n_s \cdot N \cdot |\mathcal{X}| \cdot m)$ and $O(n_s \cdot N \cdot |\mathcal{X}| \cdot \log m' + n_s \cdot N \cdot |\mathcal{X}|^2 \cdot m)$ operations ($O(\log m')$ operations are needed by binary search, which we use when projecting a transition lattice); doing this when using the look-up table value function approximation $\phi_i(x_i, s_i, F_{i,i+1})$ involves $O(n_s \cdot N \cdot \log m' \cdot m + n_s \cdot N \cdot |\mathcal{X}| \cdot m^2)$ and $O(n_s \cdot N \cdot |\mathcal{X}| \cdot \log m' \cdot m + n_s \cdot N \cdot |\mathcal{X}|^2 \cdot m^2)$ operations, respectively.

Table 2.2: Computational complexity of estimating a greedy lower bound and a dual upper bound with look-up table value function approximations.

Number of Prices in the Look-up Table	Computational Complexity	
	Greedy Lower Bound	Dual Upper Bound
1	$O\left(n_s \cdot N \cdot [\log m' + \lvert\mathcal{X}\rvert \cdot m]\right)$	$O\left(n_s \cdot N \cdot \lvert\mathcal{X}\rvert \cdot [\log m' + \lvert\mathcal{X}\rvert \cdot m]\right)$
2	$O\left(n_s \cdot N \cdot m \cdot [\log m' + \lvert\mathcal{X}\rvert \cdot m]\right)$	$O\left(n_s \cdot N \cdot \lvert\mathcal{X}\rvert \cdot m \cdot [\log m' + \lvert\mathcal{X}\rvert \cdot m]\right)$

Table 2.2 summarizes the outcome of this analysis. This table shows that estimating dual upper bounds is more costly than estimating greedy lower bounds, due to the computation of the dual value function in (2.8) at each inventory level in the set $\mathcal{X}$ and for all the stages in set $\mathcal{I}$ given a price sample path P_0. Reasonable values of the parameters n^s, $|\mathcal{X}|$, and m' satisfy $n^s \cdot |\mathcal{X}| \geq m'$. Hence, estimating dual upper bounds is also more costly than solving each of SADP, ADP1, and ADP2.

2.8 Numerical Results

In this section we discuss the computational performance of the models presented in §2.5.2 applied to the 24-stage LMS instances. Appendix A.3 contains additional numerical results related to SADP. These instances are based on natural gas data from the New York Mercantile Exchange (NYMEX) and the energy trading literature. Each instance is identified

by a season (Spring, Summer, Fall, or Winter) and one of three injection and withdrawal capacity pairs, with their labels 1, 2, and 3 denoting a heavy, intermediate, and mild capacity restriction, respectively. These instances are based on the multi-maturity Black model (2.4)-(2.5). The details of these instances are available in LMS.

In §§2.8.1-2.8.2 we investigate the upper and lower bounding performance of the models summarized in Table 2.3. We discuss their run times in §2.8.3.

Table 2.3: Models used in our numerical study.

ALP	Constraint-based ALP Relaxation	Number of Prices in the Look-up Table
ADP0	SADP, ADP1	1
ADP0′	ADP2	2

2.8.1 Upper Bounds

As LMS, we use 10,000 forward curve sample paths to obtain our dual upper bound estimates on the value of storage in the initial stage and state. Across all the considered instances, the ADP0-based dual upper bound estimates are between 30% and 690% larger than the worst dual upper bound estimates obtained with ADP1 and SADP, and the ADP0′-based dual upper bound estimates are between 21% and 600% larger than the ADP2-based dual upper bound estimates. Thus, on these instances, the value function approximations of the considered ALP relaxations lead to substantially tighter dual upper bound estimates than the value function approximations of their respective ALPs. These findings are consistent with our error bound analysis carried out in §2.6.

We denote by UBS, UB1, and UB2 the dual upper bound estimates associated with SADP, ADP1, and ADP2, respectively. Figure 2.3 displays UBS and UB1 on all the considered instances as percentages of UB2, which is tighter than all the other estimated upper bounds. The error bars in this figure indicate standard errors, also reported as percentages of UB2. UBS and UB1 match on all the instances after accounting for sampling variability. UB2 is better than both UBS and UB1 by an average of 2.82% on the Winter instances, while this average is smaller on the other instances. We are thus able to obtain substantially improved upper bound estimates compared to LMS on the Winter instances. The observed performance of UB2 relative to UBS and UB1 is consistent with our error bound analysis performed in §2.6.

2.8.2 Lower Bounds

We also use $10,000$ sample paths to obtain our lower bound estimates on the value of storage in the initial stage and state. Across all the considered instances, the ADP0-based lower bound estimates are between 25% and 100% smaller than the worst lower bound

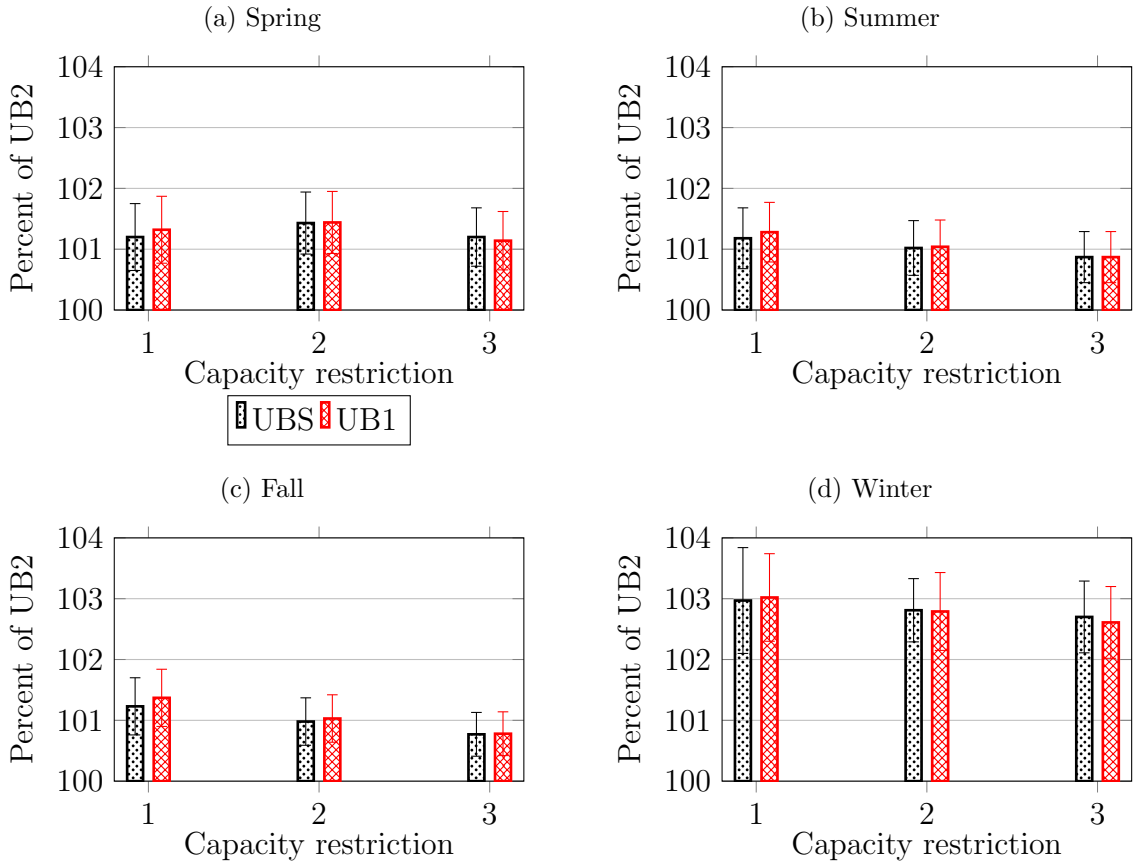


Figure 2.3: Estimated upper bounds and their standard errors (error bars).

estimates obtained with ADP1 and SADP, and the ADP0′-based lower bound estimates are between 5% and 89% smaller than the ADP2-based lower bound estimates. The control policies obtained from the ALP relaxations are thus substantially better than the control policies based on their respective ALPs on these instances. These results are in line with our error bound analysis performed in §2.6.

We denote by LBS, LB1, and LB2 the lower bound estimates obtained using SADP, ADP1, and ADP2, respectively. Figure 2.4 displays these estimates as percentages of UB2. The error bars in this figure indicate the standard errors of these estimates as percentages of UB2. The difference between LBS and LB1 is less than one standard error (expressed as a ratio of UB2) on the Spring, Summer, and Fall instances, while LB1 is weaker than LBS by no more than 2.44% of UB2 on the Winter instances. LB2 outperforms both LBS and LB1 on all the considered instances: The improvement of LB2 on LBS is 2.00-3.36% across the Spring, Summer, and Fall instances, and 6.72-8.43% on the Winter instances. The improvements of LB2 on LB1 are similar on the Spring, Summer, and Fall instances, but are larger on the Winter instances. These results suggest that ADP2 is a better model than SADP and ADP1, with maximum suboptimality gaps of 3.03% of UB2 on the Spring, Summer, and Fall instances, and 9.03% of UB2 on the Winter instances. In contrast, these suboptimalities are 5.77% and 17.46% for SADP, and 6.11% and 19.89% for ADP1.

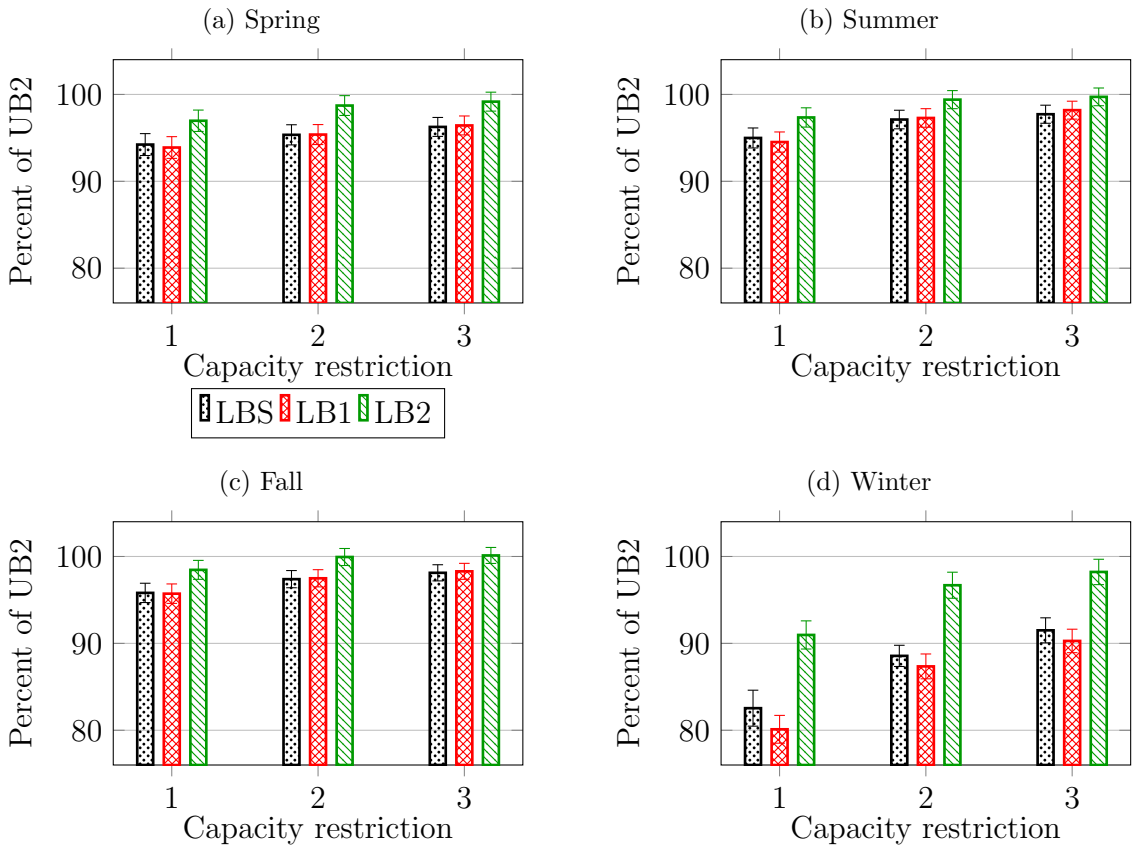


Figure 2.4: Estimated lower bounds and their standard errors (error bars) without reoptimization.

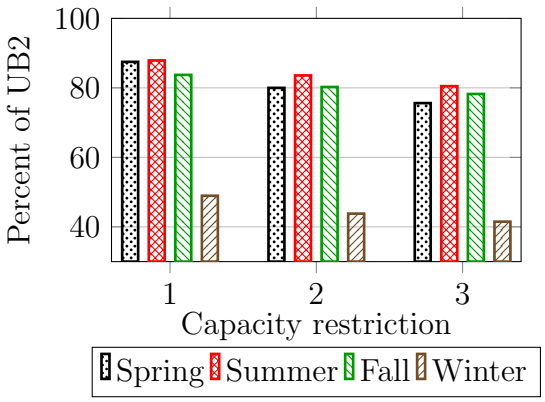


Figure 2.5: Intrinsic values.

The relative performance of ADP2 against ADP1 and SADP is consistent with part (ii) of Proposition 7. To shed some more light on the difference between the ADP2-based and SADP/ADP1-based lower bounds on the Winter instances relative to the other instances, Figure 2.5 reports the intrinsic value for each instance, that is, the value of storage due to seasonality (deterministic variability). This value is obtained by solving a deterministic

version of EDP, (2.3), based only on the initial (time 0) forward curve (see §3.2 in LMS for further details). The computed intrinsic values are less than 50% of their respective UB2 values on the Winter instances, while they are at least 75% of UB2 on the remaining instances. Thus, a substantially larger portion of the storage value is attributable to price uncertainty for the Winter instances than for the other instances. In other words, capturing the evolution of the forward curve appears to be more important on the Winter instances than on the other instances. Because the ADP2 value function approximation depends on both the spot and prompt futures prices while the ones of SADP and ADP1 depend only on the spot price, ADP2 is better able to capture the evolution of the forward curve.

We denote by RLBS, RLB1, and RLB2 the estimates of the reoptimization versions of LBS, LB1, and LB2, respectively. Figure 2.6 displays these reoptimization-based lower bound estimates and their standard errors as percentages of the UB2 values (some of the reported lower bound estimates exceed UB2 due to Monte Carlo sampling error). RLBS, RLB1, and RLB2 are almost tight on the Spring, Summer, and Fall instances. RLB2 is slightly better than RLBS and RLB1 on the Winter instances, with a maximum optimality gap of 2.38% of UB2 compared to 3.51% for RLBS and 2.58% for RLB1. Further, LB2 is worse than RLB2 by 0.20-6.65% of UB2 on all the instances, while LBS and LB1, respectively, fall below RLBS and RLB1 by 2.29-13.94% and 1.28-14.51% of UB2 on all the instances. Thus, while reoptimization can be useful even for ADP2, it appears to be less critical for ADP2 than it is for SADP and ADP1 to obtain near optimal lower bounds and policies.

We now compare the ADP2-based lower bounds against the ones estimated using two state-of-the-art approaches for commodity storage real option valuation and management: The rolling intrinsic policy and least squares Monte Carlo (see §2.1 for relevant references). Our implementation of the least squares Monte Carlo method uses basis functions that for every stage and inventory level include polynomials of orders one and two in each futures price. Across all the considered instances, the averages of the lower bounds (as percentages of UB2) estimated by the rolling intrinsic policy and least squares Monte Carlo, respectively, are 99.14% and 98.83% (the standard errors of the individual lower bound estimates vary between 0.77% and 1.76% of UB2). The analogous averages for LB2 and RLB2 are 97.98% and 99.59%, respectively. The ADP2-based lower bounds are thus competitive with the ones obtained by these state-of-the-art techniques.

2.8.3 CPU Times

The models that we solve numerically are formulated on discretized state and action spaces. As in LMS, we optimally discretize the feasible inventory set into 21 equally spaced points. We further reduce the considered inventory levels by eliminating the ones that cannot be feasibly reached in each stage from the initial stage and state. We obtain discretized price sets from the multi-maturity Black (1976) price model (2.4)-(2.5) using Rubinstein (1994) binomial lattices (see Appendix 2.7 for details) and also apply lattice restrictions (Levy, 2004) to shorten the CPU time required to solve ADP2. This approach, standard in computational finance, is effective: We obtain a speed up equal to one order of magnitude

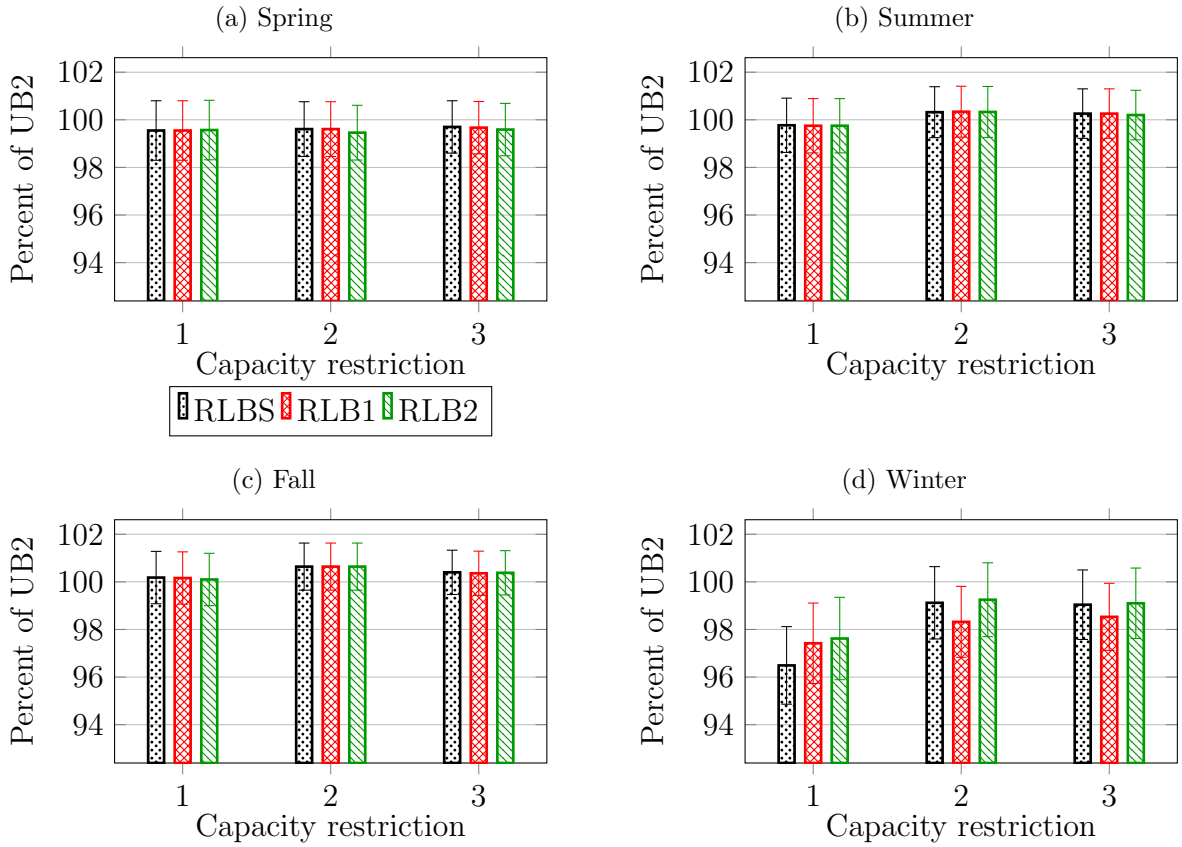


Figure 2.6: Estimated lower bounds and their standard errors (error bars) with reoptimization.

while the estimated lower and upper bounds change by less than 0.2% with this restriction in place.

Our experiments are based on the following computational setup: A 64 bits PowerEdge R515 with twelve AMD Opteron 4176 2.4GHz processors, of which we used only one, with 64GB of memory, the Linux Fedora 15 operating system, and the g++ 4.6.1 20110908 (Red Hat 4.6.1-9) compiler. The SADP results that we report are obtained with the code of LMS run within our computational setup.

The CPU seconds required to solve SADP ranges from 120 to 122. Solving ADP1 and ADP2 takes between 0.11 and 0.12 and 36 and 53 CPU seconds, respectively. Thus, on all the considered instances, the ADP1 and ADP2 computational requirements are at least 1,000 times and 2 times smaller, respectively, than the ones of SADP (recall that we use lattice restrictions when solving ADP2).

The SADP overall CPU seconds, that is, also including the time required for bound estimation, vary from 272 to 314. The ADP1 and ADP2 overall CPU seconds range between 10 and 17 and 154 and 225, respectively. Therefore, the ADP1 overall CPU run times are at least 1 order of magnitude smaller than the ones of SADP on all the considered instances. The ADP2 overall CPU times are between 76% and 53% of the ones

of SADP. However, solving ADP2 is 12 to 16 times slower than solving ADP1. Given a value function approximation, the upper bound estimation is more costly than the lower bound estimation. For example, on average, the upper bound estimation requires roughly 87% and 75% of the total bounding CPU time for ADP1 and ADP2, respectively.

Computing RLBS takes between 544 and 619 CPU seconds, while this range for RLB1 is 90-93 CPU seconds, that is, roughly 6 times smaller. The RLB2 CPU seconds range from 1,222 to 1,248. Thus, the RLB2 run times are roughly 1 order of magnitude and 2 times larger than the RLB1 and RLBS run times, respectively.

2.9 Conclusions

Real option management of commodity storage assets is an important practical problem that, in general, gives rise to an intractable MDP when using high dimensional models of commodity forward curve evolution. We develop a novel approximate dynamic programming approach to derive ALP relaxations. Our approach relies on approximately enforcing on the ALP dual a property of the exact dual. We derive tractable ALP relaxations by applying our approach using low dimensional look-up table value function approximations, subsuming an existing approximate dynamic programming model. We derive error bounds that provide theoretical support for using our ALP relaxations over their respective ALPs. Our numerical results on existing natural gas instances are promising, showing that our ALP relaxations substantially outperform their respective ALPs, with our best ALP relaxation matching or improving on the best lower and upper bounds available in the literature for these instances, and being competitive with state-of-the-art methods for obtaining heuristic policies and estimating lower bounds on the value of commodity storage.

Chapter 3

Improved Least Squares Monte Carlo for Term Structure Option Valuation with Energy Applications

(Joint work with François Margot and Nicola Secomandi)

3.1 Introduction

The pricing of options with multiple exercises is an important area of financial engineering, with applications including commodity, energy, and interest rate derivatives. Examples include chooser flexible caps (Meinshausen and Hambly, 2004), portfolio liquidation (Gyurko et al., 2011), swing options (Barbieri and Garman, 1996, Jaillet et al., 2004, Chandramouli and Haugh, 2012), switching options (Cortazar et al., 2008), and commodity processing and storage (Maragos, 2002, Boogert and De Jong, 2008, 2011/12, Secomandi, 2010, Lai et al., 2010, Arvesen et al., 2013, Boogert and Mazières, 2011, Devalkar et al., 2011, Thompson, 2012, Wu et al., 2012). In particular, our focus is on energy swing and storage options.

Term structure models are widespread both in practice and in the literature that deals with applications in commodity, energy, and fixed income industries (Ho and Lee, 1986, Cortazar and Schwartz, 1994, Clewlow and Strickland, 2000, Maragos, 2002, Eydeland and Wolyniec, 2003, Veronesi, 2010). Valuing multiple exercise options using these models generally gives rise to intractable Markov decision problems (MDPs). The intractability here is due to two curses of dimensionality that affect the stochastic dynamic programs (SDPs) corresponding to these MDPs: (i) The high dimensionality of the state spaces of these SDPs and (ii) the inability to exactly compute the expectations that are present in these SDPs (Powell, 2011, §4.1).

The financial engineering literature typically approaches the solution of these SDPs using Monte Carlo based approximate dynamic programming (ADP) techniques, which compute a heuristic exercise policy and greedy lower and dual upper bounds on the option value (see Rogers 2002, Andersen and Broadie 2004, Chapter 8 in Glasserman 2004, Haugh

and Kogan 2004, Detemple 2006, Haugh and Kogan 2007, Brown et al. 2010, and references therein).

The least squares Monte Carlo (LSM) approach, pioneered by Carriere (1996), Longstaff and Schwartz (2001), and Tsitsiklis and Van Roy (2001), has become the norm for valuing multiple exercise options (see Appendix B in Eydeland and Wolyniec 2003, Glasserman and Yu 2004, Meinshausen and Hambly 2004, Detemple 2006, Boogert and De Jong 2008, 2011/12, Bender 2011, Gyurko et al. 2011). This method approximates the SDP continuation (value) function. We thus refer to this method as LSMC, where C stands for continuation (function). LSMC uses basis functions and a convenient sample average approximation, respectively, to overcome the first and second curses of dimensionality.

A known difficulty with LSMC is the estimation of dual upper bounds (see Chapter 8 in Glasserman 2004). To overcome this difficulty, Gyurko et al. (2011) and Desai et al. (2012b) propose an LSMC variant that approximates a value function based on the LSMC continuation function approximation. We refer to this LSMC variant as LSMH, where H denotes hybrid.

An appealing feature of LSMC and LSMH is that they can be used with any term structure model from which term structure elements can be sampled. However, this generality suggests that it may be possible to improve on these methods by developing an LSM method that exploits properties of specific families of term structure models.

We develop an LSM variant to be used in conjunction with term structure models commonly used both in practice and the literature (Ho and Lee, 1986, Cortazar and Schwartz, 1994, Clewlow and Strickland, 2000, Maragos, 2002, Eydeland and Wolyniec, 2003, Veronesi, 2010). As LSMC and LSMH, our approach uses basis functions to solve the first curse of dimensionality, but it differs from LSMC because it approximates a value function and from LSMH because it does so directly. In stark contrast to LSMC and LSMH, the key idea behind our approach is to overcome the second curse of dimensionality by choosing basis functions that allow us to compute expectations in essentially closed-form when employing term structure models; that is, this choice of basis functions allows us to avoid employing the sample average approximation used by both LSMC and LSMH. Examples include common basis functions used in the LSM literature, such as polynomials of term structure elements and prices of call and put options on the term structure (Longstaff and Schwartz, 2001, Andersen and Broadie, 2004, Boogert and De Jong, 2008, 2011/12, Cortazar et al., 2008, Gyurko et al., 2011, Desai et al., 2012b). A catalog of other candidate basis functions can be found in Haug (2006). We refer to our LSM variant as LSMV, where V abbreviates value (function).

We numerically compare the relative performance of LSMC, LSMH, and LSMV on instances of realistic energy swing and storage options. We observe that LSMV needs a considerably smaller number of regression samples than both LSMC and LSMH to obtain near optimal bound estimates with roughly the same accuracy and precision. This improvement leads to moderate computational savings. However, for a given number of evaluation samples, the LSMV computational effort required to estimate dual upper bounds is between one and three orders of magnitude smaller than the analogous singular effort of both LSMC and LSMH, while all three LSM methods exhibit comparable computational effort

when estimating greedy lower bounds. We also perform a worst case error bounding analysis that offers a theoretical view on the relative quality of the bounds that these methods estimate on our instances. The relevance of our proposed method and our findings extends beyond the specific applications considered in this chapter.

Glasserman and Yu (2004) propose an LSMC variant that solves the second curse of dimensionality, only for dual upper bound estimation, by choosing basis functions that satisfy a martingale condition. In contrast, our approach is not based on such martingale condition and overcomes the second curse of dimensionality when estimating both VFAs and bounds. Moreover, these authors focus on the valuation of American options, whereas our method has broader applicability.

The remainder of this chapter is organized as follows. In §3.2 we formulate an MDP for multiple exercise option valuation, apply it to energy swing and storage options, and discuss the two curses of dimensionality that arise when attempting to solve this MDP using stochastic dynamic programming. In §3.3 we discuss the estimation of greedy lower bounds and dual upper bounds. In §3.4 we present LSMC and LSMH. We describe LSMV in §3.5. We perform our error bounding analysis in §3.6. We conduct our numerical study in §3.7. We conclude in §3.8. All proofs are in Appendix B.1. Appendix B.2 includes additional material related to our error bound analysis. Appendix B.3 reports lower and upper bounds estimated using LSMV on the instances of Lai et al. (2010) used in §2.8.

3.2 Option Valuation Model and Curses of Dimensionality

We describe our MDP framework for the valuation of multiple exercise options in §3.2.1 and discuss applications of this MDP to energy swing and storage options in §3.2.2. In §3.2.3 we formulate two SDPs that can (in theory) be used to compute an optimal policy of this MDP and discuss the two curses of dimensionality that make these SDPs intractable.

3.2.1 MDP

There are N exercise dates, each denoted as T_i, $i \in \mathcal{I} := \{0, \ldots, N-1\}$. The set $\mathcal{I}$ is the stage set. The state of our MDP at stage i is partitioned into *endogenous* and *exogenous* components. The endogenous component is the scalar x_i. It belongs to the finite set $\mathcal{X}_i$ that represents information about the number of remaining exercise rights at stage i. The *exogenous* component is the vector $F_i \in \mathbb{R}^{N-i}$ that represents the option underlying term structure $(F_{i,i}, F_{i,i+1}, \ldots, F_{i,N-1})$, where $F_{i,j}$ is the element of the term structure associated with date T_j at time T_i. We define $F_N := 0$. In commodity and energy applications, F_i is a forward curve, $F_{i,i}$ is the time T_i spot price, and $F_{i,j}$ is the date T_i futures price with maturity at time $T_j > T_i$. For fixed income applications, F_i is a bond yield curve and $F_{i,j}$ is the date T_i interest rate of the bond with maturity at date T_j.

At stage i and state (x_i, F_i), the decision maker chooses an exercise action a from the finite set $\mathcal{A}_i(x_i)$, which includes the number of rights that can be exercised at stage i,

and receives the reward $r_i : (a, F_i) \mapsto \mathbb{R}$. Subsequently, the endogenous part of the state transitions from x_i to $x_{i+1} := x_i - a$, and the exogenous part of the state evolves from F_i to F_{i+1} according to a known risk-neutral (risk adjusted) stochastic process. In this chapter we assume that the dynamics of the exogenous information are governed by a term structure model of the type discussed in §3.5.1. However, the models formulated in this section have wider applicability.

Let $\mathbb{E}$ denote expectation under the risk-neutral probability measure for the exogenous information stochastic process (such measure is unique in our setup). A policy π is the collection of decision functions $\{A_0^\pi, \ldots, A_{N-1}^\pi\}$, where $A_i^\pi : (x_i, F_i) \mapsto \mathcal{A}_i(x_i)$, $\forall (i, x_i, F_i) \in \mathcal{I} \times \mathcal{X}_i \times \mathbb{R}^{N-i}$. We let Π be the set of all feasible policies. We denote by $\delta \in (0, 1]$ the risk-free discount factor from each time T_i back to time T_{i-1}, $i \in \mathcal{I} \setminus \{0\}$, that is, the discount factor is constant across stages. This assumption can be relaxed in a straightforward manner. Define $T_0 := 0$. Let (x_0, F_0) be the time T_0 state. Computing the option value $V_0(x_0, F_0)$ and an optimal exercise policy entails solving the MDP

$$\max_{\pi \in \Pi} \sum_{i \in \mathcal{I}} \delta^i \mathbb{E}\left[r_i(A_i^\pi(x_i^\pi, F_i), F_i) | x_0, F_0\right], \tag{3.1}$$

where x_i^π is the random endogenous part of the state at stage i when using policy π.

To simplify our notation, for the most part in the rest of the chapter we omit the sets that index a tuple. For example, we write (i, x_i, F_i, a) in lieu of $(i, x_i, F_i, a) \in \mathcal{I} \times \mathcal{X}_i \times \mathbb{R}^{N-i} \times \mathcal{A}_i(x_i)$. We write $(\cdot)_{-(i)}$ to indicate that i is excluded from $\mathcal{I}$ in the tuple ground set.

3.2.2 Energy Applications

We consider two applications: Energy swing and storage options. They are the focus of our numerical study in §3.7.

Energy swing option. Swing options are common in energy applications (Barbieri and Garman, 1996, Jaillet et al., 2004). We focus on a purchase swing option. This option could be used, for example, by a producer of ethylene that requires an amount $q_i > 0$ of crude oil as input to a thermal cracking process at time T_i for each i. The contract has two parts: A *purchase* part that involves buying the quantity q_i at the strike price K_i on each date T_i; and a *swing* part that endows the producer with $n \leq N$ *swing rights* to increase or decrease each purchase amount q_i by a fixed constant $Q_i \in (0, q_i]$ at the strike price K_i at each stage i. At most one swing right can be exercised at a given stage i.

The incentive to exercise this swing option in stage i stems from the producer's ability to transact in the spot market at the prevailing spot price $F_{i,i}$. If $K_i > F_{i,i}$ the producer has the incentive to purchase a quantity $q_i - Q_i$ from the purchase swing contract at the strike price K_i and purchase a quantity Q_i from the spot market at the price $F_{i,i}$. This combined trade results in a gain of $Q_i(K_i - F_{i,i})$ relative to procuring q_i at the strike price K_i. Similarly, if $K_i < F_{i,i}$, the producer has the incentive to purchase a quantity $q_i + Q_i$

from the purchase swing contract at the strike price K_i and sell a quantity Q_i into the spot market at price $F_{i,i}$, for a gain of $Q_i(F_{i,i} - K_i)$.

Valuing the purchase part of this contract is trivial. The valuation of the swing part of this contract can be modeled using our MDP by defining the endogenous state variable x_i to be the number of available swing rights at stage i. The set $\mathcal{X}_i$ is thus $\{i, \ldots, n\}$. The feasible action set is $\mathcal{A}_i(x_i) := \{0, 1\}$ if $x_i > 0$, and $\mathcal{A}_i(x_i) := \{0\}$ if $x_i = 0$. That is, exercise is allowed only when there is at least one swing right available. The stage i reward function $r_i(a, F_i)$ is defined as $Q_i \cdot |K_i - F_{i,i}| \cdot a$.

Energy storage option. Consider a finite-term lease contract on a portion of the space and capacity of an energy (e.g., natural gas) storage facility (see Secomandi 2010 and Lai et al. 2010 for details). At each of a given number of dates, the contract owner can buy energy from the wholesale spot market and inject it into this facility or withdraw from the leased facility previously purchased and injected energy and sell it into the wholesale spot market. The valuation of the energy storage contract can be modeled using our MDP by appropriately defining the state and action spaces and the reward function.

The endogenous state x_i is the inventory in storage at stage i. The maximum amount of inventory allowed by the storage contract is $\bar{x}$. The feasible inventory set in stage i is $\mathcal{X}_i := [0, \bar{x}]$. In each stage, the storage contract withdrawal and injection capacities are $\underline{a}$ and $\overline{a}$. They satisfy $0 \leq \underline{a}, \overline{a} \leq \bar{x}$. At stage i, a positive action is an energy withdrawal and sell decision, a negative action is an energy purchase and inject decision, and zero is the do nothing decision. The set of feasible injections, withdrawals, and overall actions are $\mathcal{A}_i^I(x_i) := [\max\{-\overline{a}, (x_i - \bar{x})\}, 0]$, $\mathcal{A}_i^W(x_i) := [0, \min\{x_i, \underline{a}\}]$, and $\mathcal{A}_i(x_i) := \mathcal{A}_i^I(x_i) \cup \mathcal{A}_i^W(x_i)$, respectively. Although the sets $\mathcal{X}_i$, $\mathcal{A}_i^I(x_i)$, and $\mathcal{A}_i^W(x_i)$ are intervals, by Lemma 1 in Secomandi et al. (2012) they can be optimally discretized if $\underline{a}$, $\overline{a}$, and $\bar{x}$ are rational. We assume this to be the case in this chapter.

Let the coefficients $\alpha^W \in (0, 1]$ and $\alpha^I \geq 1$ model energy losses associated with energy withdrawals and injections, respectively, and the coefficients ς^W and ς^I represent withdrawal and injection marginal costs, respectively. The immediate reward function is $r_i(a, F_i) := (\alpha^I F_{i,i} + \varsigma^I)a$ if $a \in \mathbb{R}_-$; and $r_i(a, F_i) := (\alpha^W F_{i,i} - \varsigma^W)a$ if $a \in \mathbb{R}_+$.

3.2.3 SDPs and Curses of Dimensionality

We formulate two SDPs to solve, at least in theory, the MDP (3.1): The *value function* SDP and the *continuation function* SDP. The LSM methods discussed in §§3.4-3.5 approximate these SDPs.

Let f_i denote a generic function with support on the stage i state space $\mathcal{X}_i \times \mathbb{R}^{N-i}$. We define the stage and state dependent operator

$$\mathcal{L}_{(i,x_i,F_i)} f_{i+1} := \max_a r_i(a, F_i) + \delta \mathbb{E}\left[f_{i+1}\left(x_i - a, F_{i+1}\right) | F_i\right]. \tag{3.2}$$

In theory, an optimal policy to the MDP (3.1) can be obtained by stochastic dynamic programming. The value function SDP, $\forall(i, x_i, F_i)$, is

$$V_i(x_i, F_i) = \mathcal{L}_{(i,x_i,F_i)} V_{i+1}, \tag{3.3}$$

with boundary conditions $V_N(x_N, F_N) := 0$, $\forall x_N$, where $V_i(x_i, F_i)$ is the optimal value function in stage i and state (x_i, F_i).

The continuation function SDP is based on the continuation function $C_i(x_{i+1}, F_i)$, $\forall(i, x_{i+1}, F_i)$, which is defined as

$$C_i(x_{i+1}, F_i) := \delta \mathbb{E}\left[V_{i+1}\left(x_{i+1}, F_{i+1}\right) | F_i\right]. \tag{3.4}$$

Let g_i denote a generic function with support on $\mathcal{X}_{i+1} \times \mathbb{R}^{N-i}$. We define the operator

$$\mathcal{H}_{(i,x_i,F_i)} g_i := \max_a r_i(a, F_i) + g_i(x_i - a, F_i). \tag{3.5}$$

The continuation function SDP, $\forall(i, x_{i+1}, F_i)$, is

$$C_i(x_{i+1}, F_i) = \delta \mathbb{E}\left[\mathcal{H}_{(i+1,x_{i+1},F_{i+1})} C_{i+1} | F_i\right], \tag{3.6}$$

with boundary conditions $C_{N-1}(x_N, F_{N-1}) := 0, \forall(x_N, F_{N-1})$. In this case, the option value $V_0(x_0, F_0)$ is $\mathcal{H}_{(0,x_0,F_0)} C_0$. The SDP (3.6) can be derived by substituting for $V_{i+1}\left(x_{i+1}, F_{i+1}\right)$ in the right hand side of (3.4) using the right hand side of (3.3) expressed for $i+1$ and simplifying the resulting expression using the operator defined in (3.5).

Solving the SDPs (3.3) and (3.6) is typically intractable due to two curses of dimensionality: (i) The high dimensionality of the value function and continuation function caused by the presence of the high dimensional term structure in the states of these SDPs; (ii) the inability to evaluate exactly the expectations $\mathbb{E}\left[V_{i+1}\left(x_{i+1}, F_{i+1}\right) | F_i\right]$ and $\mathbb{E}\left[\mathcal{H}_{(i+1,x_{i+1},F_{i+1})} C_{i+1} | F_i\right]$ in these SDPs when using common term structure models (see §3.5.1). Of course, these expectations can be computed exactly when using discretization techniques, such as lattices, but this approach is limited to models with one or two stochastic factors with specific structure, in which case, the first curse of dimensionality is also solved (see, e.g., Schwartz and Smith 2000 and Jaillet et al. 2004 for energy applications). In contrast, dealing with more realistic term structure models of the type used in this chapter requires adopting a different approach to break these two curses of dimensionality. We discuss Monte Carlo based methods in §§3.3-3.5.

3.3 Bounding the Option Value

In this section we discuss standard Monte Carlo ADP approaches for heuristically solving the two curses of dimensionality discussed in §3.2.3. These approaches determine a heuristic exercise policy and estimate bounds on the option value $V_0(x_0, F_0)$ (see §6.1.1 in Bertsekas 2007, Brown et al. 2010, and Powell 2011). These methods rely on (i) low dimensional value function approximations (VFAs) and continuation function approximations (CFAs)

for breaking the first curse of dimensionality and (ii) sample average approximations of expectations for breaking the second curse of dimensionality. Let $\hat{V}_i(x_i, F_i)$ and $\hat{C}_i(x_{i+1}, F_i)$ be *given* VFA and CFA, respectively. We discuss methods to compute VFAs and CFAs in §§3.4-3.5.

To estimate a lower bound on the option value $V_0(x_0, F_0)$ one generates a set of W term structure *evaluation* sample paths $\{F_i^w, \forall(i,w)\}$ $(w \in \{1, \ldots, W\})$ starting from the term structure F_0 at time T_0, and simulates the greedy policy induced by the VFA or CFA . That is, on each sample path, at each stage i and state (x_i, F_i) a greedy action is computed by solving $\mathcal{L}_{(i,x_i,F_i)}\hat{V}_{i+1}$ when using a VFA and $\mathcal{H}_{(i,x_i,F_i)}\hat{C}_i$ when using a CFA, with the understanding that the expectation $\mathbb{E}\left[\hat{V}_{i+1}\left(x_{i+1}, F_{i+1}\right)|F_i\right]$ appearing in $\mathcal{L}_{(i,x_i,F_i)}\hat{V}_{i+1}$ is replaced by its sample average approximation, which requires additional inner simulation (see Gyurko et al. 2011 and Desai et al. 2012b, no such approximation is needed when a CFA is used). A greedy lower bound on the option value is estimated by averaging the sums of time T_0 discounted rewards obtained from implementing the greedy actions computed along each sample path. Obviously, greedy optimizations can be used to determine a heuristic control policy, that is, a sequence of feasible actions for the stages and states encountered when managing the option.

The quality of the estimated greedy lower bound can be assessed by estimating dual upper bounds by applying the information relaxation and duality framework (see Brown et al. 2010, and references therein). This approach relies on the availability of *feasible* dual penalties $p_i(x_{i+1}, F_{i+1}, F_i)$ that penalize knowledge at time T_i of the future information F_{i+1}: The feasibility requirement is $\mathbb{E}[p_i(x_{i+1}, F_{i+1}, F_i)|F_i] \leq 0$ (see Brown et al. 2010 for details). Such penalties can be defined using a VFA as follows:

$$\hat{V}_{i+1}(x_{i+1}, F_{i+1}) - \mathbb{E}\left[\hat{V}_{i+1}(x_{i+1}, F_{i+1})|F_i\right], \tag{3.7}$$

where the first term is the stage $i+1$ VFA and the second term is the undiscounted stage i CFA induced by the stage $i+1$ VFA. Analogous to the VFA-based lower bound estimation, the expectation $\mathbb{E}\left[\hat{V}_{i+1}(x_{i+1}, F_{i+1})|F_i\right]$ in (3.7) is replaced by its sample average approximation.

A CFA penalty analogous to (3.7) is typically obtained by replacing the first and second terms in (3.7) by the stage $i+1$ VFA $\mathcal{H}_{(i+1,x_{i+1},F_{i+1})}\hat{C}_{i+1}$, which is *induced* by the CFA $\hat{C}_{i+1}$, and the undiscounted stage i CFA $\hat{C}_i(x_{i+1}, F_i)/\delta$, respectively. This dual penalty is

$$\mathcal{H}_{(i+1,x_{i+1},F_{i+1})}\hat{C}_{i+1} - \hat{C}_i(x_{i+1}, F_i)/\delta. \tag{3.8}$$

When using the penalties (3.8), it holds that

$$\begin{aligned}\mathbb{E}\left[\mathcal{H}_{(i+1,x_{i+1},F_{i+1})}\hat{C}_{i+1}|F_i\right] - \hat{C}_i(x_{i+1}, F_i)/\delta \geq \\ \mathbb{E}[\hat{C}_{i+1}(x_{i+1}, F_{i+1})|F_i] - \hat{C}_i(x_{i+1}, F_i)/\delta,\end{aligned} \tag{3.9}$$

where the inequality is obtained by assuming $r_{i+1}(0, F_{i+1}) = 0$, $\forall F_{i+1}$ (this assumption is satisfied for the applications discussed in §3.2.2). In general, the right hand side of (3.9)

can be strictly positive, which implies that the dual penalties (3.8) can be infeasible, that is, they can lead to invalid dual upper bounds on the option value.

In contrast, feasible dual penalties can be defined by replacing $\hat{V}_{i+1}(x_{i+1}, F_{i+1})$ in both the first and second terms of (3.7) by the induced VFA:

$$\mathcal{H}_{(i+1,x_{i+1},F_{i+1})}\hat{C}_{i+1} - \mathbb{E}\left[\mathcal{H}_{(i+1,x_{i+1},F_{i+1})}\hat{C}_{i+1}|F_i\right]. \tag{3.10}$$

The expectation $\mathbb{E}\left[\mathcal{H}_{(i+1,x_{i+1},F_{i+1})}\hat{C}_{i+1}|F_i\right]$ in (3.10) cannot be computed exactly because of the presence of a maximization in the operator $\mathcal{H}_{(i+1,x_{i+1},F_{i+1})}$ inside this expectation. It is standard to replace this expectation by its sample average approximation, an approximation that requires additional inner simulation and can be burdensome (Andersen and Broadie, 2004, Haugh and Kogan, 2007).

Consider the same set of W term structure sample paths $\{F_i^w, \forall(i,w)\}$ employed for greedy lower bound estimation. Once dual feasible penalties are specified, a point estimate $U_0^w(x_0)$ of a dual upper bound on the option value $V_0(x_0, F_0)$ can be obtained by solving the following deterministic dynamic program defined on the w-th term structure sample path:

$$U_i^w(x_i) = \max_a r_i(a, F_i^w) - p_i(x_i - a, F_{i+1}^w, {F_i}^w) + \delta U_{i+1}^w(x_i - a),$$

$\forall(i, x_i)$, with boundary conditions $U_N^w(x_N) := 0$, $\forall x_N$. A dual upper bound estimate on the sought option value is the average of the point estimates $U_0^w(x_0)$, $\forall w$.

3.4 Standard LSM Method and Variant

In §3.4.1 we discuss an ideal template for computing a CFA. In §§3.4.2-3.4.3 we describe LSMC and one of its variants, LSMH.

We define a CFA as a linear combination of a number B_i^C of basis functions. These types of functions are commonly used in the ADP literature (e.g., see §6.1.1 in Bertsekas 2007 and page 326 in Powell 2011). Let $\psi_{i,b}$ denote the b-th CFA basis function in stage i and $\theta_{i,b}$ its associated weight. For each stage $i_{-(N-1)}$, we define the vector of CFA basis functions $\Psi_i := (\psi_{i,1}, \ldots, \psi_{i,B_i^C})$ and the vector of basis function weights $\theta_i := (\theta_{i,1}, \ldots, \theta_{i,B_i^C})$. We define a CFA as

$$(\Psi_i\theta_i)(x_{i+1}, F_i) := \sum_b \psi_{i,b}(x_{i+1}, F_i)\theta_{i,b}. \tag{3.11}$$

Thus, the problem of determining the stage i CFA reduces to computing the vector of weights θ_i.

3.4.1 Ideal Template

Template I (I denotes ideal) describes the steps of an ideal LSM procedure for computing the weights of a CFA. The inputs to this procedure are the number of sample paths and

Template I: Ideal LSM procedure for computing a CFA

Inputs: Number of sample paths P and set of basis function vectors $\{\Psi_i, \forall i_{-(N-1)}\}$.

Initialization: Generate the set of P term structure sample paths $\{F_i^p, \forall(i,p)\}$; $\bar{\theta}_{N-1} := 0$.

For each $i = N-2$ to 0 **do**:

(i) **For** each (x_{i+1}, p) **do**: Compute the stage i CFA estimate

$$c_i(x_{i+1}, p) := \delta \mathbb{E}\left[\mathcal{H}_{(i+1,x_{i+1},F_{i+1})}(\Psi_{i+1}\bar{\theta}_{i+1})|F_i^p\right].$$

(ii) Perform a 2-norm regression on the CFA estimates in set $\{c_i(x_{i+1}, p), \forall(x_{i+1}, p)\}$ to determine the weights $\bar{\theta}_i$.

the basis function sets at each stage. Template I begins by generating P term structure *regression* sample paths $\{F_i^p, \forall(i,p)\}$ ($p \in \{1, \ldots, P\}$) and initializing the stage $N-1$ weight vector $\bar{\theta}_{N-1}$ to zero. Then, at each stage i, starting from stage $N-2$ and moving backwards to stage 0, this procedure performs the following steps: In Step (i), it computes estimates $c_i(x_{i+1}, p)$ of the stage i CFA obtained by replacing the stage $i+1$ continuation function C_{i+1} in the SDP (3.6) with the known stage $i+1$ CFA ($\Psi_{i+1}\bar{\theta}_{i+1}$); in Step (ii), it performs a 2-norm regression on these estimates to determine the stage i CFA weights.

Template I is ideal because in general the expectation $\delta\mathbb{E}[\mathcal{H}_{(i+1,x_{i+1},F_{i+1})}(\Psi_{i+1}\bar{\theta}_{i+1})|F_i^p]$ in Step (i) cannot be computed exactly, due to the second curse of dimensionality. Approximations of this expectation are thus required to make the template practical.

Algorithm 1: LSMC

Inputs: Number of sample paths P and set of basis function vectors $\{\Psi_i, \forall i_{-(N-1)}\}$.

Initialization: Generate the set of P term structure sample paths $\{F_i^p, \forall(i,p)\}$; $\bar{\theta}_{N-1} := 0$.

For each $i = N-2$ to 0 **do**:

(i) **For** each (x_{i+1}, p) **do**: Compute the stage i CFA estimate

$$\hat{c}_i(x_{i+1}, p) := \delta\mathcal{H}_{(i+1,x_{i+1},F_{i+1}^p)}(\Psi_{i+1}\bar{\theta}_{i+1}).$$

(ii) Perform a 2-norm regression on the CFA estimates in set $\{\hat{c}_i(x_{i+1}, p), \forall(x_{i+1}, p)\}$ to determine the weights $\bar{\theta}_i$.

3.4.2 LSMC

The LSMC procedure proposed by Longstaff and Schwartz (2001) and Tsitsiklis and Van Roy (2001) computes a CFA by approximating the expectation $\mathbb{E}[\mathcal{H}_{(i+1,x_{i+1},F_{i+1})}(\Psi_{i+1}\bar{\theta}_{i+1})|F_i^p]$ in Step (i) of Template I using a sample average approximation based only on the p-th sample path:

$$\mathbb{E}\left[\mathcal{H}_{(i+1,x_{i+1},F_{i+1})}(\Psi_{i+1}\bar{\theta}_{i+1})|F_i^p\right] \approx \mathcal{H}_{(i+1,x_{i+1},F_{i+1}^p)}(\Psi_{i+1}\bar{\theta}_{i+1}).$$

Using this approximation, which is based on the already available p-th sample path, allows LSMC to avoid inner simulations in Step (i). However, a drawback of this approximation is its high variance. We summarize LSMC in Algorithm 1, which differs from Template I only in the definition of the CFA estimates in Step (i).

As discussed in §3.3, estimating a greedy lower bound using the LSMC CFA is easy but estimating a dual upper bound using this CFA is challenging.

Algorithm 2: LSMH

Inputs: Number of sample paths P and sets of basis function vectors $\{\Psi_i, \forall i_{-(N-1)}\}$ and $\{\Phi_i, \forall i_{-(0)}\}$.

Initialization: Generate the set of P term structure sample paths $\{F_i^p, \forall(i,p)\}$; $\bar{\theta}_{N-1} := 0$, and $\bar{\gamma}_N := 0$.

For each $i = N-2$ to 0 **do**:

(i) **For** each (x_{i+1}, p) **do**: Compute the stage i CFA estimate

$$\hat{c}_i(x_{i+1},p) := \delta\mathcal{H}_{(i+1,x_{i+1},F_{i+1}^p)}(\Psi_{i+1}\bar{\theta}_{i+1}).$$

(ii) Perform a 2-norm regression on the CFA estimates in set $\{\hat{c}_i(x_{i+1},p), \forall(x_{i+1},p)\}$ to determine the weights $\bar{\theta}_i$.

(iii) Perform a 2-norm regression on the VFA estimates in set $\{(1/\delta)\hat{c}_i(x_{i+1},p), \forall(x_{i+1},p)\}$ to determine the weights $\bar{\gamma}_{i+1}$.

3.4.3 LSMH

LSMH is a variant of LSMC proposed by Gyurko et al. (2011) and Desai et al. (2012b) that overcomes the LSMC computational burden of estimating dual upper bounds. LSMC does so by using the LSMC CFA to compute a VFA. Analogous to our CFA definition, a VFA is a linear combination of B_i^V basis functions. Let $\phi_{i,b}$ be the b-th VFA basis function in stage i and $\gamma_{i,b}$ the weight associated with this basis function. Let $\Phi_i := (\phi_{i,1}, \ldots, \phi_{i,B_i^V})$

and $\beta_i := (\beta_{i,1}, \ldots, \beta_{i,B_i^V})$. The VFA is

$$(\Phi_i \gamma_i)(x_i, F_i) := \sum_b \phi_{i,b}(x_i, F_i)\gamma_{i,b}. \tag{3.12}$$

Algorithm 2 outlines the LSMH steps. Comparing Algorithm 2 with Algorithm 1 shows that the only differences between LSMH and LSMC are the additional LSMH input set of VFA basis function vectors and the regression Step (iii) in Algorithm 2 that estimates a stage $i+1$ VFA from the stage i CFA of LSMC. This VFA estimation is based on the stage $i+1$ VFA estimates induced by the stage $i+1$ LSMC CFA, that is, $(1/\delta)\hat{c}_i(x_{i+1}, p) \equiv \mathcal{H}_{(i+1,x_{i+1},F_{i+1}^p)}(\Psi_{i+1}\bar{\theta}_{i+1})$.

Dual upper bounds and greedy lower bounds can be estimated using the LSMH VFA as discussed in §3.3. Gyurko et al. (2011) and Desai et al. (2012b) employ the LSMH VFA to estimate dual upper bounds. Gyurko et al. (2011) also estimate greedy lower bounds using the LSMH VFA. In contrast, Desai et al. (2012b) estimate greedy lower bounds using the LSMC CFA.

3.5 LSM Method for Term Structure Models

In §3.5.1 we describe term structure models. In §3.5.2 we introduce LSMV, which exploits a key property of these models.

3.5.1 Term Structure Models

Term structure models are widespread in commodity, energy, and fixed income applications both in practice and in the academic literature (Ho and Lee, 1986, Cortazar and Schwartz, 1994, Clewlow and Strickland, 2000, Maragos, 2002, Eydeland and Wolyniec, 2003, Veronesi, 2010). In these models, the term structure evolution is governed by the risk-neutral dynamics of a multidimensional diffusion model. In this continuous time setting, we denote by $F(t, T_j)$ the value of the element of the term structure at time $t \in [T_0, T_j]$ with maturity at time T_j, $\forall j \in \mathcal{I}$. Hence, if $t = T_i$, $i \in \mathcal{I}$, and $j > i$, then $F(t, T_j) \equiv F_{i,j}$. Given a fixed number $K \in \{1, \ldots, N-1\}$ of stochastic factors, the evolution of $F(t, T_j)$, $\forall j \in \mathcal{I} \setminus \{0\}$ and $t \in (0, T_j]$ is governed by the following stochastic differential equations:

$$\frac{dF(t, T_j)}{F(t, T_j)} = \sum_{k=1}^{K} \sigma_{j,k}(t) dW_k(t), \ \forall j \in \mathcal{I} \setminus \{0\}, t \in (0, T_j], \tag{3.13}$$

$$dW_k(t) dW_{k'}(t) = 0, \ \forall k, k' \in \{1, \ldots, K\}, \ k \neq k', \tag{3.14}$$

where $\sigma_{j,k}(t)$ is the time t loading coefficient on the Brownian motion W_k for the term structure element $F(t, T_j)$.

The model (3.13)-(3.14) is quite general. It captures seasonality in the variances and covariances of changes in the term structure elements because the loading factors are time

dependent, and seasonality in the term structure levels from the seasonality in the initial (time T_0) term structure.

Under model (3.13)-(3.14), it is possible to compute (sometimes approximate) conditional expectations of certain classes of functions of future term structure elements as essentially closed form functions of current term structure elements, which is due to the joint lognormality of the relevant distributions. This is a key property exploited by LSMV. We provide three examples of such classes of functions below (see Haug 2006 for a catalog):

1. All polynomials of term structure elements. For example, when $i' > i$, we can use the property $\mathbb{E}[F_{i',j}|F_{i,j}] = F_{i,j}$ to compute expectations of functions that are linear in the term structure elements, and the property $\mathbb{E}[F_{i',j}^2|F_{i,j}] = F_{i,j}^2 \exp(\sum_{k\in\mathcal{K}} \int_{T_i}^{T_{i'}} \sigma_{j,k}^2(t)dt)$ to compute expectations of quadratic functions of such elements (these properties are easy to verify).

2. Prices of call and put options on the term structure elements: $\mathbb{E}[(F_{i',j} - \overline{K})^+|F_{i,j}]$ and $\mathbb{E}[(\overline{K} - F_{i',j})^+|F_{i,j}]$, where $i' > i$, and $\overline{K} \in \mathbb{R}$ is the given strike price (see §1.1.3 of Haug 2006 for explicit formulas for these prices).

3. Prices of spread options on term structure elements: $\mathbb{E}[(\lambda_1 F_{i',l} - \lambda_2 F_{i',j} - \overline{K})^+|F_{i,j}, F_{i,l}]$, where $l > j \geq i' > i$ and λ_1 and λ_2 are given constants. Since a closed form expression for this price is not available under model (3.13)-(3.14) when $\overline{K} \neq 0$, one can instead use the near-optimal and essentially closed form lower bound on this price developed by Bjerksund and Stensland (2011) (see Margrabe 1978 for the case $\overline{K} = 0$).

3.5.2 LSMV

LSMV computes a VFA by approximating the value function SDP (3.3) using basis functions. We define the stage i LSMV VFA analogously to (3.12) but we denote its basis function weights by β_i; that is, the stage i LSMV VFA is $\Phi_i\beta_i$. Approximating the SDP (3.3) using these basis functions requires computing expectations of next stage VFAs, that is, the terms $\mathbb{E}[(\Phi_{i+1}\beta_{i+1})(\cdot, F_{i+1})|F_i]$ in stage i. The main idea behind LSMV is to choose basis functions in the context of the term structure model (3.13)-(3.14) to avoid approximating these expectations. We do this in two steps:

1. We choose basis functions $\phi_{i,b}(x_i, F_i)$ that are separable in the endogenous (EN) and exogenous (EX) components of the state, that is,

$$\phi_{i,b}(x_i, F_i) \equiv \phi_{i,b}^{EN}(x_i)\phi_{i,b}^{EX}(F_i) \tag{3.15}$$

for given functions $\phi_{i,b}^{EN}(x_i)$ and $\phi_{i,b}^{EX}(F_i)$. The expectation $\mathbb{E}[(\Phi_{i+1}\beta_{i+1})(x_{i+1}, F_{i+1})|F_i]$ is then

$$\sum_b \phi_{i+1,b}^{EN}(x_{i+1})\mathbb{E}[\phi_{i+1,b}^{EX}(F_{i+1})|F_i]\beta_{i+1,b}.$$

Since in this expression $\phi_{i+1,b}^{EN}(x_{i+1})$ is outside the expectation, it can be any function of x_{i+1} that can be evaluated.

2. We choose the function of the exogenous state component $\phi^{EX}_{i+1,b}(F_{i+1})$ such that each expectation $\mathbb{E}[\phi^{EX}_{i+1,b}(F_{i+1})|F_i]$ is a function of the term structure F_i that can be computed in essentially closed form by exploiting the property of the term structure model (3.13)-(3.14) discussed at the end of §3.5.1, that is,

$$\mathbb{E}[\phi^{EX}_{i+1,b}(F_{i+1})|F_i] = h_{i,b}(F_i) \tag{3.16}$$

for some known function $h_{i,b}(F_i)$.

The three choices of basis functions discussed at the end of §3.5.1 can thus be used as basis functions of the exogenous part of the state that satisfy (3.16). Look-up tables can be used as basis functions of the finite endogenous part of the state. These choices of basis functions are commonly used in the literature to instantiate a CFA when using LSMC (Longstaff and Schwartz, 2001, Boogert and De Jong, 2008, Cortazar et al., 2008). In contrast, we also use them to instantiate a VFA in our numerical study in §3.7.

Algorithm 3: LSMV

Inputs: Number of sample paths P and set of basis function vectors $\{\Phi_i, \forall i_{-(0)}\}$ that satisfy (3.15)-(3.16).

Initialization: Generate the set of P term structure sample paths $\{F^p_i, \forall(i,p)\}$; $\bar{\beta}_N := 0$.

For each $i = N-1$ to 1 **do**:

(i) **For** each (x_i, p) **do**: Compute the stage i VFA estimate

$$v_i(x_i, p) := \mathcal{L}_{(i,x_i,F^p_i)}(\Phi_{i+1}\bar{\beta}_{i+1}).$$

(ii) Perform a 2-norm regression on the VFA estimates in set $\{v_i(x_i,p), \forall(x_i,p)\}$ to determine the weights $\bar{\beta}_i$.

The LSMV steps are summarized in Algorithm 3. The inputs to LSMV are the number of sample paths and VFA basis function sets that satisfy (3.15)-(3.16). LSMV starts by generating the set of P term structure sample paths and initializing the stage N weight vector $\bar{\beta}_N$ to zero. Then, at each stage i, starting from stage $i = N-1$ and moving backward to stage $i = 1$, it computes in Step (i) estimates $v_i(x_i, p)$ of the stage i VFA by replacing the stage $i+1$ value function V_{i+1} in $\mathcal{L}_{(i,x_i,F^p_i)}V_{i+1}$ – the right hand side of (3.3) – by the known stage $i+1$ VFA $\Phi_{i+1}\bar{\beta}_{i+1}$. In Step (ii), LSMV performs a 2-norm regression on these estimates to determine the stage i regression weights $\bar{\beta}_i$.

The operator $\mathcal{L}_{(i,x_i,F^p_i)}(\Phi_{i+1}\bar{\beta}_{i+1})$ used to compute the VFA estimates in Step (i) of LSMV includes the expectation $\mathbb{E}[(\Phi_{i+1}\beta_{i+1})(\cdot, F_{i+1})|F_i]$. It is our choice of basis functions that satisfy (3.15)-(3.16) and our use of the term structure model (3.13)-(3.14) that allows us to compute this expectation. Moreover, we can exactly compute similar expectations

that arise when estimating greedy lower bounds and dual upper bounds using the LSMV VFA (see §3.3). Thus, LSMV eliminates the second curse of dimensionality in §3.3 without resorting to sample average approximations, as done by LSMC and LSMH.

3.6 Error Bounding Analysis

In this section, we analyze LSMC, LSMH, and LSMV by deriving and comparing worst case (that is, ∞-norm) bounds on the errors incurred when using these methods for a fixed number of both regression and evaluation samples. The premise behind our analysis is that a smaller worst case bound on the error associated with a given method compared to the error associated with another method provides some theoretical support for the claim that the former method should likely outperform the latter method. In §3.6.1 we discuss the assumptions underlying the analysis performed in §§3.6.2-3.6.4. In §3.6.2 we establish a preliminary result that we use in §3.6.3 and §3.6.4 to investigate the greedy lower bounds and the dual upper bounds, respectively, estimated by LSMC, LSMH, and LSMV. We summarize the theoretical predictions regarding the likely relative performance of each method in §3.6.5.

3.6.1 Assumptions

The SDPs (3.3) and (3.6) have finite action spaces but partially continuous state spaces. To simplify our analysis, we focus on sampled versions of these SDPs with finite state and action spaces so that all norms used in our analysis are defined over finite domains. This finiteness assumption could possibly induce some discretization error but this error will be the same across all methods and thus will not affect statements regarding their relative performance. Our sampled versions of SDPs (3.3) and (3.6) are formulated using the same regression sample paths of the term structure F_i used by LSMC, LSMH, and LSMV, that is, $\{F_i^p, \forall (x_i, p)\}$. Throughout this section we also refer to these sample paths as *formulation* sample paths. Denote by $\mathbb{E}^s$ expectation with respect to a probability distribution on these sampled term structures (the superscript s denotes sampled). Let $\mathcal{L}_i^s$ be the stage i operator defined by (3.2) with $\mathbb{E}$ replaced by $\mathbb{E}^s$. The sampled finite state value function SDP is

$$V_i^s(x_i, F_i^p) = \mathcal{L}_{(i,x_i,F_i^p)}^s V_{i+1}^s, \tag{3.17}$$

$\forall (i, x_i, p)$, with boundary conditions $V_N^s(x_N, F_N) := 0, \forall x_N$, where $V_i^s(x_i, F_i^p)$ is the optimal value function in stage i and state (x_i, F_i^p). The sampled finite state continuation function SDP is

$$C_i^s(x_{i+1}, F_i^p) = \delta \mathbb{E}^s \left[\mathcal{H}_{(i+1,x_{i+1},F_{i+1})} C_{i+1}^s | F_i^p \right], \tag{3.18}$$

$\forall (i, x_{i+1}, p)$, with boundary conditions $C_{N-1}^s(x_N, F_{N-1}^p) := 0$, $\forall (x_N, p)$, where $C_i^s(x_{i+1}, F_i^p)$ is the optimal continuation function in stage i and state (x_{i+1}, F_i^p).

We analyze versions of LSMV, LSMC, and LSMH to approximately solve the SDPs (3.17) and (3.18). For notational simplicity, we continue to refer to their respective

VFA/CFA regression weights by $\bar{\beta}_i$, $\bar{\gamma}_i$, and $\bar{\theta}_i$, that is, we do not superscript these quantities with s. We also do not superscript the estimates v_i, c_i, and $\hat{c}_i$ and continue to refer to the considered LSM versions as LSMV, LSMC, and LSMH.

These LSM methods solve sequences of 2-norm regression problems to compute the weights corresponding to their basis function approximations. For ease of analysis, we assume that each of these regression problems has a unique optimal solution. For example, at stage i, consider the 2-norm regression problem solved in Step (ii) of LSMV to determine the weights $\bar{\beta}_i$:

$$\min_{\beta_i} \|(\Phi_i \beta_i)(\cdot) - v_i(\cdot)\|_2 \,, \tag{3.19}$$

where $\|g(\cdot)\|_2 := \left(\sum_{d \in \mathcal{D}_g} (g(d))^2\right)^{1/2}$ is the 2-norm of a function g with finite domain $\mathcal{D}_g$. Define Y as the regression matrix with $P \cdot |\mathcal{X}_i|$ rows and B_i^V columns and whose element in row (x_i, p) and column b is $\phi_{i,b}(x_i, F_i^p)$. We assume that Y has full column rank. Under this assumption, the unique optimal solution to (3.19) is $\bar{\beta}_i = (Y^T Y)^{-1} Y^T v_i$. We can thus define the projection operator associated with the 2-norm regression problem (3.19) as $\Pi_2^\Phi := \Phi_i (Y^T Y)^{-1} Y^T$, where we suppress the dependence of Π_2^Φ on stage i and Y for notational simplicity. This operator allows us to succinctly represent a 2-norm regression on v_i involving the VFA basis functions as $\Pi_2^\Phi v_i = \Phi_i \bar{\beta}_i$. We make analogous assumptions for the 2-norm regression problems involving the vector of CFA basis functions Ψ_i and denote the associated projection operators by Π_2^Ψ.

3.6.2 VFA/CFA Estimation

Denote the ∞-norm of a generic function g with domain $\mathcal{D}_g$ by $\|g(\cdot)\|_\infty := \max_{d \in \mathcal{D}_g} g(d)$. We define the error $\tilde{e}_i^C$ incurred when approximating C_i^s with the LSMC stage i CFA and the errors $\tilde{e}_i^V$ and $\tilde{e}_i^H$ incurred when approximating V_i^s with the LSMV and LSMH stage i VFAs, respectively, as

$$\begin{aligned}
\tilde{e}_i^C &:= \left\|(\Psi_i \bar{\theta}_i)(\cdot) - C_i^s(\cdot)\right\|_\infty \,, \\
\tilde{e}_i^H &:= \|(\Phi_i \bar{\gamma}_i)(\cdot) - V_i^s(\cdot)\|_\infty \,, \\
\tilde{e}_i^V &:= \left\|(\Phi_i \bar{\beta}_i)(\cdot) - V_i^s(\cdot)\right\|_\infty \,.
\end{aligned}$$

We bound these errors in terms of the following errors:

$$e_i^* := \left\|\Pi_2^\Phi V_i^s(\cdot) - V_i^s(\cdot)\right\|_\infty \,, \tag{3.20}$$

$$e_i^{**} := \left\|\Pi_2^\Psi C_i^s(\cdot) - C_i^s(\cdot)\right\|_\infty \,, \tag{3.21}$$

$$\bar{e}_i^C := \left\|\Pi_2^\Psi \hat{c}_i(\cdot) - \Pi_2 c_i(\cdot)\right\|_\infty \,, \tag{3.22}$$

$$\bar{e}_i^H := \left\|\Pi_2^\Phi \hat{c}_{i-1}(\cdot)/\delta - \hat{c}_{i-1}(\cdot)/\delta\right\|_\infty \,. \tag{3.23}$$

The terms e_i^* and e_i^{**} are regression errors incurred when using the basis functions ϕ_i and ψ_i to approximate the value function V_i^s and the continuation function C_i^s, respectively. The term $\bar{e}_i^C$ is the error incurred by LSMC when estimating a CFA by regressing on the

estimates $\hat{c}_i(x_{i+1}, p)$ of the stage i continuation function $\delta\mathbb{E}^s[\mathcal{H}_{(i+1,x_{i+1},F_{i+1})}(\Psi_{i+1}\bar{\theta}_{i+1})|F_i^p]$, instead of regressing on the evaluations $c_i(x_i, p)$ of this continuation function. The term $\bar{e}_i^H$ is the regression error incurred in Step (iii) of LSMH when computing a VFA by regressing on the value function estimates $\hat{c}_{i-1}(x_i, p)/\delta$ of the LSMC CFA.

Consider LSMV, which approximates the SDP (3.17). LSMV differs from (3.17) at stage i in two ways: (a) in Step (i) LSMV computes the estimates $v_i(x_i, p)$ by using the stage $i+1$ value function approximation $\Phi_{i+1}\bar{\beta}_{i+1}$ as an argument to the operator $\mathcal{L}^s_{(i,x_i,F_i^p)}(\cdot)$, whereas the SDP (3.17) computes V_i^s using V_{i+1}^s as an argument to the same operator; and (b) in Step (ii) LSMV regresses over these estimates to compute the stage i VFA, whereas the SDP (3.17) is not based on regression. These differences introduce separate errors in the LSMV VFA, that is, they contribute to $\tilde{e}_i^V$ differently. However, both these errors, and hence $\tilde{e}_i^V$, can be *bounded* by sums of discounted regression errors (3.20), as shown in Part (a) of Lemma 1.

The CFA estimated by LSMC includes two analogous errors, that is, these errors contribute to $\tilde{e}_i^C$. These errors can be bounded by sums of discounted regression errors (3.21). In addition, LSMC incurs an error to replace the expectation on the right hand side of the SDP (3.18) with a single sample approximation (see Step (i) of LSMC). This error also contributes to $\tilde{e}_i^C$ and can be bounded by $\bar{e}_i^C$. Part (b) of Lemma 1 presents the resulting bound on $\tilde{e}_i^C$.

LSMH computes its stage i VFA using a regression on value function estimates obtained from the stage i LSMC CFA. Thus, the same errors that contribute to $\tilde{e}_i^C$ also contribute to $\tilde{e}_i^H$, but this regression adds to $\tilde{e}_i^H$. This additional regression error can be bounded by $\bar{e}_i^H$. Part (c) of Lemma 1 reports the resulting bound on $\tilde{e}_i^H$.

Lemma 1. *It holds that*

(a) $\tilde{e}_i^V \leq \sum_{j=i}^{N-1} \delta^{j-i} e_j^*,\ \forall(i)_{-(0)},$

(b) $\tilde{e}_i^C \leq \sum_{j=i}^{N-2} \delta^{j-i}(e_j^{**} + \bar{e}_j^C),\ \forall(i)_{-(N-1)},$

(c) $\tilde{e}_i^H \leq \bar{e}_i^H + \sum_{j=i}^{N-2} \delta^{j-i}(e_j^{**} + \bar{e}_j^C),\ \forall(i)_{-(0)}.$

We use Lemma 1 in the proofs of Propositions 8 and 9 in §3.6.3 and §3.6.4, which establish error bounds related to the dual upper bounds and the greedy lower bounds estimated by LSMV, LSMC, and LSMH.

3.6.3 Dual Upper Bound Estimation

We denote by $u_i^s(x_{i+1}, F_{i+1}^p, F_i^p)$ the dual penalties (3.7) instantiated using the value function V_i^s. Assuming identical formulation and evaluation samples, using these dual penalties

for upper bound estimation results in a tight upper bound estimate on the option value, that is, the estimated dual upper bound equals $V_i^s(x_0, F_0)$ (Theorem 2.3 in Brown et al. 2010). Let $u_i^{\bar{\beta}}(x_{i+1}, F_{i+1}^p, F_i^p)$ and $u_i^{\bar{\gamma}}(x_{i+1}, F_{i+1}^p, F_i^p)$ denote the dual penalties (3.7) instantiated using the VFAs of LSMV and LSMH, respectively. Further, let $u_i^{\bar{\theta}}(x_{i+1}, F_{i+1}^p, F_i^p)$ be the dual penalties (3.10) instantiated using the LSMC CFA. The worst case errors between the dual penalties of LSMV, LSMC, and LSMH, respectively, and the optimal dual penalties are

$$\tilde{e}_i^{V,DP} := \|u_i^{\bar{\beta}}(\cdot) - u_i^s(\cdot)\|_\infty, \tag{3.24}$$

$$\tilde{e}_i^{C,DP} := \|u_i^{\bar{\theta}}(\cdot) - u_i^s(\cdot)\|_\infty, \tag{3.25}$$

$$\tilde{e}_i^{H,DP} := \|u_i^{\bar{\gamma}}(\cdot) - u_i^s(\cdot)\|_\infty, \tag{3.26}$$

where the superscript DP indicates dual penalty. Proposition 8 establishes bounds on these errors. These bounds reflect an error structure analogous to the one bounded in Lemma 1.

Proposition 8. *It holds that*

(a) $\tilde{e}_i^{V,DP} \leq 2\sum_{j=i}^{N-2} \delta^{j-i} e_{j+1}^*, \ \forall (i)_{-(N-1)},$

(b) $\tilde{e}_i^{C,DP} \leq 2\sum_{j=i}^{N-3} \delta^{j-i}(e_{j+1}^{**} + \bar{e}_{j+1}^C), \ \forall i \in \mathcal{I} \setminus \{N-1, N-2\},$

(c) $\tilde{e}_i^{H,DP} \leq 2\left[\bar{e}_{i+1}^H + \sum_{j=i}^{N-3} \delta^{j-i}(e_{j+1}^{**} + \bar{e}_{j+1}^C)\right], \ \forall (i)_{-(N-1)}.$

Under some technical conditions, discussed in Appendix §B.2.1, the bound on $\tilde{e}_i^{V,DP}$ in Part (a) of Proposition 8 is smaller than both the bounds on $\tilde{e}_i^{C,DP}$ and $\tilde{e}_i^{H,DP}$ in parts (b) and (c) of this proposition. The bound on $\tilde{e}_i^{C,DP}$ in Part (b) of Proposition 8 dominates the bound on $\tilde{e}_i^{H,DP}$ in Part (c) of this proposition. We thus conclude that (i) the LSMV-based dual upper bound estimate should likely be better than both the LSMC-based and LSMH-based dual upper bound estimates and (ii) the dual upper bound estimated by LSMC should outperform the one estimated by LSMH.

3.6.4 Greedy Lower Bound Estimation

LSMC uses its CFA for greedy lower bound estimation. We can think of both LSMH and LSMV as estimating greedy lower bounds using the CFAs induced by their respective VFAs, which we respectively define as

$$C_i^{\bar{\beta}}(x_{i+1}, F_i^p) := \delta \mathbb{E}^s \left[(\Phi_{i+1}\bar{\beta}_{i+1})(x_{i+1}, F_{i+1})|F_i^p\right], \tag{3.27}$$

$$C_i^{\bar{\gamma}}(x_{i+1}, F_i^p) := \delta \mathbb{E}^s \left[(\Phi_{i+1}\bar{\gamma}_{i+1})(x_{i+1}, F_{i+1})|F_i^p\right]. \tag{3.28}$$

Obviously, assuming identical formulation and evaluation samples, the greedy lower bound estimated using the continuation function C_i^s is tight, that is, it equals $V_i^s(x_0, F_0)$. To understand the relative greedy lower bounding performance of LSMV, LSMC, and LSMH, we derive and compare bounds on the errors between the, possibly induced, CFAs associated with each of these methods and the continuation function C_i^s. Part (b) of Lemma 1 already provides such an error bound for the LSMC CFA. Proposition 9, based on parts (a) and (c) of Lemma 1, establishes error bounds on the errors incurred by the LSMV and LSMH induced CFAs. We define these respective errors as

$$\tilde{e}_i^{V,IC} := \left\| C_i^{\bar{\beta}}(\cdot) - C_i^s(\cdot) \right\|_\infty, \tag{3.29}$$

$$\tilde{e}_i^{H,IC} := \left\| C_i^{\bar{\gamma}}(\cdot) - C_i^s(\cdot) \right\|_\infty, \tag{3.30}$$

where the superscript IC stands for induced continuation function.

Proposition 9. *It holds that*

(a) $\tilde{e}_i^{V,IC} \leq \delta \sum_{j=i}^{N-2} \delta^{j-i} e_{j+1}^*, \forall (i)_{-(N-1)},$

(b) $\tilde{e}_i^{H,IC} \leq \delta \left[\bar{e}_{i+1}^H + \sum_{j=i}^{N-3} \delta^{j-i} (e_{j+1}^{**} + \bar{e}_{j+1}^C) \right], \forall (i)_{-(N-1)}.$

Under some technical conditions discussed in Appendix §B.2.2, we can show that the bound on $\tilde{e}_i^{V,IC}$ in Part (a) of Proposition 9 is no worse than both the bound on $\tilde{e}_i^C$ in Part (b) of Lemma 1 and the bound on $\tilde{e}_i^{H,IC}$ in Part (b) of Proposition 9. Hence, the greedy lower bounds estimated by LSMV should likely outperform the ones estimated by both LSMC and LSMH.

Intuitively, one would expect that the error bound on $\tilde{e}_i^{H,IC}$ in Part (b) of Proposition 9 should be larger than the error bound on $\tilde{e}_i^C$ in Part (b) of Lemma 1 because the LSMH VFA is estimated using regression based on the LSMC CFA. However, under some technical conditions discussed in Appendix §B.2.2 we find that this intuition is wrong. Hence, we conclude that the greedy lower bound estimated by LSMH should likely outperform the one estimated by LSMC.

Table 3.1: Summary of our predictions on the relative bounding performance of LSMV, LSMC, and LSMH ($\succcurlyeq$ denotes weakly better than).

Dual Upper Bounds	Greedy Lower Bounds
LSMV $\succcurlyeq$ LSMC $\succcurlyeq$ LSMH	LSMV $\succcurlyeq$ LSMH $\succcurlyeq$ LSMC

3.6.5 Summary

Table 3.1 summarizes our predictions on the relative bounding performance of LSMV, LSMC, and LSMH ($\succcurlyeq$ means weakly better than). Because our predictions are based on worst case bounds they need not match the observed numerical performance of these methods. However, they provide a theoretical perspective on the numerical investigation that we conduct in §3.7. In addition, our predictions focus on the quality of the bounds estimated using different LSM methods but ignore their respective computational efforts. Considering this aspect is important for determining the practical usefulness of an LSM method. In §3.7, we numerically investigate both the bounding quality and the computational burden of LSMC, LSMH, and LSMV.

3.7 Computational Results

In this section we benchmark the computational performance of LSMC, LSMH, and LSMV on crude oil swing option and natural gas storage option instances. The term structure in these applications is an energy forward curve. In §3.7.1 we discuss a specific term structure model and its calibration. We describe the crude oil swing option and natural gas storage option instances in §3.7.2. In §3.7.3 we present the basis functions that we use in §3.7.4 to investigate the upper and lower bounding performance of LSMC, LSMH, and LSMV.

3.7.1 Price Model and Calibration

We choose each function $\sigma_{m,k}(\cdot)$ in the term structure model (3.13)-(3.14) to be right continuous and piecewise constant during each interval $[T_i, T_{i+1})$ (Blanco et al., 2002, Secomandi et al., 2012). That is, we set $\sigma_{j,k}(t)$ equal to the constant $\sigma_{j,k,i}$, $\forall t \in [T_i, T_{i+1})$. Under this specification, we can equivalently rewrite (3.13)-(3.14) as

$$F(t', T_j) = F(t, T_j) \exp\left[-\frac{1}{2}(t'-t)\sum_{k=1}^{K}\sigma_{j,k,i}^2 + \sqrt{t'-t}\sum_{k=1}^{K}\sigma_{j,k,i}Z_k\right], \tag{3.31}$$

for all $i \in \mathcal{I}$, $j \in \{i+1, \ldots, N-1\}$, $t \in [T_i, T_{i+1})$ and $t' \in (T_i, T_{i+1}]$ with $t' > t$, and with $Z := (Z_k, k = 1, \ldots, K)$ a vector of K independent standard normal random variables. Notice that the prices in (3.31) are correlated in general because they are driven by common factors. We use (3.31) to generate forward curve sample paths by Monte Carlo simulation.

We use ten years of NYMEX crude oil and natural gas futures prices, observed from 1997 to 2006, to estimate sample variance-covariance matrices of the daily log futures price returns for each month for both commodities. We perform a principal component analysis of all these matrices to estimate the loading coefficients $\sigma_{j,k,i}$ (see Blanco et al. 2002 and Secomandi et al. 2012 §6.1 for more details). We choose the number of factors K equal to 7 and 3 for natural gas and crude oil, respectively, as these are the smallest numbers

of factors explaining more than 99% of the total observed variance in each of our monthly data sets.

3.7.2 Instances

We create four 24-stage *price* instances for both crude oil and natural gas by defining the time zero forward curve, F_0, as the forward curve for these energy sources observed on the first trading date of April, July, October, and January 2006, respectively, because we take these months as representative of Spring, Summer, Fall, and Winter. Following Lai et al. (2010), who use the same convention, we use risk free interest rates equal to 4.74%, 5.05%, 5.01%, and 4.87% for the Spring, Summer, Fall, and Winter price instances, respectively. We refine these price instances with application specific information to create our crude oil swing option and natural gas storage option instances.

We create our swing option instances by adding to each crude oil price instance the number of swing rights n, which we vary between 1 and 10 in increments of 1, and setting the base load capacity q_i equal to 1 and the swing capacity Q_i equal to 0.2 for each of the 24 stages (we do not consider different values for Q_i as this parameter simply scales the reward function and, hence, the value of the swing option). Each strike price K_i is set equal to the price at time 0 of the futures with maturity at time T_i, $F_{0,i}$. We thus obtain forty swing option instances.

Our storage option instances are based on our natural gas price instances and follow Lai et al. (2010) for the specification of their operational parameters. In particular, we add to each such price instance a normalized storage capacity $\bar{x}$ equal to 1, and high, moderate, and low injection and withdrawal capacity pairs as defined in Lai et al. (2010). The initial inventory x_0 is set to 0. This process results in twelve natural gas storage instances.

3.7.3 Basis Functions

We define VFAs as described in §3.5.2 with $\phi_{i,b}^{EN}(x_i)$ defined as look-up tables. This modeling choice corresponds to specifying a different set of basis functions for each value of the endogenous state variable in a given stage. We define the CFA basis function sets analogously. Therefore, the VFA and CFA basis function sets can be represented as Φ_{i,x_i}, $\forall(i,x_i)_{-(0)}$, and $\Psi_{i,x_{i+1}}$, $\forall(i,x_{i+1})_{-(N-1)}$, respectively.

We use three sets of VFA basis functions for our swing option instances. Table 3.2 reports the functions included in sets 1 and 2. Set 1 is standard (Longstaff and Schwartz, 2001, Boogert and De Jong, 2008, Cortazar et al., 2008). Set 2 is based on the observation that the reward function and the *optimal* value function when the number of swing rights equals the number of exercise dates ($n = N$) can be modeled using pairs of call and put option prices. Set 3 is the union of sets 1 and 2. We use three analogous specifications for the sets of CFA basis functions.

For our storage option instances, we also use three VFA and CFA basis function sets. Set 1 is identical to the swing option set 1. Defining $F_{j,j+1}^{W} := \alpha^{W} F_{j,j+1} - \varsigma^{W}$ and $F_{j,j}^{I} :=$

Table 3.2: Basis functions in sets 1 and 2 in stage i and state (x_i, F_i).

Set 1
1
$F_{i,j}, \forall j \in \{i, \ldots, N-1\}$
$F_{i,j}^2, \forall j \in \{i, \ldots, N-1\}$
$F_{i,j}F_{i,m}, \forall m, j \in \{i, \ldots, \min\{i+4, N-1\}\}, m > j$
Set 2
1
$\mathbb{E}[(F_{j,j} - F_{0,j})^+ \vert F_{i,j}], \forall j \in \{i, \ldots, N-1\}$
$\mathbb{E}[(F_{0,j} - F_{j,j})^+ \vert F_{i,j}], \forall j \in \{i, \ldots, N-1\}$

$\alpha^I F_{j,j} + \varsigma^I$, set 2 includes the functions 1, $F_{i,i}$, and $\mathbb{E}[(\delta F^W_{j,j+1} - F^I_{j,j})^+ | F_{i,j}, F_{i,j+1}], \forall j \in \{i, \ldots, N-2\}$. This choice is based on the finding by Secomandi (2014) that the optimal value function of SDP (3.3) applied to storage with unitary injection and withdrawal loss coefficients, zero injection and withdrawal marginal costs, and injection and withdrawal capacities equal to the space ($\alpha^I = \alpha^W = 1$, $\varsigma^I = \varsigma^W = 0$, and $|\underline{a}| = \overline{a} = \overline{x}$) is of this form. Set 3 is the union of sets 1 and 2. We use three analogous CFA basis function sets.

3.7.4 Results

For a given set of basis functions, we estimate greedy lower bounds and dual upper bounds using $W = 100{,}000$ evaluation sample paths as this choice ensures that the standard errors of all the estimates are less than 0.5% of the tightest upper bound estimate. Our sample average approximations of expectations are based on 100 inner sample paths when estimating bounds using the CFA/VFA of LSMC and LSMH (see §3.3 for details). As Gyurko et al. (2011), when using LSMH we estimate greedy lower bounds based on the VFA computed by this method.

When LSMV, LSMC, and LSMH are combined with the three definitions of basis function sets presented in §3.7.3, we obtain nine versions of these methods, labeled as LSMV1, LSMV2, and LSMV3; LSMC1, LSMC2, and LSMC3; and LSMH1, LSMH2, and LSMH3. Irrespective of the LSM approach, sets 1 and 3 result in the best converged (dual upper and greedy lower) bound estimates on the storage option and swing option instances, respectively (convergence means that the bound estimates do not change as the number of regression samples, P, used to compute VFAs and CFAs increases). The converged estimates for sets 1 and 3 are essentially the same on the storage option instances, but using set 3 is computationally more expensive. Thus, we report the LSMV3, LSMC3, and LSMH3 results on the swing option instances and the LSMV1, LSMC1, and LSMH1 results on the storage option instances.

Based on our error bound analysis conducted in §3.6, for fixed values of the numbers of regression and evaluation samples (P and W) we expect that (i) LSMV should outperform LSMC and LSMH in both dual upper and greedy lower bounding performance; (ii) the

LSMC dual upper bound estimates should be better than the ones of LSMH; and (iii) the LSMH greedy lower bound estimates should be of higher quality than the ones of LSMC. We now numerically assess the quality differences between the bounds estimated by LSMV, LSMC, and LSMH and compare our findings with these predictions. In addition, we assess the changes in the quality of the bounds estimated by these LSM methods as functions of the number of regression samples, P, keeping the number of evaluation samples, W, fixed.

Figures 3.1 and 3.2 display the dual upper bounds and greedy lower bounds estimated by LSMV3, LSMC3, and LSMH3 on the swing option instances with three exercise rights ($n = 3$) as percentages of the LSMV3 dual upper bound estimates. The relative performance of these methods on the instances with more exercise rights are similar and are not reported here for brevity. Figures 3.3 and 3.4 display the dual upper bounds and greedy lower bounds estimated by LSMV1, LSMC1, and LSMH1 on the January and April storage option instances as percentages of the LSMV1 dual upper bound estimates. The results for the July and October instances are similar and, we do not report them here for conciseness.

These numerical findings are for the most part consistent with our theoretical predictions, except for the relative upper bounding performance of LSMC and LSMH on some instances. (In Appendix B.2.3, we verify numerically for the case $P = 1{,}000$ that the conditions that form the premises of these predictions appear to be verified.) These discrepancies may be due to the nature of our analysis, which is based on comparing bounds on worst case errors. Moreover, the differences between the bounds estimated by the considered LSMV and LSMH versions are small, being at most 2% for the dual upper bounds and 3.5% for the greedy lower bounds. The analogous differences between the bounds estimated by LSMV and LSMC are more pronounced, being at most 2.5% for the dual upper bounds and 10.5% for the greedy lower bounds. For sufficiently large values of P, the accuracy and precision of the bounds estimated by LSMV, LSMC, and LSMH are comparable and these bounds are near optimal.

More interesting is the insensitivity of the accuracy and precision of the LSMV estimated upper and lower bounds to changes in the number of regression samples, that is, these bounds converge when P is at most 1,000. This is not the case for the other methods. This observed behavior suggests that the sample path approximation errors $\bar{e}_i^C$ that are present in the error bounds for LSMC and LSMH are the dominant errors that determine the quality of the bounds that these methods estimate (see (3.22) and Propositions 8 and 9), because, intuitively, we expect that these errors would decrease as more regression samples are used. In contrast, such an error is not present in the LSMV error bounds.

Figure 3.5 displays the CPU times required to estimate a VFA/CFA for different values of the number of regression samples using each LSM method on the swing option instances with one and ten exercise rights as they are the instances that require the least and most computational effort, respectively. For analogous reasons, Figure 3.6 displays the computational effort to estimate a VFA/CFA when varying the number of regression samples using each LSM method on the storage option instances with high and low capacities. These times are essentially the same for LSMV and LSMC while they are slightly larger for LSMH. In particular, LSMV requires less than one CPU second to estimate a VFA that delivers accurate and precise bound estimates, that is, it requires no more than 1,000

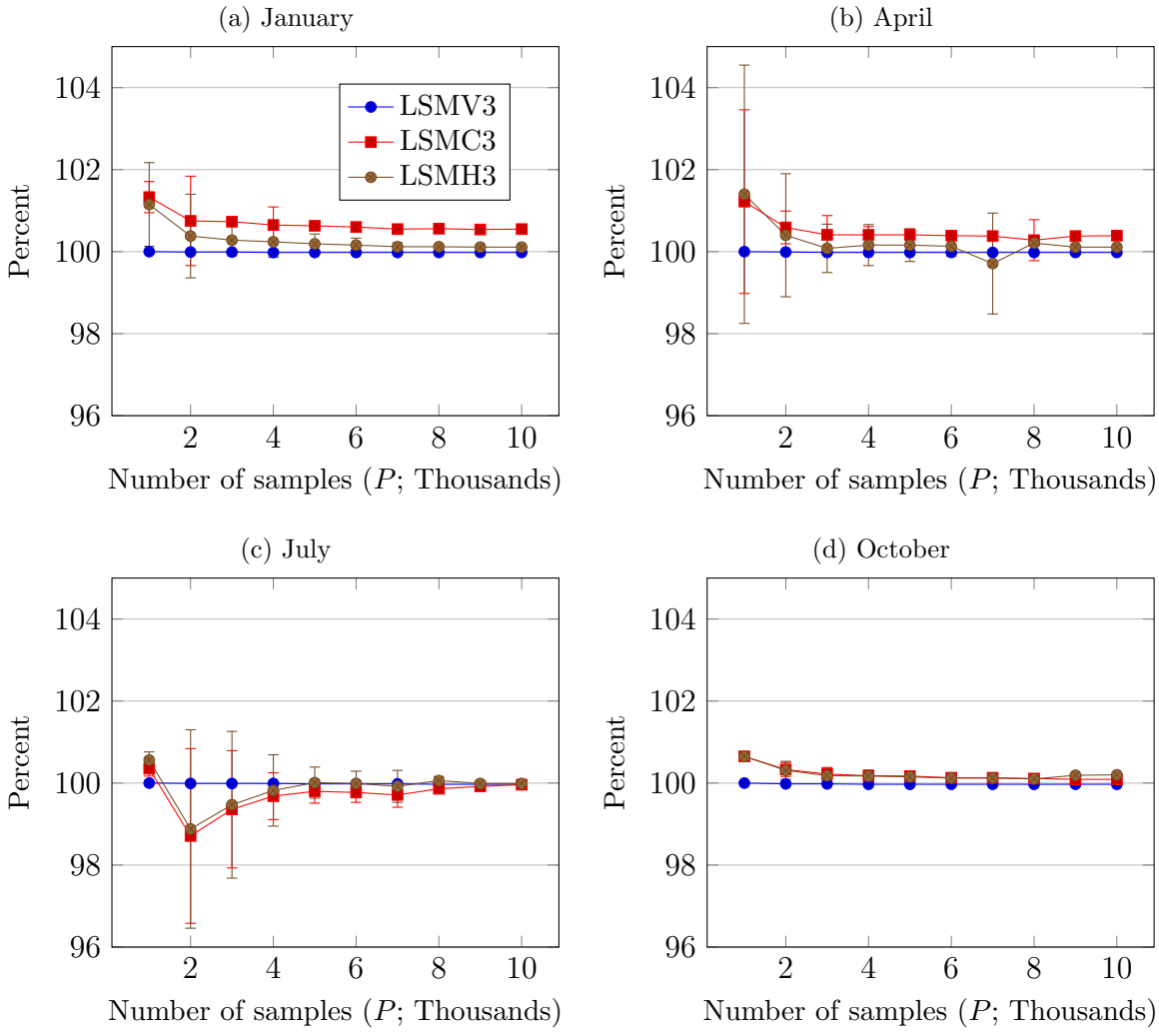


Figure 3.1: Convergence of the dual upper bounds estimated by LSMV3, LSMC3, and LSMH3 as percentages of the LSMV3 dual upper bound estimates on the swing option instances with three exercise rights ($n = 3$) and one hundred thousand evaluation samples ($W = 100{,}000$).

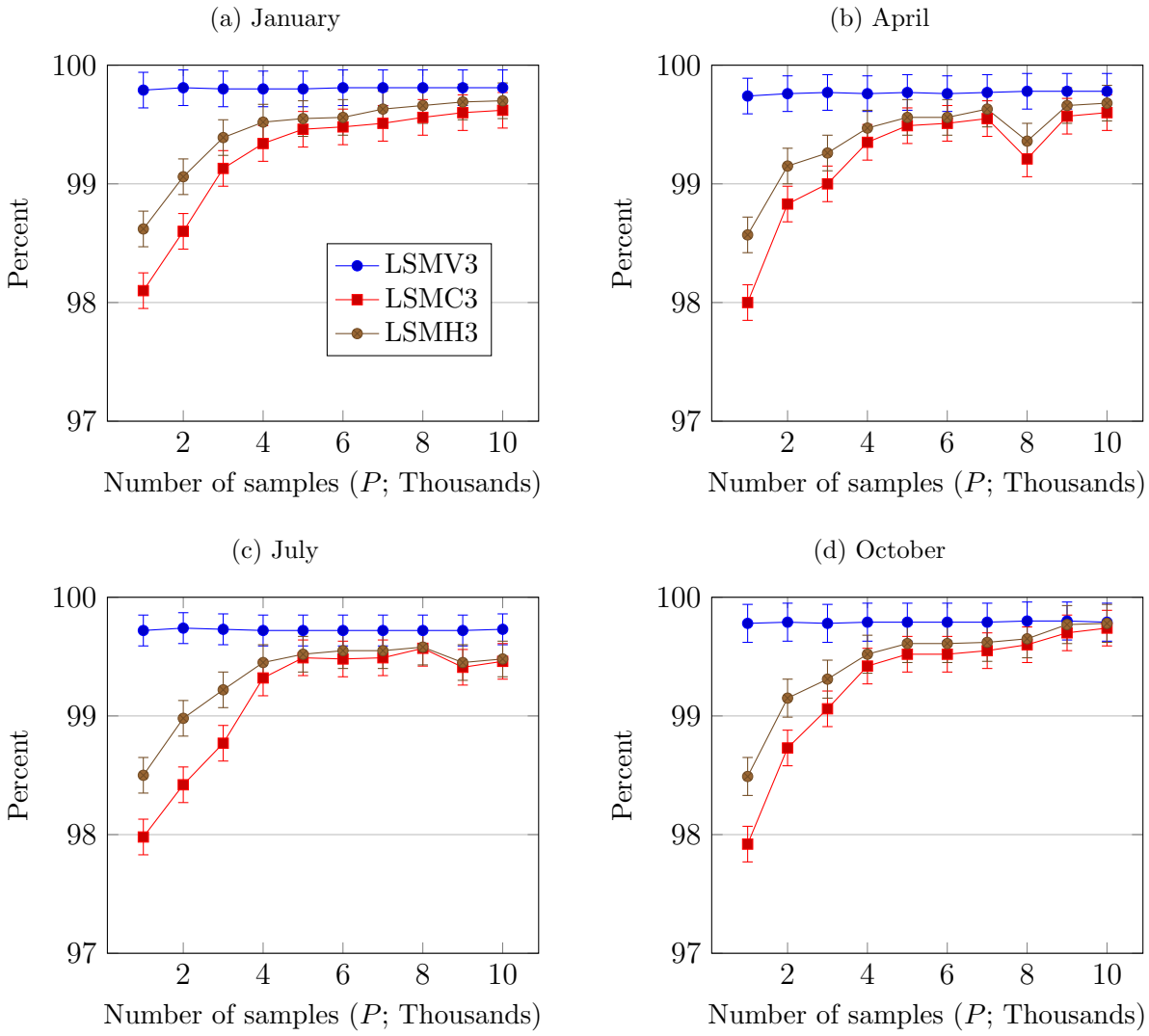


Figure 3.2: Convergence of the greedy lower bounds estimated by LSMV3, LSMC3, and LSMH3 as percentages of the LSMV3 dual upper bound estimates on the swing option instances with three exercise rights ($n = 3$) and one hundred thousand evaluation samples ($W = 100{,}000$).

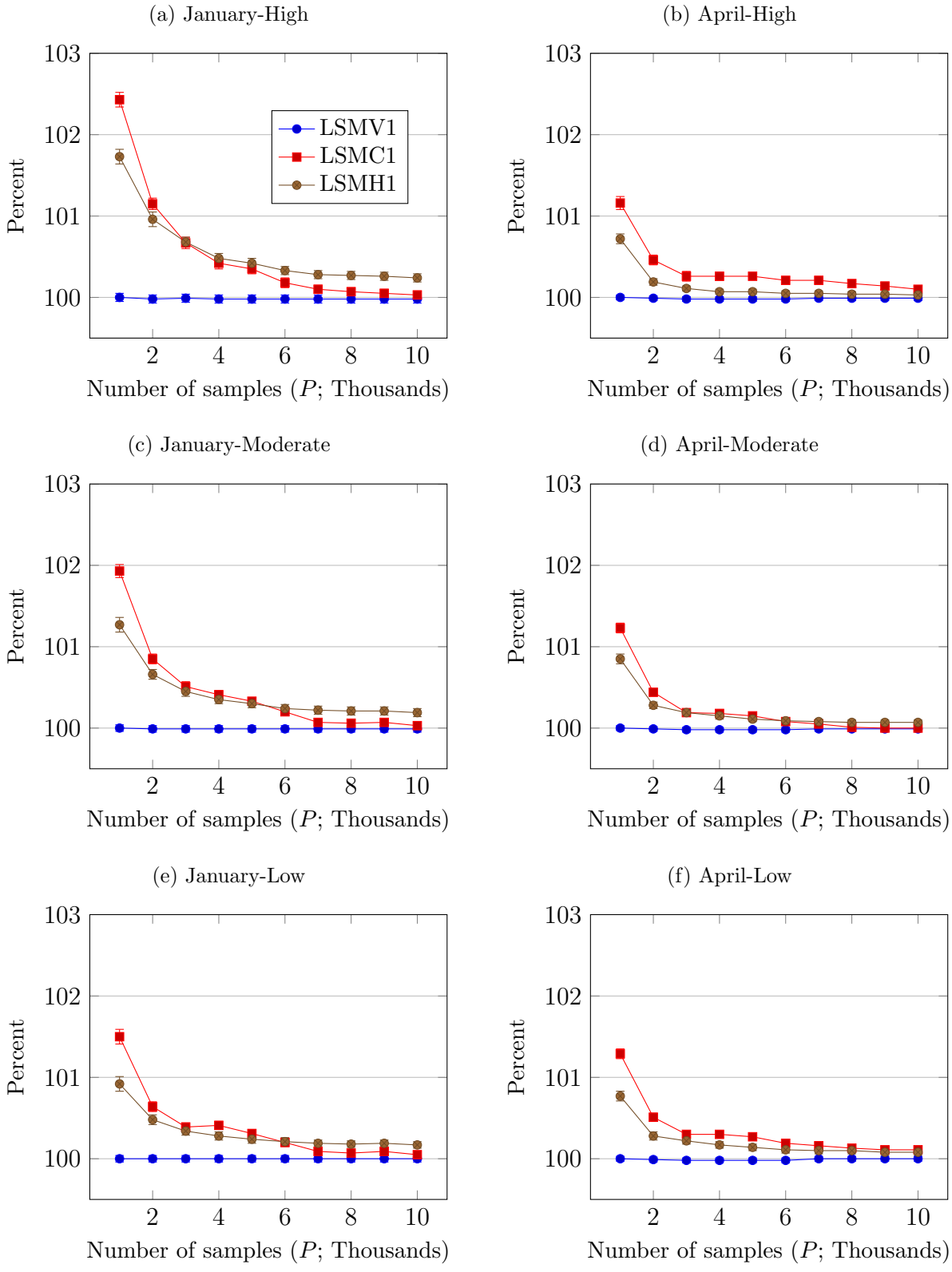


Figure 3.3: Convergence of the dual upper bounds estimated by LSMV1, LSMC1, and LSMH1 as percentages of the LSMV1 dual upper bound estimates on the January and April storage option instances with one hundred thousand evaluation samples (W = 100,000).

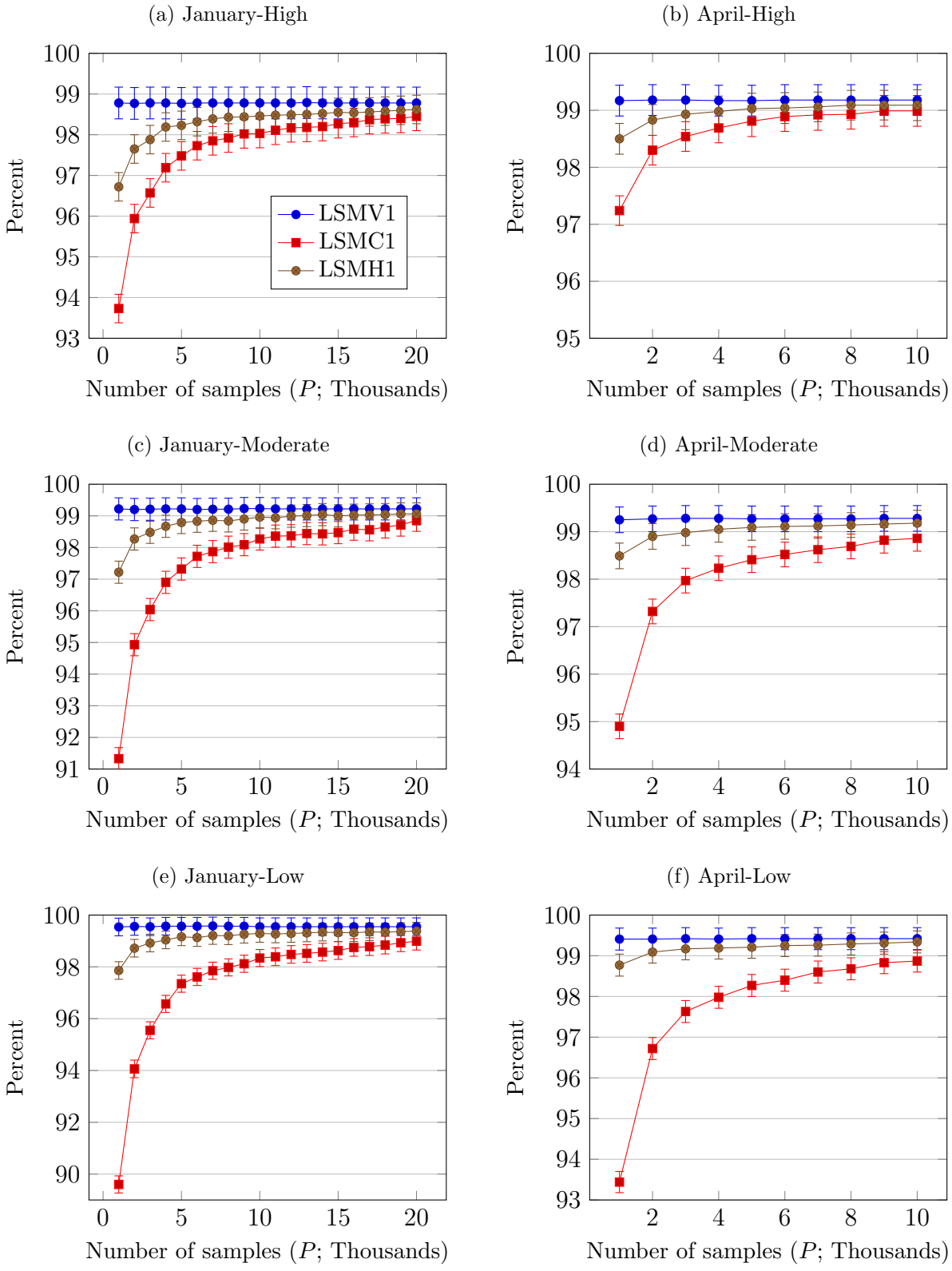


Figure 3.4: Convergence of the greedy lower bounds estimated by LSMV1, LSMC1, and LSMH1 as percentages of the LSMV1 dual upper bound estimates on the January and April storage option instances with one hundred thousand evaluation samples ($W = 100{,}000$).

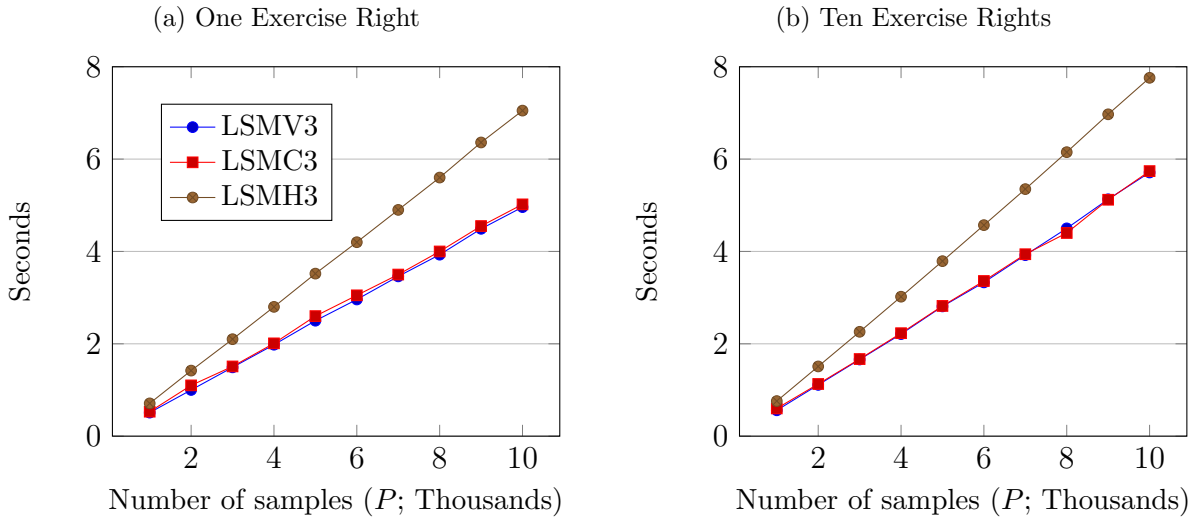


Figure 3.5: Average CPU seconds required for computing a VFA/CFA on the swing option instances with one ($n = 1$) and ten ($n = 10$) exercise rights.

regression samples, while the remaining methods may require up to 20 additional seconds to achieve the same bounding performance.

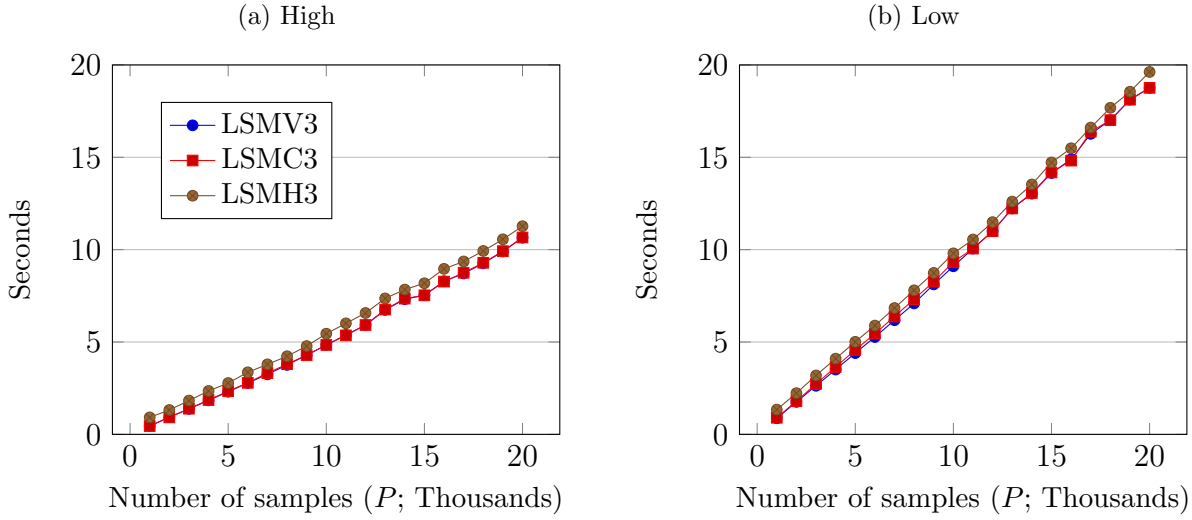


Figure 3.6: Average CPU seconds required for computing a VFA/CFA on the storage option instances with high and low capacity.

Table 3.3 reports the average CPU time incurred by the three considered LSM methods to estimate greedy lower bounds and dual upper bounds on our swing option and storage option instances (this table excludes the CFA/VFA estimation times). Our implementation of LSMH computes once the same inner sample path averages that need to be evaluated when estimating lower and upper bounds. We attribute this CPU time to the upper bound estimation. We find that the computational effort exerted by LSMV, LSMC, and LSMH to

estimate greedy lower bounds is low and essentially equal. In particular, the resulting CPU times vary between 2-5 and 13-17 seconds on the swing option and storage option instances, respectively. The differences in the CPU times taken by these methods to estimate upper bounds are instead substantial. LSMV estimates these bounds in a much faster fashion than both LSMC and LSMH: The CPU time required by LSMV for dual upper bound estimation is roughly 20-100 times smaller than the one of LSMH and 120-3200 times smaller than the one of LSMC. Thus, the absence of sample average approximations that distinguishes LSMV from both LSMC and LSMH makes LSMV between 1 and 3 orders of magnitude faster than the two competing methods for dual upper bound estimation.

Table 3.3: Average CPU seconds needed for estimating lower and upper bounds on a subset of the swing option instances and on the storage option instances using one hundred thousand evaluation samples (W = 100,000).

Swing Option						
	Lower Bound			Upper Bound		
n	LSMV	LSMC	LSMH	LSMV	LSMC	LSMH
1	15.07	13.49	14.01	14.06	1,699.40	1,568.07
10	16.40	14.50	15.12	16.82	2,424.97	1,701.56
Storage Option						
	Lower Bound			Upper Bound		
Capacity	LSMV	LSMC	LSMH	LSMV	LSMC	LSMH
High	3.53	2.29	2.50	13.62	20,149.28	601.55
Moderate	4.17	3.31	3.46	22.35	56,550.55	646.03
Low	4.65	4.14	4.32	27.63	87,828.51	659.36

In summary, our numerical results suggest that (i) the three LSM methods are all equivalent in terms of the quality of the estimated lower and upper bounds provided that a sufficient number of regression samples are used to estimate the LSMC CFA and the LSMH VFA, and (ii) the resulting computational savings that LSMV obtains relative to both LSMC and LSMH during CFA/VFA estimation are overshadowed by the analogous savings that arise when estimating dual upper bounds, while there are no substantial differences in the computational requirements of these three methods when estimating greedy lower bounds.

3.8 Conclusions

We develop an LSM method for valuing multiple exercise options in conjunction with term structure models, that are widespread among practitioners and academics. We benchmark our LSM method against the standard LSM method and a state-of-the-art variant thereof on realistic energy swing and storage option instances. We find that all these LSM methods estimate near optimal, accurate, and precise greedy lower and dual upper bounds. However, the existing approaches require a significantly larger number of regression samples than our LSM approach to achieve such bounding performance. The computational savings

that result from this improvement are dominated by the analogous savings obtained by our method when estimating dual upper bounds. In particular, our LSM approach is between one and three orders of magnitude faster than the existing LSM approaches when estimating dual upper bounds. The computational effort of all the considered LSM methods are comparable for greedy lower bound estimation. Thus, we provide numerical support for the use of our LSM method for valuing multiple exercise options in conjunction with term structure models. We also conduct a worst case error bounding analysis that provides a theoretical perspective on the relative quality of the bounds estimated by these methods on our instances.

Chapter 4

Joint Merchant Management of Natural Gas Storage and Transport Assets

(Joint work with Nicola Secomandi)

4.1 Introduction

Natural gas is an important commodity. It served more than one quarter of the 2012 energy consumption in the United States (EIA, 2013). The availability and importance of natural gas is growing with the shale boom (Smith, 2013). It is projected that natural gas consumption in North America will increase by 18% between 2008 and 2030 and be accompanied by a need for 130-210 billion US dollars worth of midstream natural gas infrastructure (INGAA, 2009). Eighty percent of this projected infrastructure cost is for building new natural gas pipeline systems (INGAA, 2009).

Pipeline systems give merchants the ability to trade natural gas across time and geographical markets. That is, these systems embed two types of assets that merchants manage as real options: storage and transport. Storage assets allow merchants to trade natural gas over time by buying natural gas and injecting it into a storage facility and withdrawing previously injected natural gas from the storage facility and selling it. Transport assets allow merchants to trade natural gas across different geographical locations by contemporaneously transporting natural gas along pipelines connecting multiple geographical markets. Merchants acquire storage and transport assets by purchasing from pipeline companies contracts on their capacity.

The extant literature has studied the merchant management of natural gas storage and transport assets in isolation. Charnes et al. (1966) study a fast commodity storage asset that can be completely filled up or emptied in a single period. That is, the asset has no constraining injection or withdrawal capacities. Secomandi (2010) studies a slow commodity storage asset that requires multiple periods to be filled up or emptied. These authors

show that the optimal storage policy has a basestock (target) structure. Irrespective of the speed of the asset, computing an optimal storage policy is intractable when using a model of the evolution of the natural gas price with more than a few stochastic factors. When the evolution of this price is modeled using multi-factor price models, several authors focus on computing near optimal heuristic storage operating policies and lower bounds on the storage asset value (Lai et al. 2010, Boogert and De Jong 2011/12, Boogert and Mazières 2011, Thompson 2012, Wu et al. 2012, Chapters 2 and 3). Lai et al. (2010), Secomandi (2012), and Chapters 2 and 3 also compute dual upper bounds on this value by applying the information relaxation and duality framework (Brown et al., 2010, and references therein). All these authors assume that storage is operated in a single market. The valuation and merchant management of natural gas transport assets is studied by Secomandi (2010) and Secomandi and Wang (2012). Secomandi (2010) provides empirical evidence for the use of the real options approach to value the point-to-point version of these assets. Secomandi and Wang (2012) propose a linear programming and Monte Carlo simulation based method to value and manage such assets when they have a network structure. These authors do not consider storage.

In contrast to the extant literature, we consider the joint merchant management of natural gas storage and transport assets. We formulate a finite horizon discrete time Markov Decision Problem (MDP), the states of which include, in every stage, the storage inventory level and the forward curves of a set of geographically interconnected markets where natural gas can be traded – a forward curve is a vector of futures prices. At each stage and state, the MDP multidimensional action is a vector of storage and transport decisions. Our MDP thus substantially generalizes the single market natural gas storage MDP so far considered in the literature, in which the states include a single forward curve and the action is a scalar (Secomandi and Seppi 2014, ch. 5 and ch. 6, and references therein). Our MDP also models more general transport assets than the model of Secomandi and Wang (2012).

Computing an optimal policy of our MDP is intractable. We thus leverage our structural analysis of this model and obtain a heuristic policy by extending a least squares Monte Carlo (LSM) method (Longstaff and Schwartz 2001, Tsitsiklis and Van Roy 2001, Rogers 2002, Glasserman 2004, ch. 8, Chapter 3). When applied to realistic instances developed in conjunction with an energy trading company, our heuristic policy is near optimal compared to various dual upper bounds that we estimate, a finding consistent with the results of Chapter 3.

Using our heuristic policy and realistic instances we investigate various managerial aspects of our business problem. We find that (i) the joint, rather than decoupled, merchant management of storage and transport assets has substantial value; (ii) this management can be nearly optimally simplified by prioritizing storage relative to transport, despite the substitutability between these activities, a property that we formally establish, being considerable; (iii) the value due to price uncertainty is appreciable, can be almost entirely captured by sequentially reoptimizing the deterministic version of our MDP, an approach included in existing commercial software (FEA, 2013) – a finding that extends what is known for the single market storage asset (Lai et al., 2010, Secomandi, 2010) – and a limited look-ahead, and hence faster to compute, version of this heuristic policy is also near

optimal; and (iv) the value of transport trading across multiple pipelines is substantial.

Beyond natural gas storage and transport assets, our research has potential relevance for the merchant management of assets used to store and transport commodities such as crude oil and refined products, metals, agricultural products, and even electricity (Markland, 1975, Markland and Newett, 1976, Smith and McCardle, 1998, 1999, Deng et al., 2001, Kleindorfer and Wu, 2003, Rømo et al., 2009, Boyabatli, 2011, Boyabatli et al., 2011, Devalkar et al., 2011, Kim and Powell, 2011, Lai et al., 2011, Arvesen et al., 2013, Zhou et al., 2013a,b).

We proceed by presenting in §4.2 some background material on the business problem that we study. In §4.3 we formulate our MDP and reformulate it as a stochastic dynamic program (SDP). In §4.4 we analyze the value function and an optimal storage policy of this SDP, also establishing the substitutability between the storage and transport assets. In §4.5 we discuss our LSM based policy and how to use it to estimate a lower bound on the combined value of the storage and transport assets. In §4.6 we consider the estimation of dual upper bounds on this value. In §4.7 we conduct our numerical analysis. We conclude in §4.8. Proofs are in Appendix C.

4.2 Background Material

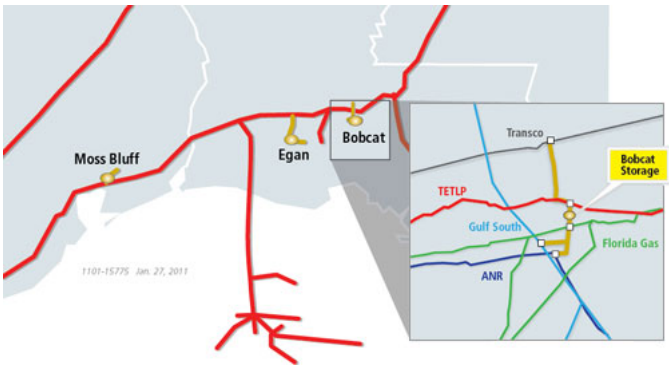


Figure 4.1: The Bobcat storage facility and connecting pipelines (Source: Spectra Energy website).

Pipeline systems comprise of storage facilities, compressor stations, metering stations, and interconnect stations that link different pipelines (Pipeline Knowledge & Development, 2010). Figure 4.1 illustrates the connections of the Bobcat storage facility, located in Louisiana, to five major pipeline systems: Texas Eastern Transmission Company (TETCO; also referred to as TETLP), Transcontinental Gas Pipeline Company (TRANSCO), Gulf South Pipeline Company, Florida Gas Transmission Company, and ANR Pipeline Company. Natural gas can be shipped across different pipelines through *interconnect stations*. Figure 4.1 shows that the Bobcat storage facility is an off-pipeline interconnect station. In contrast, Figure 4.2 illustrates that TETCO and the Algonquin Gas Transmission pipeline

(AGT) are directly connected at on-pipeline interconnect stations on the AGT pipeline. Transferring natural gas between pipelines is referred to as *wheeling* (EIA, 1996, ch. 3).

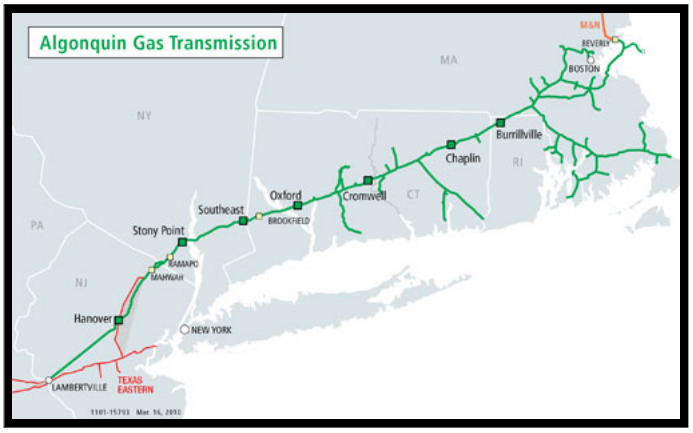


Figure 4.2: Interconnect stations between the TETCO and AGT pipeline systems (Source: Spectra Energy website).

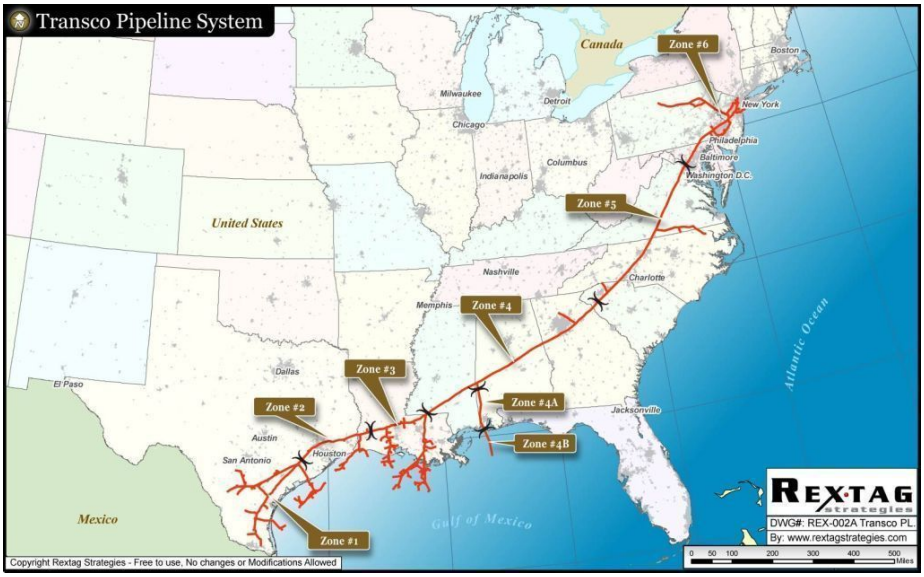


Figure 4.3: The TRANSCO pipeline system.

Merchant trading of natural gas occurs on commercial networks that are simplified representations of the physical pipeline systems where several pipeline segments, storage facilities, and compressor and metering stations are aggregated into zones. Figures 4.3 and 4.4 display the zones of TRANSCO and TETCO. Figure 4.5 shows the AGT pipeline, which is smaller than both TRANSCO and TETCO and is treated as a single zone for merchant trading purposes. Natural gas is traded on more than one hundred physical markets in North America. Derivative contracts on this commodity are traded on organized

exchanges. Prominent examples are the New York Mercantile Exchange (NYMEX) natural gas futures contracts with delivery at Henry Hub, Louisiana, and basis swaps at about forty geographical locations in North America – the price of a basis swap for a given maturity represents an offset with respect to the NYMEX natural gas futures price for the same maturity, and hence the futures price for such a location and maturity is the sum of its basis swap price and the NYMEX futures price for this maturity. From a merchant trading perspective, the zones of major pipelines in North America are associated with the NYMEX futures and basis swaps.

Figure 4.4: The TETCO pipeline system.

The trading activity of natural gas merchants on these commercial networks is based on acquiring contracts on the storage and transport capacity of pipelines. A storage contract specifies a collection of time periods during which storage can be used; the storage space accessible at a storage facility; injection and withdrawal capacities for each time period; and variable and fuel costs. A transport contract specifies a collection of time periods during which transport can be performed; a set of points where natural gas can be received into the pipeline (receipt points) or delivered from the pipeline (delivery points); capacity limits at each of these points; and variable and fuel costs to ship natural gas from receipt to delivery points. Commercially, the transport of natural gas is contemporaneous because natural gas is shipped by displacement using compressor stations that maintain pressure differentials between pipeline segments. We refer to these contracts as storage and transport *assets*. Merchants manage these assets as real options on natural gas prices that give them the ability to change the temporal or geographical availability of natural gas (Maragos, 2002, Lai et al., 2010, Secomandi, 2010, Secomandi and Wang, 2012).

The merchant value of a storage asset originates from trading natural gas between time periods where price differences exceed the cost of storage. Such price differences can exist in a competitive equilibrium because of the volatility in production and demand and

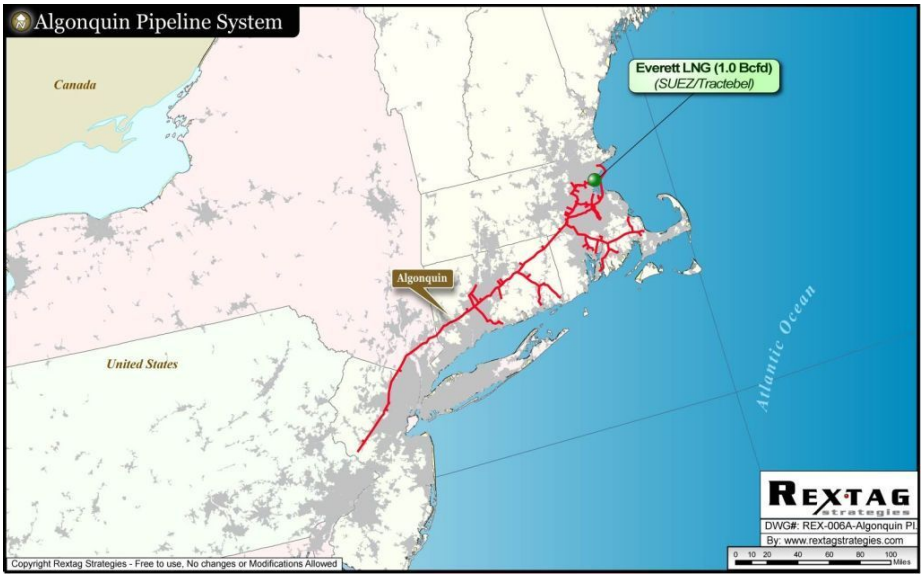


Figure 4.5: The AGT pipeline system.

the high costs associated with changing production to satisfy demand (Pindyck, 2004). A competitive equilibrium perspective on natural prices is reasonable because there is empirical support that natural gas markets have become increasingly competitive since deregulation in the early 1990s (De Vany and Walls, 1993, Cuddington and Wang, 2006). The number of merchants has also considerably increased (Dahl and Matson, 1998).

Analogously, the merchant value of a transport asset originates from trades between geographical locations where the price differences exceed the cost of transportation. Although transport occurs contemporaneously, such price differences can occur, as explained next. Our discussion is based on §5 of Secomandi (2010) (see also Cuddington and Wang 2006 and Marmer et al. 2007). For simplicity consider two markets m_1 and m_2. Without transport, the equilibrium natural gas prices at m_1 and m_2 are determined by local production and consumption at each location. Now suppose for simplicity that uncapacitated transport is possible between m_1 and m_2 at a constant transportation cost. In this case, at equilibrium, the natural gas price at m_1 (m_2) will be at most the sum of the natural gas price at m_2 (m_1) and the transportation cost. Otherwise, there will be an arbitrage opportunity. However, in practice, transport assets have finite capacity, which can be tight (Marmer et al., 2007, Friedman and Philbin, 2014). When this happens, the price difference between markets m_2 and m_1 can be larger than the transport cost by the congestion value of transport capacity. In this case, even though the price difference is larger than the transport cost, arbitrage is not possible because of the tight capacity constraint. Therefore, pricing a transport asset can be viewed as computing the congestion value of its capacity.

4.3 Model

We consider a natural gas commercial network that consists of one storage facility and several pipeline zones, and respective geographical markets, some of which can be interconnect stations. We represent the storage facility and these markets as nodes on a network. The geographical markets are represented by the set $\mathcal{M}$. Although storage may be co-located with one of these geographical markets, we represent it as a separate node to be able to distinguish between storage trades and transport trades, defined below. Labeling storage by ST, the node set of the commercial network is $\mathcal{M} \cup \{\text{ST}\}$. Figure 4.6 illustrates this representation using a realistic commercial network that includes (from left to right) three markets corresponding to TRANSCO zones 3, 4, and 6 (see Figure 4.3), which we label Z3, Z4, and Z6; the Bobcat storage facility located at the interconnect station (IC) between TRANSCO and TETCO (see Figure 4.1); four markets on TETCO, corresponding to its zones 1 through 3 and East Louisiana (see Figure 4.4), which we label M1, M2, and M3, and ELA; and the AGT zone (see Figure 4.5). An edge linking two nodes in this network indicates the possibility of transporting natural gas between these nodes (in both directions).

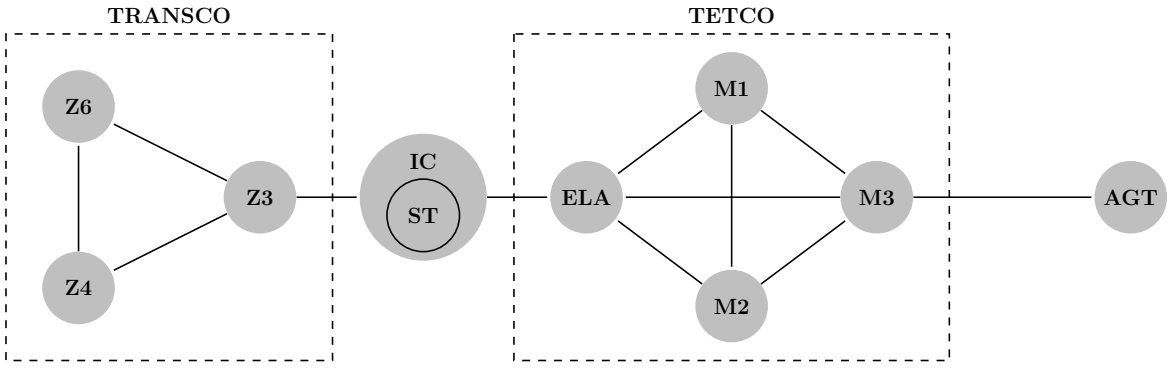


Figure 4.6: Commercial network based on the Bobcat storage facility, portions of TRANSCO and TETCO, and AGT.

A merchant trade specifies the type of activity (storage or transport), the quantity of natural gas stored or shipped, the date of service, and a unique path (sequence of nodes) in the commercial network. We denote the set of merchant trades by $\mathcal{J}$. In particular, $\mathcal{J}^I$ and $\mathcal{J}^W$ are the subsets of $\mathcal{J}$ that include the injection and withdrawal storage trades, respectively. We denote by p_j the path of trade j, and by $p_j(n)$ the n-th node in this path. The number of nodes in the path p_j is denoted by $|p_j|$. The set of trades can be partitioned into transport trades within a pipeline system, wheeling trades across pipeline systems, storage injection trades, and storage withdrawal trades. In Figure 4.6, the paths Z3-Z6 and M2-M3 are transport trades and the paths Z3-IC-ELA and M3-AGT are wheeling trades. The path of an injection trade starts at ST. The path of a withdrawal trade ends at ST. In Figure 4.6, paths Z3-IC-ST and AGT-M3-ELA-IC-ST belong to storage injection trades and paths ST-IC-Z3-Z6 and ST-IC-ELA to storage withdrawal trades.

We formulate an MDP to optimize the trading of natural gas on a given commercial

network. Trades can be performed at each of N times. Denote the i-th such time as T_i with i belonging to the set $\mathcal{I} := \{0, 1, \ldots, N-1\}$. We use the set $\mathcal{I}$ as the stage set of our MDP. Let $\bar{y}$ denote the maximum inventory allowed in storage. The inventory in storage at stage i is $y_i \in \mathcal{Y} := [0, \bar{y}]$. At time T_i the market m futures price with maturity at time $T_{i'} > T_i$ is $F^m_{i,i'} \in \mathbb{R}_+$ and the forward curve of this market is $\boldsymbol{F}^m_i := (F^m_{i,i}, F^m_{i,i+1}, \ldots, F^m_{i,N-1})$. We denote the time T_i spot price at market m by $s^m_i \equiv F^m_{ii}$. For notational convenience, we define the array of forward curves and the vector of spot prices across all markets at time T_i as $\boldsymbol{F}_i := (\boldsymbol{F}^m_i, \forall m \in \mathcal{M})$ and $\boldsymbol{s}_i := (s^m_i, \forall m \in \mathcal{M})$, respectively. We also define $\boldsymbol{F}_N := 0$. The stage i state is the pair $(y_i, \boldsymbol{F}_i)$.

The cash flows generated from performing a trade on a given date include the cost of purchasing or revenue from selling natural gas at a given market at its prevailing spot price and two types of variables costs: commodity charges and compressor fuel costs.

The commodity charge is a fee that the merchant pays to the pipeline company on each unit of transported, injected, or withdrawn natural gas. The commodity charge for transporting natural gas between node m and node m' is denoted by $c^{m,m'}$. The storage injection and withdrawal commodity charges are denoted by c^I and c^W, respectively (I and W abbreviate injection and withdrawal, respectively).

Compressor stations create pressure differentials between pipeline segments, enabling the transport of natural gas. Storage injections and withdrawals are also based on pressure differentials obtained by the use of pumps, which, however, sometimes are not needed for withdrawal. Merchants pay the pipeline company in kind for the fuel used for compression. The fuel consumed to transport 1 unit of natural gas from node m to node m' is $(1 - \phi^{m,m'})/\phi^{m,m'}$, where $\phi^{m,m'} \in (0, 1)$: $1/\phi^{m,m'} \equiv 1 + (1 - \phi^{m,m'})/\phi^{m,m'}$ units of natural gas needs to be received at node m in order to deliver 1 unit of this commodity at node m'. We assume that this fuel is purchased at the market corresponding to node m. The fuel consumed to inject or withdraw 1 unit of natural gas into or out of storage is $(1-\phi^I)/\phi^I$ and $(1-\phi^W)/\phi^W$, respectively, where $\phi^I \in (0, 1)$ and $\phi^W \in (0, 1)$ have interpretations analogous to the transport fuel loss. We assume that the fuel used for injection or withdrawal is monetized at the spot price of the market closest to storage.

Denote by x_j the amount of natural gas transacted under trade $j \in \mathcal{J}$. For notational convenience, we define $\boldsymbol{x} := (x_j, j \in \mathcal{J})$. The reward $r(\boldsymbol{x}, \boldsymbol{s})$ from executing the vector of trade amounts $\boldsymbol{x}$ given the vector of spot prices $\boldsymbol{s}$ is defined as

$$r(\boldsymbol{x}, \boldsymbol{s}) := \sum_{j \in \mathcal{J}} \sum_{n=1}^{|p_j|} \left[\alpha^j_n(\boldsymbol{s}) + \gamma^j_n\right] x_j, \tag{4.1}$$

where

$$\alpha_n^j(\boldsymbol{s}) := \begin{cases} -s^{p_j(2)}\dfrac{(1-\phi^W)}{\phi^W}\mathbb{1}(j \in \mathcal{J}^W) - \dfrac{s^{p_j(1)}}{\phi^{p_j(1),p_j(2)}}\mathbb{1}(j \in \mathcal{J} \setminus \mathcal{J}^W), & \text{if } n = 1, \\ s^{p_j(l)}\dfrac{(1-\phi^{p_j(n),p_j(n+1)})}{\phi^{p_j(n),p_j(n+1)}}, & \text{if } 1 < n < |p_j|, \\ -s^{p_j(|p_j|-1)}\dfrac{(1-\phi^I)}{\phi^I}\mathbb{1}(j \in \mathcal{J}^I) + s^{p_j(|p_j|)}\mathbb{1}(j \in \mathcal{J} \setminus \mathcal{J}^I), & \text{if } n = |p_j|; \end{cases}$$

$$\gamma_n^j := \begin{cases} -c^W\mathbb{1}(j \in \mathcal{J}^W) - c^{p_j(1),p_j(2)}\mathbb{1}(j \in \mathcal{J} \setminus \mathcal{J}^W), & \text{if } n = 1, \\ -c^{p_j(n),p_j(n+1)}, & \text{if } 1 < n < |p_j|, \\ -c^I\mathbb{1}(j \in \mathcal{J}^I), & \text{if } n = |p_j|. \end{cases}$$

When executing trade j, the term $\alpha_n^j(\boldsymbol{s})$ in (4.1) includes the cost and revenue from buying and selling natural gas, respectively, and the value of the corresponding fuel loss incurred: When $n = 1$ this term equals the value of the withdrawal fuel loss if j is a withdrawal trade, and the sum of the purchase costs of one unit of natural gas and of the transport fuel on the edge $(p_j(1), p_j(2))$ otherwise; when $1 < n < |p_j|$ this term equals the value of the transport fuel loss on the edge $(p_j(n), p_j(n+1))$; and when $n = |p_j|$ this term equals the value of the injection fuel loss if j is an injection trade, and the revenue from selling one unit of natural gas otherwise. The term γ_n^j in (4.1) represents the commodity charge incurred when executing trade j: When $n = 1$ this term equals the withdrawal commodity charge if j is a withdrawal trade, and the transport commodity charge on the edge $(p_j(1), p_j(2))$ otherwise; when $1 < n < |p_j|$ this term equals the transport commodity charge on the edge $(p_j(n), p_j(n+1))$; and when $n = |p_j|$ this term equals the injection commodity charge if j is an injection trade.

We now define the set of feasible traded quantities. The receipt and delivery capacities of node m are denoted as $C^{R,m}$ and $C^{D,m}$ (R is for receipt and D is for delivery), respectively. The storage injection and withdrawal capacities are C^I and C^W, respectively. Let $\mathcal{J}^R(m)$ and $\mathcal{J}^D(m)$, respectively, be the set of trades for which node m is a receipt point and a delivery point in their respective paths. We denote by $\vee$ a logical disjunction (or). A vector of trade amounts $\boldsymbol{x}$ is feasible at inventory level $y \in \mathcal{Y}$ if it satisfies the following conditions:

$$\sum_{j \in \mathcal{J}^R(m)} x_j \le C^{R,m}, \forall m \in \mathcal{M}, \tag{4.2}$$

$$\sum_{j \in \mathcal{J}^D(m)} x_j \le C^{D,m}, \forall m \in \mathcal{M}, \tag{4.3}$$

$$\begin{pmatrix} \sum_{j \in \mathcal{J}^I} x_j \le \min\{C^I, \bar{y} - y\} \\ \sum_{j \in \mathcal{J}^W} x_j = 0 \end{pmatrix} \vee \begin{pmatrix} \sum_{j \in \mathcal{J}^I} x_j = 0 \\ \sum_{j \in \mathcal{J}^W} x_j \le \min\{C^W, y\} \end{pmatrix}, \tag{4.4}$$

$$x_j \geq 0, \forall j \in \mathcal{J}. \tag{4.5}$$

The receipt and delivery capacities at each market are imposed by constraints (4.2) and (4.3), respectively. The left and right hand sides of the disjunction (4.4) express the storage constraints: When the storage decision is to inject, (4.4) ensures that (i) the sum of the withdrawal trade amounts is zero and (ii) the sum of the injection trade amounts is less than both the storage injection capacity and the available space in storage; if the storage decision is to withdraw, (4.4) ensures that (i) the sum of the injection trade amounts is zero and (ii) the sum of the withdrawal trade amounts is less than both the storage withdrawal capacity and the available inventory in storage. Constraints (4.5) enforce nonnegativity of the trade amounts. The set of feasible trade amounts is thus defined as $\mathcal{X}(y) := \{\boldsymbol{x} | (4.2)\text{-}(4.5)\}$.

Given the stage i spot price vector $\boldsymbol{s}_i$, executing a feasible collection of trade amounts $\boldsymbol{x}$ at inventory level y_i results in an immediate reward of $r(\boldsymbol{x}, \boldsymbol{s}_i)$ and an inventory transition from y_i to $y_i + \sum_{j \in \mathcal{J}^I} x_j - \sum_{j \in \mathcal{J}^W} x_j$. In contrast, the evolution of the stage i array of forward curves $\boldsymbol{F}_i$ into the stage $i+1$ array of forward curves $\boldsymbol{F}_{i+1}$ is governed by a known risk-adjusted stochastic process, to be discussed shortly, which is assumed to be unaffected by the merchant trades (that is, the merchant is a small player). Let $\mathbb{E}$ denote expectation under the corresponding risk-neutral probability measure for the forward curve evolution (Secomandi and Seppi, 2014, ch3). A policy π is the collection of decision rules $\{A_0^\pi, A_1^\pi, \ldots, A_{N-1}^\pi\}$, where $A_i^\pi : (y_i, \boldsymbol{F}_i) \mapsto \mathcal{X}(y_i)$, $\forall (i, y_i, \boldsymbol{F}_i) \in \mathcal{I} \times \mathcal{Y} \times \mathbb{R}_+^{M \cdot (N-i)}$. We let Π be the set of all feasible policies. We denote by δ the risk-free discount factor from each time T_i back to time T_{i-1}, $\forall i \in \mathcal{I} \setminus \{0\}$. Let $(y_0, \boldsymbol{F}_0)$ be the time $T_0 := 0$ state. Maximizing the time T_0 value of the storage and transport assets entails solving the following MDP:

$$\max_{\pi \in \Pi} \sum_{i \in \mathcal{I}} \delta^i \mathbb{E}\left[r(A_i^\pi(y_i^\pi, \boldsymbol{F}_i), \boldsymbol{s}_i) | y_0, \boldsymbol{F}_0 \right], \tag{4.6}$$

where y_i^π is the random inventory level in stage i when using policy π. Denoting the value function in stage i and state $(y_i, \boldsymbol{F}_i)$ by $V_i(y_i, \boldsymbol{F}_i)$, an optimal policy of (4.6) can in theory be computed by solving the following SDP, $\forall (i, y_i, \boldsymbol{F}_i) \in \mathcal{I} \times \mathcal{Y} \times \mathbb{R}_+^{M \cdot (N-i)}$:

$$V_i(y_i, \boldsymbol{F}_i) = \max_{\boldsymbol{x} \in \mathcal{X}(y_i)} r(\boldsymbol{x}, \boldsymbol{s}_i) + \delta \mathbb{E}\left[V_{i+1}\left(y_i + \sum_{j \in \mathcal{J}^I} x_j - \sum_{j \in \mathcal{J}^W} x_j, \boldsymbol{F}_{i+1} \right) \middle| \boldsymbol{F}_i \right], \tag{4.7}$$

with boundary conditions $V_N(y_N, \boldsymbol{F}_N) := 0, \forall y_N \in \mathcal{Y}$.

We model the continuous time *risk neutral* dynamics of the array of forward curves using a reduced-form price model, which is popular in academia and practice for modeling natural gas forward curve evolution. Reduced-form models for natural gas forward curve evolution are popular because their calibration relies on natural gas futures and options that have significant trading volume. In these models, it is assumed that the impact of factors such as demand, supply, and capacity on natural gas prices at equilibrium (see discussion at the end of §4.3) are embedded in the prices of futures and options and can be (approximately) captured via calibration (see Secomandi 2010, page 401, and Eydeland and Wolyniec 2003, ch. 4). The specific reduced-form model we use is a multi-market

extension of the multifactor term structure model that is widespread among practitioners in commodity industries (Cortazar and Schwartz, 1994, Clewlow and Strickland, 2000, Blanco et al., 2002, Secomandi et al., 2012, Secomandi and Seppi, 2014, ch. 4). The one and two factor models of Jaillet et al. (2004) and Schwartz and Smith (2000), respectively, are special cases of this model. In this continuous time setting, we denote by $F^m(t,T_i)$ the market m futures price at time $t \in [T_0,T_i]$ with maturity on date $T_i \geq t$. We denote by K the number of stochastic factors driving the evolution of this price; by $dW_k(t)$ the standard geometric Brownian motion increment corresponding to factor k at time t; and by $\sigma_{m,i,k}(t)$ the time t loading factor on the k-th Brownian motion increment for the evolution of the price of the market m futures with maturity at time T_i. The evolution of $F^m(t,T_i)$, $\forall (m,i) \in \mathcal{M} \times \mathcal{I} \setminus \{0\}$ and $t \in (T_0,T_i]$, is governed by the following stochastic differential equations:

$$\frac{dF^m(t,T_i)}{F^m(t,T_i)} = \sum_{k=1}^{K} \sigma_{m,i,k}(t) dW_k(t),\ \forall (m,i) \in \mathcal{M} \times \mathcal{I} \setminus \{0\}, t \in (T_0,T_i], \tag{4.8}$$

$$dW_k(t) dW_{k'}(t) = 0,\ \forall k,k' \in \{1,2,\ldots,K\},\ k \neq k'. \tag{4.9}$$

This model captures the seasonality in price levels via the initial array of forward curves, and the seasonalities in the price changes through the dependence of the loading factors on the trading time (t). The price changes can be correlated because they are functions of common factors. Our analysis in §4.4 does not depend on this specific price model. In contrast, our algorithm developed in §4.5 to obtain a heuristic policy and one of the upper bounds presented in §4.6 rely on a particular specification of this price model, which we use in our numerical analysis carried out in §4.7.

4.4 Structural Analysis

In this section we analyze the SDP (4.7). In §4.4.1 we reformulate it to facilitate our structural analysis of the value function and of an optimal storage policy of this SDP in §4.4.2 and the development of our LSM approach in §4.5. We formally establish the relationship between the value of storage and transport in §4.4.3.

4.4.1 Reformulation of SDP (4.7)

Our reformulation of the SDP (4.7) explicitly optimizes the storage inventory change assuming the storage and transport decisions are made optimally for every feasible storage inventory change. We also refer to the storage inventory change as the storage action. We point out that a given storage action may result from executing more than one storage trade.

Given the inventory levels y_i and y_{i+1}, define the storage action (inventory change) a as $y_i - y_{i+1}$: A positive storage action corresponds to a withdrawal of natural gas from the storage asset, a negative storage action corresponds to an injection of natural gas

into this asset, and a zero storage action corresponds to leaving the inventory in storage unchanged (doing nothing). Let $\mathcal{X}'(a)$ denote the collections of all (storage and transport) trade amounts that satisfy the receipt and delivery capacity constraints and result in an inventory change equal to a. A vector of trade amounts $\boldsymbol{x}$ belongs to this set if it satisfies

$$\sum_{j\in\mathcal{J}^I} x_j = \begin{cases} -a, & \text{if } a < 0, \\ 0, & \text{otherwise,} \end{cases} \tag{4.10}$$

$$\sum_{j\in\mathcal{J}^W} x_j = \begin{cases} a, & \text{if } a > 0, \\ 0, & \text{otherwise,} \end{cases} \tag{4.11}$$

$$(4.2), (4.3), (4.5). \tag{4.12}$$

Constraints (4.10) and (4.11) ensure that the sums of the injection and the withdrawal trade amounts, respectively, are consistent with the storage action a.

Given a spot price vector $\boldsymbol{s} \in \mathbb{R}_+^M$ and a storage action a, an optimal collection of storage and transport trade amounts in set $\mathcal{X}'(a)$ can be computed by solving the linear program

$$\bar{r}(a, \boldsymbol{s}) := \max_{\boldsymbol{x}\in\mathcal{X}'(a)} r(\boldsymbol{x}, \boldsymbol{s}). \tag{4.13}$$

We define a^I and a^W as the maximum injection and withdrawal amounts, respectively, ignoring the storage asset capacity constraints:

$$a^I := \max_{\boldsymbol{x}} \sum_{j\in\mathcal{J}^I} x_j \text{ s.t. } (4.2), (4.3), (4.5);$$

$$a^W := \max_{\boldsymbol{x}} \sum_{j\in\mathcal{J}^W} x_j \text{ s.t. } (4.2), (4.3), (4.5).$$

Given a storage action a, it is straightforward to check that it is feasible, that is, $\mathcal{X}'(a) \neq \emptyset$, if and only if $a \in [-a^I, a^W]$. Thus, the set of feasible stage $i+1$ inventory levels reachable from the stage i inventory level y_i is

$$\mathcal{Z}(y_i) := \left[\max\{0, y_i - C^W, y_i - a^W\}, \min\{\overline{y}, y_i + C^I, y_i + a^I\}\right].$$

Define the continuation function $W_i(y_{i+1}, \boldsymbol{F}_i)$, $\forall (i, y_{i+1}, \boldsymbol{F}_i) \in \mathcal{I} \times \mathcal{Y} \times \mathbb{R}_+^{M\cdot(N-i)}$, as

$$W_i(y_{i+1}, \boldsymbol{F}_i) := \delta\mathbb{E}\left[V_{i+1}\left(y_{i+1}, \boldsymbol{F}_{i+1}\right) | \boldsymbol{F}_i\right]. \tag{4.14}$$

Our reformulated SDP (4.7) is

$$V_i(y_i, \boldsymbol{F}_i) = \max_{y_{i+1}\in\mathcal{Z}(y_i)} \bar{r}(y_i - y_{i+1}, \boldsymbol{s}_i) + W_i(y_{i+1}, \boldsymbol{F}_i), \tag{4.15}$$

$\forall (i, y_i, \boldsymbol{F}_i) \in \mathcal{I} \times \mathcal{Y} \times \mathbb{R}_+^{M\cdot(N-i)}$, with boundary conditions $V_N(y_N, \boldsymbol{F}_N) := 0, \forall y_N \in \mathcal{Y}$. In contrast to SDP (4.7), the maximization in SDP (4.15) is over the feasible next stage

inventory level and the function $\bar{r}(\cdot, \boldsymbol{s}_i)$ used in this maximization returns the value of the optimal storage and transport trade amounts conditional on a feasible storage action.

4.4.2 Value and Continuation Functions and Optimal Storage Policy Structure

We now characterize the value and continuation functions and an optimal policy of the SDP (4.15). Specifically, we characterize the structure of an optimal storage policy. Under a mild assumption, this analysis leads to a substantial simplification of the SDP (4.15). Let $\mathcal{I}_i := \{i+1, i+2, \ldots, N-1\}$. We begin by establishing the concavity of the value and continuation functions of this SDP in Lemma 2.

Lemma 2. *For each given $(i, \boldsymbol{F}_i) \in \mathcal{I} \times \mathbb{R}_+^{M\cdot(N-i)}$, the functions $V_i(y_i, \boldsymbol{F}_i)$ and $W_i(y_{i+1}, \boldsymbol{F}_i)$ are concave in $y_i \in \mathcal{Y}$ and $y_{i+1} \in \mathcal{Y}$, respectively.*

We state a technical assumption with no practical impact that allows us to refine Lemma 2 and establish the structure of an optimal storage policy in Proposition 10.

Assumption 3. *The parameters $C^{R,m}$, $\forall m \in \mathcal{M}$, $C^{D,m}$, $\forall m \in \mathcal{M}$, y_0, $\bar{y}$, C^W, and C^I are rational numbers.*

Let $\bar{G}$ be the largest rational number such that the *transport* capacity values in sets $\{C^{R,m}, \forall m \in \mathcal{M}\}$ and $\{C^{D,m}, \forall m \in \mathcal{M}\}$ are integer multiples of $\bar{G}$ ($\bar{G}$ exists by Assumption 3). We interpret $\bar{G}$ as a lot size. Lemma 3 characterizes the function $\bar{r}(\cdot, \boldsymbol{s})$ in terms of this lot size.

Lemma 3. *Suppose Assumption 3 holds. For each given spot price vector $\boldsymbol{s} \in \mathbb{R}_+^M$, the function $\bar{r}(\cdot, \boldsymbol{s})$ is piecewise linear concave on the interval $a \in [-a^I, a^W]$ with slope changes at integer multiples of $\bar{G}$.*

The optimal storage policy structure established shortly relies on the *target* function $b_i(y_i, \boldsymbol{F}_i)$, defined as the smallest element of

$$\underset{y_{i+1}\in\mathcal{Y}}{\operatorname{argmax}}\, \bar{r}(y_i - y_{i+1}, \boldsymbol{s}_i) + W_i(y_{i+1}, \boldsymbol{F}_i). \tag{4.16}$$

The *target* qualification of this function is due to its output being a stage $i+1$ inventory level that might not be reachable from the stage i inventory level y_i – because the optimization in (4.16) is over the set of all feasible inventory levels, $\mathcal{Y}$, which potentially *strictly* includes the set of feasible next stage inventory levels for the given inventory level y_i, $\mathcal{Z}(y_i)$. We also define the functions $\underline{b}_i(\boldsymbol{F}_i)$ and $\bar{b}_i(\boldsymbol{F}_i)$ that return the smallest and largest feasible inventory levels, respectively, for which the do nothing storage action is optimal in stage i given the array of forward curves $\boldsymbol{F}_i$:

$$\underline{b}_i(\boldsymbol{F}_i) \;:=\; \min_{y_i\in\mathcal{Y}} y_i \text{ s.t. } y_i = b_i(y_i, \boldsymbol{F}_i),$$

$$\bar{b}_i(\boldsymbol{F}_i) \;:=\; \max_{y_i \in \mathcal{Y}} y_i \text{ s.t. } y_i = b_i(y_i, \boldsymbol{F}_i).$$

We show the existence of these functions in Proposition 10.

Under Assumption 3, Proposition 10, based on Lemmas 2 and 3, refines in Part (a) the characterizations of the value and continuation functions of the SDP (4.15) established in Lemma 2 and characterizes in Part (b) the structure of an optimal storage policy. This characterization depends on the lot size G that is defined as the largest rational number such that $\bar{G}$, y_0, $\bar{y}$, C^W, and C^I are all integer multiples of G (G exists by Assumption 3). This lot size is smaller, potentially strictly, than the lot size $\bar{G}$ because it also depends on the storage injection and withdrawal capacities and the initial and maximum inventory levels. We denote an optimal storage decision rule as $\bar{A}_i^*(y_i, \boldsymbol{F}_i)$.

Proposition 10. *Suppose Assumption 3 holds.*

(a) For each given $(i, \boldsymbol{F}_i) \in \mathcal{I} \times \mathbb{R}_+^{M\cdot(N-i)}$, the value function $V_i(y_i, \boldsymbol{F}_i)$ and the continuation function $W_i(y_{i+1}, \boldsymbol{F}_i)$ are piecewise linear concave in $y_i \in \mathcal{Y}$ and $y_{i+1} \in \mathcal{Y}$, respectively, with slope changes at integer multiples of G.

(b) For each given $(i, \boldsymbol{F}_i) \in \mathcal{I} \times \mathbb{R}_+^{M\cdot(N-i)}$ and for every $q \in \{0, G, \ldots, (\bar{y}/G) - 1\}$, the target function $b_i(\cdot, \boldsymbol{F}_i)$ equals a constant or varies linearly with slope 1 for all $y_i \in [qG, (q+1)G]$:

$$b_i(y_i, \boldsymbol{F}_i) = b_i(qG, \boldsymbol{F}_i) + (y_i - qG)\theta_q, \forall y_i \in [qG, (q+1)G],$$

with $\theta_q \in \{0, 1\}$. The functions $\underline{b}_i(\boldsymbol{F}_i)$ and $\bar{b}_i(\boldsymbol{F}_i)$ exist and partition the feasible inventory set $\mathcal{Y}$ into the three regions $[0, \underline{b}_i(\boldsymbol{F}_i))$, $[\underline{b}_i(\boldsymbol{F}_i), \bar{b}_i(\boldsymbol{F}_i)]$, and $(\bar{b}_i(\boldsymbol{F}_i), \bar{y}]$, such that (i) an optimal storage action is to inject when $y_i \in [0, \underline{b}_i(\boldsymbol{F}_i))$, do nothing when $y_i \in [\underline{b}_i(\boldsymbol{F}_i), \bar{b}_i(\boldsymbol{F}_i)]$, and withdraw when $y_i \in (\bar{b}_i(\boldsymbol{F}_i), \bar{y}]$, and (ii) the function $b_i(y_i, \boldsymbol{F}_i)$ returns a value that lies in the region of this partition that y_i belongs to. Specifically, an optimal storage policy is defined by the decision rules

$$\bar{A}_i^*(y_i, \boldsymbol{F}_i) = \begin{cases} \max\{-C^I, y_i - b_i(y_i, \boldsymbol{F}_i)\}, & \text{if } y_i \in [0, \underline{b}_i(\boldsymbol{F}_i)), \\ 0, & \text{if } y_i \in [\underline{b}_i(\boldsymbol{F}_i), \bar{b}_i(\boldsymbol{F}_i)], \\ \min\{C^W, y_i - b_i(y_i, \boldsymbol{F}_i)\}, & \text{if } y_i \in (\bar{b}_i(\boldsymbol{F}_i), \bar{y}], \end{cases} \tag{4.17}$$

$\forall (i, y_i, \boldsymbol{F}_i) \in \mathcal{I} \times \mathcal{Y} \times \mathbb{R}_+^{M\cdot(N-i)}$.

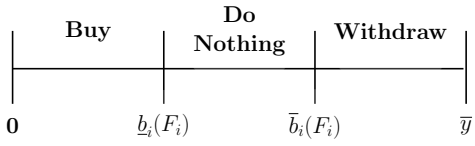


Figure 4.7: Partition of the feasible inventory set based on type of storage action

Under Assumption 3, Part (a) of Proposition 10 establishes the piecewise linearity of the value and continuation functions of the SDP (4.15) with possible slope changes at a predetermined set of values. Related results have been established in Bannister and Kaye (1991) and Nascimento and Powell (2013a). However, these papers do not characterize an optimal storage policy structure, as we do in Part (b) of Proposition 10.

Before interpreting our optimal storage policy structure, we briefly describe the *double basestock* target structure of an optimal storage policy that is known in the single market case (Secomandi, 2010, Secomandi et al., 2012). This structure includes two stage and forward curve, but not inventory, dependent basestock target functions. Given a stage and a forward curve it is optimal to inject up to the smaller basestock target function value for inventory levels below this target value, withdraw down to the larger basestock target function value for inventory levels above this target value, and do nothing for inventory levels between these target function values. Thus, for each given stage and forward curve, the feasible inventory set is partitioned into inject, do nothing, and withdraw regions, and the optimal storage action within the inject and withdraw regions *strictly* increases in inventory. Moreover, under an assumption analogous to Assumption 3, but which excludes the initial inventory level and transport capacities, these target values are integer multiples of a given lot size.

Under Assumption 3, Part (b) of Proposition 10 establishes the structure of an optimal storage policy in the multiple market setting. Analogous to the single market case, for each given stage and array of forward curves, the feasible inventory set is partitioned into inject, do nothing, and withdraw regions (see Figure 4.7). However, this partition is defined by the functions $\underline{b}_i(\boldsymbol{F}_i)$ and $\bar{b}_i(\boldsymbol{F}_i)$ that may not be target functions. In other words, in general these are only threshold functions that define this partition. Moreover, an optimal storage action is determined by the target function $b_i(y_i, \boldsymbol{F}_i)$, which can be interpreted as a piecewise linear *basestock* target function that also depends on the inventory level y_i, and can thus take *infinitely* many values for each given array of forward curves $\boldsymbol{F}_i$. This aspect is in stark contrast to the *double* basestock optimal policy structure for the single market case. As in this case, however, these basestock target functions do not bring a given inventory level outside of the region it belongs to in the inject, do nothing, and withdraw partition, and the resulting next stage inventory level is a weakly increasing function of this inventory level. Moreover, different from this case, the storage injection and withdrawal actions can be *weakly* increasing in the inventory level, that is, it is possible to have subregions where these actions are equal to a constant. Figure 4.8 conceptually illustrates this structure for the injection case.

Our optimal storage policy structure has useful computational implications. By Part (b) of Proposition 10, our optimal storage policy only visits inventory levels in the finite set

$$\mathcal{Y}' := \{0, G, \ldots, \bar{y}\},$$

with $\bar{y}/G+1$ values, and for each inventory level y_i in this set chooses a next stage inventory level in the finite set

$$\mathcal{Z}'(y_i) := \left\{y_i - \min\{C^W, a^W\}, y_i - \min\{C^W, a^W\} + G, \ldots, y_i + \min\{C^I, a^I\}\right\}.$$

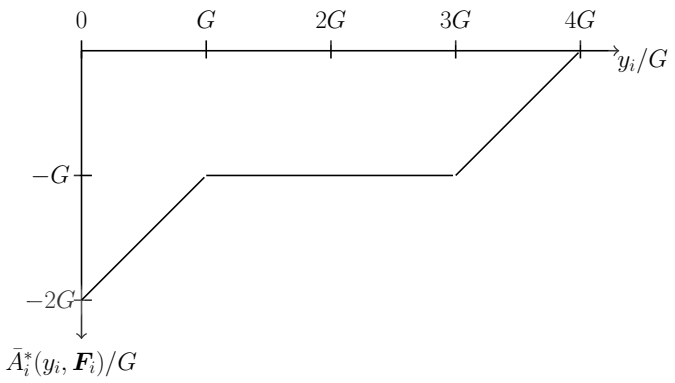


Figure 4.8: Conceptual illustration of the piecewise linearity of an optimal policy structure in the injection region with $\underline{b}_i(\boldsymbol{F}_i) = 4G$.

Thus, in principle, the optimal storage policy associated with the decision rules (4.17) could be computed by solving the following SDP:

$$V_i(y_i, \boldsymbol{F}_i) = \max_{y_{i+1} \in \mathcal{Z}'(y_i)} \bar{r}(y_i - y_{i+1}, \boldsymbol{s}_i) + W_i(y_{i+1}, \boldsymbol{F}_i), \tag{4.18}$$

$\forall (i, y_i, \boldsymbol{F}_i) \in \mathcal{I} \times \mathcal{Y}' \times \mathbb{R}_+^{M \cdot (N-i)}$, with boundary conditions $V_N(y_N, \boldsymbol{F}_N) := 0, \forall y_N \in \mathcal{Y}'$. The SDP (4.18) critically differs from the SDP (4.15) because it optimizes over the finite set $\mathcal{Z}'(y_i)$ rather than the interval $\mathcal{Z}(y_i)$.

Given the pair $(i, \boldsymbol{F}_i)$ and *assuming* knowledge of the function $W_i(\cdot, \boldsymbol{F}_i)$, the optimization on the right hand side of (4.18) can be performed efficiently *for all* the feasible inventory levels y_i by exploiting the results established in Proposition 10. Specifically, combining the structure of the functions $\bar{r}(\cdot, \boldsymbol{s}_i)$ and $W_i(\cdot, \boldsymbol{F}_i)$ and the decision rules (4.17) allows us to efficiently compute the set of optimal next stage inventory levels for all inventory levels y_i by scanning *once* the next stage inventory levels in the set $\mathcal{Y}'$. This efficient search is possible because the optimal next stage inventory level weakly increases in the current stage i inventory level, y_i, and thus we can restrict our search for the optimal next stage inventory level for a given y_i in set $\mathcal{Y}' \setminus \{0\}$ to the subset of $\mathcal{Y}'$ delimited by the optimal next stage inventory level computed for $y_i - G$ and the maximum inventory level $\bar{y}$.

Although the function $W_i(\cdot, \boldsymbol{F}_i)$ is unknown, the same scheme remains applicable when this function is replaced by another piecewise linear concave function of the current inventory level with slope changes at integer multiples of the lot size G. We use this approach in §4.5 when developing our approximate solution method, which relies on approximating the continuation function $W_i(\cdot, \boldsymbol{F}_i)$ with a function that satisfies this property. Moreover, we take advantage of this property when using our estimated approximate continuation function to compute a heuristic policy and estimate a lower bound on the combined value of the storage and transport assets, as detailed in §4.5. One of the approaches presented in §4.6 to estimate an upper bound on this value also critically relies on our estimated value function approximations, which we use to obtain our approximate continuation functions,

changing slope at integer multiples of the lot size G.

Finally, it is important to point out that the lot size used to define the sets $\mathcal{Y}'$ and $\mathcal{Z}'(y_i)$ depends on the initial inventory level. This dependence implies that different initial inventory levels potentially require different discretizations of these sets. In contrast, the lot size used to discretize the feasible inventory and storage action sets in the single market case is independent of the initial inventory level (Secomandi, 2010, Secomandi et al., 2012).

4.4.3 Relationship Between the Values of Storage and Transport

Storage and transport compete for the same receipt and delivery capacity. Thus, intuitively, they are substitute activities. Proposition 11 establishes this property formally. Let Π^{TR} and Π^{ST} be the subsets of policies in Π that allow only transport and storage trades, respectively. We define the transport and storage MDPs as

$$V_0^{TR}(x_0, \boldsymbol{F}_0) := \max_{\pi \in \Pi^{TR}} \sum_{i \in \mathcal{I}} \delta^i \mathbb{E}\left[r(A_i^\pi(y_i^\pi, \boldsymbol{F}_i), \boldsymbol{s}_i) | y_0, \boldsymbol{F}_0\right], \tag{4.19}$$

$$V_0^{ST}(x_0, \boldsymbol{F}_0) := \max_{\pi \in \Pi^{ST}} \sum_{i \in \mathcal{I}} \delta^i \mathbb{E}\left[r(A_i^\pi(y_i^\pi, \boldsymbol{F}_i), \boldsymbol{s}_i) | y_0, \boldsymbol{F}_0\right]. \tag{4.20}$$

Proposition 11. *It holds that*

$$V_0(x_0, \boldsymbol{F}_0) \leq V_0^{TR}(x_0, \boldsymbol{F}_0) + V_0^{ST}(x_0, \boldsymbol{F}_0). \tag{4.21}$$

Inequality (4.21) is consistent with the definition of substitutes in §2.6.1 of Topkis (1998). When (4.21) holds as a strict inequality, storage and transport are strict substitutes and jointly managing these activities is necessary to obtain an optimal policy. In other words, a pair of optimal policies to (4.19) and (4.20) cannot form an optimal policy to (4.6). On the other hand, when (4.21) holds as an equality there is no substitution between storage and transport, and hence these activities can be optimally managed independently of each other. We estimate numerically the degree of substitutability between storage and transport in §4.7.2.3.

4.5 LSM Heuristic Policy and Lower Bound

In theory, an optimal storage action at a given stage and state can be computed by solving the optimization problem defined by the right hand side of the SDP (4.18) – we suppose that Assumption 3 holds in the rest of this chapter. However, this approach is not practical because computing the continuation function in this optimization is intractable due to (i) the high dimensionality of the state space and (ii) the inability to compute the expectation in the definition of this function. To overcome both these issues, we follow a popular approximate dynamic programming (ADP) approach to compute heuristic, but hopefully near optimal, decisions, by replacing the unknown continuation function in the right hand

side of the SDP (4.18) by a *tractable* continuation function approximation (Powell 2011, Bertsekas 2007). A lower bound on the option value can be estimated by Monte Carlo simulation of the policy defined by these heuristic decisions. We explain the details of the lower bound estimation after we describe the LSM approach (Longstaff and Schwartz, 2001, Tsitsiklis and Van Roy, 2001) that we use to compute a continuation function approximation.

We extend the version of the LSM approach proposed by Chapter 3 to compute a continuation function approximation by first computing a *value function* approximation, which is also useful for upper bound estimation, as discussed in §4.6. We consider value function approximations that are linear combinations of a given set of basis functions. For each stage i and inventory level $y_i \in \mathcal{Y}'$ we specify B_i basis functions. The b-th basis function is $\phi_{i,y_i,b} : \boldsymbol{F}_i \mapsto \mathbb{R}$ and its linear combination weight is $\beta_{i,y_i,b}$. We define the row and column vectors $\Phi_{i,y_i} := (\phi_{i,y_i,1}, \phi_{i,y_i,2}, \ldots, \phi_{i,y_i,B_i})$ and $\beta_{i,y_i} := (\beta_{i,y_i,1}, \beta_{i,y_i,2}, \ldots, \beta_{i,y_i,B_i})^\top$. The value function approximation is

$$\hat{V}_i(y_i, \boldsymbol{F}_i; \beta_{i,y_i}) := (\Phi_{i,y_i}\beta_{i,y_i})(\boldsymbol{F}_i) \equiv \sum_{b=1}^{B_i} \phi_{i,y_i,b}(\boldsymbol{F}_i)\beta_{i,y_i,b}. \tag{4.22}$$

We define a continuation function approximation by replacing V_i by $\hat{V}_i$ on the right hand side of (4.14):

$$\begin{aligned} \hat{W}_i(y_{i+1}, \boldsymbol{F}_i; \beta_{i+1,y_{i+1}}) &:= \delta\mathbb{E}\left[\hat{V}_{i+1}(y_{i+1}, \boldsymbol{F}_{i+1}; \beta_{i+1,y_{i+1}})|\boldsymbol{F}_i\right] \\ &= \delta \sum_{b=1}^{B_{i+1}} \mathbb{E}\left[\phi_{i+1,y_{i+1},b}(\boldsymbol{F}_{i+1})|\boldsymbol{F}_i\right] \beta_{i+1,y_{i+1},b}. \end{aligned} \tag{4.23}$$

Following §3.5.2 we choose basis functions such that the expectations in (4.23) can be computed exactly when using the price model (4.8)-(4.9), as discussed in §4.7.2.1. We extend the LSM variant of §3.5.2 to approximate the SDP (4.18). We refer to our extension as extended LSM (ELSM). Our extension consists of a concavification step, discussed below, which yields a continuation function approximation $\hat{W}_i^{conc}(\cdot, \boldsymbol{F}_i; \beta_{i+1,\cdot})$ that is piecewise linear concave with break points in set $\mathcal{Y}'$. This step has not been used with LSM methods but it has appeared in other ADP contexts (Powell 2011, ch. 13, and Nascimento and Powell 2013a)

Algorithm 3 summarizes the ELSM steps. The inputs to ELSM are the number of sample paths and basis functions that allow the exact computation of the expectation in (4.23) (see §4.7.2.1 for an example). ELSM outputs the weights $\bar{\beta}_{i,y_i}$ that define a value function approximation via (4.22) and a continuation function approximation via (4.23). ELSM starts by generating H *regression* sample paths of the arrays of forward curves from stage 1 through N starting from $\boldsymbol{F}_0$, which we include in set $\{\boldsymbol{F}_i^h, i \in \mathcal{I} \setminus \{0\}, h = 1, 2, \ldots, H\}$, and initializing the stage N weight vector $\bar{\beta}_{N,y_N}$ to zero. At each stage $i \in \mathcal{I} \setminus \{0\}$, starting from stage $N-1$ and moving backward to stage 1: In Step 1(i), for each $h \in \{1, 2, \ldots, H\}$, ELSM computes the stage i continuation function approximation using the stage $i+1$ basis function weights. In Step 1(ii) ELSM concavifies this continuation function approximation

Algorithm 3: ELSM

Inputs: Number of sample paths H and basis functions that allow the exact computation of the expectation in (4.23).

Outputs: Weights $\bar{\beta}_{i,y_i}$, $\forall (i, y_i) \in \mathcal{I} \times \mathcal{Y}'$.

Initialization: Generate H *regression* sample paths of the arrays of forward curves $\{\boldsymbol{F}_i^h, i \in \mathcal{I} \setminus \{0\}, h = 1, \ldots, H\}$ starting from $\boldsymbol{F}_0$; $\bar{\beta}_{N,y_N} := 0$, $\forall y_N \in \mathcal{Y}'$.

For each $i = N - 1$ to 1 **do**:

1. **For** each $h \in \{1, 2, \ldots, H\}$ **do**:

   (i) **For** each $y_{i+1} \in \mathcal{Y}'$ **do**:

   $$\hat{W}_i(y_{i+1}, \boldsymbol{F}_i^h; \bar{\beta}_{i+1,y_{i+1}}) := \delta \sum_{b=1}^{B_{i+1}} \mathbb{E}\left[\phi_{i+1,y_{i+1},b}(\boldsymbol{F}_{i+1}) | \boldsymbol{F}_i^h\right] \bar{\beta}_{i+1,y_{i+1},b}.$$

   (ii) Concavify $\hat{W}_i(\cdot, \boldsymbol{F}_i^h; \bar{\beta}_{i+1,\cdot})$ to obtain $\hat{W}_i^{conc}(\cdot, \boldsymbol{F}_i^h; \bar{\beta}_{i+1,\cdot})$.

   (iii) **For** each $y_i \in \mathcal{Y}'$ **do**:

   $$v_i(y_i, \boldsymbol{F}_i^h) := \max_{y_{i+1} \in \mathcal{Z}'(y_i)} \bar{r}(y_i - y_{i+1}, \boldsymbol{s}_i^h) + \hat{W}_i^{conc}(y_{i+1}, \boldsymbol{F}_i^h; \bar{\beta}_{i+1,y_{i+1}}). \tag{4.24}$$

2. **For** each $y_i \in \mathcal{Y}'$ **do**: Perform a 2-norm regression on the set of value function estimates $\{v_i(y_i, \boldsymbol{F}_i^h), \forall h \in \{1, 2, \ldots, H\}\}$ to determine the weights $\bar{\beta}_{i,y_i}$.

by a simple scanning and modification procedure illustrated in Figure 4.9. After concavification, for each $y_i \in \mathcal{Y}'$, in Step 1(iii) ELSM computes the approximate value function estimates for each inventory level $y_i \in \mathcal{Y}'$ by solving a set of greedy optimization problems (4.24). These optimizations can be performed efficiently using the scheme discussed at the end of §4.4.2, because the functions $\hat{W}_i^{conc}(y_{i+1}, \boldsymbol{F}_i^h; \bar{\beta}_{i+1,y_{i+1}})$ and $\bar{r}(y_i - y_{i+1}, \boldsymbol{s}_i^h)$ are both piecewise linear concave in y_{i+1} with break points in set $\mathcal{Z}'(y_i)$ given a pair $(i, \boldsymbol{F}_i)$. In Step 2, ELSM performs a 2-norm regression on these value function estimates to determine the weight vectors $\bar{\beta}_{i,y_i}$ at each inventory level y_i in set $\mathcal{Y}'$.

We now explain the details of how the continuation function approximation estimated by ELSM is used to estimate a lower bound. At stage i and state $(y_i, \boldsymbol{F}_i)$, we replace the continuation function W_i on the right hand side of the SDP (4.18) with $\hat{W}_i^{conc}(y_{i+1}, \boldsymbol{F}_i; \bar{\beta}_{i+1,y_{i+1}})$ to obtain the following optimization problem, which is greedy with respect to $\hat{W}_i$:

$$\max_{y_{i+1} \in \mathcal{Z}'(y_i)} \bar{r}_i(y_i - y_{i+1}, \boldsymbol{s}_i) + \hat{W}_i^{conc}(y_{i+1}, \boldsymbol{F}_i; \bar{\beta}_{i+1,y_{i+1}}). \tag{4.25}$$

Given the solution y_{i+1}^g to (4.25) (breaking ties in favor of the smallest maximizer), the

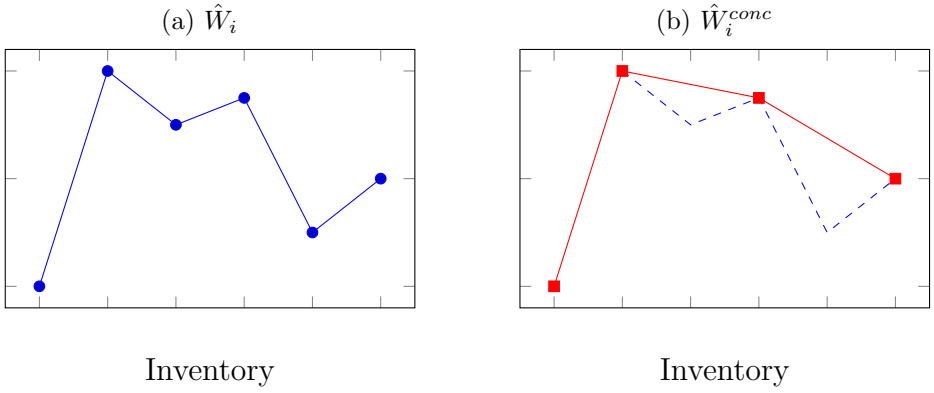


Figure 4.9: Illustration of the ELSM concavification step.

corresponding greedy storage action is $a_i^g := y_i - y_{i+1}^g$. The collection of greedy actions at all stages and states defines the greedy policy associated with $\hat{W}_i^{conc}$. Because $\hat{W}_i^{conc}(\cdot, \boldsymbol{F}_i^h; \bar{\beta}_{i+1,\cdot})$ is piecewise linear concave with slope changes at integer multiples of the lot size G, each y_{i+1}^g can be found efficiently, and the corresponding greedy policy has the same structure of the optimal policy identified in Part (b) of Proposition 10. A lower bound on the value of an optimal policy can be estimated by applying the greedy policy along L Monte Carlo *simulation* sample paths of the arrays of forward curves, which we include in set $\{\boldsymbol{F}_i^l, i \in \mathcal{I} \setminus \{0\}, l = 1, 2, \ldots, L\}$, starting from the time 0 inventory level y_0 and the array of forward curves $\boldsymbol{F}_0$.

4.6 Dual Upper Bounds

In this section we discuss the estimation of dual upper bounds on the value of an optimal policy (Brown et al. 2010 and references therein). Such a bound is based on performing hindsight optimizations in which knowledge of future information is penalized using dual penalties. We denote by $u_i(y_{i+1}, \boldsymbol{F}_i, \boldsymbol{F}_{i+1})$ the dual penalty in stage i given y_{i+1}, $\boldsymbol{F}_i$, and $\boldsymbol{F}_{i+1}$. Specifically, these penalties penalize knowledge in stage i of the array of forward curves $\boldsymbol{F}_{i+1}$ and must satisfy the feasibility condition $\mathbb{E}[u_i(y_{i+1}, \boldsymbol{F}_i, \boldsymbol{F}_{i+1})|\boldsymbol{F}_i] \leq 0$ (see Brown et al. 2010 for details). Once feasible dual penalties are specified, we estimate dual upper bounds in Monte Carlo simulation using the same set of L simulation sample paths $\{\boldsymbol{F}_i^l, i \in \mathcal{I} \setminus \{0\}, l = 1, 2, \ldots, L\}$ employed for lower bound estimation. A point estimate $U_0^l(y_0)$ of an upper bound on $V_0(y_0, \boldsymbol{F}_0)$ can be computed based on the l-th sample path of the arrays of forward curves by solving the following dynamic program, $\forall (i, y_i) \in \mathcal{I} \times \mathcal{Y}'$:

$$U_i^l(y_i) = \max_{y_{i+1} \in \mathcal{Z}_i'(y_i)} \bar{r}(y_i - y_{i+1}, \boldsymbol{s}_i^l) - u_i(y_{i+1}, \boldsymbol{F}_i^l, \boldsymbol{F}_{i+1}^l) + \delta U_{i+1}^l(y_{i+1}),$$

with boundary conditions $U_N^l(y_N) := 0$, $\forall y_N \in \mathcal{Y}'$. An upper bound estimate on the optimal policy value $V_0(y_0, \boldsymbol{F}_0)$ is obtained by averaging the point estimates $U_0^l(y_0)$, $\forall l \in \{1, 2, \ldots, L\}$.

We consider the following feasible dual penalties instantiated using the ELSM value function approximation (4.22):

$$\hat{V}_{i+1}(y_{i+1}, \boldsymbol{F}_{i+1}; \bar{\beta}_{i+1,y_{i+1}}) - \mathbb{E}\left[\hat{V}_{i+1}(y_{i+1}, \boldsymbol{F}_{i+1}; \bar{\beta}_{i+1,y_{i+1}})|\boldsymbol{F}_i\right]. \tag{4.26}$$

As alluded to in §4.5, our choice of basis functions allows the exact computation of the expectation in (4.26). We also consider two types of linear dual penalties that are not based on such a function, and hence are easier to instantiate. The first type of linear dual penalties are

$$(s^m_{i+1} - F^m_{i,i+1})y_{i+1}. \tag{4.27}$$

These penalties are feasible because $\mathbb{E}[s^m_{i+1}|F^m_{i,i+1}] = F^m_{i,i+1}$ by the martingale property of futures prices under a risk neutral measure (Shreve, 2004, page 244), and in particular under the price model (4.8)-(4.9). These penalties are motivated by the encouraging upper bounding results of Secomandi (2012) who uses analogous penalties for the case of storage with a single market. The second type of linear dual penalties are averages of the penalties (4.27) for each of the M markets:

$$(1/M)\sum_{m\in\mathcal{M}}(s^m_{i+1} - F^m_{i,i+1})y_{i+1}. \tag{4.28}$$

The feasibility of these dual penalties follows from the feasibility of the dual penalties (4.27).

4.7 Numerical Analysis

In this section we perform our numerical investigation. In §4.7.1 we describe our instances. In §4.7.2 we discuss our findings.

4.7.1 Instances

We developed our instances in conjunction with a major natural gas trading company that operates in the United States. These instances are based on the commercial network displayed in Figure 4.6 and discussed in §4.3. We do not explicitly model the interconnect station IC, in addition to ST, because the commodity charges and fuel losses for transporting natural gas between IC and ST are zero. We use a time horizon equal to a year subdivided into monthly periods (that is, $N = 12$). Our instances include operational and price model parameters.

Operational parameters. The storage asset parameters are normalized maximum inventory ($\bar{y}$) equal to 1 MMBtu; normalized monthly injection capacity (C^I) and withdrawal capacity (C^W) equal to 0.45 MMBtu/month and 0.75 MMBtu/month, respectively; injection and withdrawal fuel adjustment factors (ϕ^I and ϕ^W) equal to 1 and 0.985, respectively; and injection and withdrawal commodity charges (c^I and c^W) equal to 0.02 \$/MMBtu and 0.01 \$/MMBtu, respectively.

Table 4.1: Transport fuel losses ($\phi^{m,m'}$) for the months April to November.

	ST	Z3	Z4	Z6	ELA	M1	M2	M3	AGT
ST	-	1	-	-	1	-	-	-	-
Z3		-	0.9823	0.9638	-	-	-	-	-
Z4			-	0.9672	-	-	-	-	-
Z6				-	-	-	-	-	-
ELA					-	0.9557	0.9406	0.9305	-
M1						-	0.9632	0.9531	-
M2							-	0.9602	-
M3								-	0.9907

Table 4.2: Transport fuel losses ($\phi^{m,m'}$) for the months December to March.

	ST	Z3	Z4	Z6	ELA	M1	M2	M3	AGT
ST	-	1	-	-	1	-	-	-	-
Z3		-	0.9823	0.9638	-	-	-	-	-
Z4			-	0.9672	-	-	-	-	-
Z6				-	-	-	-	-	-
ELA					-	0.9523	0.9316	0.9179	-
M1						-	0.956	0.9423	-
M2							-	0.952	-
M3								-	0.99

Table 4.3: Commodity charges ($c^{m,m'}$, \$/MMBtu).

	ST	Z3	Z4	Z6	ELA	M1	M2	M3	AGT
ST	-	0.05	-	-	0.0103	-	-	-	-
Z3		-	0.02253	0.04454	-	-	-	-	-
Z4			-	0.04027	-	-	-	-	-
Z6				-	-	-	-	-	-
ELA					-	0.0353	0.0762	0.1044	-
M1						-	0.0659	0.0941	-
M2							-	0.0743	-
M3								-	0.013

The operational parameters of the transport assets are commodity charges and fuel losses ($c^{m,m'}$ and $\phi^{m,m'}$, respectively) equal to the values given in Tables 4.1-4.3; receipt and delivery capacities at both markets Z3 and M3 equal to 0.45 MMBtu/month ($= C^W$) and 0.75 MMBtu/month ($= C^I$), respectively; receipt and delivery capacities at all the TRANSCO markets other than market Z3 equal to 0.15 MMBtu/month ($= C^W/3$) and 0.25 MMBtu/month ($= C^I/3$), respectively; and receipt and delivery capacities at the AGT market and all the TETCO markets other than market M3 equal to 0.09 MMBtu/month ($= C^W/5$) and 0.15 MMBtu/month ($= C^I/5$), respectively. Thus, the lot size G is equal to 0.01.

Price model parameters. We calibrated a specification of the price model (4.8)-(4.9) using data made available to us by the energy trading company mentioned above. In this specification each function $\sigma_{m,i',k}(t)$ is right continuous and piecewise constant within a trading month, that is, during each interval in the set $\{[T_i, T_{i+1}), \forall i \in \mathcal{I} \setminus \{N-1\}\}$ (see Blanco et al. 2002 and Secomandi et al. 2012 for more details). We denote by $\sigma_{m,i',k,i}$ the value of each such function taken in the interval $[T_i, T_{i+1})$. Pick $t \in [T_i, T_{i+1})$ and $t' \in (T_i, T_{i+1}]$ with $t' > t$. Recall that $\mathcal{I}_i \equiv \{i+1, i+2, \ldots, N-1\}$. Under this specification, we can reexpress (4.8)-(4.9) in a form suitable for simulation using a vector $(Z_k, k = 1, 2, \ldots, K)$ of K independent standard normal random variables as

$$F^m(t', T_{i'}) = F^m(t, T_{i'}) \exp\left[-\frac{1}{2}(t'-t)\sum_{k=1}^{K}\sigma^2_{m,i',k,i} + \sqrt{t'-t}\sum_{k=1}^{K}\sigma_{m,i',k,i}Z_k\right], \tag{4.29}$$

$\forall (m, i, i') \in \mathcal{M} \times \mathcal{I} \times \mathcal{I}_i$. Notice that prices can be correlated because they are functions of common factors.

Our data set includes 1 year and 3 months (June 2011 to August 2012) of natural gas closing futures prices for Henry Hub, Louisiana, and basis swaps for each of the 8 markets in Figure 4.6. From this information we created a futures price data set for each of these 8 markets. We first estimated monthly sample variance-covariance matrices of the daily log futures price returns across maturities and markets. We then performed a principal component analysis of these matrices and estimated the loading coefficients $\sigma_{m,i',k,i}$ accordingly (see Clewlow and Strickland 2000 §8.6 for more details). We chose the number of factors K equal to 6 because this is the smallest value that explains more than 99% of the total observed variance in each of our monthly data sets.

We created 12 instances by choosing 12 valuation dates corresponding to the first trading date of each month from June 2011 to May 2012. We set the discount factor for each instance based on the following one year United States treasury rates corresponding to our valuation dates: 0.18%, 0.20%, 0.22%, 0.10%, 0.12%, 0.13%, 0.12%, 0.12%, 0.13%, 0.18%, 0.18%, and 0.19%.

The details of the estimated loading coefficients and the initial forward curves are available from the authors upon request.

4.7.2 Findings

In §4.7.2.1 we estimate bounds on the combined value of the storage and transport assets. In §§4.7.2.2-4.7.2.5 we provide insights into the merchant management of these assets.

4.7.2.1 Bound Estimation

For a given stage and inventory pair (i, y_i), we implement ELSM using the following polynomials of futures prices as basis functions:

$$1,$$

$$\{F^m_{i,i'}, \forall i' \in \mathcal{I}_i\},$$
$$\{(F^m_{i,i'})^2, \forall i' \in \mathcal{I}_i\},$$
$$\{F^m_{i,i'}F^{m'}_{i,i'}, \forall i' \in \mathcal{I}_i; m, m' \in \mathcal{M}, m \neq m'\},$$
$$\{F^m_{i,i'}F^m_{i,i'+1}, i' \in \mathcal{I}_i \setminus \{N-1\}, m \in \mathcal{M}\}.$$

This choice of basis functions is common in the LSM literature (Longstaff and Schwartz, 2001). Chapter 3 uses it for valuing storage in a single market. Define $\Delta T_i := T_{i+1} - T_i$. For $w \geq i' > i$ and under price model (4.29), it is easy to verify that the expectation in (4.23) for each of these basis functions is

$$\mathbb{E}[F^m_{i',i''}|F^m_{i,i''}] = F^m_{i,i''},$$
$$\mathbb{E}[(F^m_{i',i''})^2|F^m_{i,i''}] = (F^m_{i,i''})^2 \exp\left\{\sum_{w=i}^{i'-1} \Delta T_w \sum_{k\in\mathcal{K}} \sigma^2_{m,i'',k,w}\right\},$$
$$\mathbb{E}[F^m_{i',i''}F^{m'}_{i',i''}|F^m_{i,i''}, F^{m'}_{i,i''}] = F^m_{i,i''}F^{m'}_{i,i''} \exp\left\{\sum_{w=i}^{i'-1} \Delta T_w \sum_{k\in\mathcal{K}} \sigma_{m,i'',k,w}\sigma_{m',i'',k,w}\right\},$$
$$\mathbb{E}[F^m_{i',i''}F^m_{i',i''+1}|F^m_{i,i''}, F^m_{i,i''+1}] = F^m_{i,i''}F^m_{i,i''+1} \exp\left\{\sum_{w=i}^{i'-1} \Delta T_w \sum_{k\in\mathcal{K}} \sigma_{m,i'',k,w}\sigma_{m,i''+1,k,w}\right\}.$$

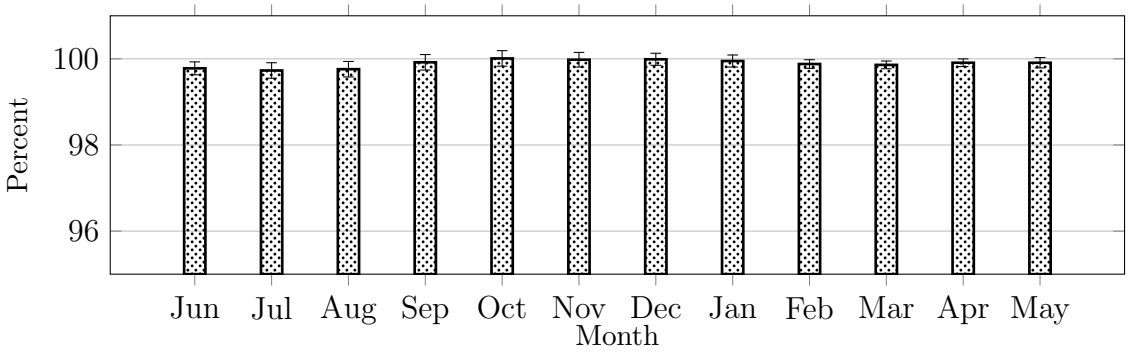


Figure 4.10: Comparison of the estimated ELSM-based lower bounds as percentages of the estimated ELSM-based dual upper bounds.

We estimate the ELSM value function approximation using 10,000 regression forward curve samples ($H = 10{,}000$). We use this value function approximation to estimate greedy lower bounds and dual upper bounds, the latter based on the penalties (4.26), on the time T_0 combined value of the storage and transport assets, $V_0(x_0, \boldsymbol{F}_0)$, by employing 30,000 simulation forward curve samples ($L = 30{,}000$). Figure 4.10 reports the lower bound estimates as percentages of the dual upper bound estimates. The error bars in this figure are standard errors, which are at most 0.15% of their respective estimates. The ELSM-based lower bound and dual upper bound estimates are essentially tight on all the instances. We refer to these lower and upper bound estimates as LBL and UBL, respectively. Chapter 3 uses its LSM approach, on which ELSM is based on, to estimate almost tight bounds

on the value of natural gas storage with access to a single market. Thus, despite the more complicated structure of the optimal policy when jointly managing storage and transport assets, ELSM continues to estimate effectively tight bounds.

We also estimate dual upper bounds on the linear dual penalties (4.27) and (4.28), which are not based on the ELSM value function approximation. We consider the linear dual penalties (4.27) specified for markets Z3 and ELA. We denote their respective dual upper bound estimates as UBZ3 and UBELA. We label the dual upper bound obtained with the average penalties (4.28) as UBA. Figure 4.11 displays these dual upper bounds as percentages of UBL along with their standard errors, both estimated on the same forward curve samples used to estimate UBL. These bounds are somewhat weaker than UBL, but perform quite well: They are respectively at most 0.86%, 0.65%, and 0.44% larger than UBL. UBA is tighter than UBZ3 and UBELA, with two exceptions. Linear dual penalties are thus effective in our application. This finding is consistent with a result of Secomandi (2012) obtained in the context of the merchant management of natural gas storage in a single market.

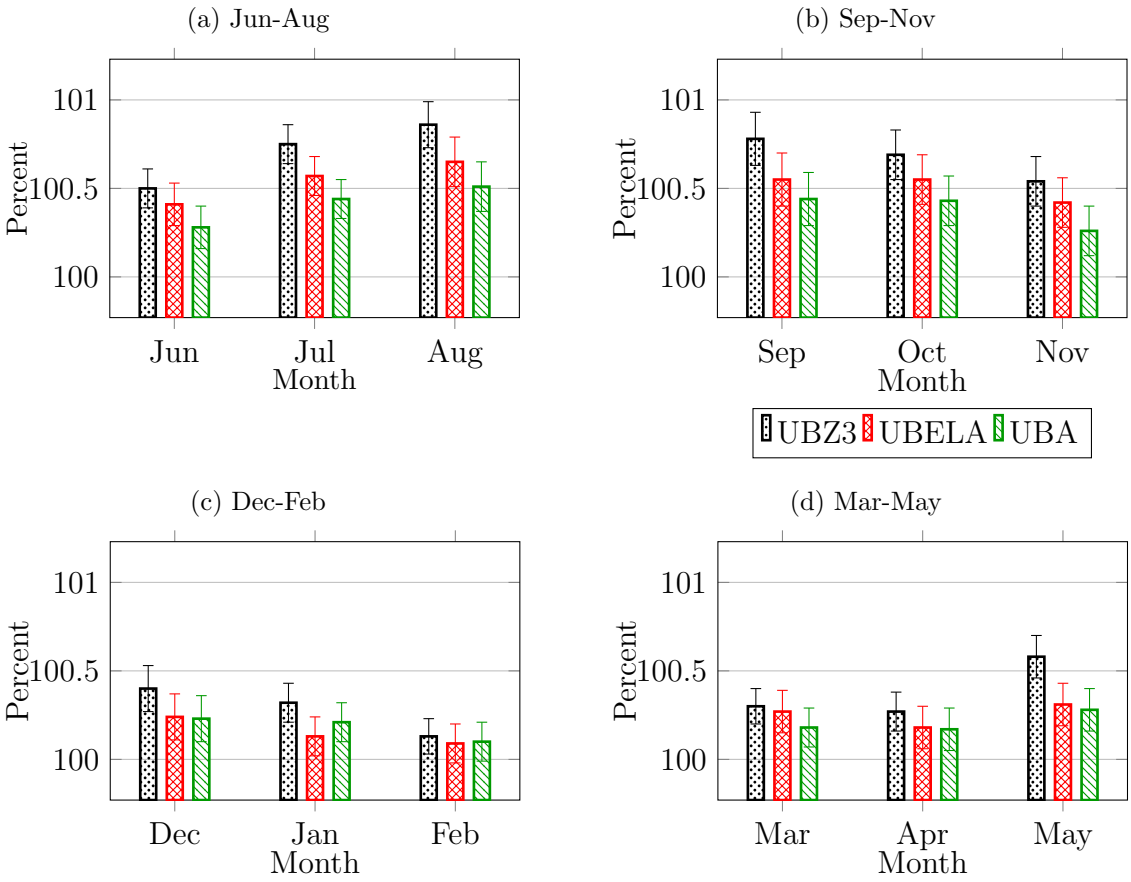


Figure 4.11: Comparison of the estimated dual upper bounds based on linear dual penalties as percentages of UBL.

Our computational setup is a 64 bits PowerEdge R515 with twelve AMD Opteron 4176

2.4GHz processors with 64GB of memory, the Linux Fedora 17 operating system, and the gcc version 4.7.2 20120921 (Red Hat 4.7.2-2) compiler. We use the LAPACK 3.X library with a single processor for ordinary least squares regression and Gurobi 5.0 (Gurobi Optimization, 2012) for solving linear programs. Estimating the value function approximations and the ELSM-based bounds takes 14 minutes on average across our instances. Roughly, 17%, 31%, and 52% of this time is used for estimating a value function approximation, a lower bound, and an upper bound, respectively. The CPU times required to estimate upper bounds using the penalties (4.27) and (4.28) are equal to the analogous CPU time when using the penalties (4.26) based on the ELSM value function approximation, which we assume is available from lower bound estimation.

4.7.2.2 Relevance of Jointly Managing Storage and Transport

We assess the relevance of the joint merchant management of storage and transport assets by comparing the value of the near optimal policy discussed in §4.7.2.1, LBL, against the value of a policy that manages these assets in an almost decoupled fashion and forces storage to have access to a single market. Specifically, (i) we estimate a value function approximation using the LSM method of Chapter 3, using the same basis functions of our ELSM implementation, by only considering storage trades at a given single market; and (ii) at each stage and state, we first use this value function approximation to determine a storage trade amount that is greedy with respect to this function, and then optimize the transport trade amounts using the residual receipt and delivery capacities at the market assigned to storage and the full receipt and delivery capacities at the remaining markets. By choosing the best storage to market assignment, we quantify the average value of this (almost) decoupled policy to be 89.60% of LBL on our instances. Thus, we estimate the incremental value of jointly managing the storage and transport assets to be 11.40%.

We provide some operational intuition for this benefit by analyzing the flow rates of these two policies. The flow rate of a given policy is the average amount of natural gas sold across all the stages and states visited in the Monte Carlo simulation used to estimate the value of this policy. The flow rate of the policy that jointly manages storage and transport, the ELSM-based policy, is 0.85. Compared to this flow rate, the flow rate of the almost decoupled policy is only 2.5% smaller, but its transport and storage components are 8.1% larger and 24.3% smaller than their respective ELSM-based policy counterparts. In other words, the flow rates of these policies are close to each other but the mix of their storage and transport trade amounts is substantially different.

The almost decoupled policy allows for some interaction between the management of storage and transport assets, because the transport assets can use the capacity not utilized by the storage asset. The value of the best completely decoupled policy, which ignores this interaction, is a meager 52.07% of LBL. There is thus substantial benefit from even partially integrating the merchant management of the storage and transport assets.

4.7.2.3 Effectiveness of Sequential Policies

We now investigate the possibility of near optimally simplifying the joint merchant management of the storage and transport assets by considering sequential policies. A sequential policy prioritizes one of the storage and transport decisions over the other. Such a policy is optimal if and only if there is no substitution between storage and transport activities (a simple corollary of Proposition 11). The findings discussed in §4.7.2.2 suggest that there is substantial substitution between these activities on our instances. Indeed, our estimates of the average values, across all our instances, of the values of the transport only and storage only policies, $V^{TR}(x_0, \boldsymbol{F}_0)$ and $V^{ST}(x_0, \boldsymbol{F}_0)$, are 74.95% and 55.47% of LBL, respectively. This substantial substitution suggests that the sequential policies might not fair well. Nonetheless, we now evaluate the performance of these policies.

We first consider a sequential policy that gives preference to transport over storage. Specifically, at a given stage and state this policy first optimizes the transport trade amounts and then optimizes the storage trade amounts on the residual capacity based on a value function approximation estimated using an obvious ELSM modification. The suboptimality of this policy is substantial, its value being on average 78.17% of LBL. Similar to the almost decoupled policy, the flow rate of this policy is a mere 1.2% smaller than the flow rate of the ELSM-based policy, but their mix of storage and transport trade amounts are very different: The average transport and storage flow rates of this sequential policy are 32.0% larger and 68.4% smaller than the analogous flow rates of the ELSM-based policy.

We now consider a policy that prioritizes storage over transport. We implemented this policy using a modification of the ELSM method analogous to the one we used to implement the other sequential policy. This alternative sequential policy performs near optimally, achieving 98.63% of LBL on average across our instances. Compared to the ELSM-based policy, the flow rate of this policy is 5.5% smaller *and* its transport and storage flow rate components are only 2.6% and 11.6% smaller, respectively; that is, these figures suggest that these policies have a similar mix of trades.

Our findings indicate that obtaining a near optimal sequential policy depends critically on how storage and transport are prioritized. Moreover, the near optimal sequential policy that gives priority to storage over transport is simpler to compute than the policy that does not prioritize these activities because in every stage and state it solves a single linear program rather than several such programs.

4.7.2.4 Value of Price Uncertainty and Performance of the Extended Rolling Intrinsic Policy

The intrinsic policy solves the deterministic version of the MDP (4.6) that only depends on the time T_0 array of forward curves $\boldsymbol{F}_0$. The value of this policy is referred to as the intrinsic value and denoted by $V_0^I(y_0; \boldsymbol{F}_0)$. The intrinsic value can be computed by solving the following deterministic version of SDP (4.18):

$$V_i^I(y_i; \boldsymbol{F}_0) = \max_{y_{i+1} \in \mathcal{Z}'(y_i)} \bar{r}(y_i - y_{i+1}, \boldsymbol{F}_{0,i}) + \delta V_{i+1}^I(y_{i+1}; \boldsymbol{F}_0), \tag{4.30}$$

$\forall (i, y_i) \in \mathcal{I} \times \mathcal{Y}'$, with boundary conditions $V_N^I(y_N; \boldsymbol{F}_0) := 0$, $\forall y_N \in \mathcal{Y}'$ – model (4.30) can also be equivalently formulated as a linear program.

The intrinsic value excludes the value due to price uncertainty, known as the extrinsic value. This value is the difference $V_0(y_0, \boldsymbol{F}_0) - V_0^I(y_0; \boldsymbol{F}_0)$ and measures how much can be gained by adapting the operating policy to the uncertain evolution of the array of forward curves. We estimate the extrinsic value on our instances by subtracting $V_0^I(y_0; F_0)$ from LBL. The estimated extrinsic values range between 7.17% and 14.47% of LBL across our instances (the standard errors of these estimates are at most 0.2% of their respective values of LBL). The average of these estimates is 10.41%. Although substantial, this average is smaller than both the 27.93% and 21% average extrinsic values estimated by Lai et al. (2010) and Secomandi (2010), respectively, in the case of storage operating in a single market and the 64.51% estimated by Secomandi and Wang (2012) in the case of transport without storage. Explaining these differences is difficult because our instances (i) couple storage and transport and (ii) use different parameters than the ones considered by these authors.

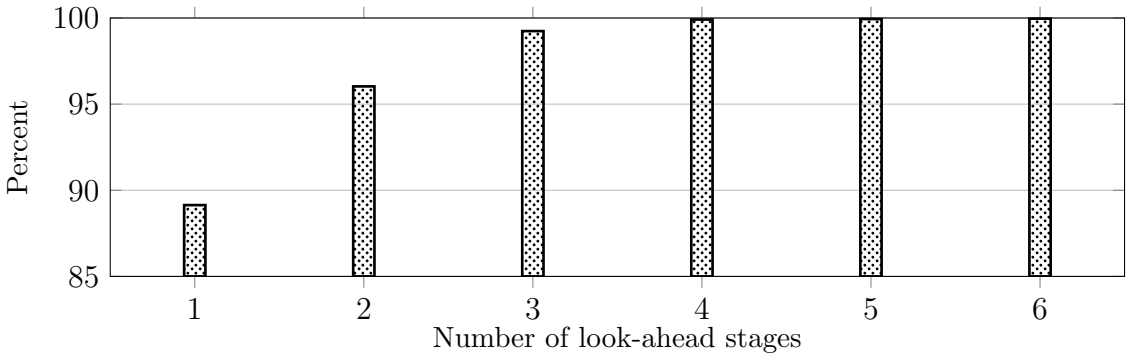


Figure 4.12: Comparison of the average estimated lower bounds corresponding to the limited look-ahead extended rolling intrinsic policy as percentages of UBL.

When managing storage in a single market setting, a common approach among practitioners to capture the extrinsic value of storage is to reoptimize the intrinsic policy at every stage and observed state (Gray and Khandelwal, 2004, Breslin et al., 2009, Lai et al., 2010, Secomandi, 2010, Secomandi et al., 2012). The policy generated by this approach is referred to as the rolling intrinsic policy and is known to be near optimal. We analyze a rolling intrinsic policy extended for jointly managing storage and transport in a multiple market setting by reoptimizing the deterministic dynamic program (4.30) at every stage and state. Specifically, in stage i and state $(y_i, \boldsymbol{F}_i)$ this dynamic program depends on $\boldsymbol{F}_i$ rather than $\boldsymbol{F}_0$. This extended rolling intrinsic policy is part of the commercial software StoragePLUS developed by Financial Engineering Associates, Inc. (FEA, 2013). We find that this policy performs near optimally on our instances, its estimated value being within 1 standard error of UBL on each instance. This finding provides support for using this policy in practice.

Despite the equivalence of LBL and the estimated value of the extended rolling intrinsic policy (recall that LBL is near optimal), estimating the latter lower bound requires at

least 2 orders of magnitude more CPU time than the former. We thus investigate ways to reduce this computational effort. In particular, this computational burden can be reduced by reoptimizing a limited look-ahead version of the intrinsic model (4.30). Figure 4.12 plots as percentages of UBL the average estimated lower bounds corresponding to different choices of the number of look-ahead stages. Near optimal performance is achieved with a look-ahead of as few as 4 stages. This simplified approach leads to computational savings of about 1 order of magnitude relative to the extended rolling intrinsic policy implemented with a full look-ahead, a finding that has practical relevance. However, estimating a lower bound on the combined value of storage and transport even with this limited look-ahead rolling intrinsic policy remains substantially slower compared to the ELSM approach. Moreover, estimating an upper bound on this value based on the discussed linear penalty is appealing when using the extended rolling intrinsic approach, because instantiating these dual penalties does not require the computation of a value function approximation.

4.7.2.5 Value of Transport Trading Across Multiple Pipelines

Our instances are based on three different pipelines (see Figure 4.6). We now assess the value of inter-pipeline transport trading, that is, the value of wheeling (see §4.3; we do not estimate the value of wheeling across multiple pipelines using storage because estimating this value requires keeping track of the injection source of the inventory in storage, which is challenging to do). We do this by considering a policy that restricts transport trades to occur within zones of a single pipeline. This policy has no trades involving AGT, because AGT is a single zone, and ignores all the contemporaneous transfers of natural gas between TETCO and TRANSCO. However, transfer of natural gas between TETCO and TRANSCO is still possible over time using storage. Comparing the value of this restricted policy, we find that the average value of wheeling is 36.45% of LBL on our instances. Allowing transport trades between AGT and TETCO alone reduces this value to 28.81%. These figures suggest that transport trading between TETCO and TRANSCO has substantially more value than doing the same between TETCO and AGT. This finding reflects the smaller AGT capacity compared to TRANSCO and the fact that natural gas shipped from TRANSCO to AGT must go through TETCO.

4.8 Conclusion

We investigate the joint merchant management of natural gas storage and transport assets, thus extending the current literature that studies the management of these assets in isolation. We model the management of these assets as an MDP and characterize the structure of the value function and an optimal policy of this model. Because computing such a policy is intractable, we leverage this structural analysis to extend an existing storage LSM method to compute a near optimal policy for this MDP. We apply our method to realistic instances and find that (i) the joint, rather than decoupled, merchant management of storage and transport assets has substantial value; (ii) this management can be nearly optimally simplified by prioritizing storage relative to transport, despite the substitutability

between these activities being considerable; (iii) the value due to price uncertainty is large but can be almost entirely captured by sequentially reoptimizing a deterministic version of our MDP, an approach included in existing commercial software; and (iv) the value of transport trading across different pipelines is substantial. Our research has potential relevance beyond the specific application considered in this chapter.

Chapter 5

A Critical Review of Least Squares and Math Programming Based ADP Methods

(Joint work with François Margot and Nicola Secomandi)

5.1 Introduction

Finite horizon and discrete time stochastic optimization problems arise commonly in stochastic control applications, including energy real options such as commodity and energy conversion assets (Lai et al., 2010, Boogert and De Jong, 2011/12, Devalkar et al., 2011, Powell et al., 2012b, Thompson, 2013). Solving these problems directly is typically difficult because their deterministic versions may be mathematical programs containing non convex features such as integer variables. A popular approach is therefore to reformulate dynamic stochastic optimization problems as stochastic dynamic programs (SDPs). In theory, SDPs can be solved recursively in the finite horizon and discrete time setting, and using a (possibly infinite) linear program in the general case. These exact approaches are typically not practical due to the well known curses of dimensionality caused by high dimensional state and action spaces and expectations that are potentially challenging to compute (Powell 2011, §4.1, and Chapter 2 of this thesis).

An active area of research focuses on approximate methods for solving high dimensional SDPs. This area is broadly referred to as approximate dynamic programming (ADP). A prevalent idea in ADP is to compute an approximation to the value function of the SDP and use this approximation together with the Bellman operator to obtain feasible actions. We consider ADP methods based on least Squares Monte Carlo (LSM) and math programming that rely on this idea of computing a value function approximation.

Least squares Monte Carlo is a popular ADP approach for managing energy real options in that literature. Examples include the popular Longstaff and Schwartz (2001) method and recent variants such as the ones from Gyurko et al. (2011) and Desai et al. (2012b) and

LSMV of Chapter 3. The popularity of these methods can be attributed to their simplicity and excellent practical performance. We focus in this chapter on LSMV.

Math programming based ADP methods are known for their strong theoretical guarantees. The standard approach in this literature computes a value function approximation by solving a linear program referred to as an approximate linear program (ALP; Schweitzer and Seidmann 1985, de Farias and Van Roy 2003). Recent research has focused on improving ALP by solving math programs obtained by relaxing the ALP constraints. The smoothed ALP of Desai et al. (2012a) and the constrained-based and multiplier-based ALP relaxations in Chapter 2 use ALP relaxations to improve the ALP value function approximation. The iterated Bellman linear program due to Wang and Boyd (2010) relaxes ALP to improve its upper bounds on the exact value function and the optimal policy value. Different from these linear programming based methods, Petrik (2012) develops a mixed integer programming based ADP approach referred to as distributionally robust ADP.

We critically review the above mentioned ADP methods. We classify them based on the objectives driving their computation of a value function approximation: (i) minimize value function approximation error; and (ii) minimize greedy policy loss. Value function approximation error is the difference between the exact value function and the value function approximation under an appropriate norm. Greedy policy loss is the difference between the optimal policy value and the value of the greedy policy induced by the value function approximation.

LSMV, ALP, and the four discussed ALP relaxations minimize value function approximation error. We unify these methods by showing that they can all be derived using the ALP relaxation framework of Chapter 2, which is based on restricting the ALP dual. In particular, we bridge the financial-engineering based LSMV method and the constraint-based ALP relaxations by showing that they are special cases of a family of ALP relaxations. This ALP relaxation is parameterized by distributions used for constraint-aggregation that can be viewed as basis functions approximating the dual variables. We also specify restrictions of the ALP dual to obtain the multiplier-based ALP relaxations, the smoothed ALP, and the iterated Bellman linear program. Our ALP-dual-based derivation of the iterated Bellman linear program shows that a parameter choice commonly used in finite horizon and discrete time problems makes its upper bound on the optimal policy value equal to the one from ALP.

Distributionally robust ADP (Petrik, 2012) approximately minimizes policy loss by minimizing an upper bound on this quantity. Using this approach entails solving large mixed integer programs, which is challenging. We develop a new mixed integer programming formulation that *exactly* minimizes the greedy policy loss for a class of structured SDPs arising in energy real options applications. The size of our formulation is comparable to the mixed integer program of Petrik (2012) and is thus challenging to solve, but it guarantees the best value function approximation in terms of greedy policy loss. We numerically test our mixed integer program on 6 month versions of the 24 month natural gas storage option instances discussed in §3.7.2 of Chapter 3. We warm start our mixed integer program using the LSMV value function approximation, which we know is essentially optimal in terms of policy loss, and check if its near optimality can be proven. We

find that the mixed integer program proves the optimality of the LSMV value function approximation on roughly 42% of the instances and marginally improves the LSMV lower bound estimate on the samples used to compute the value function approximation. These results motivate further research into advanced techniques for solving this mixed integer program more efficiently.

The remainder of this chapter is organized as follows. In §5.2 we formulate our SDP and discuss exact solution approaches. In §5.3 we define value function approximations and greedy operating policies. In §5.4 we present ADP methods that minimize value function approximation error and unify them in §5.5. We describe the distributionally robust ADP approach in §5.6 and our mixed integer program in §5.7. We conclude in §5.8.

5.2 SDP Formulation and Exact Solution

We consider solving a finite horizon discrete time stochastic optimization problem with N stages belonging to the set $\mathcal{I} := \{0, \ldots, N-1\}$ and indexed by i. The stage i state space is denoted by $\mathcal{Y}_i$ and the initial state is y_0, that is, $\mathcal{Y}_0 := \{y_0\}$. The set $\mathcal{A}_i(y_i)$ includes the feasible actions in stage i and state y_i. We assume that the state and action spaces are finite. A feasible action a_i at a state y_i results in an immediate reward and a state transition to a stage $i+1$ state y_{i+1} with probability $\Pr(y_{i+1}|i, y_i, a_i)$. We represent the stage i immediate reward for each state-action pair by a function $r_i(y_i, a_i)$. The transition kernel when taking action a_i at stage i and state y_i is expectation $\mathbb{E}_{i,y_i,a_i}$ defined by probabilities $\Pr(y_{i+1}|i, y_i, a_i), \forall y_{i+1}$. Rewards at states in future stages are deflated using a discount factor $\delta \in (0, 1)$. Let Π define the set of all policies and $\pi \in \Pi$ a policy with its action at stage i and state y_i denoted by $a_i^\pi(y_i)$. The value of a policy is defined as

$$\rho(\pi) := \sum_{i \in \mathcal{I}} \delta^i \mathbb{E}[r(y_i^\pi, a_i^\pi(y_i^\pi))],$$

where $\mathbb{E}$ is expectation under policy π starting from the initial state y_0 and y_i^π is the random state at stage i under policy π. The stochastic optimization problem is

$$\max_{\pi \in \Pi} \rho(\pi).$$

In theory, an optimal policy π^* can be obtained by reformulating the above problem as an SDP. Denoting by $V_i \in \mathbb{R}^{|\mathcal{Y}_i|}$ the stage i value function of the SDP, we write its associated recursion as

$$V_i(y_i) = \max_{a_i \in \mathcal{A}_i(y_i)} r_i(y_i, a_i) + \delta \mathbb{E}_{i,y_i,a_i} V_{i+1}, \quad \forall (i, y_i) \in \mathcal{I} \times \mathcal{Y}_i, \tag{5.1}$$

where we use the vector product $P_{i,y_i,a_i} V_{i+1}$ instead of $\sum_{y_{i+1}} \Pr(y_{i+1}|i, y_i, a_i) V_{i+1}(y_{i+1})$. Similarly, we also replace sums by vector products in other parts of this chapter. For ease of notation, in the rest of this chapter we write (i, y_i, a_i) instead of $(i, y_i, a_i) \in \mathcal{I} \times \mathcal{Y}_i \times \mathcal{A}_i(y_i)$. We write $(\cdot)_{-(i)}$ to indicate that i is excluded from $\mathcal{I}$ in the tuple ground set.

The value function of SDP (5.1) can also be computed by reformulating it as an equivalent linear program (Manne, 1960). Defining the nonnegative vectors $\{c_i \in \mathbb{R}_+^{|\mathcal{Y}_i|}, \forall i\}$ with $c_0(y_0) > 0$, this linear program is

$$\min_{\overline{V}} \sum_i c_i^\intercal \overline{V}_i \tag{5.2}$$

$$\text{s.t. } \overline{V}_{N-1}(y_{N-1}) \geq r_{N-1}(y_{N-1}, a_{N-1}), \quad \forall (y_{N-1}, a_{N-1}), \tag{5.3}$$

$$\overline{V}_i(y_i) \geq r_i(y_i, a_i) + \delta \mathbb{E}_{i,y_i,a_i} \overline{V}_{i+1}, \quad \forall (i, y_i, a_i)_{-(N-1)}. \tag{5.4}$$

We refer to the linear program (5.2)-(5.4) as PLP (P and LP abbreviate primal and linear program, respectively). The PLP variables are $\overline{V}_i(y_i)$, $\forall (i, y_i)$. The objective function (5.2) minimizes the weighted sum of these variables over all stages and states, using the vectors c_i's as weights. Ignoring the difference in notation between the variables of PLP and the value function of SDP (5.1), the constraints of PLP can be obtained from this SDP as follows: For each pair (i, y_i) express the maximization over the set $\mathcal{A}_i(y_i)$ in (5.1) as $|\mathcal{A}_i(y_i)|$ inequalities, one for each a_i. At optimality, the values of the PLP variables match the SDP value function in every stage and state that is reachable starting from a state where $c_i(y_i) > 0$. (This can be easily verified using complementary slackness).

The dual of PLP, denoted DLP (D abbreviates dual), is

$$\max_u \sum_i r_i^\intercal u_i \tag{5.5}$$

$$\text{s.t. } \sum_{a_0} u_0(y_0, a_0) = c_0(y_0), \tag{5.6}$$

$$\sum_{a_i} u_i(y_i, a_i) = c_i(y_i) + \delta \sum_{(y'_{i-1}, a_{i-1})} \Pr(y_i | y'_{i-1}, a_{i-1}) u_{i-1}(y_{i-1}, a_{i-1}), \quad \forall (i, y_i)_{-(0)}, \tag{5.7}$$

$$u_i(y_i, a_i) \geq 0, \forall (i, y_i, a_i). \tag{5.8}$$

The value of the dual variable $u_i(y_i, a_i)$ in a DLP feasible solution can be interpreted as the frequency of visiting stage i and state y_i and taking action a_i. The objective function (5.5) for a feasible DLP solution $u := \{u_i(y_i, a_i), \forall (i, y_i, a_i)\}$ is the weighted sum of the immediate rewards using these frequencies as weights. The constraint (5.6) ensures that the frequency of visiting the initial state in the initial stage is $c_0(y_0)$. Constraints (5.7) are discounted flow balance constraints. Constraints (5.8) impose nonnegativity conditions on the decision variables. Property 2 states an easily verifiable condition and another well known property of DLP basic feasible solutions (see Theorem 6.9.1 in Puterman 1994, page 224).

Property 2. *Any feasible DLP solution u satisfies*

$$\sum_{(y_i, a_i)} u_i(y_i, a_i) = \sum_{j=0}^{i} \delta^{i-j} \sum_{y_j} c_j(y_j), \forall i. \tag{5.9}$$

Moreover, every basic feasible DLP solution u is such that for any stage-and-state pair

(i, y_i) there exists exactly one action a_i for which $u_i(y_i, a_i) > 0$.

5.3 Value Function Approximations and Greedy Policies

Solving the SDP recursion (5.1), PLP, or DLP is intractable in many important applications due to the curse of dimensionality. ADP provides a wide array of techniques to overcome this issue when solving the exact models discussed in §5.2 (see Bertsekas 2007 and Powell 2011). A common approach in this literature is to compute a tractable approximation to the value function V_i of SDP (5.1) that is a linear parameterization of a set of *basis* functions. The b-th basis function at stage i, $\phi_{i,b} : \mathcal{Y}_i \mapsto \mathbb{R}$, is a mapping from the stage i state space to the real line. We denote the set of stage i basis functions by $\Phi_i = \{\phi_{i,b}, b = 1, \ldots, B_i\}$, where B_i is a positive integer. Given weights $\beta_{i,b}$, $\forall b$, the stage i value function approximation is defined as $\sum_{b=1}^{B_i} \phi_{i,b}(y_i)\beta_{i,b}$, which we abbreviate to $(\Phi_i\beta_i)(y_i)$. Methods to compute the weight vector $\beta_i := (\phi_{i,b}, b = 1, \ldots, B_i)$ are discussed in §5.4, §5.6, and §5.7. A widespread measure of the quality of a value function approximation is its deviation from the exact value function under a weighted 1-norm. This is referred to as the value function approximation error and is defined as

$$\|V - (\Phi\beta)\|_{1,\mu} := \sum_{(i,y_i)} \mu_i(y_i)|V_i(y_i) - (\Phi_i\beta_i)(y_i)|, \tag{5.10}$$

where $\mu_i(y_i) \geq 0$, $\forall (i, y_i)$.

Given a value function approximation, the Bellman operator associated with the SDP (5.1) can be used to compute a feasible action at every stage i and state y_i by solving the following optimization problem over feasible actions:

$$\underset{a_i \in \mathcal{A}_i(y_i)}{\operatorname{argmax}}\, r_i(y_i, a_i) + \delta \mathbb{E}_{i,y_i,a_i}(\Phi_{i+1}\beta_{i+1}). \tag{5.11}$$

Thus, a value function approximation implicitly defines a feasible policy through (5.11). This policy is referred to as a *greedy* policy in the ADP literature. Denoting a greedy policy given the weights $\beta := (\beta_i, i \in \mathcal{I})$ as $\pi^g(\beta)$, an alternate measure of quality to the value function approximation error is the greedy policy loss

$$\rho(\pi^*) - \rho(\pi^g(\beta)). \tag{5.12}$$

5.4 ADP Methods for Minimizing VFA Error

In this section, we review ADP methods that compute a value function approximation by heuristically minimizing the value function approximation error (5.10). We discuss an LSM method in §5.4.1. The ALP approach is presented in §5.4.2. Various ALP relaxation approaches are described in §§5.4.3-5.4.5.

5.4.1 LSM

LSM methods are recursive procedures that are ubiquitous in financial engineering for approximating SDP (5.1) or its continuation value function formulation (see §3.2.3 of Chapter 3 for details of these two formulations). This approach was pioneered by Carriere (1996), Longstaff and Schwartz (2001), and Tsitsiklis and Van Roy (2001). Chapter 3 treats these methods in detail.

We focus in this chapter on our LSM variant, LSMV, described in §3.5 and outlined in Algorithm 3 on Page 55. To facilitate our unification results in §5.5 we rewrite this algorithm succinctly as follows. Let $\mathcal{T}_{i,y_i}$ represent the stage i Bellman operator given a set of basis functions, that is,

$$\mathcal{T}_{i,y_i}(\Phi_{i+1}\beta_{i+1}) := \max_{a_i \in \mathcal{A}_i(y_i)} r_i(y_i, a_i) + \delta \mathbb{E}_{i,y_i,a_i}(\Phi_{i+1}\beta_{i+1}).$$

We will drop the index y_i in this operator when using it in vector form. Consistent with Algorithm 3, let $\bar{\beta}_N \equiv 0$. Starting from stage $N-1$ and moving back to stage 0, LSMV is a recursive procedure that computes the weights $\bar{\beta}_i$ by solving

$$\bar{\beta}_i := \min_{\beta_i \in \mathbb{R}^{B_i}} \|(\Phi_i \beta_i) - \mathcal{T}_i(\Phi_{i+1}\bar{\beta}_{i+1})\|_2, \tag{5.13}$$

where $\|\cdot\|_2$ represents the 2-norm.

5.4.2 ALP

Schweitzer and Seidmann (1985) and de Farias and Van Roy (2003) proposed a linear program, ALP, to compute a value function approximation. The ALP is

$$\min_{\beta} \sum_i c_i^{\intercal}(\Phi_i \beta_i) \tag{5.14}$$

$$\text{s.t. } (\Phi_{N-1}\beta_{N-1})(y_{N-1}) \geq r(y_{N-1}, a_{N-1}), \forall (y_{N-1}, a_{N-1}), \tag{5.15}$$

$$(\Phi_i \beta_i)(y_i) \geq r(y_i, a_i) + \delta \mathbb{E}_{i,y_i,a_i}(\Phi_{i+1}\beta_{i+1}), \forall (i, y_i, a_i)_{-(N-1)}. \tag{5.16}$$

The objective function (5.14) minimizes a weighted combination of the value function approximation at each stage and state using the weights $c_i(y_i)$. For this reason, the vectors c_i, $\forall i$, are referred to as state-relevance weights. Without loss of generality we assume that $\sum_{i,y_i} c_i(y_i) = 1$. Constraints (5.15)-(5.16) approximate the SDP recursion and can be viewed as a linearization of the convex constraints

$$(\Phi_i \beta_i) \geq \mathcal{T}_i(\Phi_{i+1}\beta_{i+1}), \forall i, \tag{5.17}$$

which can be derived from (5.1) by replacing V_i by $(\Phi_i \beta_i)$ and changing the equalities to inequalities. Constraints (5.17) are referred to as *Bellman inequalities* (see, e.g., Wang and Boyd 2010). We make the following standard assumption on the basis functions to ensure the feasibility of ALP (de Farias and Van Roy, 2003):

Assumption 4. $\phi_{i,1}(y_i) := 1,\ \forall (i, y_i)$.

Because the Bellman operator is monotonic and a contraction mapping (Puterman, 1994), a value function approximation satisfying constraints (5.17) is guaranteed to be a pointwise upper bound on the exact value function, that is,

$$(\Phi_i \beta_i) \geq \mathcal{T}_i(\Phi_{i+1}\beta_{i+1}) \geq \mathcal{T}_i^2(\Phi_{i+1}\beta_{i+1}) \geq \ldots \geq V_i, \forall i. \tag{5.18}$$

In particular, since the exact value function at the initial stage and state coincides with the optimal policy value, these inequalities establish that the ALP value function approximation at the initial stage and state provides an upper bound on the value of an optimal policy. This upper bound can be combined with the greedy policy lower bound, also computed using the ALP value function approximation, to obtain a performance guarantee on the greedy policy.

The dual of ALP, denoted DALP, is

$$\max_{w} \sum_{i} r_i^{\intercal} w_i \tag{5.19}$$

$$\text{s.t.} \quad \sum_{a_0} w_0(y_0, a_0) = c_0(y_0), \tag{5.20}$$

$$\sum_{y_i} \phi_{i,b}(y_i) \sum_{a_i} w_i(y_i, a_i) = \sum_{y_i} \phi_{i,b}(y_i) \left[c_i(y_i) + \delta \sum_{(y_{i-1}, a_{i-1})} \Pr(y_i | y_{i-1}, a_{i-1}) w_{i-1}(y_{i-1}, a_{i-1}) \right], \forall (i, b)_{-(0)}, \tag{5.21}$$

$$w_i(y_i, a_i) \geq 0, \forall (i, y_i, a_i). \tag{5.22}$$

The DALP variables can be interpreted as the frequencies at which a state is visited and an action is taken by some policy, possibly infeasible. The DALP objective function is the sum of returns weighted by the DALP state-action frequency vector. Constraint (5.20) is identical to the DLP constraint (5.6), while constraints (5.21) at stage i are B_i different aggregations of the DLP flow balance constraints (5.7) using weights defined by each stage i basis function. Constraints (5.22) enforce nonnegativity. This relaxation breaks flow balance at each stage, which was present in DLP. Nevertheless, the aggregate flow at each stage is conserved as stated in Property 3, which can be easily verified using the DALP constraints corresponding to $b = 1$ under Assumption 4.

Property 3. *Any feasible DALP solution w satisfies*

$$\sum_{(y_i, a_i)} w_i(y_i, a_i) = \sum_{j=0}^{i} \delta^{i-j} \sum_{y_j} c_j(y_j), \forall i. \tag{5.23}$$

Solving ALP (or DALP) can be potentially challenging because it has a large number of constraints. However, since ALP has a manageable number of variables, it may be possible

to employ a row-generation scheme to solve it exactly or employ a column-generation heuristic to solve DALP (Adelman, 2003, 2004, 2007). Alternatively, sampling schemes can be used to solve approximate versions of ALP (de Farias and Van Roy, 2003, 2004). In other words, ALP is constructed using the sampled state sets $\hat{\mathcal{Y}}_i \subseteq \mathcal{Y}_i$, $\forall i$, where the cardinality of $|\hat{\mathcal{Y}}_i|$ is chosen to be manageable at each stage i.

5.4.3 Smoothed ALP

Desai et al. (2012a) observe that the value function approximation from ALP can be potentially poor because it must be an upper bound on the exact value function at each stage and state (see (5.18)). These authors relax this condition by solving a linear program with constraints that relax the ALP constraints. In other words, feasible solutions of this ALP relaxation can violate the ALP constraints (5.17) with the goal of obtaining a value function approximation that potentially approximates better the exact value function compared to the ALP value function approximation. The ALP relaxation model of Desai et al. (2012a) is referred to as smoothed ALP (SALP).

We now discuss SALP. Let $\Pr^{\pi^*}(i, y_i|j, y_j)$ be the probability of reaching state y_i from state y_j, $i > j$, under an optimal policy π^*. Define

$$\Pr^{\pi^*,c,\delta}(i, y_i) := \sum_{j=0}^{i} \delta^{i-j} \sum_{y_j} \Pr^{\pi^*}(i, y_i|j, y_j) c_j(y_j).$$

The term $\Pr^{\pi^*,c,\delta}(i, y_i)$ can be interpreted as the discounted probability of visiting state y_i at stage i assuming that the initial distribution of states is given by the probability mass function defined by $c_i(y_i)$, $\forall(i, y_i)$. Using these probabilities, SALP is

$$\min_{\beta} \sum_{i} c_i^{\intercal}(\Phi_i \beta_i) + d^{\intercal} \Pr^{*,c,\delta} \tag{5.24}$$

$$\text{s.t. } (\Phi_{N-1}\beta_{N-1})(y_{N-1}) \geq r_i(y_{N-1}, a_{N-1}), \forall(y_{N-1}, a_{N-1}), \tag{5.25}$$

$$(\Phi_i \beta_i)(y_i) + d_i(y_i) \geq r_i(y_i, a_i) + \delta \mathbb{E}_{i,y_i,a_i}(\Phi_{i+1}\beta_{i+1}), \forall(i, y_i, a_i)_{-(N-1)}, \tag{5.26}$$

$$d_i(y_i) \geq 0, \forall(i, y_i)_{-(N-1)}. \tag{5.27}$$

SALP differs from ALP (5.14)-(5.16) in two ways: (i) a slack variable $d_i(y_i)$ is present in the constraints (5.16) and allows their violations, and (ii) the amount of violation is controlled by a new objective function term that penalizes a weighted combination of these slack variables. The value function of SALP may not provide an upper bound on the exact value function or the optimal policy value.

To operationalize SALP, the conceptual probabilities $\Pr^{\pi^*,c,\delta}(\cdot, \cdot)$ in its objective function (5.24) need to be approximated. After such an approximation, solving SALP is potentially challenging because it has a large number of new variables, one for every state. Hence, in contrast to ALP, SALP has both a large number of variables and constraints. The presence of a large number of variables precludes the use of row generation for its solution. Thus, obtaining a value function approximation in practice necessitates solving

a sampled approximation of SALP constructed by replacing $\mathcal{Y}_i$ by a sampled set of states $\hat{\mathcal{Y}}_i$ at each stage i. Such sampled versions of SALP can be solved efficiently using a barrier method – with the same computational complexity per iteration as solving an analogous sampled version of ALP but with a sublinear increase in the number of iterations – because of the sparsity of the constraint matrix associated with (5.26) and the slack variables (see §5.1 of Desai et al. 2012a for details).

5.4.4 Iterated Bellman Linear Program

Wang and Boyd (2010), O'Donoghue et al. (2011), and O'Donoghue et al. (2013) propose an ALP relaxation to compute a value function approximation that we refer to as the iterated Bellman linear program. Similar to SALP, this linear program relaxes the ALP constraints, but, despite this relaxation, its value function approximation remains an upper bound on the exact value function at all stages and states. In other words, the value function approximation from the iterated Bellman linear program may violate (5.17) but is guaranteed to satisfy $(\Phi_i \beta_i) \geq V_i$.

The key idea behind relaxing the right hand side of (5.17) is to replace $\mathcal{T}_i(\Phi_{i+1}\beta_{i+1})$ by a function f_i that satisfies $\mathcal{T}_i(\Phi_{i+1}\beta_{i+1}) \geq f_i \geq V_i$. As shown in (5.18), such a function can be obtained by iteratively applying the Bellman operator to the right hand side of (5.17). For instance applying the Bellman operator K times results in the following iterated Bellman inequalities:

$$(\Phi_i \beta_i) \geq \mathcal{T}_i^k(\Phi_{i+1}\beta_{i+1}), \forall i. \tag{5.28}$$

Constraints (5.28) are convex but their right hand side is difficult to evaluate. Moreover, linearizing these constraints requires the addition of a large number of new variables.

To overcome these issues, Wang and Boyd (2010) propose a system of Bellman inequalities such that any value function approximation satisfying this system is bounded below by the right hand side of an iterated Bellman inequality such as (5.28). We now discuss this system of Bellman inequalities, in particular, we use the most recent variant in O'Donoghue et al. (2013). Let $\beta_{j,i}$, $\forall i$ and $j = 0, 1, \ldots, J$, denote the weights of $J+1$ different value function approximations, each constructed using the basis functions in set $\Phi := (\Phi_i, i \in \mathcal{I})$. These value function approximations are used to define the convex constraint system

$$(\Phi_i \beta_{j,i}) \geq \mathcal{T}_i(\Phi_{i+1}\beta_{j+1,i+1}), \forall (i,j), \tag{5.29}$$

$$\beta_{J,i} = \beta_{J+1,i}, \forall i. \tag{5.30}$$

The constraints (5.29) corresponding to stage i are $J+1$ Bellman inequalities. The j-th stage i Bellman inequality differs from the corresponding ALP Bellman inequality (5.17) because the value function approximations on the left and right hand sides of (5.29) are different. In particular, the j-th value function approximation appears on the left hand side of (5.29) while the $j+1$-st value function approximation appears on its right hand side. Chaining these inequalities for increasing values of j establishes that the following

iterated Bellman inequalities are implied by (5.29):

$$(\Phi_i \beta_{0,i}) \geq \mathcal{T}_i^J(\Phi_{i+1}\beta_{J,i+1}), \forall i. \tag{5.31}$$

Further, using equalities (5.30) in the inequalities (5.29) for $j = J$ gives

$$(\Phi_i \beta_{J,i}) \geq \mathcal{T}_i(\Phi_{i+1}\beta_{J,i+1}), \forall i. \tag{5.32}$$

Finally, combining (5.31) and (5.32) shows that the value function approximation $(\Phi_i\beta_{0,i})$ is bounded below by the right hand side of an iterated version of the Bellman inequality (5.32).

To obtain the iterated Bellman linear program, O'Donoghue et al. (2013) linearize the convex system (5.29)-(5.30), and obtain

$$\min_{\beta} \sum_i c_i^\intercal (\Phi_i \beta_{0,i}) \tag{5.33}$$

$$\text{s.t. } (\Phi_{N-1}\beta_{j,N-1})(y_{N-1}) \geq r(y_{N-1}, a_{N-1}), \forall (y_{N-1}, a_{N-1}), j \in \{0, \ldots, J\}, \tag{5.34}$$

$$(\Phi_i \beta_{j,i})(y_i) \geq r(y_i, a_i) + \delta \mathbb{E}_{i,y_i,a_i}(\Phi_{i+1}\beta_{j+1,i+1}), \forall (i, y_i, a_i)_{-(N-1)}, j \in \{0, \ldots, J-1\}, \tag{5.35}$$

$$(\Phi_J \beta_{J,i})(y_i) \geq r(y_i, a_i) + \delta \mathbb{E}_{i,y_i,a_i}(\Phi_{i+1}\beta_{J,i+1}), \forall (i, y_i, a_i)_{-(N-1)}. \tag{5.36}$$

This linear program has J times the number of variables as ALP, but a potentially smaller number of new variables than SALP. It is also easy to see that (5.33)-(5.36) is a relaxation of ALP because adding the equalities $\beta_{j,i} = \beta_{j+1,i}$, $\forall i$ and $j = 0, \ldots, J$, makes this linear program equivalent to the ALP (5.14)-(5.16). Similar to ALP, (5.33)-(5.36) can be solved by using row generation or approximated using a tractable sampled version.

5.4.5 Constraint-based and Multiplier-based ALP Relaxations

Section 2.4 of Chapter 2 identifies a potential issue with ALP by comparing its dual, DALP, with the exact dual, DLP (see §5.2). In particular, Proposition 1 establishes that the set of DALP optimal solutions are potentially distorted compared to DLP optimal solutions. Because optimal DLP solutions are in one-to-one correspondence with optimal policies of SDP (5.1), this implies that DALP solutions are also inconsistent with the SDP optimal policies. In particular, DALP optimal solutions may all correspond to infeasible policies to the SDP. By complementary slackness, this inconsistency implies that the ALP value function approximation is determined by states visited by an infeasible policy.

To overcome this inconsistency, §2.5 proposes adding constraints to DALP enforcing properties of DLP optimal solutions in an exact or approximate manner. The primal linear program corresponding to this DALP restriction is an ALP relaxation and can be solved to compute a value function approximation. This approach is used to derive two classes of ALP relaxations when using look-up table value function approximations: constraint-based and multiplier-based relaxations. To facilitate our unification of methods in §5.5.2, we present these ALP relaxation classes here in the context of general basis functions. To be consistent with the treatment in Chapter 2, we make the following assumptions:

(A1) The SDP state y_i at each stage i can be partitioned into an endogenous component y_i^d and an exogenous component, y_i^r (the superscripts d and r denote deterministic and random, respectively). (Changes in the endogenous component are caused by the decision makers actions while the exogenous component evolves according to a given stochastic process that does not depend on these actions.)

(A2) Probabilities in set $\{\Pr(y_i^r|y_{i-1}^r), \forall y_i^r\}$ defining the probability mass function over the support of y_i^r given y_{i-1}^r are available. (Typically from a stochastic process such as (2.4)-(2.5) used in Chapter 2.)

(A3) State relevance weights are chosen to be $c_0(y_0) = 1$, and $c_i(y_i) = 0$, $\forall (i, y_i)_{-(0)}$.

Constraint-based ALP Relaxations. Let $y_i^{r,+}$ be a subset of the components of the exogenous state y_i^r and $y_i^{r,-}$ be its remaining elements. Given a generic probability mass function $p(y^{r,-}|y^{r,+}, y_i^d)$ (to be defined shortly), the constraints added to DALP are

$$\sum_{a_i} w_i(y_i, a_i) = p(y_i^{r,-}|y_i^d, y_i^{r,+})\theta_i(y_i^d, y_i^{r,+}), \forall (i, y_i), \tag{5.37}$$

where $\theta_i(y_i^d, y_i^{r,+})$ are new variables that will be introduced in DALP. The primal linear program corresponding to the DALP restriction with constraints (5.37) is

$$\min_{\beta} \sum_i c_i^\intercal(\Phi_i\beta_i) \tag{5.38}$$

$$\text{s.t. } (\Phi_{N-1}\beta_{N-1})(y_{N-1}) \geq r_{N-1}(y_{N-1}, a_{N-1}), \forall (y_{N-1}, a_{N-1}), \tag{5.39}$$

$$(\Phi_i\beta_i)(y_i) + d_i(y_i) \geq r_i(y_i, a_i) + \delta\mathbb{E}_{i,y_i,a_i}(\Phi_{i+1}\beta_{i+1}), \forall (i, y_i, a_i)_{-(N-1)}, \tag{5.40}$$

$$\sum_{y_i^{r,-}} p(y_i^{r,-}|y_i^d, y_i^{r,+})d_i(y_i^d, y_i^{r,+}, y_i^{r,-}) = 0, \forall (i, y_i^d, y_i^{r,+}). \tag{5.41}$$

By choosing $p(y_i^{r,-}|y_i^d, y_i^{r,+})$, the linear program (5.38)-(5.41) specifies different constraint-based ALP relaxations. Section 2.5 considers the following practical choices for these probabilities

$$\Pr(y_i^{r,-}|y_i^{r,+}),$$
$$\mathbb{1}(y_i^{r,-} = \mathbb{E}[y_i^{r,-}|y_i^{r,+}]).$$

Both these probability mass functions can be derived from the probability mass function that we assume is available from the exogenous information stochastic process, because they only depend on the exogenous part of the state.

Similar to SALP, in general, solving linear program (5.38)-(5.41) is more challenging than solving ALP and entails solving a sampled approximation instead. However, when the value function approximation is chosen to be low dimensional look up tables, Proposition 4 in Chapter 2 shows that constraint-based ALP relaxations become tractable approximate SDPs.

Multiplier-based ALP Relaxations. The idea here is to add constraints to DALP that match discounted probabilities associated with the exogenous part of the state y_i^r or moments of the components of this state. Addition of such constraints is justified under (A3) because the value of the variable $u_i^*(y_i, a_i)$ in an DLP optimal solution can be interpreted as the discounted probability of reaching stage i and state y_i and taking action a_i starting from the initial stage and state. Denoting the probability of reaching the stage i exogenous state y_i^r starting from the initial state y_0 by $\Pr(i, y_i^r)$, optimal DLP solutions thus match the discounted probability of the exogenous part of the state, $\delta^i \Pr(i, y_i^r)$, that is, $\sum_{(a_i, y_i^d)} u_i^*(y_i, a_i) \equiv \delta^i \Pr(i, y_i^r)$.

To match moments in DALP, we interpret the term $\sum_{(a_i, y_i^d)} w_i(y_i, a_i)$ as the discounted probability mass function of the exogenous part of the state. Matching the entire discounted probability mass function of the exogenous part of the state involves adding the following constraints to DALP:

$$\sum_{(a_i, y_i^d)} w_i(y_i, a_i) = \delta^i \Pr(i, y_i^r), \forall (i, y_i^r). \tag{5.42}$$

The primal linear program corresponding to DALP restricted with these constraints is

$$\min_{\beta} \sum_i c_i^{\intercal}(\Phi_i \beta_i) + \sum_{(i, y_i^r)} \Pr(i, y_i^r) d_i(y_i^r) \tag{5.43}$$

$$\text{s.t. } (\Phi_{N-1}\beta_{N-1})(y_{N-1}) \geq r_{N-1}(y_{N-1}, a_{N-1}), \forall (y_{N-1}, a_{N-1}), \tag{5.44}$$

$$(\Phi_i \beta_i)(y_i) + d_i(y_i^r) \geq r_i(y_i, a_i) + \delta \mathbb{E}_{i, y_i, a_i}(\Phi_{i+1}\beta_{i+1}), \forall (i, y_i, a_i)_{-(N-1)}. \tag{5.45}$$

Compared to the constraint-based ALP relaxation, the multiplier-based ALP relaxation (5.43)-(5.45) controls the amount of relaxation using the additional term in its objective function as opposed to constraints. Because the number of decision variables $d_i(y_i^r)$ in a multiplier-based ALP relaxation is equal to the number of restricting constraints added to DALP, one could simply add fewer constraints to DALP and obtain a smaller ALP relaxation. For instance, such constraints might depend only on a few moments of the probability mass function rather than all the probabilities that define it, as discussed next.

Let $y_{i,j}^r$ be the j-th element of y_i^r and $|y_i^r|$ the number of elements of y_i^r. We denote by $y_i^{r,-j}$ the elements of y_i^r other than the j-th element and by $y_i^{r,-\{j,k\}}$ the elements of y_i^r with both the j-th and k-th elements removed. The probabilities of the element $y_{i,j}^r$ and the pair $y_{i,j}^r y_{i,k}^r$ are given by $\Pr(i, y_{i,j}^r)$ and $\Pr(i, y_{i,j}^r y_{i,k}^r)$, respectively. Recalling the interpretation of $\sum_{(a_i, y_i^d)} w_i(y_i, a_i)$ used in (5.42), the following constraints enforce the first, second, and cross moments of the elements of y_i^r at each stage i:

$$\sum_{y_{i,j}^r} y_{i,j}^r \sum_{(y_i^d, y_i^{r,-j}, a_i)} w_i(y_i, a_i) = \sum_{y_{i,j}^r} y_{i,j}^r \Pr(i, y_{i,j}^r), \forall (i, j) \in \mathcal{I} \times \{1, \ldots, |y_i^r|\}, \tag{5.46}$$

$$\sum_{y_{i,j}^r} (y_{i,j}^r)^2 \sum_{(y_i^d, y_i^{r,-j}, a_i)} w_i(y_i, a_i) = \sum_{y_{i,j}^r} (y_{i,j}^r)^2 \Pr(i, y_{i,j}^r), \forall (i, j) \in \mathcal{I} \times \{1, \ldots, |y_i^r|\}, \tag{5.47}$$

$$\sum_{(y^r_{i,j},y^r_{i,k})} y^r_{i,j}y^r_{i,k} \sum_{(y^d_i,y^{r,-\{j,k\}}_i,a_i)} w_i(y_i,a_i) = \sum_{(y^r_{i,j},y^r_{i,k})} y^r_{i,j}y^r_{i,k} \Pr(i,y^r_{i,j}y^r_{i,k}),$$
$$\forall i \in \mathcal{I}, (j,k) \times \{1,\ldots,|y^r_i|\}, k > j. \tag{5.48}$$

Multiplier-based ALP relaxations are more challenging to solve than ALP, but, as discussed above, we can better control the number of new variables in this ALP relaxation class. Thus, we may be able to employ row generation when the multiplier-based ALP relaxations have only a small to moderate number of new variables relative to ALP; resorting to solving a sampled approximation when row generation is not tractable.

5.5 Unification of ADP Methods in §5.4

By complementary slackness, the set of all optimal DALP solutions determine which constraints in ALP must hold as equalities at optimality. In other words, the DALP optimal solution set implicitly samples the set of states that determines the ALP value function approximation. This set is

$$\mathcal{Y}^{DALP} := \{(i,y_i) | \exists \text{ an optimal DALP solution } w^* \text{ such that } \textstyle\sum_{a_i} w^*_i(y_i,a_i) > 0\},$$

where the term $\sum_{a_i} w_i(y_i,a_i)$ is the state occupancy frequency of state y_i in stage i under a DALP feasible solution w (see discussion after the DALP formulation in §5.4.2 for an interpretation of the DALP variables).

When the DALP optimal solution set is distorted relative to the DLP optimal solution set, as discussed in §2.4 and §5.4.5, the cardinality of the set $\mathcal{Y}^{DALP}$ may be small, thus leading to an under-exploration of the state space when determining the value function approximation. Property 3 states that the sum of DALP state occupancy frequencies is equal to a positive constant. Moreover, DALP variables are nonnegative by (5.22) and states outside $\mathcal{Y}^{DALP}$ have zero state occupancy frequencies. Therefore, a small cardinality of $\mathcal{Y}^{DALP}$ implies that the state occupancy frequencies induced by optimal DALP solutions are artificially high only for certain states. We restrict DALP by adding constraints with the goal of avoiding the concentration of state occupancy frequencies at a few states. By applying this principle in different ways, we will show in §5.5 that the methods considered are ALP relaxations.

5.5.1 SALP and Multiplier-based ALP Relaxations

The simplest way to avoid state occupancy frequencies becoming too large at a subset of states is to place an upper bound on their value.

We find that SALP bounds the state-action frequencies by adding the following inequality constraint $\forall(i,y_i)$

$$\sum_{a_i} w_i(y_i,a_i) \leq \Pr^{\pi^*,c,\delta}(y_i). \tag{5.49}$$

The probabilities on the right hand side of (5.49) are valid upper bounds in the sense that there is a DLP optimal solution that satisfies this upper bound. The primal linear program corresponding to DALP with constraints (5.49) is the linear program (5.24)-(5.27), which is the finite horizon specification of the conceptual model (14) stated and analyzed in §4.3 by Desai et al. (2012a).

Multiplier-based ALP relaxations bound state occupancy frequencies using a different strategy. Because the state occupancy frequencies $\sum_{a_i} w_i(y_i, a_i)$ are positive by (5.22), they can be bounded by equality constraints that set their weighted sum equal to a constant. Constraints (5.42) and constraints (5.46)-(5.48) are such examples with probabilities and moments of state elements on their right hand sides, respectively. For a fixed state $(\bar{y}_i^d, \bar{y}_i^r)$, this bounding property is easy to see for constraints (5.46) because

$$\begin{aligned}\sum_{a_i} w_i(\bar{y}_i^d, \bar{y}_i^r, a_i) &= \Pr(y_i^r) - \sum_{y_i^d \neq \bar{y}_i^d, a_i} w_i(y_i^d, \bar{y}_i^r, a_i) \\ &\leq \Pr(y_i^r).\end{aligned}$$

5.5.2 LSM and Constraint-based ALP Relaxations

In this subsection, we will represent the DALP state occupancy frequencies as a linear combination of a family of pre-specified state occupancy frequency distributions. If the chosen state occupancy frequency distributions do not concentrate the occupancy frequencies to a few states, then the optimal DALP solution under this restriction will not under explore the state space, that is, the cardinality of $\mathcal{Y}^{DALP}$ will not be small.

Let $\Psi_i = \{\psi_{i,b'}, b' = 1, \ldots, B_i'\}$ be a set of B_i' stage i basis functions. Each basis function defines state occupancy frequencies over the stage i state space $\mathcal{Y}_i$. Denoting by $\theta_{i,b'}$ the weight of the b'-th stage i basis function, we approximate the state occupancy frequencies using constraints parametrized by basis functions:

$$\sum_{a_i} w_i(y_i, a_i) = (\Psi_i \theta_i)(y_i), \forall (i, y_i), \tag{5.50}$$

where $(\Psi_i \theta_i)(y_i) := \sum_{b'} \psi_{i,b'}(y_i)\theta_{i,b'}$. Adding these constraints to DALP and taking the dual of the resulting DALP restriction gives the following ALP relaxation:

$$\min_{\beta, d} \sum_i c_i^\intercal (\Phi_i \beta_i) \tag{5.51}$$

$$(\Phi_i \beta_i)(y_i) + d_i(y_i) \geq r_i(y_i, a_i) + \delta \mathbb{E}_{i, y_i, a_i}(\Phi_{i+1}\beta_{i+1}), \quad \forall (i, y_i, a_i), \tag{5.52}$$

$$\sum_{y_i} \psi_{i,b}(y_i) d_i(y_i) = 0, \quad \forall (i, b), \tag{5.53}$$

where constraints (5.53) in the ALP relaxation are a consequence of the dual variables θ_i in the constraints (5.50) that we added to restrict DALP.

We begin by stating in Proposition 12 the relationship between LSMV and the ALP

Algorithm 3: Stage decomposed solution of (5.51)-(5.53)

1. Initialize $\hat{\beta}_N \equiv 0$.
2. For each $i = N-1$ to 0 set $\hat{\beta}_i$ to the optimal solution of the linear program

$$\min_{\beta_i, d_i} \; c_i^{\intercal}(\Phi_i \beta_i) \tag{5.54}$$

$$(\Phi_i \beta_i)(y_i) + d_i(y_i) \geq r_i(y_i, a_i) + \delta \mathbb{E}_{i,y_i,a_i}(\Phi_{i+1}\hat{\beta}_{i+1}), \quad \forall (y_i, a_i), \tag{5.55}$$

$$\sum_{y_i} \psi_{i,b}(y_i) d_i(y_i) = 0, \quad \forall b. \tag{5.56}$$

3. Return $\hat{\beta}_i$, $\forall i$.

relaxation (5.51)-(5.53) parameterized by basis functions in Ψ_i, $\forall i$. Let $\bar{\beta}_i$, $\forall i \in \mathcal{I} \setminus \{0\}$ be the solution to LSMV, and Z_i be the $|\mathcal{Y}_i| \times B_i$ regression matrix, where $Z_{i,y_i,b} = \phi_{i,b}(y_i)$. Given basis function weights β_i, $\forall i$, define

$$d_i(y_i; \beta) := \max_{a_i} r_i(y_i, a_i) + \delta \mathbb{E}_{i,y_i,a_i}(\Phi_{i+1}\beta_{i+1}) - (\Phi_i \beta_i)(y_i), \forall (i, y_i).$$

Proposition 12. *Suppose that for all i the regression matrix Z_i has full column rank, the basis functions in Φ_i are nonnegative, and $\Phi_i \equiv \Psi_i$. Then there exists state-relevance weights $\{\hat{c}_i, \forall i\}$ such that $(\bar{\beta}_i, d_i(\cdot; \beta))$, $\forall i$, is an optimal solution to (5.51)-(5.53).*

Proof. Consider the following convex program:

$$\min_{\beta} \; \sum_i c_i^{\intercal}(\Phi_i \beta_i) \tag{5.57}$$

$$\sum_{y_i} \phi_{i,b'}(y_i)(\Phi_i \beta_i)(y_i) \geq \sum_{y_i} \phi_{i,b'}(y_i) \left[\max_{a_i \in \mathcal{A}_i(y_i)} r_i(y_i, a_i) + \delta \mathbb{E}_{i,y_i,a_i}(\Phi_{i+1}\beta_{i+1}) \right], \quad \forall (i, b'). \tag{5.58}$$

Let the set of feasible β_i, $\forall i$, in systems (5.58) and (5.52)-(5.53) be denoted by $\mathcal{FS}^1$ and $\mathcal{FS}^2$, respectively. We will show that $\mathcal{FS}^1 \equiv \mathcal{FS}^2$. Notice that the constraints (5.52) define the lower bound $d_i(y_i; \beta)$ on $d_i(y_i)$. Constraints (5.58) can be obtained by using this lower bound in (5.53) because $\psi_{i,b}(y_i) = \phi_{i,b}(y_i) \geq 0$ by our assumption. Thus, $\mathcal{FS}^1 \supseteq \mathcal{FS}^2$. Next, given a $\bar{\beta}_i$ satisfying (5.58), it is easy to verify that $(\bar{\beta}_i, \bar{d}_i(\cdot; \bar{\beta}))$, $\forall i$, is a feasible solution to (5.52)-(5.53). Hence, we also have $\mathcal{FS}^1 \subseteq \mathcal{FS}^2$. Given that $\mathcal{FS}^1 \equiv \mathcal{FS}^2$, the equivalence between math programs (5.57)-(5.58) and (5.51)-(5.53) follows from their objective functions being the same.

Consider the inequalities (5.58) corresponding to stage i expressed as equalities. When the vector β_{i+1} on the right hand side of these equalities is chosen to be the stage $i+1$

LSMV vector of weights $\bar{\beta}_{i+1}$, they define the well known first order conditions for the 2-norm regression problem (5.13) used to compute the stage i LSMV vector of weights β_i. Using the full rank assumption on the regression matrices, it is easy to verify through an induction argument, starting from stage $N-1$ and moving backwards to stage 1, that $\bar{\beta}_i$, $\forall i$, is the unique solution to the equality system defined by the inequalities (5.58).

Moreover, $\bar{\beta}_i$, $\forall i$, is a boundary point of the convex feasible region (5.58) because all the inequalities hold as equalities at this point. Then it follows from the linear support function characterization of convex sets (see §2.4 in Hiriart-Urruty and Lemaréchal 2001) that there exists a non zero vector $\hat{c} \equiv \{c_i, \forall i\}$ such that $\bar{\beta}_i$, $\forall i$ is an optimal solution to the linear program (5.51)-(5.53). □

Proposition 12 states that the LSMV weights define an optimal solution to the ALP relaxation (5.51)-(5.53) when $\Phi \equiv \Psi$ and under the mild condition of positive basis functions, which can be easily satisfied by replacing basis functions by their positive and negative parts. Thus, there is an ALP relaxation (5.51)-(5.53) that recovers the LSMV weights but we are unable to characterize these state relevance weights. Nevertheless, as established in Proposition 13, the LSMV weights can be obtained by solving a stage-decomposed version of (5.51)-(5.53) using the readily available state-relevance weights $\hat{c}_i(y_i) = \sum_b \phi_{i,b}(y_i)$, $\forall (i, y_i)$ defined by basis functions in Φ.

Proposition 13. *Suppose that for all i the regression matrix Z_i has full column rank, the basis functions in Φ_i are nonnegative, and $\Phi_i \equiv \Psi_i$. Then for state-relevance weights $\hat{c}_i(y_i) = \sum_b \phi_{i,b}(y_i)$, $\forall (i, y_i)$, Algorithm 3 returns the LSMV weights $\bar{\beta}_i$, $\forall i$.*

Proof. Using essentially the same arguments as in the proof of Proposition 12 it can be shown that (5.55)-(5.56) is equivalent to the convex constraints

$$\sum_{y_i} \phi_{i,b'}(y_i)(\Phi_i \beta_i)(y_i) \geq \sum_{y_i} \phi_{i,b'}(y_i) \left[\max_{a_i \in \mathcal{A}_i(y_i)} r_i(y_i, a_i) + \delta \mathbb{E}_{i,y_i,a_i}(\Phi_{i+1} \hat{\beta}_{i+1}) \right], \quad \forall (i, b'). \tag{5.59}$$

The only difference between the above constraints and the constraints (5.58) is that the right hand side of (5.59) is a constant because $\hat{\beta}_{i+1}$ is specified. Under the assumed choice of state relevance weights, the objective function (5.54) is a positive combination of the left hand sides of (5.59). Moreover, $\bar{\beta}_i$, $\forall i$, is the unique solution to (5.59) expressed as equalities because of the full rank assumption on the regression matrices. Then the objective sense being minimization implies that the inequalities (5.59) must hold as equalities at optimality. □

Although we focus in this subsection on LSMV, the Longstaff and Schwartz (2001) LSM approach discussed in §3.4.2 can also be derived as a relaxation of an approximate linear program based on the continuation function SDP formulation (3.6).

We now turn to deriving the constraint-based ALP relaxation (5.38)-(5.41). This ALP relaxation differs from the parametric ALP relaxation (5.51)-(5.53) only in its constraints

(5.41). We will show that constraints (5.41) can be derived from constraints (5.53) by choosing basis functions Ψ_i in (5.53) to be different from Φ_i. Following the treatment in §5.4.5, we will assume (A1)-(A3) hold. Let $\hat{Y}_i = \{(y_i^d, y_i^{r,+}) | \exists\ y_i^{r,-} \text{ s.t. } (y_i^d, y_i^{r,+}, y_i^{r,-}) \in \mathcal{Y}_i\}$. Consider the choice

$$\psi_{i,b}(y_i) = \mathbb{1}(y_i^d = \bar{y}_i^d[b], y_i^{r,+} = \bar{y}_i^{r,+}[b]) p(y_i^{r,-} | \bar{y}_i^d[b], \bar{y}_i^{r,+}[b]),$$

$\forall b = 1, \ldots, |\hat{Y}_i|$, where $(\bar{y}_i^d[b], \bar{y}_i^{r,+}[b])$ denote the b-th element in set $\hat{Y}_i$. Under this choice we have

$$\begin{aligned}\sum_{y_i} \psi_{i,b}(y_i) d_i(y_i) &= \sum_{y_i} \mathbb{1}(y_i^d = \bar{y}_i^d, y_i^{r,+} = \bar{y}_i^{r,+}) p(y_i^{r,-} | \bar{y}_i^d, \bar{y}_i^{r,+}) d_i(y_i) \\ &= \sum_{y_i^{r,-}} p(y_i^{r,-} | \bar{y}_i^d, \bar{y}_i^{r,+}) d_i(\bar{y}_i^d, \bar{y}_i^{r,+}, y_i^{r,-}),\end{aligned}$$

which is a constraint in (5.41). Thus, we recover the constraint-based ALP relaxation (5.38)-(5.41).

5.5.3 Iterated Bellman Linear Program

As explained in §5.4.4, the iterated Bellman linear program is an ALP relaxation that is defined in a lifted primal space. We will establish relationships between the dual of the iterated Bellman linear program and both DALP and DLP. These relationships highlight how the dual of the iterated Bellman linear program attempts to increase the number of states explored by DALP. We also show a potential weakness of the upper bound on the optimal policy value from the iterated Bellman linear program for a commonly encountered sufficient condition in finite horizon and discrete time problems.

We begin by presenting the dual of the iterated Bellman linear program. Consistent with the indexing of constraints (5.35)-(5.36), we denote by $w_{j,i}^L(y_i, a_i)$ the dual variables associated with the j-th set of inequalities corresponding to the triple (i, y_i, a_i) (the superscript L abbreviates lifted space). The dual of the iterated Bellman linear program is

$$\max_{w^L} \sum_{(j,i)} r_i^\intercal w_{j,i}^L \tag{5.60}$$

$$\text{s.t.} \sum_{a_0} w_{0,0}^L(y_0, a_0) = c_0(y_0), \tag{5.61}$$

$$\sum_{y_i} \phi_{i,b}(y_i) \sum_{a_i} w_{0,i}^L(y_i, a_i) = \sum_{y_i} \phi_{i,b}(y_i) c_i(y_i), \forall (i,b)_{-(0)}, \tag{5.62}$$

$$\sum_{a_0} w_{j,0}^L(y_0, a_0) = 0, \forall j = 1, \ldots, J, \tag{5.63}$$

$$\sum_{y_i} \phi_{i,b}(y_i) \sum_{a_i} w_{j,i}^L(y_i, a_i) = \sum_{y_i} \phi_{i,b}(y_i) \delta \sum_{(y_{i-1}, a_{i-1})} \Pr(y_i | y_{i-1}, a_{i-1}) w_{j-1,i-1}^L(y_{i-1}, a_{i-1}),$$

$$\forall (i,b)_{-(0)}, j = 1, \ldots, J-1, \tag{5.64}$$

$$\sum_{y_i} \phi_{i,b}(y_i) \sum_{a_i} w^L_{J,i}(y_i, a_i) = \sum_{y_i} \phi_{i,b}(y_i) \delta \sum_{(y_{i-1}, a_{i-1})} \Pr(y_i | y_{i-1}, a_{i-1}) [w^L_{J-1,i-1}(y_{i-1}, a_{i-1}) + w^L_{J,i-1}(y_{i-1}, a_{i-1})],$$

$$\forall (i,b)_{-(0)}, \tag{5.65}$$

$$w^L_{j,i}(y_i, a_i) \geq 0, \forall (i, y_i, a_i), j = 1, \ldots, J. \tag{5.66}$$

Proposition 14 states our main results. We summarize the findings of Proposition 14 in Figure 5.1.

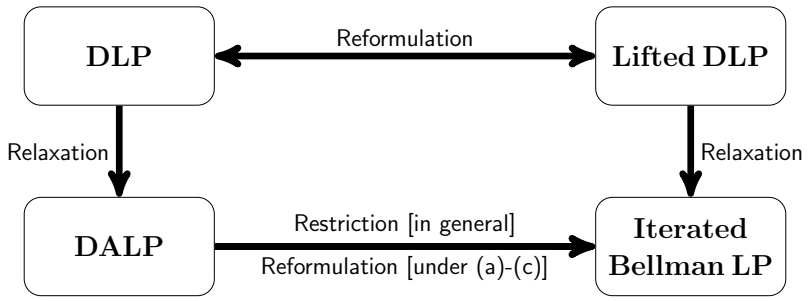


Figure 5.1: Schematic illustration of results in Proposition 14.

Proposition 14. *It holds that the dual (5.60)-(5.66) of the iterated Bellman linear program is (i) a relaxation of an equivalent lifted reformulation of DLP, and (ii) a restriction of an equivalent lifted reformulation of DALP using constraints (5.63)-(5.66). Moreover, (iii) if*

(a) $c_0(y_0) > 0$ *and* $c_i(y_i) = 0$, $\forall (i, y_i)_{-(0)}$,

(b) for each (i, y_i), $\exists b$ *such that* $|\phi_{i,b}(y_i)| > 0$, *and*

(c) for each i, *either* $\phi_{i,b}(y_i) \geq 0$, $\forall y_i$ *or* $\phi_{i,b}(y_i) \leq 0$, $\forall y_i$,

then the dual of the iterated Bellman linear program is equivalent to DLP and the optimal objective function values of the iterated Bellman linear program and ALP are identical.

Proof. (i) Consider the following linear program:

$$\max_{u^L} \sum_{(j,i)} r_i^\intercal u^L_{j,i} \tag{5.67}$$

$$\text{s.t.} \sum_{a_i} u^L_{0,i}(y_i, a_i) = c_i(y_i), \forall (i, y_i), \tag{5.68}$$

$$\sum_{a_0} u^L_{j,0}(y_0, a_0) = 0, \forall j = 1, \ldots, J, \tag{5.69}$$

$$\sum_{a_i} u^L_{j,i}(y_i, a_i) =$$

$$\delta \sum_{(y_{i-1},a_{i-1})} \Pr(y_i|y_{i-1},a_{i-1}) u^L_{j-1,i-1}(y_{i-1},a_{i-1}), \forall (i,y_i)_{-(0)}, j=1,\ldots,J-1, \quad (5.70)$$

$$\sum_{a_i} u^L_{J,i}(y_i,a_i) =$$

$$\delta \sum_{(y_{i-1},a_{i-1})} \Pr(y_i|y_{i-1},a_{i-1})[u^L_{J-1,i-1}(y_{i-1},a_{i-1}) + u^L_{J,i-1}(y_{i-1},a_{i-1})], \forall (i,y_i)_{-(0)}, \quad (5.71)$$

$$u^L_{j,i}(y_i,a_i) \geq 0, \forall (i,y_i,a_i), j=1,\ldots,J. \quad (5.72)$$

Constraints (5.68) for stage 0, (5.69), and (5.72) are identical to constraints (5.61), (5.63), and (5.66), except for a notational difference. Each of the remaining constraints in (5.61)-(5.66) is a linear combination of constraints in (5.68)-(5.72) using weights specified by a basis function evaluated at different states at a given stage. Thus, the dual (5.60)-(5.66) of the iterated Bellman linear program is a relaxation of (5.67)-(5.72). To prove the claimed result we now show that (5.67)-(5.72) is an equivalent lifted reformulation of DLP.

Let $\hat{u}^L_{j,i}$, $\forall (i,j)$, be a feasible solution to (5.67)-(5.72). Define the terms $\bar{u}_i(y_i,a_i) = \sum_{j=0}^J \hat{u}^L_{j,i}(y_i,a_i)$, $\forall (i,y_i,a_i)$. It can be easily verified that $\bar{u}_i$, $\forall i$, defines a feasible solution to DLP with the same objective function value as $\hat{u}^L_{j,i}$, $\forall (i,j)$.

For the converse, let $\bar{u}_i$, $\forall i$, be a basic feasible solution of DLP. Consider the system obtained by adding the following conditions to (5.67)-(5.72): $u^L_{j,i}(y_i,a_i) = 0$, $\forall (j,i,y_i,a_i)$ such that $\bar{u}_i(y_i,a_i) = 0$. Given the triple (i,j,y_i), the restricted system will have at most one a_i for which $u^L_{j,i}(y_i,a_i) > 0$ because of Property 2. As a result, this restricted system becomes a simple recursion which can be solved starting from stage 0 and moving forward to stage $N-1$. We label this solution $\hat{u}^L_{j,i}$, $\forall (i,j)$. Because flow over stages and states is conserved in both DLP and (5.67)-(5.72), we have $\sum_{j=0}^J \hat{u}^L_{j,i}(y_i,a_i) = \bar{u}_i(y_i,a_i)$, $\forall (i,y_i,a_i)$. Thus, $\hat{u}^L_{j,i}$, $\forall (i,j)$, has the same objective function value as $\bar{u}_i$, $\forall i$.

(ii) To derive the ALP relaxation (5.33)-(5.36) we first lift DALP by using the substitution $w_i(y_i,a_i) = \sum_{j=0}^J w^L_{j,i}(y_i,a_i)$. The resulting lifted DALP is

$$\max_{w^L} \sum_{(j,i)} r_i^\intercal w^L_{j,i} \quad (5.73)$$

$$\text{s.t.} \sum_{a_0} \sum_{j=0}^J w^L_{j,0}(y_0,a_0) = c_0(y_0), \quad (5.74)$$

$$\sum_{y_i} \phi_{i,b}(y_i) \sum_{a_i} \sum_{j=0}^J w^L_{j,i}(y_i,a_i) =$$

$$\sum_{y_i} \phi_{i,b}(y_i) \left[c_i(y_i) + \delta \sum_{(y_{i-1},a_{i-1})} \Pr(y_i|y_{i-1},a_{i-1}) \sum_{j=0}^J w^L_{j,i-1}(y_{i-1},a_{i-1}) \right], \forall (i,b)_{-(0)}, \quad (5.75)$$

$$\sum_{j=0}^J w^L_{j,i}(y_i,a_i) \geq 0, \forall (i,y_i,a_i). \quad (5.76)$$

We restrict (5.73)-(5.74) by adding constraints (5.63)-(5.66). The resulting restriction is equivalent to the dual of the iterated Bellman linear program, except for the following cosmetic differences: (i) Constraints (5.76) are absent in (5.61)-(5.66), (ii) constraints (5.74) are different from constraints (5.61), and constraints (5.75) are different from constraints (5.62). The first difference is immaterial because constraints (5.76) are implied by constraints (5.66). The second difference can be reconciled by substituting (5.63) in (5.74). Finally, the third difference can be handled by using (5.64) and (5.65) to eliminate terms from (5.75) and obtain (5.62). It is evident from our derivation that for any feasible solution $\hat{w}^L_{j,i}$, $\forall i, j$, of (5.60)-(5.66) the terms $\bar{w}_i(y_i, a_i) := \sum_{j=0}^{J} \hat{w}^L_{j,i}(y_i, a_i)$ define a feasible solution to DALP.

(iii) Let $\mathcal{W} := \{(i, j) \in \mathcal{I} \times \{0, \ldots, J\} | i \neq j \text{ if } 0 < j < J \text{ or } i < j \text{ if } j = J\}$. We will show that conditions (a)-(c) and constraints (5.61)-(5.66) force $w_{j,i}(y_i, a_i) = 0$, $\forall (i, j) \in \mathcal{W}$ and $\forall (y_i, a_i)$. To verify this claim, notice that assumption (a) implies that the right hand side of constraints (5.62) are zero. This together with conditions (b) and (c) imply that $w^L_{0,i}(y_i, a_i) = 0$, $\forall (i, y_i, a_i)_{-(0)}$. Moreover, by (5.63), the only nonzero variables at stage 0 correspond to $j = 0$. Because of these observations, conditions (b) and (c), and (5.64), the only nonzero variables at stage 1 correspond to $j = 1$. Moving forward in this manner shows that $w_{j,i}(y_i, a_i) = 0$, $\forall (i, j) \in \mathcal{W}$ and $\forall (y_i, a_i)$, which implies that variables with non zero values have (i, j) pairs belonging to the set $\{(i, j) | i = j, \forall\, 0 < j < J\} \cup \{(i, j) | i \geq j, \forall j = J\}$. In other words, for each i there is exactly one j for which variables in the lifted DALP could be non zero. The set of constraints defined by these pairs are equivalent to the DALP constraints, after accounting for condition (a). Therefore, the optimal objective function values of DALP and (5.60)-(5.66) are equal, which implies that the optimal objective function values of ALP and the iterated Bellman linear program are also equal because of strong linear programming duality. □

Part (i) of Proposition 14 establishes that the dual (5.60)-(5.66) of the iterated Bellman linear program is a relaxation of the lifted reformulation (5.67)-(5.72) of DLP. This lifted reformulation has an intuitive network flow relationship to DLP. Notice that the flow balance constraints (5.7) of DLP equate the flow at stage i and state y_i (the left hand side) to the new flow $c_i(y_i)$ entering the system at stage i and state y_i, and the discounted sum of flows that entered into the system at prior stages and reached stage i and state y_i (the right hand side). Thus, DLP conserves flow entering into the system at different stages in an aggregate fashion (see aggregated flow illustration in Figure 5.2). An alternative to this aggregate flow conservation is to dedicate a separate set of equations to conserve flows introduced at states in each stage i for the first J periods (that is periods $0, \ldots, J-1$) that they spend in the system and then re-aggregate these flows at the next period (see disaggregated flow illustration in Figure 5.2). This is the intuition behind the lifted DLP reformulation (5.67)-(5.72).

Next, notice that the relaxation used to derive the dual (5.60)-(5.66) of the iterated Bellman linear program from the lifted DLP reformulation (5.67)-(5.72) is analogous to the relaxation used to derive DALP from DLP. Thus, one may expect DALP and the dual of iterated Bellman linear program to also be equivalent. Nevertheless, Part (ii) of Proposition 14 shows that (5.60)-(5.66) is a restriction of DALP. This is because the exact

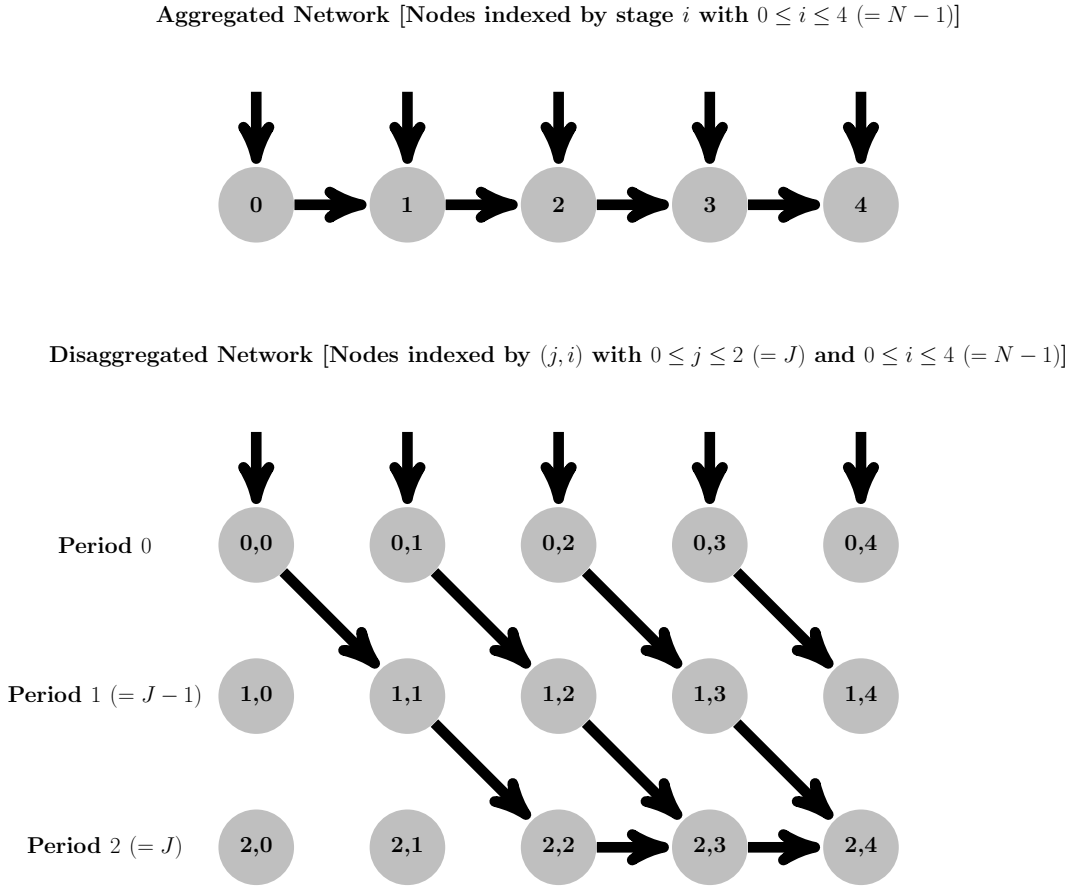


Figure 5.2: Network flow illustration of relationship between DLP and lifted DLP reformulation (5.67)-(5.72).

disaggregation of DLP flows illustrated in Figure 5.2 and used to obtain (5.67)-(5.72) is not exact when applied to DALP because DALP lacks the network flow structure of DLP. This particular form of restriction has implications on the number of states explored by the dual of the iterated Bellman linear program relative to DALP. Recall from the proof of Proposition 14 that every feasible solution $\hat{w}^L$ of (5.60)-(5.66) defines a feasible solution $\bar{w}$ of DALP via the equivalence relationship $\bar{w}_i(y_i, a_i) \equiv \sum_{j=0}^{J} \hat{w}^L_{j,i}(y_i, a_i)$. This feasible DALP solution conserves flow at each stage by Property 3, but $\bar{w}$ is defined by a restriction of DALP that disaggregates flow as shown in Figure 5.2. Thus, $\bar{w}$ is likely to have nonzero entries at a larger number of states than DALP optimal solutions.

Finally, Part (iii) of Proposition 14 shows that the choice of state relevance weights plays an important role in how the dual of the iterated Bellman linear program restricts DALP under the mild conditions (a)-(c): Condition (a) is natural for finite horizon problems because the initial state distribution is concentrated on a given stage 0 state (see Chapter 2 and Adelman 2003), condition (b) is almost always true, and condition (c) can be easily satisfied for practical purposes by splitting a basis function violating this condition into its positive and negative parts. In particular, this proposition shows that the dual of the

iterated Bellman linear program does not restrict DALP under conditions (a)-(c). As a consequence, the upper bound on the optimal policy value from the iterated Bellman linear program is equal to the upper bound from ALP (see §5.4.2 and §5.4.4 for a discussion of the upper bounds from ALP and the iterated Bellman linear program). Thus, we identify a likely parameter combination that must be avoided when using the iterated Bellman linear program to improve the ALP upper bound on the optimal policy value.

5.6 Mixed Integer Programming ADP Methods for Approximately Minimizing Greedy Policy Loss

The methods considered in §§5.4-5.5 attempt to minimize the value function approximation error (5.10). If the primary purpose of a value function approximation is to compute a greedy policy, then attempting to find a value function approximation that minimizes greedy policy loss (5.12) seems logical. In this subsection, we discuss the math programming based ADP approach of Petrik (2012) that approximates this objective by minimizing an upper bound on policy loss. To simplify our exposition we will choose state relevance weights such that $c_0(y_0) = 1$ and $c_i(y_i) = 0$, $\forall (i, y_i)_{-(0)}$. All the results below have their respective counterparts when using general state relevance weights.

Let $\mathcal{SA} := \cup_{(i,y_i)} \mathcal{A}_i(y_i)$. We define the set of policies as $\Pi := \{\pi \in [0,1]^{|\mathcal{SA}|} : \sum_{a_i} \pi(a_i|y_i) = 1, \forall (i, y_i)\}$, where the notation $\pi_i(a_i|y_i)$ stresses that a policy in Π is represented using conditional probabilities of taking action a_i given that we are in stage i and state y_i. This representation is different from the representation of a policy in DLP as a set of discounted joint probabilities of visiting stage i and state y_i and taking action a_i. Petrik (2012) constructs an upper bound on the policy loss using a lower bound $\tilde{\rho}(\pi)$ on the value of a policy $\rho(\pi)$, that is, for an optimal policy π^*,

$$\rho(\pi^*) - \rho(\pi) \leq \rho(\pi^*) - \tilde{\rho}(\pi).$$

The lower bound $\tilde{\rho}(\pi)$ is defined as

$$\tilde{\rho}(\pi) = \max_{\beta} \left((\Phi_0 \beta_0) - \max_{w \in \tilde{W}(\pi)} \sum_{(i,y_i,a_i)} w_i(y_i, a_i) \left((\Phi_i \beta_i)(y_i) - r_i(y_i, a_i) - \delta \mathbb{E}_{i,y_i,a_i} (\Phi_{i+1} \beta_{i+1}) \right) \right),$$

where $\tilde{W}(\pi) := \{w \in \mathbb{R}^{|\mathcal{SA}|} | (5.20)\text{-}(5.22), w_i(y_i, a_i) \leq \delta^i \pi_i(a_i|y_i), \forall (i, y_i, a_i)\}$. The set $\tilde{W}(\pi)$ is a restriction of the DALP feasible set because it is defined using the DALP constraints (5.20)-(5.22) and the additional constraints $w_i(y_i, a_i) \leq \delta^i \pi_i(a_i|y_i), \forall (i, y_i, a_i)$. For a deterministic policy π these additional constraints ensure that $w_i(y_i, a_i)$ is non zero for a given (i, y_i) pair at precisely the action prescribed by π. This property will be critical for finding a value function approximation that is greedy with respect to a deterministic policy.

In order to find the policy in Π that minimizes the upper bound $\rho(\pi^*) - \tilde{\rho}(\pi)$ on the

policy loss, Petrik (2012) solves the equivalent problem of maximizing $\tilde{\rho}(\pi)$:

$$\max_{\pi \in \Pi} \tilde{\rho}(\pi). \tag{5.77}$$

The primal version of the optimization problem (5.77) can be reformulated as the following bilinear program:

$$\max_{\beta,d,\pi} (\Phi_0 \beta_0) - \sum_i \pi_i^\intercal d_i \tag{5.78}$$

$$\text{s.t. } (\Phi_{N-1}\beta_{N-1})(y_{N-1}) \leq r(y_{N-1}, a_{N-1}) + d_{N-1}(y_{N-1}, a_{N-1}), \forall (y_{N-1}, a_{N-1}), \tag{5.79}$$

$$(\Phi_i \beta_i)(y_i) \leq r(y_i, a_i) + \delta \mathbb{E}_{i,y_i,a_i}(\Phi_{i+1}\beta_{i+1}) + d_i(y_i, a_i), \forall (i, y_i, a_i)_{-(N-1)}, \tag{5.80}$$

$$\sum_{a_i} \pi_i(y_i|a_i) = 1, \forall (i, y_i), \tag{5.81}$$

$$0 \leq \pi_i(y_i|a_i) \leq 1, \forall (i, y_i, a_i). \tag{5.82}$$

Solving this bilinear program returns both a policy and a value function approximation. Petrik (2012) shows that there exists an optimal solution triple (π', β', d') to (5.78)-(5.82) such that π' is the greedy policy induced by the value function approximation $(\Phi\beta')$. Moreover, by optimality, it follows that β' defines the value function approximation that induces the greedy (hence deterministic) policy maximizing $\tilde{\rho}(\pi)$, but not necessarily maximizing $\rho(\pi)$. This connection between the value function approximation and its greedy policy is a result of the constraints $w_i(y_i, a_i) \leq \delta^i \pi_i(a_i|y_i), \forall (i, y_i, a_i)$ in $\tilde{W}(\pi)$ that make this set a restriction of the DALP feasible set.

For computational purposes, Petrik (2012) reformulates this bilinear program as a mixed integer linear program, which he refers to as the distributionally robust mixed integer program because it is based on a relaxation of the DLP state occupancy frequencies. We abbreviate this mixed integer program as DRMIP. Given an upper bound $\tau_i(y_i, a_i)$ on $d_i(y_i, a_i)$, DRMIP is

$$\max_{\beta,d,\pi,z} (\Phi_0 \beta_0) - \sum_{(i,y_i,a_i)} z_i(y_i, a_i) \tag{5.83}$$

$$\text{s.t. } z_i(y_i, a_i) \geq d_i(y_i, a_i) - \tau_i(y_i, a_i)(1 - \pi_i(y_i|a_i)), \forall (i, y_i, a_i), \tag{5.84}$$

$$(\Phi_{N-1}\beta_{N-1})(y_{N-1}) \leq r(y_{N-1}, a_{N-1}) + d_{N-1}(y_{N-1}, a_{N-1}), \forall (y_{N-1}, a_{N-1}), \tag{5.85}$$

$$(\Phi_i \beta_i)(y_i) \leq r(y_i, a_i) + \delta \mathbb{E}_{i,y_i,a_i}(\Phi_{i+1}\beta_{i+1}) + d_i(y_i, a_i), \forall (i, y_i, a_i)_{-(N-1)}, \tag{5.86}$$

$$\sum_{a_i} \pi_i(y_i|a_i) = 1, \forall (i, y_i), \tag{5.87}$$

$$\pi_i(y_i|a_i) \in \{0, 1\}, \forall (i, y_i, a_i). \tag{5.88}$$

DRMIP has a large number of variables and constraints. Specifically, it has $|\mathcal{SA}|$ binary variables and $2|\mathcal{SA}| + \sum_i B_i$ free variables. Similar to the ALP relaxations discussed in §5.4, using DRMIP in practice entails solving a sampled approximation. However, because DRMIP has binary variables, solving these sampled versions of DRMIP is substantially more challenging than solving sampled ALP relaxations.

5.7 Mixed Integer Programming ADP Methods for Exactly Minimizing Greedy Policy Loss

The DRMIP model discussed in §5.6 computes a value function approximation by minimizing an upper bound on the greedy policy loss, but this value function approximation may not minimize greedy policy loss. Motivated by this shortcoming, we propose a mixed integer programming formulation that exactly minimizes greedy policy loss for a certain class of structured SDPs. These SDPs have (i) each state at stage i partitioned into an endogenous component y_i^d and an exogenous component y_i^r; and (ii) a deterministic stage i endogenous state transition rule $f_i(y_i^d, a_i)$ which maps state y_i^d and action a_i to a unique stage $i+1$ state in $\mathcal{Y}_{i+1}$. These assumptions are satisfied by the applications studied in Chapters 2-4, and other common energy real option applications

Our mixed integer program attempts to mimic the process of estimating the value of a greedy policy $\pi^g(\beta)$ associated with the value function approximation $(\Phi\beta)$ via Monte Carlo simulation (see §5.3 for a discussion of value function approximations and greedy policies). We explain this estimation process before presenting our formulation. We first sample L paths of the exogenous states y_i^r over stages in set $\mathcal{I} \setminus \{0\}$. Let $y_i^{r,p}, \forall i_{-(0)}$ denote the p-th sample path. To estimate the policy value we require the following two quantities: (i) a greedy action $a_{p,i}(\beta)$ taken at stage i and on sample path p, which can be computed using (5.11), and (ii) the endogenous state $y_{p,i}^d$ reached at stage i on sample path p as a result of following greedy actions. Given the value function approximation weights $\beta_i, \forall i_{-(0)}$, these two quantities can be computed using the following forward recursion:

1. Initialize $y_{p,0}^d(\beta) = y_0$ and $a_{p,0}(\beta) = \underset{a_0 \in \mathcal{A}_0(y_0)}{\text{argmax}}\, r_0(y_0, a_0) + \delta \mathbb{E}_{y_0,a_0}(\Phi_1\beta_1)$ for all p;
2. For stages 1 to $N-1$ do
   (a)
   $$y_{p,i}^d(\beta) = f_{i-1}(y_{p,i-1}^d(\beta), a_{p,i-1}(\beta)), \forall p, \tag{5.89}$$
   (b)
   $$a_{p,i}(\beta) \in \underset{a_i \in \mathcal{A}_i(y_{p,i}^d(\beta), y_{p,i}^r)}{\text{argmax}} r_i(y_{p,i}^d(\beta), y_{p,i}^r, a_i) + \delta \mathbb{E}_{y_{p,i}^d(\beta), y_{p,i}^r, a_i}(\Phi_{i+1}\beta_{i+1}), \forall p. \tag{5.90}$$

The triples $(i, y_{p,i}^d(\beta), a_{i,p}(\beta))$, $\forall i$, define the trajectory through the endogenous-state and action spaces taken by the greedy policy on sample path p. An estimate of the greedy policy value can be obtained by averaging the discounted sum of rewards on the L trajectories:

$$(1/L) \sum_{(p,i)} \delta^i r_i(y_{p,i}^d(\beta), y_{p,i}^r, a_{p,i}(\beta)). \tag{5.91}$$

.

The following mixed integer program finds the value function approximation that maximizes the greedy policy value estimate:

$$\max_{\beta,d,z} (1/L) \sum_{(i,y_i^d,y_{p,i}^r,a_i)} \delta^i r_i(y_i^d, y_{p,i}^r, a_i) z_i(y_i^d, y_{p,i}^r, a_i) \tag{5.92}$$

$$\text{s.t. } d_i(y_i^d, y_{p,i}^r, a_i) = r_i(y_i^d, y_{p,i}^r, a_i) + \mathbb{E}_{y_i^d, y_{p,i}^r, a_i}(\Phi_{i+1}\beta_{i+1}), \quad \forall (i, y_i^d, p, a_i), \tag{5.93}$$

$$d_i(y_i^d, y_{p,i}^r, a_i) - d_i(y_i^d, y_{p,i}^r, a_i') \geq (1 - z_i(y_i^d, y_{p,i}^r, a_i))\ell_i(y_i^d, y_{p,i}^r, a_i, a_i'), \quad \forall (i, y_i^d, p, a_i, a_i'), a_i' \neq a_i, \tag{5.94}$$

$$\sum_{y_i^d, a_i} z_i(y_i^d, y_{p,i}^r, a_i) = 1, \quad \forall (i,p), \tag{5.95}$$

$$\sum_{y_i^d, a_i} f(y_i^d, a_i) z_i(y_i^d, y_{p,i}^r, a_i) = \sum_{y_{i+1}^d, a_{i+1}} y_{i+1}^d z_{i+1}(y_{i+1}^d, y_{p,i}^r, a_{i+1}), \quad \forall (i,p)_{-(N-1)}, \tag{5.96}$$

$$z_i(y_i^d, y_{p,i}^r, a_i) \in \{0,1\}, \quad \forall (i, y_i^d, p, a_i). \tag{5.97}$$

This mixed integer program has three sets of variables: (i) the weights β_i define the stage i value function approximation; (ii) the binary variables z_i track the endogenous states visited and greedy actions taken along each sample path when following the greedy policy, that is, $z_i(y_i^d, y_{p,i}^r, a_i)$ equals 1, if $y_i^d = y_{p,i}^d(\beta)$ and $a_i = a_{i,p}(\beta)$, and zero otherwise; and (iii) the variables d_i store the sum of the immediate reward and continuation value for the stage-state-action triple (i, y_i, a_i). The parameter $\ell_i(y_i^d, y_{p,i}^r, a_i, a_i')$ is a lower bound on the difference $d_i(y_i^d, y_{p,i}^r, a_i) - d_i(y_i^d, y_{p,i}^r, a_i')$ and needs to be specified. Given the definition of variables z_i, the objective function (5.92) is analogous to (5.91). Constraints (5.93) enforce the definition of the d_i variables. Constraints (5.94) ensure that $z_i(y_i^d, y_{p,i}^r, a_i)$ equals 1 for an action that is greedy with respect to the value function approximation, that is, they mimic (5.90). Constraints (5.95) ensure that a single state-action pair is selected at stage i along the p-th sample path starting from x_0. Constraints (5.96) ensure that the state transition law $f_i(y_i^d, a_i)$ is enforced, that is, they model conditions (5.89).

Denoting the sampled version of the set $\mathcal{SA}$ by $\hat{\mathcal{SA}}$, the mixed integer program (5.92)-(5.97) has $|\hat{\mathcal{SA}}|$ binary variables and $|\hat{\mathcal{SA}}| + \sum_i B_i$ free variables. When compared to DRMIP, our formulation has the same number of binary variables but fewer free variables. Nevertheless, (5.92)-(5.97) has a large number of binary variables and constraints. Table 5.1 reports the number of variables and constraints in (5.92)-(5.97) when representing 6-month truncations of the twelve 24-month natural gas storage instances of §3.7.2 and using a modest 400 sample paths, that is, $L = 400$ (see §3.7.2 for details of these instances). The number of binary variables and constraints ranges are between 150,000-800,000 and 1.9-14.5 million, respectively.

Given the large size of the mixed integer program (5.92)-(5.97), it is unreasonable to expect its performance to be competitive with the LSMV method of Chapter 3, which computes value function approximations in under 2 seconds and estimates tight lower bounds on both the 6 and 24-stage instances (we independently verified the optimality of the LSMV lower bound estimates on the 6 stage instances). However, because our mixed integer program guarantees a value function approximation that maximizes the policy

Table 5.1: Number of variables and constraints in (5.92)-(5.97)

Capacity	Binary Variables	Free Variables	Constraints
High	226,404	227,538	1,989,617
Medium	158,404	159,394	1,248,017
Low	786,410	788,600	14,322,501

value (that is, the lower bound estimate) it can potentially serve as a benchmark for ADP algorithms in the literature, including LSMV, which lack this guarantee. To understand the viability of our mixed integer program for this purpose, we check if it can prove the known near optimality of the LSMV value function approximation on these 6-month instances.

To solve the mixed integer program (5.92)-(5.97) we need to choose the number of sample paths L and the lower bound parameter $\ell_i(y_i^d, y_{p,i}^r, a_i, a_i')$ in constraints (5.94). We define an in-sample lower bound estimate as one that is obtained on the L sample paths used for computing a value function approximation. Because the objective function (5.92) of our mixed integer program is an in-sample estimate of the greedy lower bound, its optimal objective function value provides a reliable proof of optimality only when this in-sample estimate has small standard error. Thus, our choice of L should ideally result in the LSMV in-sample lower bound estimate having a small standard error. For our choice of L equals 400, the standard error of the LSMV in-sample lower bound estimate ranges between 7-12% of the tight LSMV dual upper bound estimate. This standard error range is not small but, as we will see shortly, the mixed integer program for this value of L is already very challenging to solve. We choose $\ell_i(y_i^d, y_{p,i}^r, a_i, a_i')$ as follows:

$$\begin{aligned} d_i(y_i^d, y_{p,i}^r, a_i) - d_i(y_i^d, y_{p,i}^r, a_i') =& r_i(y_i^d, y_{p,i}^r, a_i) - r_i(y_i^d, y_{p,i}^r, a_i') \\ &+ \mathbb{E}_{y_{p,i}^d(\beta), y_{p,i}^r, a_i}(\Phi_{i+1}\beta_{i+1}) - \mathbb{E}_{y_{p,i}^d(\beta), y_{p,i}^r, a_i'}(\Phi_{i+1}\beta_{i+1}) \\ \geq\ & r_i(y_i^d, y_{p,i}^r, a_i) - V_i^{UB}(y_i) \qquad (5.98) \\ =\ & \ell_i(y_i^d, y_{p,i}^r, a_i, a_i'), \end{aligned}$$

where we obtain (5.98) by assuming that $\mathbb{E}_{y_{p,i}^d(\beta), y_{p,i}^r, a_i}(\Phi_{i+1}\beta_{i+1}) \geq 0$ (the exact value function is nonnegative) and replacing $r_i(y_i^d, y_{p,i}^r, a_i') + \mathbb{E}_{y_{p,i}^d(\beta), y_{p,i}^r, a_i'}(\Phi_{i+1}\beta_{i+1})$ by an upper bound $V_i^{UB}(y_i)$ on the exact value function at stage i and state y_i. We set $V_i^{UB}(y_i)$ equal to the discounted sum of rewards from selling up to the withdrawal capacity at each stage at the maximum spot price over all L sample paths, but other upper bounds on the exact value function could also be used.

For our computational experiments we use the Gurobi 5.0 (Gurobi Optimization, 2012) solver with up to 12 threads on a 64 bits PowerEdge R515 with twelve AMD Opteron 4176 2.4GHz processors with 64GB of memory, the Linux Fedora 17 operating system, and the gcc version 4.7.2 20120921 (Red Hat 4.7.2-2) compiler. We attempt to solve the twelve 6-month natural gas storage instances under default Gurobi settings and using the LSMV value function approximation as a warm start solution. We set a time limit of 48 hours. Gurobi failed to solve all the instances under these default settings. In particular, it was unable to solve the linear relaxation at the root node on all instances. Therefore, we tried

changing the root node solver from its default choice of barrier to dual simplex. This change was useful. Gurobi solved the root node relaxation in five out of the twelve instances by taking between 11 and 16 hours, but once again failed to prove optimality in all cases. Although some root node relaxations were solved, Gurobi reported numerical difficulties and changed to quadratic precision on all the instances. To address this numerical issue, we cleaned the coefficients of the mixed integer program to avoid values very close to zero and restrict all coefficients to have at most four decimal places. With this change, Gurobi solved the root node relaxation after roughly 1-10 hours for all the instances and also continued to solved the mixed integer program and prove the optimality of the LSMV value function approximation for five out of the twelve instances in 3-16 hours. The in-sample lower bound from the mixed integer program value function approximation was 0.5% larger than the LSMV in-sample lower bound estimate on three out of these five instances. The optimality gap of the unsolved instances ranged from 28% to 94%.

Our results show that the mixed integer program (5.92)-(5.97) proves the optimality of the LSMV value function approximation on roughly 42% of the instances that we considered, but remains quite challenging to solve for the remaining instances. This motivates future research into tighter formulations and/or more advanced solution techniques such as row or column generation.

5.8 Conclusions

In this chapter, we review some recent ADP methods for solving high dimensional finite horizon and discrete time SDPs arising in energy real options applications. Minimizing value function approximation error is a common objective for an LSM method, constrained-based and multiplier based ALP relaxations, the smoothed ALP, and the iterated Bellman linear program. We unify these ADP methods by deriving them as ALP relaxations obtained by restricting the ALP dual. The dual restrictions we use for unification are different strategies for avoiding the set of ALP dual optimal solutions from potentially under-exploring the state space. Our unification results identify an underlying thread among different ALP relaxations and the considered LSM method. We also review the distributionally robust ADP approach for heuristically minimizing greedy policy loss. Motivated by this method, we develop a mixed integer program that exactly minimizes policy loss for a certain class of structured SDPs arising in energy real options applications. We perform computational experiments on small natural gas storage instances and find that our mixed integer program is still challenging to solve, but is able to prove the known near optimality of the considered LSM value function approximation on some small instances. These results motivates further research into advanced solution techniques.

Chapter 6

Conclusions

The merchant operations of commodity and energy conversion assets is an important area of business applications that give rise to intractable MDPs, especially when using common practice-based price evolution models. This thesis leveraged tools from operations research and financial engineering to design novel ADP methods based on ALP relaxations, LSM, and mixed integer programming for approximating these MDPs. We provided theoretical support for the performance of some of these ADP methods. We benchmarked our ADP methods on realistic instances of existing and new applications, such as natural gas storage, crude oil swing options, and natural gas pipeline systems. Our methods were either competitive with or outperformed state-of-the-art techniques used in practice and academia, such as the rolling intrinsic approach and existing LSM variants. We also used an ALP relaxation perspective to unify different ADP methods.

Overall, our findings highlight the role of a math programing ADP approach for managing commodity and energy conversion assets. Although we focus on these assets, the techniques in this thesis have potential broader relevance for solving MDPs in other application contexts, such as inventory control with demand forecast updates, multiple sourcing, and optimal medical treatment design.

In §6.1, we summarize the insights gained from this thesis. In §6.2, we discuss directions for future research.

6.1 Summary of Insights

At a high level, this thesis explored the question of whether a financial engineering or math programming ADP approach should be used for approximating the value (continuation) function of SDPs with large exogenous state spaces. Such SDPs arise in the merchant operations of commodity and energy conversion assets. The financial engineering literature has a long history of approximating SDPs with large exogenous state spaces but small endogenous state spaces. The standard approach in this literature is LSM. In contrast, the operations research literature has focused on approximating SDPs with large endogenous state spaces but small exogenous state spaces. ALP is a popular operations research ADP

approach. We thus studied the LSM and ALP approaches for shedding light on the above question.

We identified that ALP may perform poorly when approximating the value function of SDPs, especially with large exogenous state spaces. We prescribed ALP relaxations as a way of overcoming this issue. Different from the recent work on ALP relaxations, our key observation was about the importance of correcting inconsistencies in the ALP dual to derive effective ALP relaxations. This led to our main result, a general ALP relaxation framework. We used this framework to derive effective ALP relaxations that delivered near optimal policies and tight lower and upper bounds on realistic natural gas storage instances. In addition, our best ALP relaxation was competitive with the standard LSM approach for lower bound estimation but computationally better suited for upper bound estimation.

Our next contribution was to overcome this upper bounding deficiency in the standard LSM approach, and a related issue with a recent LSM variant. By developing a new LSM variant, LSMV, we showed that upper bounds can be estimated efficiently by combining value function approximations with term structure price models, which are popular both in practice and academia for modeling commodity price evolution. LSMV was between one and three orders of magnitude faster than existing LSM variants for estimating upper bounds on the market value of realistic natural gas storage and crude oil swing option instances. Moreover, LSMV outperformed our best ALP relaxation model in both quality of the estimated lower and upper bounds and computational time.

The above findings suggests the use of LSMV over ALP relaxations. Interestingly, we showed that these seemingly different approaches are closely related. Specifically, our ALP relaxation framework based on correcting the ALP dual subsumes LSMV, as well as the existing ALP relaxations in the literature. Thus, the excellent performance of LSMV in fact supports the use of our ALP relaxation framework, even though LSMV has appealing computational behavior.

Although our ALP relaxations, including LSMV, are effective, they do not find a value function approximation that maximizes the greedy policy value (lower bound) estimate. To the best of our knowledge, other methods in the literature also handle this objective heuristically. This is a modeling drawback. We showed that a mixed integer program can be formulated to exactly handle this objective for a class of structured SDPs. Our mixed integer program is challenging to solve and motivates further research into reformulations and advanced solution techniques.

We also applied ADP to the novel application of jointly managing storage and transport assets in a pipeline system; the current literature studies these assets in isolation. We leveraged structural analysis to extend LSMV to compute near optimal operating policies and tight lower and upper bounds on realistic instances created in collaboration with a major natural gas trading company. We found that on these instances (i) the joint, rather than decoupled, merchant management of storage and transport assets has substantial value; (ii) this management can be nearly optimally simplified by prioritizing storage relative to transport, despite the considerable substitutability between these activities; (iii) the value due to price uncertainty is large but can be almost entirely captured by sequentially reop-

timizing a deterministic version of our MDP, an approach included in existing commercial software; and (iv) the value of transport trading across different pipelines is substantial.

Overall, our findings indicate that math programing provides a general framework for thinking about ADP methods for approximately solving SDPs with large exogenous state spaces. In particular, this approach subsumes LSM methods, which, however, based on our computational experiments appear to be the most effective methods for solving these SDPs.

6.2 Future Research Directions

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

Exploring the ALP relaxation framework of Chapter 2 from an upper bounding perspective is yet to be done. To this end, the multiplier-based ALP relaxations discussed in Appendix A.1 are appealing as they provide an upper bound on the optimal policy value that is no worse than the ALP upper bound, which has been used in the literature (see the discussion in §1.3.3). It would be interesting to quantify the upper bound improvement from using multiplier-based ALP relaxations instead of ALP. Theoretical bounds on the value function approximation error of these relaxations would further support their use.

Another research question related to ALP relaxations is the possibility of constructing them in an automated fashion. The ALP relaxations in this thesis were derived by adding pre-specified sets of constraints to the ALP dual. Instead, it may be possible to design an algorithm for adding constraints to the ALP dual in an adaptive manner by analyzing the current value function approximation obtained in a given iteration. Such an approach would provide greater flexibility when applying ALP relaxations.

As discussed in §5.7, additional research is also required to efficiently solve the mixed integer program (5.92)-(5.97). This could involve applying advanced solution techniques and exploring alternate reformulations.

The excellent performance of LSMV in our numerical experiments motivates studying this approach in more detail. Extending the applicability of LSMV to a broader class of price models that include features such as jumps and negative prices observed in electricity markets is practically useful. From a theoretical perspective, it would be interesting to investigate if LSMV is the best method that can be derived from the ALP relaxation (5.51)-(5.53), which we used to unify LSMV and the constraint-based ALP relaxations. Answering this question may also shed light on ways to further improve LSMV.

A broader methodological question relates to the selection of an approximation architecture for constructing value function approximations. Currently this selection is application specific. More research into automating this selection would reduce the barrier to apply ADP. For example, when representing a value function approximation as a linear combination of basis functions, we would want to automate the choice of basis functions.

Finally, the merchant management of a single storage asset in a natural gas pipeline system (see Chapter 4) could be extended to the case of several storage assets. It would be interesting to characterize the optimal storage policy structure in this setting and develop ADP methods for computing near optimal policies and estimating lower and upper bounds on the optimal policy value.

Appendix A

Additional Material for Chapter 2

A.1 Multiplier-based ALP Relaxations

This appendix discusses a model that belongs to a class of ALP relaxations that limits the amount by which an ALP is relaxed using objective function multipliers that weigh the variables that relax the ALP constraints. Hence, we label the ALP relaxations in this class as *multiplier-based* ALP relaxations. We propose derive a multiplier-based ALP relaxation by adding constraints to DALP that match the discounted first, second, and cross moments of the futures prices in every stage conditional on F_0. We refer to this ALP relaxations as MMR, where MM and R abbreviate moment matching and relaxation, respectively.

Every feasible DALP solution induces a discounted probability mass function on the feasible state and action spaces. Given a feasible solution w to DALP corresponding to ALP (2.26)-(2.28), the discounted probability of the stage i price pair $(s_i, F_{i,i+1})$ under this probability mass function is thus $\sum_{(x_i,a_i)} w_i(x_i, s_i, F_{i,i+1}, a_i)$. Recalling that $s_i \equiv F_{i,i}$, define $\{s_i, F_{i,i+1}\} \setminus \{F_{i,j}\}$ as $F_{i,i+1}$ if $j = i+1$, s_i if $j = i$, and $\{s_i, F_{i,i+1}\}$ otherwise. We restrict DALP by using the following three sets of discounted moment matching constraints:

$$\begin{aligned}
\sum_{F_{i,j}} F_{i,j} \sum_{\{s_i,F_{i,i+1}\}\setminus\{F_{i,j}\}} \sum_{(x_i,a_i)} w_i(x_i, s_i, F_{i,i+1}, a_i) &= \delta^i \mathbb{E}[F_{i,j}|F_{0,j}], \quad \forall i, j \in \{i, i+1\}, \\
\sum_{F_{i,j}} F_{i,j}^2 \sum_{\{s_i,F_{i,i+1}\}\setminus\{F_{i,j}\}} \sum_{(x_i,a_i)} w_i(x_i, s_i, F_{i,i+1}, a_i) &= \delta^i \mathbb{E}[F_{i,j}^2|F_{0,j}], \quad \forall i, j \in \{i, i+1\}, \\
\sum_{(s_i,F_{i,i+1})} s_i F_{i,i+1} \sum_{(x_i,a_i)} w_i(x_i, s_i, F_{i,i+1}, a_i) &= \delta^i \mathbb{E}[s_i F_{i,i+1}|F_{0,i}, F_{0,i+1}], \quad \forall i.
\end{aligned}$$

We denote by $d_{i,j}^f$, $d_{i,j}^s$, and d_i^c the dual variables corresponding to the constraints enforcing the first (f), second (s), and cross (c) discounted moments of the futures prices,

respectively, conditional on F_0. MMR is

$$\min_{\phi} \phi_0(x_0, s_0) + \sum_i \delta^i \left[\sum_{j \in \{i,i+1\}} (\mathbb{E}[F_{i,j}|F_{0,j}]d^f_{i,j} + \mathbb{E}[F^2_{i,j}|F_{0,j}]d^s_{i,j}) + \mathbb{E}[s_i F_{i,i+1}|F_{0,i}, F_{0,i+1}]d^c_i \right]$$

$$\text{s.t. } \phi_N(x_N, s_N) = 0, \forall x_N,$$

$$\phi_i(x_i, s_i) + \sum_{j \in \{i,i+1\}} F_{i,j}d^f_{i,j} + \sum_{j \in \{i,i+1\}} F^2_{i,j}d^s_{i,j} + s_i F_{i,i+1}d^c_i \geq$$

$$r(a_i, s_i) + \delta \mathbb{E}\left[\phi_{i+1}(x_i - a_i, s_{i+1})|F_{i,i+1}\right], \forall (i, x_i, s_i, F_{i,i+1}, a_i).$$

The second entry in the MMR objective function is the sum of the discounted expectations of the terms $\sum_{j \in \{i,i+1\}} F_{i,j}d^f_{i,j}$, $\sum_{j \in \{i,i+1\}} F^2_{i,j}d^s_{i,j}$, and $s_i F_{i,i+1}d^c_i$ that appear in the second set of MMR constraints. The presence of the sum of these terms in these constraints precludes the maximization over $F_{i,i+1}$ that is implicit in the ALP (2.26)-(2.28) from occurring in MMR.

The constraints added to DALP to derive MMR are implied by constraints (2.22), which match the probability mass function of the forward curve at each stage. In other words, (i) the restricted DALP feasible solutions are a subset of the DALP feasible solutions, and (ii) constraints (2.22) are violated by solutions that are in the DALP feasible set but not in the restricted DALP feasible set. Thus, more solutions in the restricted DALP feasible set satisfy constraints (2.22) than do solutions in the DALP feasible set. Moreover, as discussed at the beginning of §2.5.1, the implied nature of the constraints that restrict DALP ensures that the optimal objective function value of MMR, z^{MMR}, is a no weaker upper bound on the DDP value function at the initial stage and state than the ADP0 upper bound: $V^D_0(x_0, F_0) \leq z^{MMR} \leq \phi^{ADP0}_0(x_0, s_0)$.

We now briefly discuss the numerical performance of MMR applied to the instances used in §2.8 under the same computational setup. We formulate MMR using discretized price sets obtained in the same manner discussed in §2.7. Thus, the computational complexity of solving MMR using an interior point method is $O(N^{2.5} \cdot (m' \cdot |\mathcal{X}|)^{3.5} + L \cdot (N \cdot m' \cdot |\mathcal{X}|)^{0.5})$, where L is the number of bits required to store the input data (see §3 in Wright 1997 for details). This computational complexity is more onerous than the one associated with ADP2. The computational complexity of estimating lower and upper bounds using the MMR-based value function approximation can be read on the first row of Table 2.2.

Across all instances, the MMR-based upper bound estimates are between 1.50% and 12.86% larger than the worst upper bound estimates obtained with SDAP, ADP1, and ADP2. The MMR optimal objective function value is not competitive with UB2, being between 1.4 and 6 times larger than UB2, but it is at least 100 times smaller than the ADP0 optimal objective function value.

The MMR-based lower bound estimates are between 1.51% and 17.95% smaller than the worst lower bound estimates obtained with SADP, ADP1, and ADP2 across all the considered instances. Reoptimizing MMR yields better lower bounds, which are however, weaker than the other reoptimization based lower bounds, with optimality gaps ranging between 0.90% and 6.27% of UB2.

We now compare the CPU times required to estimate lower and upper bounds using MMR against the analogous CPU time requirements of SADP and ADP1, which also compute a value function approximation that depends at each stage on the inventory level and the spot price. We use Gurobi 5.0 (Gurobi Optimization, 2012) with a single thread for solving MMR. The CPU seconds required to solve MMR ranges from 2,444 to 9,546. Thus, solving MMR is at least 20 and 20,000 times slower than solving SADP and ADP1, respectively. The MMR overall CPU seconds, that is, also including the time required for bound estimation, vary from 2,457 to 9,601. Thus, using MMR is at least 7 and 140 times slower than using SADP and ADP1, respectively. The CPU seconds to estimate the MMR reoptimized lower bounds vary between 39,164 and 41,695, which are larger than the times required to estimate the other reoptimization based lower bounds by at least a factor of 30.

A.2 Proofs

Proof of Proposition 1. Suppose Assumption 2 is true. We proceed by induction to prove (2.19). The result is clearly true at stage 0. Suppose the result is also true for all stages $1, \ldots, i-1$. At stage i, for a given (i, x_i) the DALP constraint (2.15) corresponding to the first basis function, that is, $b = 1$, is

$$\sum_{(F_i,a_i)} w_i(x_i, F_i, a_i) = \delta \sum_{F_{i-1}} \underbrace{\sum_{F_i} \Pr(F_i|F_{i-1})}_{=1} \sum_{(x_{i-1},a_{i-1})} \mathbb{1}(x_{i-1} - a_{i-1} = x_i) w_{i-1}(x_{i-1}, F_{i-1}, a_{i-1}).$$

Summing over x_i on both sides of this constraint and simplifying gives

$$\begin{aligned}\sum_{(x_i,F_i,a_i)} w_i(x_i, F_i, a_i) &= \delta \sum_{(x_{i-1},F_{i-1},a_{i-1})} \sum_{x_i} \mathbb{1}(x_{i-1} - a_{i-1} = x_i) w_{i-1}(x_{i-1}, F_{i-1}, a_{i-1}) \\ &= \delta \sum_{(x_{i-1},F_{i-1},a_{i-1})} w_{i-1}(x_{i-1}, F_{i-1}, a_{i-1}) = \delta^i,\end{aligned}$$

where the last equality follows from the induction hypothesis. Condition (2.19) is thus true for stage i. The condition holds for all the stages by the principle of mathematical induction.

We proceed by contradiction to prove (2.20). Suppose there exists an optimal solution w^* to DALP and a forward curve $\bar{F}_i \notin \mathcal{F}_i^{=}(\beta^*)$ such that $\sum_{(x_i,a_i)} w_i^*(x_i, \bar{F}_i, a_i) > 0$. This implies that there exists at least one pair $(\bar{x}_i, \bar{a}_i)$ such that $w_i^*(\bar{x}_i, \bar{F}_i, \bar{a}_i) > 0$. Since (2.16) and (2.19) imply that the feasible set of DALP is bounded, we can write $w^* = \sum_{j\in\mathcal{J}} \lambda_j w^j$, where $\sum_{j\in\mathcal{J}} \lambda_j = 1$, $\lambda \geq 0$, $\mathcal{J}$ is the index set for the set of basic feasible solutions, and w^j is the j-th basic solution. The optimality of w^* implies that every w^j such that $\lambda_j > 0$ must also be a basic optimal solution. Further, the inequality $w_i^*(\bar{x}_i, \bar{F}_i, \bar{a}_i) > 0$ implies that there must be at least one basic optimal solution w^j such that $w_i^j(\bar{x}_i, \bar{F}_i, \bar{a}_i) > 0$. It follows from complementary slackness that the primal constraint corresponding to $(i, \bar{x}_i, \bar{F}_i, \bar{a}_i)$ holds as

an equality, which contradicts $\bar{F}_i \notin \mathcal{F}_i^{=}(\beta^*)$. □

Proof of Proposition 2. Suppose there exists a DDP optimal policy and a DALP optimal solution that satisfy (2.18). This implies that the DLP and DALP optimal objective function values match. As discussed in §2.3, DALP is a relaxation of DLP. Hence, every optimal solution of DLP is also optimal for DALP. Let w^* be the basic DALP optimal solution that corresponds to the deterministic DDP optimal policy π^*. Since π^* is deterministic it can be equivalently represented by the set of stage-state-action tuples $\mathcal{K} := \{(i, x_i, F_i, a_i) : w_i^*(x_i, F_i, a_i) > 0\}$. By complementary slackness, the ALP constraints corresponding to tuples in $\mathcal{K}$ hold as equalities. Hence, for the stage-state-action tuple $(i, x_i, F_i, a_i) \in \mathcal{K}$ the action a_i is greedy optimal at stage i and state (x_i, F_i) with respect to the value function approximation corresponding to β^*. Therefore, starting from state (x_0, F_0) in stage 0, at each visited state in each stage we can choose an action that is greedy optimal relative to the value function approximation given by β^* such that the encountered stage-state-action tuples belong to $\mathcal{K}$. Hence, it holds that $\pi^* \in \Pi^g(\beta^*)$. □

Proof of Proposition 3. The constraints (2.27)-(2.28) can be equivalently rewritten as

$$\phi_{N-1}(x_{N-1}, s_{N-1}) \geq \max_{a_{N-1}} r(a_{N-1}, s_{N-1}), \forall (x_{N-1}, s_{N-1}), \tag{A.1}$$

$$\phi_i(x_i, s_i) \geq \max_{F_{i,i+1}} \left\{ \max_{a_i} r(a_i, s_i) + \delta \mathbb{E}\left[\phi_{i+1}(x_i - a_i, s_{i+1}) | F_{i,i+1}\right] \right\}, \quad \forall (i, x_i, s_i)_{-(N-1)}. \tag{A.2}$$

These inequalities hold as equalities when evaluated using ϕ^{ADP0}. Moreover, (i) the variable $\phi_{N-1}(x_{N-1}, s_{N-1})$ appears on the left hand side of inequalities (A.1) is multiplied by positive coefficients; and (ii) the variable $\phi_i(x_i, s_i)$ is multiplied by positive coefficients on both the left hand sides of the inequalities (A.2) corresponding to (i, x_i, s_i) and the right hand sides of the inequalities (A.2) corresponding to stage $i-1$. Therefore, a feasible solution of (2.26)-(2.28) for which the constraints (A.1)-(A.2) do not hold as equalities has an objective function value greater than or equal to $\phi_0^{ADP0}(x_0, s_0)$. Hence, ϕ^{ADP0} is an optimal solution of (2.26)-(2.28). □

Proof of Proposition 4. The constraints (2.36) provide lower bounds on the $d_i(x_i, s_i, F_{i,i+1})$ variables. Substituting this lower bound in (2.37) yields the following inequalities

$$\phi_i(x_i, s_i) \geq \sum_{F_{i,i+1}} p(F_{i,i+1} | s_i, F_0) \left[\max_{a_i} r(a_i, s_i) + \delta \mathbb{E}\left[\phi_{i+1}(x_i - a_i, s_{i+1}) | F_{i,i+1}\right] \right], \forall (i, x_i, s_i).$$

The solution (ϕ^p, d^p) is feasible to (2.34)-(2.37) and makes these inequalities hold as equalities. The rest of the proof is analogous to the proof of Proposition 3. □

Proof of Proposition 5. (a) Summing both sides of (2.33) over x_i with $p(F_{i,i+1}|s_i, F_0) =$

$\Pr(F_{i,i+1}|s_i, F_{0,i+1})$ gives

$$\sum_{(x_i,a_i)} w_i(x_i, s_i, F_{i,i+1}, a_i) = \Pr(F_{i,i+1}|s_i, F_0) \sum_{x_i} \theta_i(x_i, s_i), \forall (i, s_i, F_{i,i+1}). \tag{A.3}$$

Comparing the right hand sides of (A.3) and (2.43) shows that it suffices to prove that the equality $\sum_{x_i} \theta_i(x_i, s_i) = \delta^i \Pr_i(s_i, F_0)$ holds for each stage i and spot price s_i and for every feasible solution to the dual of ALP (2.26)-(2.28). We proceed by induction. This equality holds at stage 0 by (2.13) and (2.33). Suppose this equality is also true for stages 0 through $i-1$. Consider stage i. When using the look-up table value function approximation $\phi_i(x_i, s_i)$, constraints (2.15) can be written as

$$\sum_{(F_{i,i+1},a_i)} w_i(x_i, s_i, F_{i,i+1}, a_i) =$$
$$\delta \sum_{(s_{i-1},F_{i-1,i})} \Pr(s_i|F_{i-1,i}) \sum_{(x_{i-1},a_{i-1})} \mathbb{1}(x_{i-1} - a_{i-1} = x_i) w_{i-1}(x_{i-1}, s_{i-1}, F_{i-1,i}, a_{i-1}),$$
$$\forall (i, x_i, s_i)_{-(0)}.$$

Summing over x_i on both sides of these constraints and simplifying gives

$$\sum_{(x_i,F_{i,i+1},a_i)} w_i(x_i, s_i, F_{i,i+1}, a_i) =$$
$$\delta \sum_{(s_{i-1},F_{i-1,i})} \Pr(s_i|F_{i-1,i}) \sum_{(x_{i-1},a_{i-1})} w_{i-1}(x_{i-1}, s_{i-1}, F_{i-1,i}, a_{i-1}). \tag{A.4}$$

Replacing the terms $\sum_{a_i} w_i(x_i, s_i, F_{i,i+1}, a_i)$ and $\sum_{a_{i-1}} w_{i-1}(x_{i-1}, s_{i-1}, F_{i-1,i}, a_{i-1})$ in (A.4) using (2.33) and simplifying the resulting equality yields

$$\begin{aligned}
\sum_{x_i} \theta_i(x_i, s_i) &= \delta \sum_{s_{i-1}} \left(\sum_{F_{i-1,i}} \Pr(s_i|F_{i-1,i}) \Pr(F_{i-1,i}|s_{i-1}, F_{0,i}) \right) \sum_{x_{i-1}} \theta_{i-1}(x_{i-1}, s_{i-1}) \\
&= \delta \sum_{s_{i-1}} \Pr(s_i|s_{i-1}, F_{0,i}) \sum_{x_{i-1}} \theta_{i-1}(x_{i-1}, s_{i-1}) \\
&= \delta \sum_{s_{i-1}} \Pr(s_i|s_{i-1}, F_{0,i}) \delta^{i-1} \Pr(s_{i-1}, F_0) = \delta^i \Pr(s_i, F_0),
\end{aligned}$$

where the third equality follows from our induction hypothesis: $\sum_{x_{i-1}} \theta_{i-1}(x_{i-1}, s_{i-1}) = \delta^{i-1} \Pr(s_{i-1}, F_0)$. Therefore, the stated claim holds for all the stages by the principle of mathematical induction.

(b) Summing over x_i on both sides of (2.33) with $p(F_{i,i+1}|s_i, F_0) = \mathbb{1}(F_{i,i+1} = \mathbb{E}[F_{i,i+1}|s_i, F_{0,i+1}])$ yields $\sum_{(x_i,a_i)} w_i(x_i, s_i, F_{i,i+1}, a_i) = \mathbb{1}(F_{i,i+1} = \mathbb{E}[F_{i,i+1}|s_i, F_{0,i+1}]) \sum_{x_i} \theta_i(x_i, s_i)$. Summing over $(x_i, F_{i,i+1})$ on both sides of (2.33) gives $\sum_{(x_i,F_{i,i+1},a_i)} w_i(x_i, s_i, F_{i,i+1}, a_i) = \sum_{x_i} \theta_i(x_i, s_i)$.

Therefore, it follows that

$$\sum_{F_{i,i+1}} F_{i,i+1} \left(\frac{\sum_{(x_i,a_i)} w_i(x_i, s_i, F_{i,i+1}, a_i)}{\sum_{(x_i,F_{i,i+1},a_i)} w_i(x_i, s_i, F_{i,i+1}, a_i)} \right) = \mathbb{E}[F_{i,i+1}|s_i, F_{0,i+1}].$$

□

Proof of Proposition 6. (i) Define $(\cdot)^+ := \max(0, \cdot)$. It holds that $\phi_{N-1}^{ADP0}(x_{N-1}, s_{N-1}) = (\alpha^W s_{N-1} - c^W)^+ x_{N-1}$ for all $x_{N-1} \in \mathcal{X}$ and $s_{N-1} \in \mathbb{R}_+$, since $\phi_N^{ADP0}(x_N, s_N) \equiv 0$ for all $x_N \in \mathcal{X}$. At stage $N-2$ for $x_{N-2} \in \mathcal{X} \setminus \{0\}$ we have

$$\begin{aligned}
\phi_{N-2}^{ADP0}(x_{N-2}, s_{N-2}) &= \sup_{F_{N-2,N-1} \in \mathbb{R}_+} \Bigg\{ \max_{a \in \mathcal{A}(x_{N-2})} r(a, s_{N-2}) \\
&\qquad + \delta \mathbb{E}\left[\phi_{N-1}^{ADP0}(x_{N-2} - a, s_{N-1}) | F_{N-2,N-1} \right] \Bigg\} \\
&= \sup_{F_{N-2,N-1} \in \mathbb{R}_+} \Bigg\{ \max_{a \in \mathcal{A}(x_{N-2})} r(a, s_{N-2}) \\
&\qquad + \alpha^W \delta (x_{N-2} - a) \mathbb{E}\left[\left(s_{N-1} - \frac{c^W}{\alpha^W} \right)^+ | F_{N-2,N-1} \right] \Bigg\} \\
&\geq \alpha^W \delta x_{N-2} \sup_{F_{N-2,N-1} \in \mathbb{R}_+} \mathbb{E}\left[\left(s_{N-1} - \frac{c^W}{\alpha^W} \right)^+ | F_{N-2,N-1} \right],
\end{aligned}$$

where we obtain the inequality by noting that the do-nothing decision is feasible and from $r(0, s_{N-2}) = 0$. The term $\mathbb{E}\left[\left(s_{N-1} - c^W \alpha^W \right)^+ | F_{N-2,N-1} \right]$ is an increasing function of $F_{N-2,N-1}$ under the assumption that the conditional distribution of s_{N-1} given $F_{N-2,N-1}$ is stochastically increasing in $F_{N-2,N-1}$ (Topkis 1998, Corollary 3.9.1 (a)). It follows that $\phi_{N-2}^{ADP0}(x_{N-2}, s_{N-2}) = \infty$, for all $x_{N-2} \in \mathcal{X} \setminus \{0\}$ and $s_{N-2} \in \mathbb{R}_+$. To show that $\phi_{N-2}^{ADP0}(0, s_{N-2}) = \infty$ we use a similar argument with the feasible action equal to C^I instead of 0. It follows that $\phi_i^{ADP0}(x_i, s_i)$ is also equal to infinity for all $i \in \mathcal{I}$, $x_i \in \mathcal{X}$, and $s_i \mathbb{R}_+$.

(ii) The proof for this part is similar to the proof of part (i).

(iii) We have $0 \leq V_i(x_i, F_i) \leq \bar{x} \mathbb{E}\left[\left(\sum_{j=i}^{N-1} \delta^{j-i} s_j \right) \middle| F_i \right] = \bar{x} \left(\sum_{j=i}^{N-1} \delta^{j-i} F_{ij} \right)$, since the first inequality holds because doing nothing at every stage and state is a feasible policy with zero value; the second inequality is true because the stage i value of selling $\bar{x}$ at every stage without incurring the withdrawal loss and marginal cost provides a trivial upper bound on the stage i optimal value function; and the equality follows from the property that $\mathbb{E}[s_j|F_i]$ is equal to $F_{i,j}$ under a risk neutral measure (Shreve, 2004, page 244). It can be shown in an analogous manner that the value functions of the ADPs in set $\mathcal{L}$ are bounded. We omit these derivations for brevity. □

We use Lemma 4 to derive inequality (2.47).

Lemma 4. *Let the functions f and g be defined on a finite set $\mathcal{Z}$. It holds that* $|\max_{z\in\mathcal{Z}} f(z) - \max_{z\in\mathcal{Z}} g(z)| \leq \max_{z\in\mathcal{Z}} |f(z) - g(z)|$.

Proof. Let $z_1 \in \operatorname{argmax}_{z\in\mathcal{Z}} f(z)$ and $z_2 \in \operatorname{argmax}_{z\in\mathcal{Z}} g(z)$. It holds that $f(z_1) - g(z_2) \leq f(z_1) - g(z_1) \leq \max_{z\in\mathcal{Z}}\{f(z) - g(z)\} \leq \max_{z\in\mathcal{Z}} |f(z) - g(z)|$. Following the same steps starting from $g(z_2) - f(z_1)$ yields $g(z_2) - f(z_1) \leq \max_{z\in\mathcal{Z}} |f(z) - g(z)|$. □

Derivation of Inequality (2.47). Consider ADP1. To prove the claimed bound it suffices to prove that

$$|V_i(x_i, F_i) - \phi_i^{ADP1}(x_i, s_i)| \leq \gamma_i^{ADP1}(x_i, F_i), \tag{A.5}$$

holds for all (i, x_i, F_i), because $\gamma_i^{ADP1}(x_i, F_i) \geq 0$. We establish (A.5) by induction on the number of stages. Inequality (A.5) holds as an equality at stage $N-1$ with both sides equal to zero. Make the induction hypothesis that this inequality also holds for stages $i+1, \ldots, N-2$. At stage i we have

$$\begin{aligned} |V_i(x_i, F_i) - \phi_i^{ADP1}(x_i, s_i)| \leq\ & |V_i(x_i, F_i) - \phi_i^{ADP1,V}(x_i, s_i)| \\ & + |\phi_i^{ADP1,V}(x_i, s_i) - \phi_i^{ADP1}(x_i, s_i)|. \end{aligned} \tag{A.6}$$

We bound $|\phi_i^{ADP1,V}(x_i, s_i) - \phi_i^{ADP1}(x_i, s_i)|$ as

$$\begin{aligned} |\phi_i^{ADP1,V}(x_i, s_i) - \phi_i^{ADP1}(x_i, s_i)| &= |\max_{a_i} r(a_i, s_i) + \delta\mathbb{E}[V_{i+1}(x_i - a_i, F_{i+1})|\bar{F}_i'(s_i, F_0)] \\ &\quad - \max_{a_i} r(a_i, s_i) + \delta\mathbb{E}[\phi_{i+1}^{ADP1}(x_i - a_i, s_{i+1})|\bar{F}_i'(s_i, F_0)]| \\ &\leq \delta \max_{a_i} |\mathbb{E}[V_{i+1}(x_i - a_i, F_{i+1}) \\ &\qquad - \phi_{i+1}^{ADP1}(x_i - a_i, s_{i+1})|\bar{F}_i'(s_i, F_0)]| \\ &\leq \delta \max_{x_{i+1}} \mathbb{E}[|V_{i+1}(x_{i+1}, F_{i+1}) - \phi_{i+1}^{ADP1}(x_{i+1}, s_{i+1})||\bar{F}_i'(s_i, F_0)] \\ &\leq \delta \max_{x_{i+1}} \mathbb{E}[\gamma_{i+1}^{ADP1}(x_{i+1}, F_{i+1})|\bar{F}_i'(s_i, F_0)], \end{aligned} \tag{A.7}$$

where the first inequality follows from Lemma 4, the second from the modulus inequality (Resnick, 1999, page 128) and $x_i - a_i \in \mathcal{X}$, and the third from the induction hypothesis. Using (A.6) and (A.7) and $\gamma_i^{ADP1}(x_i, F_i)$ yields $|V_i(x_i, F_i) - \phi_i^{ADP1}(x_i, s_i)| \leq \gamma_i^{ADP1}(x_i, F_i)$. The validity of inequality (A.5) in all other stages follows from the principle of mathematical induction. The proofs of the bounds for the remaining ADPs are similar and are omitted for brevity. □

We use Lemmas 5-7 in the proof of Proposition 7. We define $\rho_i := \{\rho_{i,j}, j > i\}$ and use the vector expressions $\rho_i = 1$ and $\rho_i \to 1$ instead of $\rho_{i,j} = 1, \forall j > i$ and $\rho_{i,j} \to 1, \forall j > i$, respectively, where 1 is a compatible vector of all ones in each case. For $i > j$, let $\bar{F}_i'(s_i, F_j)$ be shorthand notation for expectation $\mathbb{E}\left[F_i'|s_i, F_j\right]$, where the stage i futures price vector F_i' is the random variable and the conditioning information is the stage i spot price and the stage j forward curve. Similarly, we denote expectation $\mathbb{E}\left[F_i'|s_i, \bar{F}_{i-1}'(s_{i-1}, F_0)\right]$ by

$\bar{F}_i'(s_i, s_{i-1}, F_0)$. Building on this notation, we use $\bar{F}_i'^{(\rho_i=1)}(s_i, F_j)$ and $\bar{F}_i'^{(\rho_i=1)}(s_i, s_{i-1}, F_0)$ to denote the limits $\lim_{\rho_i \to 1} \mathbb{E}\left[F_i' | s_i, F_j\right]$ and $\lim_{\rho_i \to 1} \mathbb{E}\left[F_i' | s_i, \bar{F}_{i-1}'(s_{i-1}, F_0)\right]$, respectively.

Lemma 5. *Under price model (2.4)-(2.5):*

(a) The function $V_i(x_i, \cdot)$ is continuous. Given i and x_i, the random variable $V_i(x_i, F_i)$ is uniformly integrable;

(b) Given s_i and F_j, $i > j$, F_i' converges in distribution to the constant $\bar{F}_i'^{(\rho_i=1)}(s_i, F_j)$ when $\rho_i \to 1$;

(c) Given s_i and $\bar{F}_{i-1}'(s_{i-1}, F_0)$, F_i' converges in distribution to the constant $\bar{F}_i'^{(\rho_i=1)}(s_i, s_{i-1}, F_0)$ when $\rho_i \to 1$;

(d) Given s_i and $\bar{F}_{i-1}'(s_{i-1}, F_0)$, the constant $\bar{F}_i'^{(\rho_i=1)}(s_i, s_{i-1}, F_0)$ converges to the constant $\bar{F}_i'^{(\rho_i=1)}(s_i, F_0)$ as $\rho_{i-1} \to 1$;

(e) Given s_i and $\bar{F}_{i-1}'(s_{i-1}, F_0)$, $\bar{F}_i'(s_i, F_0)$ converges in distribution to the constant $\bar{F}_i'^{(\rho_i=1)}(s_i, F_0)$ as $\rho_i \to 1$.

Proof. (a) The continuity of $V_i(x_i, \cdot)$ follows from Proposition 5 in Secomandi et al. (2012). The uniform integrability of $V_i(x_i, F_i)$ holds by part (iii) of Proposition 6 and the fact that futures prices are uniformly integrable under price model (2.4)-(2.5) (see the dominated families criterion in Resnick 1999, page 183).

(b)-(e) In each case, pick an element of the random vector. This random variable is lognormal with mean and variance that are functions of the volatilities and instantaneous correlations of price model (2.4)-(2.5). In the limit, it can be easily verified that the variance function tends to zero and the mean function tends to its respective claimed constant. Thus, we have convergence in distribution (see Resnick 1999, page 249). □

Lemma 6. *Let $f(\cdot)$ be a real valued and continuous function and X_n a sequence of uniformly integrable random vectors. Suppose that the random variable $f(X_n)$ is uniformly integrable and $\lim_{n\to\infty} X_n$ converges in distribution to the constant vector $\bar{X}$. Then, $\lim_{n\to\infty} \mathbb{E}[|f(X_n) - f(\bar{X})|] = 0$.*

Proof. Because convergence in distribution to a constant implies convergence in probability (see Proposition 8.5.2 in Resnick 1999), we have that $\lim_{n\to\bar{n}} X_n$ converges in probability to the constant vector $\bar{X}$. Using this result and the continuity of f, it follows from part (ii) of Corollary 6.3.1 in Resnick (1999) that $f(X_n)$ converges in probability to the constant $f(\bar{X})$. This result and the uniform integrability of $f(X_n)$ allows us to use Theorem 6.6.1 in Resnick (1999) to prove the claimed result. □

Lemma 7 holds by the properties of (multivariate) lognormal random variables.

Lemma 7. *Under price model (2.4)-(2.5), it holds for $j > i > k$ that*

$$\bar{F}_{i,j}(s_i, F_k) = F_{k,j}\left(\frac{s_i}{F_{k,i}}\right)^{\frac{\rho_{i,j}\sigma_j}{\sigma_i}} \exp\left(\frac{\rho_{i,j}\sigma_j(i-k)\Delta t}{2}(\sigma_i - \rho_{i,j}\sigma_j)\right); \tag{A.8}$$

$$\mathbb{E}\left[s_i^{\sigma_j/\sigma_i}|F_{i-1,i}\right] = F_{i-1,i}^{\sigma_j/\sigma_i} \exp\left(\frac{\sigma_i\Delta t}{2}(\sigma_j - \sigma_i)\right); \tag{A.9}$$

$$\begin{aligned}\mathbb{E}\left[F_{i-1,i}^{\sigma_j/\sigma_i}\middle| s_{i-1}, F_0\right] &= F_{0,i}^{\sigma_j/\sigma_i}\left(\frac{s_{i-1}}{F_{0,i-1}}\right)^{\frac{\sigma_j\rho_{i-1,i}}{\sigma_{i-1}}} \\ &\exp\left(\frac{\sigma_j(i-1)\Delta t}{2}(\rho_{i-1,i}\sigma_{i-1} - \sigma_i + \sigma_j(1-\rho_{i-1,i}^2))\right).\end{aligned} \tag{A.10}$$

Proof of Proposition 7. (i) Consider ADP1. Note that the equivalence $\phi_i^{ADP1,V}(x_i, s_i) \equiv V_i(x_i, s_i, \bar{F}_i'(s_i, F_0))$ follows from the definitions of $\phi^{\ell,V}$ in §2.6 and V_i in (2.3). At a stage i, using this equivalence and the definition of γ_i^{ADP1} we have

$$\begin{aligned}\|\gamma_i^{ADP1}\|_{\mathbb{E},\infty} \equiv &\max_{x_i} \mathbb{E}\left[|V_i(x_i, F_i) - V_i(x_i, s_i, \bar{F}_i'(s_i, F_0))|\middle|F_0\right] \\ &+ \delta\mathbb{E}\left[\max_{x_{i+1}} \mathbb{E}\left[\gamma_i^{ADP1}(x_{i+1}, F_{i+1})|\bar{F}_i'(s_i, F_0)\right]\middle|F_0\right].\end{aligned} \tag{A.11}$$

We now show that the right hand side of (A.11) tends to zero as $\rho \to 1$. For the first term on the right hand side of (A.11), we have

$$\begin{aligned}&\lim_{\rho\to 1}\max_{x_i} \mathbb{E}\left[|V_i(x_i, F_i) - V_i(x_i, s_i, \bar{F}_i'(s_i, F_0))|\middle|F_0\right] \\ &\quad\le \max_{x_i}\lim_{\rho\to 1} \mathbb{E}\left[\mathbb{E}\left[|V_i(x_i, F_i) - V_i(x_i, s_i, \bar{F}_i^{\prime(\rho_i=1)}(s_i, F_0))|\middle|s_i, F_0\right]\middle|F_0\right] \\ &\qquad + \max_{x_i}\lim_{\rho\to 1} \mathbb{E}\left[\mathbb{E}\left[|V_i(x_i, s_i, \bar{F}_i^{\prime(\rho_i=1)}(s_i, F_0)) - V_i(x_i, s_i, \bar{F}_i'(s_i, F_0))|\middle|s_i, F_0\right]\middle|F_0\right] \\ &\quad= \max_{x_i}\lim_{\rho\to 1} \mathbb{E}\left[\mathbb{E}\left[|V_i(x_i, F_i) - V_i(x_i, s_i, \bar{F}_i^{\prime(\rho_i=1)}(s_i, F_0))|\middle|s_i, F_0\right]\middle|F_0\right] \\ &\quad= 0;\end{aligned}$$

where the inequality follows from swapping the limit and maximization, which is allowed because the maximization is over the finite set $\mathcal{X}$, iterating expectation and conditioning on s_i, and applying the triangle inequality; the first equality holds by the continuity of $V_i(x_i, \cdot)$ established in part (a) of Lemma 5 and the fact that, given s_i and F_0, $\bar{F}_i'(s_i, F_0)$ is a continuous function of ρ_i that tends to $\bar{F}_i^{\prime(\rho_i=1)}(s_i, F_0)$ as $\rho_i \to 1$ (see (A.8) in Lemma 7); and the final equality follows by applying parts (a) and (b) of Lemma 5 and Lemma 6. Thus, the limit of the first term on the right hand side of (A.11) must be zero.

The second term on the right hand side of (A.11) can be rewritten as an expression that depends only on the exact value function V_i by using the recursive definition of γ_i^{ADP1}. This expression is a sum of terms, one for each $j \in \{i+1, \ldots, N-2\}$, where the j-th term

is a sequence of iterated expectations and the inner-most expectation is

$$\mathbb{E}\left[\left|V_j(x_j, s_j, F'_j) - V_j(x_j, s_j, \bar{F}'_j(s_j, F_0))\right| \middle| \bar{F}'_{j-1}(s_{j-1}, F_0)\right]. \tag{A.12}$$

We show that the limit of (A.12) when $\rho_{j-1} \to 1$ and $\rho_j \to 1$ is zero for any fixed x_j as follows (we write the limit explicitly once and then suppress its argument in the remaining expressions):

$$\begin{aligned}
&\lim_{\substack{\rho_{j-1}\to 1\\ \rho_j \to 1}} \mathbb{E}\left[\left|V_j(x_j, s_j, F'_j) - V_j(x_j, s_j, \bar{F}'_j(s_j, F_0))\right| \middle| \bar{F}'_{j-1}(s_{j-1}, F_0)\right] \\
&= \lim \mathbb{E}\left[\mathbb{E}\left[\left|V_j(x_j, s_j, F'_j) - V_j(x_j, s_j, \bar{F}'_j(s_j, F_0))\right| \middle| s_j, \bar{F}'_{j-1}(s_{j-1}, F_0)\right] \middle| \bar{F}'_{j-1}(s_{j-1}, F_0)\right] \\
&\leq \lim \mathbb{E}\Big[\mathbb{E}\Big[\Big|V_j(x_j, s_j, F'_j) \\
&\qquad\qquad -V_j(x_j, s_j, \bar{F}'^{(\rho_j=1)}_j(s_j, s_{j-1}, F_0))\Big| \Big| s_j, \bar{F}'_{j-1}(s_{j-1}, F_0)\Big] \Big| \bar{F}'_{j-1}(s_{j-1}, F_0)\Big] \\
&\quad + \lim \mathbb{E}\Big[\mathbb{E}\Big[\Big|V_j(x_j, s_j, \bar{F}'^{(\rho_j=1)}_j(s_j, s_{j-1}, F_0)) \\
&\qquad\qquad -V_j(x_j, s_j, \bar{F}'^{(\rho_j=1)}_j(s_j, F_0))\Big| \Big| s_j, \bar{F}'_{j-1}(s_{j-1}, F_0)\Big] \Big| \bar{F}'_{j-1}(s_{j-1}, F_0)\Big] \\
&\quad + \lim \mathbb{E}\Big[\mathbb{E}\Big[\Big|V_j(x_j, s_j, \bar{F}'^{(\rho_j=1)}_j(s_j, F_0)) \\
&\qquad\qquad -V_j(x_j, s_j, \bar{F}'_j(s_j, F_0))\Big| \Big| s_j, \bar{F}'_{j-1}(s_{j-1}, F_0)\Big] \Big| \bar{F}'_{j-1}(s_{j-1}, F_0)\Big] \\
&= 0,
\end{aligned}$$

where the first equality follows from iterating expectation and conditioning on s_j; the first inequality from applying the triangle inequality twice; and the second equality from Lemma 6, part (a) of Lemma 5, and parts (c)-(e) of Lemma 5. Thus, the limit of the second term in the right hand side of (A.11) must be zero, which completes the proof for ADP1. The proof for ADP2 is similar and is thus omitted for brevity.

Consider SADP. At stage i, by definition of γ_i^{SADP} we have

$$\begin{aligned}
\|\gamma_i^{SADP}\|_{\mathbb{E},\infty} = &\max_{x_i} \mathbb{E}\left[\left|V_i(x_i, F_i) - V_i^{SADP}(x_i, s_i)\right| \middle| F_0\right] \\
&+ \delta \mathbb{E}\left[\mathbb{E}\left[\max_{x_{i+1}} \mathbb{E}\left[\gamma_{i+1}^{SADP}(x_{i+1}, F_{i+1}) | F'_i\right]\right] \middle| s_i, F_0\right] \middle| F_0\right].
\end{aligned} \tag{A.13}$$

We show that each of the terms on the right hand side of (A.13) tends to zero as $\rho \to 1$. The proof of this result for the first term is similar to the given proof for ADP1 above, and is thus omitted. The second term can be rewritten as an expression involving only the exact value function V_i using the recursive definition of γ_i^{SADP}. This expression is a sum of terms, one for each $j \in \{i+1, \ldots, N-2\}$, where the j-th term is a sequence of iterated expectations and the inner-most expectation is

$$\mathbb{E}\left[\max_{x_j} \mathbb{E}\left[\left|V_j(x_j, s_j, F'_j) - \phi_j^{SADP,V}(x_j, s_j)\right| \middle| F'_{j-1}\right] \middle| s_{j-1}, F_0\right]. \tag{A.14}$$

We show the limit of (A.14) as $\rho_{j-1} \to 1$ and $\rho_j \to 1$ is zero for any fixed x_j. We have:

$$
\begin{aligned}
&\lim_{\substack{\rho_{j-1}\to 1\\ \rho_j \to 1}} \mathbb{E}\left[\mathbb{E}\left[|V_j(x_j, s_j, F_j') - \phi_j^{SADP,V}(x_j, s_j)|\middle| F_{j-1}'\right]\middle| s_{j-1}, F_0\right] \\
&= \lim_{\substack{\rho_{j-1}\to 1\\ \rho_j \to 1}} \mathbb{E}\left[\mathbb{E}\left[\mathbb{E}\left[|V_j(x_j, s_j, F_j') - \phi_j^{SADP,V}(x_j, s_j)|\middle| s_j, F_{j-1}'\right]\middle| F_{j-1}'\right]\middle| s_{j-1}, F_0\right] \\
&\le \lim_{\substack{\rho_{j-1}\to 1\\ \rho_j \to 1}} \mathbb{E}\left[\mathbb{E}\left[\mathbb{E}\left[|V_j(x_j, s_j, F_j') - V_j(x_j, s_j, \bar{F}_j'^{(\rho_j=1)}(s_j, F_{j-1}'))|\middle| s_j, F_{j-1}'\right]\middle| F_{j-1}'\right]\middle| s_{j-1}, F_0\right] \\
&\quad + \lim_{\substack{\rho_{j-1}\to 1\\ \rho_j \to 1}} \mathbb{E}\left[\mathbb{E}\left[\mathbb{E}\left[|V_j(x_j, s_j, \bar{F}_j'^{(\rho_j=1)}(s_j, F_{j-1}')) \right.\right.\right. \\
&\qquad\qquad \left.\left.\left. - V_j(x_j, s_j, \bar{F}_j'^{(\rho_j=1)}(s_j, F_0))|\middle| s_j, F_{j-1}'\right]\middle| F_{j-1}'\right]\middle| s_{j-1}, F_0\right] \\
&\quad + \lim_{\substack{\rho_{j-1}\to 1\\ \rho_j \to 1}} \mathbb{E}\left[\mathbb{E}\left[\mathbb{E}\left[|V_j(x_j, s_j, \bar{F}_j'^{(\rho_j=1)}(s_j, F_0)) \right.\right.\right. \\
&\qquad\qquad \left.\left.\left. - \phi_j^{SADP,V}(x_j, s_j)|\middle| s_j, F_{j-1}'\right]\middle| F_{j-1}'\right]\middle| s_{j-1}, F_0\right] \qquad \text{(A.15)} \\
&= \lim_{\rho_{j-1}\to 1} \mathbb{E}\left[\mathbb{E}\left[|V_j(x_j, s_j, \bar{F}_j'^{(\rho_j=1)}(s_j, F_{j-1}')) - V_j(x_j, s_j, \bar{F}_j'^{(\rho_j=1)}(s_j, F_0))|\middle| F_{j-1}'\right]\middle| s_{j-1}, F_0\right] \\
&\le C' \sum_{k=j+1}^{N-2} \lim_{\rho_{j-1}\to 1} \mathbb{E}\left[\mathbb{E}\left[|\bar{F}_{j,k}'^{(\rho_{j,k}=1)}(s_j, F_{j-1}') - \bar{F}_{j,k}^{(\rho_{j,k}=1)}(s_j, F_0)|\middle| F_{j-1,j}, F_{j-1,k}\right]\middle| s_j, F_{0,j}, F_{0,k}\right] \\
&:= \text{RHS},
\end{aligned}
$$

where the first equality follows from iterated expectation and conditioning on s_j; the first inequality from applying the triangle inequality twice; the second equality from (i) the first term in (A.15) being zero by parts (a) and (b) of Lemma 5 and Lemma 6, (ii) unconditioning on s_j in the second term of (A.15), and (iii) the third term in (A.15) being zero by the equivalence $\phi_i^{SADP,V}(x_i, s_i) \equiv \mathbb{E}[V_i(x_i, s_i, F_i')|s_i, F_0]$, parts (a) and (b) of Lemma 5, and Lemma 6; the last inequality follows from the Lipschitz continuity of $V_j(x_i, \cdot)$ with Lipschitz constant $C' := \alpha^I \cdot \max\{|C^I|, C^W\}$, which is a straightforward modification of Proposition 5 in Secomandi et al. (2012).

We now proceed to show that RHS is zero. In the ensuing analysis, we use the superscript $\rho_{j,k} = 1$ over an expectation to indicate that it is being evaluated at this specific correlation value.

$$
\begin{aligned}
\text{RHS} = C' \sum_{k=j+1}^{N-2} \lim_{\rho_{j-1}\to 1} \mathbb{E}\left[\left|\frac{F_{j-1,k}}{F_{j-1,j}^{\sigma_k/\sigma_j}} \exp\left(\frac{\sigma_k \Delta t}{2}(\sigma_j - \sigma_k)\right)\right.\right. \\
\left.\left. - \frac{F_{0,k}}{F_{0,j}^{\sigma_k/\sigma_j}} \exp\left(\frac{\sigma_k j \Delta t}{2}(\sigma_j - \sigma_k)\right)\right| \mathbb{E}\left[s_j^{\sigma_k/\sigma_j}|F_{j-1,j}\right]|s_j, F_{0,j}, F_{0,k}\right]
\end{aligned}
$$

$$
\begin{aligned}
&= C' \sum_{k=j+1}^{N-2} \lim_{\rho_{j-1}\to 1} \mathbb{E}\left[\left|F_{j-1,k}\right.\right. \\
&\qquad\qquad \left.\left. - \frac{F_{0,k}}{F_{0,j}^{\sigma_k/\sigma_j}} F_{j-1,j}^{\sigma_k/\sigma_j} \exp\left(\frac{\sigma_k(j-1)\Delta t}{2}(\sigma_j - \sigma_k)\right)\right| \left| s_j, F_{0,j}, F_{0,k}\right.\right] \\
&= C' \sum_{k=j+1}^{N-2} \left(\lim_{\rho_{j-1}\to 1} \mathbb{E}\left[\left(\left(F_{j-1,k} - \frac{F_{0,k}}{F_{0,j}^{\sigma_k/\sigma_j}} F_{j-1,j}^{\sigma_k/\sigma_j} \exp\left(\frac{\sigma_k(j-1)\Delta t}{2}(\sigma_j - \sigma_k)\right)\right)^+ \right.\right.\right. \\
&\qquad\qquad \left.\left.\left. \left| s_j, F_{0,j}, F_{0,k}\right.\right]\right. \\
&\qquad + \lim_{\rho_{j-1}\to 1} \mathbb{E}\left[\left(\frac{F_{0,k}}{F_{0,j}^{\sigma_k/\sigma_j}} F_{j-1,j}^{\sigma_k/\sigma_j} \exp\left(\frac{\sigma_k(j-1)\Delta t}{2}(\sigma_j - \sigma_k)\right) - F_{j-1,k}\right)^+ \right. \\
&\qquad\qquad \left.\left. \left| s_j, F_{0,j}, F_{0,k}\right.\right]\right) \\
&= C' \sum_{k=j+1}^{N-2} \left(\left(\bar{F}_{j-1,k}^{(\rho_{j-1,k}=1)}(s_{j-1}, F_0) \right.\right. \\
&\qquad\qquad \left. - \frac{F_{0,k}}{F_{0,j}^{\sigma_k/\sigma_j}} \exp\left(\frac{\sigma_k(j-1)\Delta t}{2}(\sigma_j - \sigma_k)\right) \mathbb{E}^{(\rho_{j-1,j}=1)}\left[F_{j-1,j}^{\sigma_k/\sigma_j} | s_{j-1}, F_0\right]\right)^+ \\
&\qquad + \left(\frac{F_{0,k}}{F_{0,j}^{\sigma_k/\sigma_j}} \exp\left(\frac{\sigma_k(j-1)\Delta t}{2}(\sigma_j - \sigma_k)\right) \mathbb{E}^{(\rho_{j-1,j}=1)}\left[F_{j-1,j}^{\sigma_k/\sigma_j} | s_{j-1}, F_0\right] \right. \\
&\qquad\qquad \left.\left. - \bar{F}_{j-1,k}^{(\rho_{j-1,k}=1)}(s_{j-1}, F_0)\right)^+ \right) \\
&= C' \sum_{k=j+1}^{N-2} \left| \bar{F}_{j-1,k}^{(\rho_{j-1,k}=1)}(s_{j-1}, F_0) \right. \\
&\qquad\qquad \left. - \frac{F_{0,k}}{F_{0,j}^{\sigma_k/\sigma_j}} \exp\left(\frac{\sigma_k(j-1)\Delta t}{2}(\sigma_j - \sigma_k)\right) \mathbb{E}^{(\rho_{j-1,j}=1)}\left[F_{j-1,j}^{\sigma_k/\sigma_j} | s_{j-1}, F_0\right] \right| \\
&= 0,
\end{aligned}
$$

where the first equality follows from using (A.8) and factoring the term within the absolute value out of the inner expectation because this term is deterministic given F_{j-1}; the second follows from using (A.9) and simplifying; the third by splitting the absolute value into a sum of two positive parts; the fourth by evaluating the expectations using the exchange option formula of Margrabe (1978) and evaluating the limit $\rho_{j-1} \to 1$; the fifth by expressing the sum of the two positive parts as an absolute value; and the sixth by using (A.10) and simplifying. Hence, (A.14) tends to zero when $\rho \to 1$.

(ii) Since the limiting matrix is rank 2 and $|\overline{\rho}_{i,i+1}| < 1$, we have convergence of $F_i''|s_i, F_{i,i+1}$ in distribution to $\bar{F}_i''(s_i, F_{i,i+1})$ as $\rho \to \overline{\rho}$ (recall that $F_i'' \equiv \{F_{i,i+1}, \ldots, F_{N-1}\}$). The rest of the proof of this part is analogous to the proof of part (a), and is thus omitted for brevity. □

A.3 SADP Greedy Lower Bounds

The SADP lower bounds from LMS reported in §2.8 are not obtained from the greedy optimization described in §2.2.2. Instead, these authors estimate a lower bound using a policy computed by solving the SADP recursion (2.41) over a discretized price grid (see §2.7 for details of price discretization). Feasible actions are obtained from this policy grid in simulation by employing interpolation when the sampled price is not a price in the discretization. Contrary to this approach and consistent with §2.2.2, below we report the greedy lower bounds associated with the SADP value function $\phi_i^{SADP}(x_i, s_i)$ defined by the recursion (2.41) and the greedy lower bound associated with another value function embedded in this recursion. As in (28)-(30) of LMS, this implicit value function can be defined as

$$\phi'^{SADP}_i(x_i, s_i, F_{i,i+1}) := \max_{a_i} r(a_i, s_i) + \delta \mathbb{E}\left[\phi_{i+1}^{SADP}(x_i - a_i, s_{i+1}) | F_{i,i+1}\right].$$

The SADP recursion (2.41) can be written in terms of this value function as

$$\phi_i^{SADP}(x_i, s_i) = \mathbb{E}\left[\phi'^{SADP}_i(x_i, s_i, F_{i,i+1}) \mid s_i, F_{0,i+1}\right].$$

Table A.1 reports the LMS lower bound estimate from the policy grid and the greedy lower bounds estimated using the value functions $\phi_i^{SADP}(x_i, s_i)$ and $\phi'^{SADP}_i(x_i, s_i, F_{i,i+1})$ as percentages of the ADP2 upper bound estimate, UB2. The instance labels in the first column of this table have the season and the capacity label separated by a hyphen. These results indicate that the lower bound based on greedy optimization of the value function $\phi_i^{SADP}(x_i, s_i)$ dominates the lower bound from the policy grid, which is also based on this value function but uses the SADP recursion (2.41). The greedy lower bound based on $\phi'^{SADP}_i(x_i, s_i, F_{i,i+1})$ is the best in Table A.1, and is in fact comparable with the ADP2 greedy lower bound estimate, LB2. Finally, we report the value of the UB2 estimates in Table A.2 to allow the computation of values corresponding the percentage values reported in this section and in §2.8.

Table A.1: Lower bounds estimated using the LMS policy grid and greedy optimization of the value functions $\phi_i^{SADP}(x_i, s_i)$ and ${\phi'}_i^{SADP}(x_i, s_i, F_{i,i+1})$ reported as percentages of UB2

Instances	LMS	$\phi_i^{SADP}(x_i, s_i)$	${\phi'}_i^{SADP}(x_i, s_i, F_{i,i+1})$
Spring-1	94.23	94.53	97.11
Spring-2	95.35	95.99	98.82
Spring-3	96.26	97.02	99.32
Summer-1	94.99	95.32	97.56
Summer-2	97.11	97.64	99.71
Summer-3	97.72	98.46	99.92
Fall-1	95.81	96.08	98.41
Fall-2	97.39	97.85	100.07
Fall-3	98.12	98.63	100.21
Winter-1	82.54	83.21	91.23
Winter-2	88.56	89.90	96.92
Winter-3	91.50	92.88	98.33

Table A.2: UB2 values

Instances	UB2
Spring-1	4.20
Spring-2	5.26
Spring-3	5.72
Summer-1	4.70
Summer-2	6.26
Summer-3	6.78
Fall-1	4.14
Fall-2	6.38
Fall-3	7.50
Winter-1	1.80
Winter-2	2.48
Winter-3	2.85

Appendix B

Additional Material for Chapter 3

B.1 Proofs for §3.6

This section contains the proofs of Lemma 1 and Propositions 8 and 9. Lemma 8 and Lemma 9 are required in these proofs. Let Π_2 define a generic projection operator corresponding to 2-norm regression (see the operator Π_2^Φ at the end of §3.6.1 for an example).

Lemma 8. *Let the functions f and g be defined on a closed and bounded set $\mathcal{Z}$. It holds that*

(a) $|\max_{z\in\mathcal{Z}} f(z) - \max_{z\in\mathcal{Z}} g(z)| \leq \max_{z\in\mathcal{Z}} |f(z) - g(z)|$,

(b) $\|\Pi_2 f(\cdot) - \Pi_2 g(\cdot)\|_\infty \leq \|f(\cdot) - g(\cdot)\|_\infty$.

Proof. (a) Let $z_1 \in \operatorname{argmax}_{z\in\mathcal{Z}} f(z)$ and $z_2 \in \operatorname{argmax}_{z\in\mathcal{Z}} g(z)$. It holds that $f(z_1) - g(z_2) \leq f(z_1) - g(z_1) \leq \max_{z\in\mathcal{Z}}\{f(z) - g(z)\} \leq \max_{z\in\mathcal{Z}} |f(z) - g(z)|$. Following the same steps starting from $g(z_2) - f(z_1)$ yields $g(z_2) - f(z_1) \leq \max_{z\in\mathcal{Z}} |f(z) - g(z)|$.

(b) Since the optimal solution to a 2-norm regression problem is linear in its argument we have $\|\Pi_2 f - \Pi_2 g\|_\infty = \|\Pi_2(f - g)\|_\infty$. Now we use the fact that Π_2 is a projection operator to conclude that $\|\Pi_2(f - g)\|_\infty \leq \|f(\cdot) - g(\cdot)\|_\infty$.

□

Let $\hat{V}_i(x_i, F_i^p)$ and $\hat{C}_i(x_{i+1}, F_i^p)$ denote a generic VFA and a generic CFA, respectively. Further, define the continuation function induced by $\hat{V}_{i+1}(x_{i+1}, F_{i+1}^p)$ and the value function induced by $\hat{C}_i(x_{i+1}, F_i^p)$, respectively, as

$$C_i^I(x_{i+1}, F_i^p) := \delta \mathbb{E}^s \left[\hat{V}_{i+1}(x_{i+1}, F_{i+1}) | F_i^p \right],$$
$$V_i^I(x_i, F_i^p) := \mathcal{H}_{(i,x_i,F_i^p)} \hat{C}_i.$$

Lemma 9 bounds the ∞-norm error of these functions.

Lemma 9. *For $q^V, q^C \in \mathbb{R}_+$, suppose $\|\hat{V}_{i+1} - V^s_{i+1}\|_\infty \le q^V$ and $\|\hat{C}_i - C^s_i\|_\infty \le q^C$, then*

$$\|C^I_i - C^s_i\|_\infty \le \delta q^V,$$
$$\|V^I_i - V^s_i\|_\infty \le q^C.$$

Proof.

$$\begin{aligned}\|C^I_i(\cdot) - C^s_i(\cdot)\|_\infty &= \delta \left\| \mathbb{E}^s \left[\hat{V}_{i+1}(\cdot, F_{i+1}) - V^s_{i+1}(\cdot, F_{i+1}) | \cdot \right] \right\|_\infty \\ &\le \delta \left\| \hat{V}_{i+1}(\cdot) - V^s_{i+1}(\cdot) \right\|_\infty \\ &= \delta q^V.\end{aligned}$$

We bound the error term $\left\| \hat{V}_i(\cdot) - V^s_i(\cdot) \right\|_\infty$ as follows:

$$\begin{aligned}\left\| V^I_i(\cdot) - V^s_i(\cdot) \right\|_\infty &= \|\mathcal{H}_{(i,\cdot,\cdot)}\hat{C}_i - \mathcal{H}_{(i,\cdot,\cdot)}C^s_i\|_\infty \\ &= \max_{(x_i,p)} \left| \max_a r_i(a, s_i) + \hat{C}_i(x_i - a, F^p_i) - \max_a r_i(a, s_i) + C^s_i(x_i - a, F^p_i) \right| \\ &\le \max_{(x_i,p)} \left| \max_a \left| \hat{C}_i(x_i - a, F^p_i) - C^s_i(x_i - a, F^p_i) \right| \right| \\ &= \max_{(x_i,p,a)} \left| \hat{C}_i(x_i - a, F^p_i) - C^s_i(x_i - a, F^p_i) \right| \\ &\le \max_{(x_{i+1},p)} \left| \hat{C}_i(x_{i+1}, F^p_i) - C^s_i(x_{i+1}, F^p_i) \right| \\ &\le q^C,\end{aligned}$$

where the first inequality holds by Part (a) of Lemma 8 and the second by noticing that $x_i - a \in \mathcal{X}_{i+1}$. □

Proof of Lemma 1. (a) Proof by induction. At stage $N-1$ we have

$$\tilde{e}^V_{N-1} = \left\| (\Phi_{N-1}\bar{\beta}_{N-1})(\cdot) - V^s_{N-1}(\cdot) \right\|_\infty = \left\| \Pi^\Phi_2 V^s_{N-1}(\cdot) - V^s_{N-1}(\cdot) \right\|_\infty = e^*_{N-1},$$

where the second equality follows from noticing that LSMV performs a regression against V^s_{N-1} to compute $\bar{\beta}_{N-1}$ at stage $N-1$ (see Algorithm 3). Make the induction hypothesis that $\tilde{e}^V_j \le \sum_{j=i}^{N-1} \delta^{j-i} e^*_j$, $\forall j = i+1, \ldots, N-1$. For stage i, we have

$$\begin{aligned}\tilde{e}^V_i &= \left\| (\Phi_i\bar{\beta}_i)(\cdot) - V^s_i(\cdot) \right\|_\infty \\ &\le \left\| (\Phi_i\bar{\beta}_i)(\cdot) - \Pi^\Phi_2 V^s_i(\cdot) \right\|_\infty + \left\| \Pi^\Phi_2 V^s_i(\cdot) - V^s_i(\cdot) \right\|_\infty \\ &= \left\| (\Phi_i\bar{\beta}_i)(\cdot) - \Pi^\Phi_2 V^s_i(\cdot) \right\|_\infty + e^*_i, \qquad \text{(B.1)}\end{aligned}$$

where the inequality is true by the triangle inequality property of norms. Now we bound $\hat{e}_i := \left\| (\Phi_i\bar{\beta}_i)(\cdot) - \Pi^\Phi_2 V^s_i(\cdot) \right\|_\infty$. For ease of notation we use the placeholder $\eta = (x_i -$

a, F_{i+1}). We have

$$\begin{aligned}
\hat{e}_i &= \|\Pi_2^{\Phi}\mathcal{L}^s_{(i,\cdot,\cdot)}(\Phi_{i+1}\bar{\beta}_{i+1}) - \Pi_2^{\Phi}V_i^s(\cdot)\|_\infty \\
&\leq \max_{(x_i,p)}\left|\left\{\max_a r_i(a,F_i^p) + \delta\mathbb{E}^s\left[(\Phi_{i+1}\bar{\beta}_{i+1})(\eta)|F_i^p\right]\right\} - \left\{\max_a r_i(a,F_i^p) + \delta\mathbb{E}^s\left[V_{i+1}^s(\eta)|F_i^p\right]\right\}\right| \\
&\leq \delta\max_{(x_i,p)}\left|\max_a \mathbb{E}^s\left[(\Phi_{i+1}\bar{\beta}_{i+1})(\eta) - V_{i+1}^s(\eta)|F_i^p\right]\right| \\
&\leq \delta\max_{(x_i,p)}\left|\max_{(a,p)}\left\{(\Phi_{i+1}\bar{\beta}_{i+1})(x_{i+1},F_{i+1}^p) - V_{i+1}^s(x_{i+1},F_{i+1}^p)\right\}\right| \\
&\leq \delta\max_{(x_i,a,p)}\left|(\Phi_{i+1}\bar{\beta}_{i+1})(x_{i+1},F_{i+1}^p) - V_{i+1}^s(x_{i+1},F_{i+1}^p)\right| \\
&\leq \delta\max_{(x_{i+1},p)}\left|(\Phi_{i+1}\bar{\beta}_{i+1})(x_{i+1},F_{i+1}^p) - V_{i+1}^s(x_{i+1},F_{i+1}^p)\right| \\
&= \delta\tilde{e}_{i+1}^V, \qquad\qquad (B.2)
\end{aligned}$$

where the first two inequalities follow from parts (b) and (a) of Lemma 8, respectively; the third inequality from replacing an expectation by a maximum; the fourth inequality from the relation $|\max_{s\in S} g(s)| \leq \max_{s\in S}|g(s)|$; and the last inequality by noticing that $x_i - a \in \mathcal{X}_{i+1}$. Combining (B.1) and (B.2) gives $\tilde{e}_i^V \leq \delta\tilde{e}_{i+1}^V + e_i^*$, from which we obtain the desired bound on $\tilde{e}_i^V$ by applying the induction hypothesis to bound $\tilde{e}_{i+1}^V$. The result holds for all stages by the principle of mathematical induction.

(b) Proof by induction. At stage $N-2$, we have

$$\begin{aligned}
\tilde{e}_{N-2}^C &= \left\|(\Psi_{N-2}\bar{\theta}_{N-2})(\cdot) - C_{N-2}^s(\cdot)\right\|_\infty \\
&\leq \left\|(\Psi_{N-2}\bar{\theta}_{N-2})(\cdot) - \Pi_2^{\Psi}\delta\mathbb{E}^s[\mathcal{H}_{(N-1,\cdot,F_{N-1})}(\Psi_{N-1}\bar{\theta}_{N-1})|\cdot]\right\|_\infty \\
&\quad + \left\|\Pi_2^{\Psi}\delta\mathbb{E}^s[\mathcal{H}_{(N-1,\cdot,F_{N-1})}(\Psi_{N-1}\bar{\theta}_{N-1})|\cdot] - C_{N-2}^s(\cdot)\right\|_\infty \\
&= \bar{e}_{N-2}^C + e_{N-2}^{**}.
\end{aligned}$$

Make the induction hypothesis that $\tilde{e}_j^C \leq \sum_{j=i}^{N-2}\delta^{j-i}(e_j^{**} + \bar{e}_j^C)$, $\forall j = i+1,\ldots,N-2$. At stage i, it holds that

$$\begin{aligned}
\tilde{e}_i^C &= \left\|(\Psi_i\bar{\theta}_i)(\cdot) - C_i^s(\cdot)\right\|_\infty \\
&\leq \left\|(\Psi_i\bar{\theta}_i)(\cdot) - \Pi_2^{\Psi}C_i^s(\cdot)\right\|_\infty + \left\|\Pi_2^{\Psi}C_i^s(\cdot) - C_i^s(\cdot)\right\|_\infty \\
&\leq \left\|(\Psi_i\bar{\theta}_i)(\cdot) - \Pi_2^{\Psi}C_i^s(\cdot)\right\|_\infty + e_i^{**} \\
&\leq \left\|(\Psi_i\bar{\theta}_i)(\cdot) - \Pi_2^{\Psi}\mathbb{E}^s[\mathcal{H}_{(i+1,\cdot,F_{i+1})}(\Psi_{i+1}\bar{\theta}_{i+1})|\cdot]\right\|_\infty \\
&\quad + \left\|\Pi_2^{\Psi}\mathbb{E}^s[\mathcal{H}_{(i+1,\cdot,F_{i+1})}(\Psi_{i+1}\bar{\theta}_{i+1})|\cdot] - \Pi_2^{\Psi}C_i^s(\cdot)\right\|_\infty + e_i^{**} \\
&= \bar{e}_i^C + \left\|\Pi_2^{\Psi}\mathbb{E}^s[\mathcal{H}_{(i+1,\cdot,F_{i+1})}(\Psi_{i+1}\bar{\theta}_{i+1})|\cdot] - \Pi_2^{\Psi}C_i^s(\cdot)\right\|_\infty + e_i^{**}.
\end{aligned}$$

Further, we can show that $\left\|\Pi_2^{\Psi}\mathbb{E}^s[\mathcal{H}_{(i+1,\cdot,F_{i+1})}(\Psi_{i+1}\bar{\theta}_{i+1})(\cdot)|\cdot] - \Pi_2^{\Psi}C_i^s(\cdot)\right\|_\infty \leq \delta\tilde{e}_{i+1}^C$ following steps similar to those required to bound the term $\hat{e}_i \equiv \left\|(\Phi_i\bar{\beta}_i)(\cdot) - \Pi_2^{\Phi}V_i^s(\cdot)\right\|_\infty$ in Part (a) (see (B.2)). The rest of the proof follows in a straightforward manner from our induc-

tion hypothesis and the principle of mathematical induction.

(c) Recall that the term $v_i(\cdot)$ is defined in Algorithm 3. At stage i, we have

$$\begin{aligned}\|(\Phi_i\bar{\gamma}_i)(\cdot) - V_i^s(\cdot)\|_\infty &\leq \|(\Phi_i\bar{\gamma}_i)(\cdot) - v_i(\cdot)\|_\infty + \|v_i(\cdot) - V_i^s(\cdot)\|_\infty \\ &\leq \bar{e}_i^H + \|v_i(\cdot) - V_i^s(\cdot)\|_\infty \\ &\leq \bar{e}_i^H + \tilde{e}_i^C \\ &\leq \bar{e}_i^H + \sum_{j=i}^{N-2} \delta^{j-i}(e_j^{**} + \bar{e}_j^C),\end{aligned}$$

where the third inequality follows from Lemma 9 and the fourth inequality from Part (b). □

Proof of Proposition 8. Recall that the terms $u_i^{\bar{\beta}}(\cdot)$, $u_i^{\bar{\theta}}(\cdot)$, and u_i^s are defined in §3.6.3. We begin by bounding the error $\tilde{e}_i^{V,DP}$:

$$\begin{aligned}\tilde{e}_i^{V,DP} &= \|u_i^{\bar{\beta}}(\cdot) - u_i^s(\cdot)\|_\infty \\ &= \|(\Phi_{i+1}\bar{\beta}_{i+1})(\cdot) - V_{i+1}^s(\cdot) - \mathbb{E}^s\left[(\Phi_{i+1}\bar{\beta}_{i+1})(\cdot, F_{i+1}) - V_{i+1}^s(\cdot, F_{i+1})|\cdot\right]\|_\infty \\ &\leq \|(\Phi_{i+1}\bar{\beta}_{i+1})(\cdot) - V_{i+1}^s(\cdot)\|_\infty + \|\mathbb{E}^s\left[(\Phi_{i+1}\bar{\beta}_{i+1})(\cdot, F_{i+1}) - V_{i+1}^s(\cdot, F_{i+1})|\cdot\right]\|_\infty \\ &\leq 2\|(\Phi_{i+1}\bar{\beta}_{i+1})(\cdot) - V_{i+1}^s(\cdot)\|_\infty \\ &= 2\tilde{e}_{i+1}^V,\end{aligned}$$

where the first inequality follows from the triangle inequality property of norms. The claimed bound on $\tilde{e}_{i+1}^{V,DP}$ follows from bounding $\tilde{e}_{i+1}^V$ by using Part (a) of Lemma 1. The proof of the claimed bound on $\tilde{e}_i^{H,DP}$ is similar to the proof of the bound on $\tilde{e}_{i+1}^{V,DP}$. We bound $\tilde{e}_i^{C,DP}$ as follows:

$$\begin{aligned}\tilde{e}_i^{C,DP} &= \|u_i^{\bar{\theta}}(\cdot) - u_i^s(\cdot)\|_\infty \\ &= \|\mathcal{H}_{(i+1,\cdot,\cdot)}(\Psi_{i+1}\bar{\theta}_{i+1}) - \mathcal{H}_{(i+1,\cdot,\cdot)}C_{i+1}^s(\cdot) \\ &\quad - \mathbb{E}^s\left[\mathcal{H}_{(i+1,\cdot,F_{i+1})}(\Psi_{i+1}\bar{\theta}_{i+1}) - \mathcal{H}_{(i+1,\cdot,F_{i+1})}C_{i+1}^s|\cdot\right]\|_\infty \\ &\leq \|\mathcal{H}_{(i+1,\cdot,\cdot)}(\Psi_{i+1}\bar{\theta}_{i+1}) - \mathcal{H}_{(i+1,\cdot,\cdot)}C_{i+1}^s\|_\infty \\ &\quad + \|\mathbb{E}^s\left[\mathcal{H}_{(i+1,\cdot,F_{i+1})}(\Psi_{i+1}\bar{\theta}_{i+1}) - \mathcal{H}_{(i+1,\cdot,F_{i+1})}C_{i+1}^s|\cdot\right]\|_\infty \\ &\leq \max_{(x_{i+1},p,a)} \left|(\Psi_{i+1}\bar{\theta}_{i+1})(x_{i+1} - a, F_{i+1}^p) - C_{i+1}^s(x_{i+1} - a, F_{i+1}^p)\right| \\ &\quad + \max_{(x_{i+1},p)} \left|\mathbb{E}^s\left[\max_a \left|(\Psi_{i+1}\bar{\theta}_{i+1})(x_{i+1} - a, F_{i+1}) - C_{i+1}^s(x_{i+1} - a, F_{i+1})\right| |\cdot\right]\right| \\ &\leq 2\max_{(x_{i+2},p)} \left|(\Psi_{i+1}\bar{\theta}_{i+1})(x_{i+2}, p) - C_{i+1}^s(x_{i+2}, p)\right| \\ &= 2\tilde{e}_{i+1}^C,\end{aligned}$$

where we obtain the second inequality by applying Part (a) of Lemma 8. The claimed bound on $\tilde{e}_i^{C,DP}$ follows from bounding $\tilde{e}_{i+1}^C$ using Part (b) of Lemma 1. □

Proof of Proposition 9. The claimed bounds follow from a direct application of Lemma 9. □

B.2 Technical Conditions Used in §§3.6.3-3.6.4 and Their Numerical Validation

In §B.2.1 and §B.2.2 we discuss the technical conditions underlying our error bound analysis in §3.6.3 and §3.6.4, respectively. In §B.2.3 we numerically validate these conditions.

B.2.1 Upper Bound Estimation

Comparing the bounds on $\tilde{e}_i^{C,DP}$ and $\tilde{e}_i^{H,DP}$ in parts (b) and (c) of Proposition 8, respectively, with the bound on $\tilde{e}_i^{V,DP}$ in Part (c) of Proposition 8 is challenging because the regression errors e_{i+1}^* and e_i^{**} originate from approximating the functions V_{i+1}^s and C_i^s. These latter functions are related as $C_i^s(x_{i+1}, F_i^p) \equiv \delta \mathbb{E}^s \left[V_{i+1}^s(x_{i+1}, F_{i+1}) | F_i^p \right]$. We expect C_i^s to be smoother than V_{i+1}^s. It is plausible to expect that the inequality $e_i^{**} \leq \delta e_{i+1}^*$ should hold for similar basis functions and the same number of regression samples. In addition, consider the following two sets of inequalities:

$$e_i^* \leq \bar{e}_i^H, \forall(i), \tag{B.3}$$

$$\delta e_{i+1}^* \leq e_i^{**} + \bar{e}_i^C, \forall(i)_{-(N-1)}. \tag{B.4}$$

If inequalities (B.3)-(B.4) hold, then the bound on $\tilde{e}_i^{V,DP}$ is no larger than the bound on $\tilde{e}_i^{H,DP}$. In other words, the LSMV-based dual upper bound estimates should be no worse than the LSMH-based dual upper bound estimates. Because the LSMC error $\bar{e}_i^C$ results from regressing over single sample path estimates of expectations (see the discussion in §3.6.2), it is reasonable to expect this error to be larger than both the regression errors e_{i+1}^* and e_{i+1}^{**}, especially for low values of P. Moreover, because the regression estimates of $\bar{e}_i^H$ are based on the LSMC CFA, they are potentially noisy, especially for low values of P. Thus, we expect $\bar{e}_i^H$ to be larger than e_{i+1}^* (see Step (iii) of LSMH). These anticipations are validated numerically in §B.2.3. The above reasoning provides some support for conditions (B.3)-(B.4) to be satisfied.

Now we turn to comparing the bounds on $\tilde{e}_i^{V,DP}$ and $\tilde{e}_i^{C,DP}$. At stage $N-2$, the bound on $\tilde{e}_i^{C,DP}$ is no worse than the bound on $\tilde{e}_i^{V,DP}$ because the former bound is based on sums of $N-i-1$ terms while the latter bound is based on sums of $N-i$ terms. However, the bound on $\tilde{e}^{V,DP}$ is no worse than the bound on $\tilde{e}_i^{C,DP}$ for stages $0, \ldots, N-3$ if inequalities (B.4) hold and

$$e_{i+1}^* \leq \sum_{j=i+1}^{N-2} \delta^{j-i-1}(e_i^{**} + \bar{e}_i^C - \delta e_{j+1}^*), \forall i \in \mathcal{I} \setminus \{N-1, N-2\}. \tag{B.5}$$

Based on analogous arguments regarding the validity of (B.4), we expect inequality (B.5) to hold. Thus, the LSMV-based dual upper bound estimates should be no worse than the LSMC-based dual upper bound estimates.

B.2.2 Lower Bound Estimation

The error bound on the CFA induced by the LSMV VFA given in Part (a) of Proposition 9 is no worse than both the error bound on the LSMC CFA established in Part (b) of Lemma 1 and the error bound on the CFA induced by the LSMH VFA shown in Part (b) of Proposition 9 if inequalities (B.4) and (B.3)-(B.4) hold, respectively. Based on our discussion in §B.2.1 regarding the validity of these inequalities, the LSMV-based greedy lower bound estimates should be no worse than the ones based on LSMC and LSMH.

We now compare the error bound on the CFA induced by the LSMH VFA given in Part (b) of Proposition 9 with the error bound on the LSMC CFA established in Part (b) of Lemma 1. The error bound on $\tilde{e}_i^{H,IC}$ is no worse than the error bound on $\tilde{e}_i^C$ at each stage i if the following inequalities hold:

$$\delta \bar{e}_{i+1}^H \leq e_i^{**} + \bar{e}_i^C, \forall (i)_{-(N-1)}, \tag{B.6}$$

We expect these inequalities to be satisfied based on arguments analogous to the ones discussed in §3.6.3. The greedy lower bound estimates obtained by LSMH should thus outperform the ones determined by LSMC.

B.2.3 Numerical Validation

We provide support for the validity of inequalities (B.3)-(B.6) evaluated on the instances considered in §3.7.2 when $P = 1{,}000$. For brevity, we focus on evaluating these inequalities when LSMV3, LSMC3, and LSMH3 are applied to the swing option instances, but we have verified that the resulting qualitative insights remain essentially unchanged when LSMV1, LSMC1, and LSMH1 are applied to the natural gas storage instances.

Evaluating the left and right hand sides of inequalities (B.3)-(B.6) requires estimating the errors e_i^*, e_i^{**}, $\bar{e}_i^H$, and $\bar{e}_i^C$ (see (3.20)-(3.23)). Estimating e_i^* and e_i^{**} is challenging because these errors depend on the value function $V_i^s(x_i, F_i^p)$ of (3.17) and the continuation function $C_i^s(x_{i+1}, F_i^p)$ of (3.18), respectively, which are unknown. To overcome this issue, we *assume* that the stage $i+1$ LSMV3 VFA is equal to $V_i^s(x_i, F_i^p)$ and use the stage i VFA $\mathcal{L}_{(i,x_i,F_i^p)}(\Phi_{i+1}\bar{\beta}_{i+1})$ and the stage i CFA $\delta \mathbb{E}\left[(\Phi_{i+1}\bar{\beta}_{i+1})(x_{i+1}, F_{i+1})|F_i^p\right]$, respectively, as proxies for $V_i^s(x_i, F_i^p)$ and $C_i^s(x_{i+1}, F_i^p)$.

Table B.1 reports the estimated errors on the January, April, July, and October swing option instances for $n = 3$ and $P = 1{,}000$ when using LSMV3, LSMC3 and LSMH3. We find that the estimates of e_{i+1}^{**} are much smaller than the estimates of e_i^*, which is consistent with our expectation, discussed in §B.2.1, that C_i^s should be smoother than V_{i+1}^s. We verified that the estimates of $\bar{e}_i^H$ are almost always larger than the estimates of e_i^*, although their intervals overlap (see satisfaction of (B.3) in Figure B.1 below for

Table B.1: Errors e_i^*, e_i^{**}, $\bar{e}_i^H$, and $\bar{e}_i^C$ estimated on the swing option instances for $n = 3$, $P = 1{,}000$, and basis function set 3.

	e_i^*			e_i^{**}		
Month	Minimum	Median	Maximum	Minimum	Median	Maximum
January	0.095	0.568	1.293	0.000	0.084	0.343
April	0.280	0.548	1.160	0.000	0.071	0.226
July	0.000	0.682	1.444	0.000	0.118	0.342
October	0.130	0.573	1.385	0.000	0.073	0.279
	$\bar{e}_i^H$			$\bar{e}_i^C$		
Month	Minimum	Median	Maximum	Minimum	Median	Maximum
January	0.079	0.941	1.889	1.226	8.298	29.013
April	0.311	0.917	1.354	1.437	9.236	31.842
July	0.000	1.132	2.012	2.327	9.939	31.623
October	0.250	0.992	1.577	0.540	9.133	17.644

support). The estimates of $\bar{e}_i^C$ are substantially larger than the estimated errors e_i^*, e_i^{**}, and $\bar{e}_i^H$.

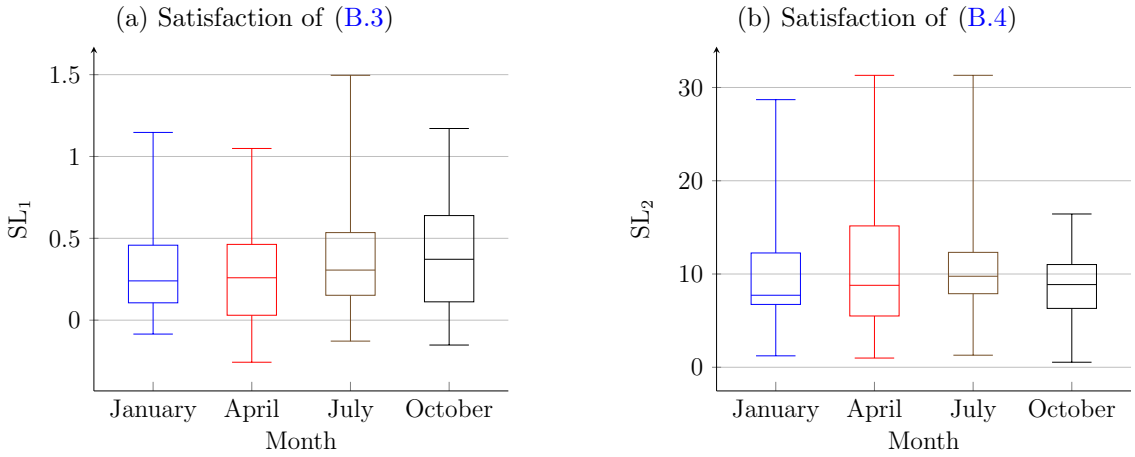


Figure B.1: Satisfaction of inequalities (B.3) and (B.4) on the swing option instances for $n = 3$, $P = 1{,}000$, and basis function set 3.

We now discuss the validity of inequalities (B.3)-(B.6). We define the slack associated with inequalities (B.3) and (B.4) by $\text{SL}_1 := \bar{e}_i^H - e_i^*$ and $\text{SL}_2 := e_i^{**} + \bar{e}_i^C - \delta e_{i+1}^*$, respectively. Thus positive values of SL_1 and SL_2 indicate satisfaction of the corresponding inequalities while negative values of these slacks represent violations. Figure B.1 is a box and whisker plot of the distributions of the estimates of SL_1 and SL_2 across stages on the swing option instances. Because the estimated SL_2 is positive across stages and instances, the inequality (B.4) always appears to be satisfied. In contrast to the estimated SL_2, the estimated SL_1 is negative in a few stages, which we verified to be at most 3 out of the 23 stages across instances.

Comparing the distribution of the estimates of SL_2 values in Figure B.1 with the distribution of estimates of e_i^* in Table B.1 suggests that the former values are substantially larger than the latter. We verified this to be true at each stage across instances. As a result, inequality (B.5) also holds for all stages $i \leq N-2$. Finally, we also verified that the inequality (B.6) is always satisfied because the estimates of $\bar{e}_i^C$ are larger than the estimated differences $\delta\bar{e}_i^H - e_i^{**}$, which is somewhat expected given the estimated values for the involved errors reported in Table B.1. Thus, our numerical error analysis for $P =$ 1,000 provides support for the validity of inequalities (B.3)-(B.6).

B.3 LSMV1 Bound Estimates on LMS Instances

Figure B.2 reports the LSMV1 dual upper bound estimate as a percentage of UB2 on the LMS instances used in Chapter 2. We find that LSMV1 improves on UB2 by 0.17% to 2.33%. Since LSMV1 is the best upper bound estimate, Figure B.3 reports as a percentage of this estimated upper bound, LB2, RLB2, and the lower bound estimate of LSMV1. We find that the lower bound estimates of LSMV1 and RLB2 are within standard error of each other and essentially optimal on all the instances. In particular, the optimality of the RLB2 lower bound estimate on all the winter instances shows that the suboptimalities of this lower bound estimate in Figure 2.6 of Chapter 2 are due to the weakness of UB2. Finally, Table B.2 reports the values of the LSMV1 lower bound and upper bound estimates.

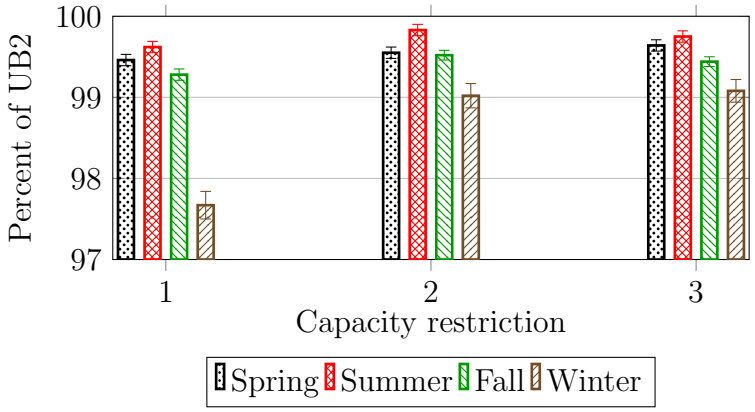


Figure B.2: LSMV1 estimated upper bounds and their standard errors (error bars).

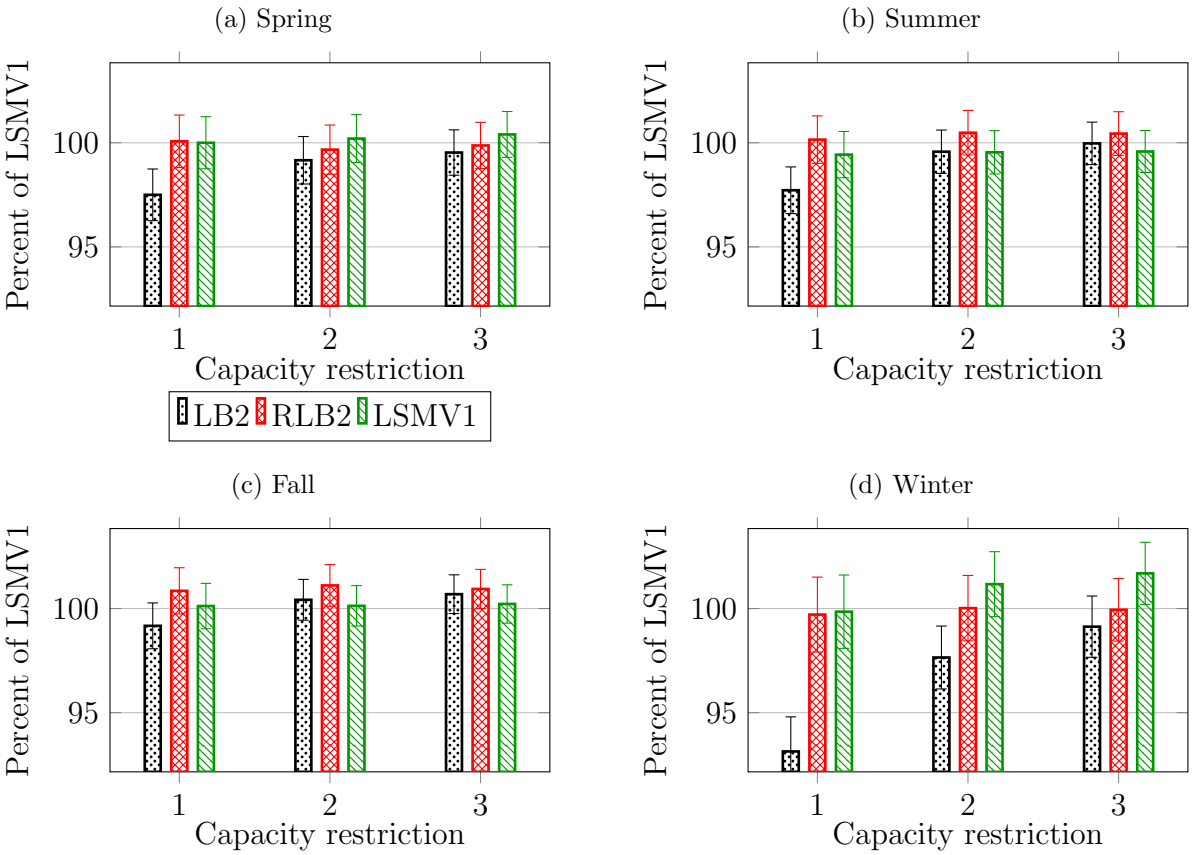


Figure B.3: Estimated lower bounds and their standard errors (error bars).

Table B.2: Values of LSMV1 lower bound and upper bound estimates.

Instance	Lower bound	Upper bound
Spring-1	4.15	4.17
Spring-2	5.21	5.23
Spring-3	5.68	5.69
Summer-1	4.64	4.68
Summer-2	6.20	6.24
Summer-3	6.71	6.76
Fall-1	4.80	4.11
Fall-2	6.32	6.35
Fall-3	7.44	7.46
Winter-1	1.72	1.75
Winter-2	2.42	2.44
Winter-3	2.79	2.81

Appendix C

Additional Material for Chapter 4

C.1 Proofs

Proof of Lemma 2. For proving the claimed characterization we require the finiteness of the value and continuation functions of SDP (4.15). It is obvious that $V_i(y_i, \boldsymbol{F}_i) \geq 0 > -\infty$, which implies that $W_i(y_{i+1}, \boldsymbol{F}_i) > -\infty$. Further, $V_i(y_i, \boldsymbol{F}_i) \leq \sum_{i' \in \mathcal{I}_i} \sum_{m \in \mathcal{M}} C^{D,m} s^m_{i'}$. Using this inequality we have

$$\begin{aligned} W_i(y_{i+1}, \boldsymbol{F}_i) &\equiv \delta \mathbb{E}\left[V_{i+1}\left(y_{i+1}, \boldsymbol{F}_{i+1}\right) | \boldsymbol{F}_i\right] \\ &\leq \delta \mathbb{E}\left[\sum_{i' \in \mathcal{I}_i} \sum_{m \in \mathcal{M}} C^{D,m} s^m_{i'} | \boldsymbol{F}_i\right] \\ &= \delta \sum_{i' \in \mathcal{I}_i} \sum_{m \in \mathcal{M}} C^{D,m} \mathbb{E}\left[s^m_{i'} | \boldsymbol{F}_i\right] \\ &= \delta \sum_{i' \in \mathcal{I}_i} \sum_{m \in \mathcal{M}} C^{D,m} F^m_{i,i'} \\ &< \infty, \end{aligned}$$

where the second equality follows from $s^m_{i'} \equiv F^m_{i',i'}$ and the martingale property of futures prices (Shreve, 2004, page 244). Thus, the value and continuation functions of SDP (4.15) are finite.

We now proceed by induction to prove the claimed result. At stage $N-1$, for a given $\boldsymbol{F}_i$, we have

$$V_{N-1}(y_{N-1}, \boldsymbol{F}_{N-1}) = \max_{y_N \in \mathcal{Z}(y_{N-1})} \bar{r}(y_{N-1} - y_N, \boldsymbol{s}_{N-1}).$$

Standard linear programming results (Bertsimas and Tsitsiklis 1997, ch. 5) imply that the function $\bar{r}(y_{N-1} - y_N, \boldsymbol{s}_{N-1})$ is concave in the pair (y_{N-1}, y_N), which belongs to a convex (polyhedral) feasible set. Further, the interval $\mathcal{Z}(y_{N-1})$ is nonempty for each given $y_{N-1} \in \mathcal{Y}$. The concavity of $V_{N-1}(\cdot, \boldsymbol{F}_{N-1})$ follows from Proposition B-4 in Heyman and Sobel (2003). The continuation function at stage $N-1$ is zero by definition and is therefore concave.

Make the induction hypothesis that the value and continuation functions are concave in their first arguments also for stages $i+1, i+2, \ldots, N-2$. We proceed to prove the claim at stage i. From the finiteness of the continuation function in every stage and the induction hypothesis, it is easy to verify that the continuation function is concave in its first argument at stage i. This property and Part (a) of this lemma imply the concavity of $\bar{r}(y_i - y_{i+1}, \boldsymbol{s}_i) + W_i(y_{i+1}, \boldsymbol{F}_i)$ in the pair (y_i, y_{i+1}), which belongs to a convex (polyhedral) feasible set. Since $\mathcal{Z}(y_i)$ is nonempty for each given $y_i \in \mathcal{Y}$, the concavity of $V_i(\cdot, \boldsymbol{F}_i)$ follows from Proposition B-4 in Heyman and Sobel (2003). The claimed concavity of the value and continuation functions at all stages for a given array of forward curves follows from the principle of mathematical induction. □

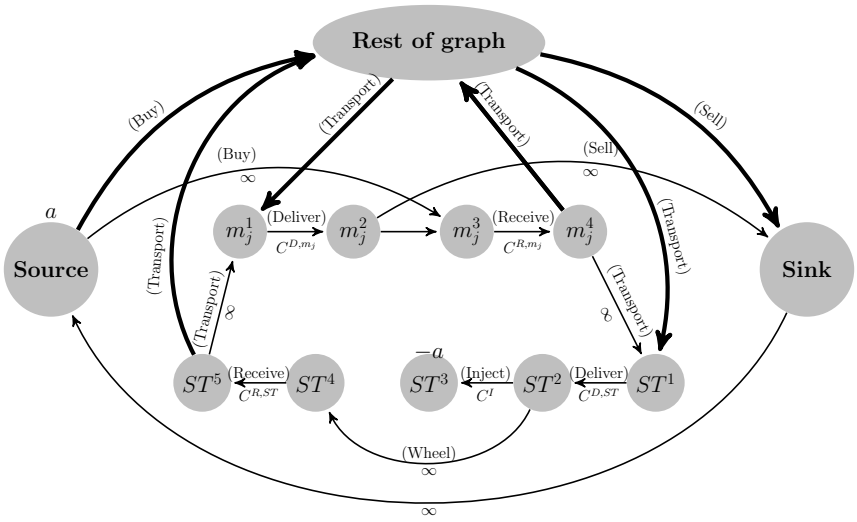


Figure C.1: Edge network formulation for the feasible set of (4.13) when $a \leq 0$.

Proof of Lemma 3. The piecewise linear concavity of $\bar{r}(\cdot, \boldsymbol{s})$ follows from standard linear programming results (Bertsimas and Tsitsiklis 1997, ch. 5). We proceed to show that the slope of $\bar{r}(\cdot, \boldsymbol{s})$ changes at integer multiples of $\bar{G}$. Our proof relies on reformulating the maximum profit flow problem (4.13), defined over the set of trade paths, as a maximum profit network flow problem on an *edge* network.

We begin by describing the edge network construction. Figure C.1 illustrates the edge network assuming a nonnegative storage action, $a \leq 0$. An analogous network exists for the case $a > 0$. Recall that $C^{R,m}$ and $C^{D,m}$ are the receipt and delivery capacities of node m, respectively. The nodes in this figure are (i) a dummy source node and a dummy sink node; (ii) nodes m_j^1-m_j^4 for modeling market m_j; (iii) nodes ST^1-ST^5 for modeling storage; and (iv) the node labeled "rest of graph" that is an aggregation of nodes and edges not represented explicitly in this figure. Thin edges denote actual edges in the network with their capacities as edge labels, while thick edges denote a collection of edges to or from the node labeled "rest of the graph". The supply of the source node is the storage action a. The demand of node ST^3 is $-a$. All other nodes have zero demand. It can be verified

that each path in this network corresponds to the path of a trade in set $\mathcal{J}$. To help verify this assertion, the labels within parentheses on the edges of this figure indicate the edge modeling purpose. For example, the label "(Buy)" on the edge (Source, m_j^3) indicates that this edge models the purchase of natural gas from market m_j. Thus, each term that makes up the cash flow of a trade can be represented as a profit on one of the edges in this network.

We introduce some generic notation to formulate the stated maximum profit flow problem on this edge network. Part of this notation uses with a different meaning notation used elsewhere in this paper, but this confined reuse of notation should not give rise to any confusion. The pair $(\mathcal{N}, \mathcal{E})$ includes the node and edge sets of this network. Let the capacity and profit on edge $e = (u, v)$ be C_e and c_e, respectively (c is the column vector of all edge profits). Denote the demand on node v by d_v. By construction, we have $d_{source} = a$, $d_{ST^3} = -a$, and all other node demands equal to zero. We use w_e to represent the flow variable associated with edge e (w is the column vector of all these flow variables). The maximum profit flow problem is

$$\max_{w} c^\top \cdot w \tag{C.1}$$

$$\sum_{e\in\mathcal{E}:e=(\cdot,v)} w_e - \sum_{e\in\mathcal{E}:e=(v,\cdot)} w_e = 0, \forall v \in \mathcal{N} \setminus \{\text{Source}, ST^3\}, \tag{C.2}$$

$$w_{(Sink,Source)} - \sum_{e\in\mathcal{E}:e=(Source,\cdot)} w_e = a, \tag{C.3}$$

$$w_{(ST_2,ST_3)} = -a, \tag{C.4}$$

$$0 \le w_e \le C_e, \forall e \in \mathcal{E}. \tag{C.5}$$

The claimed equivalence between (C.1)-(C.5) and (4.13) holds because an optimal solution of (C.1)-(C.5) can be decomposed into amounts corresponding to trades in set $\mathcal{J}$ by removing the edge from the sink to the source if $a \neq 0$ (all cycles that do not include the edge (Sink, Source) have a negative profit).

Make the change of variable $\hat{w}_e := w_e/\bar{G}$ in (C.1)-(C.5) ($\hat{w}$ is the column vector of scaled flow variables on all edges). This change yields the linear program

$$\max_{\hat{w}} \bar{G}(c^\top \cdot \hat{w}) \tag{C.6}$$

$$\sum_{e\in\mathcal{E}:e=(\cdot,v)} \hat{w}_e - \sum_{e\in\mathcal{E}:e=(v,\cdot)} \hat{w}_e = 0, \forall v \in \mathcal{N} \setminus \{\text{Source}, ST^3\}, \tag{C.7}$$

$$\hat{w}_{(Sink,Source)} - \sum_{e\in\mathcal{E}:e=(Source,\cdot)} \hat{w}_e = a/\bar{G}, \tag{C.8}$$

$$\hat{w}_{(ST_2,ST_3)} = -a/\bar{G}, \tag{C.9}$$

$$0 \le \hat{w}_e \le C_e/\bar{G}, \forall e \in \mathcal{E}. \tag{C.10}$$

From the definition of $\bar{G}$ it follows that the scaled capacity $C_e/\bar{G}$ is integer for all the edges that have finite capacity. The integrality of the scaled capacities and the unimodularity of the constraint matrices of network flow problems (see Theorems 11.11 and 11.12 in Ahuja

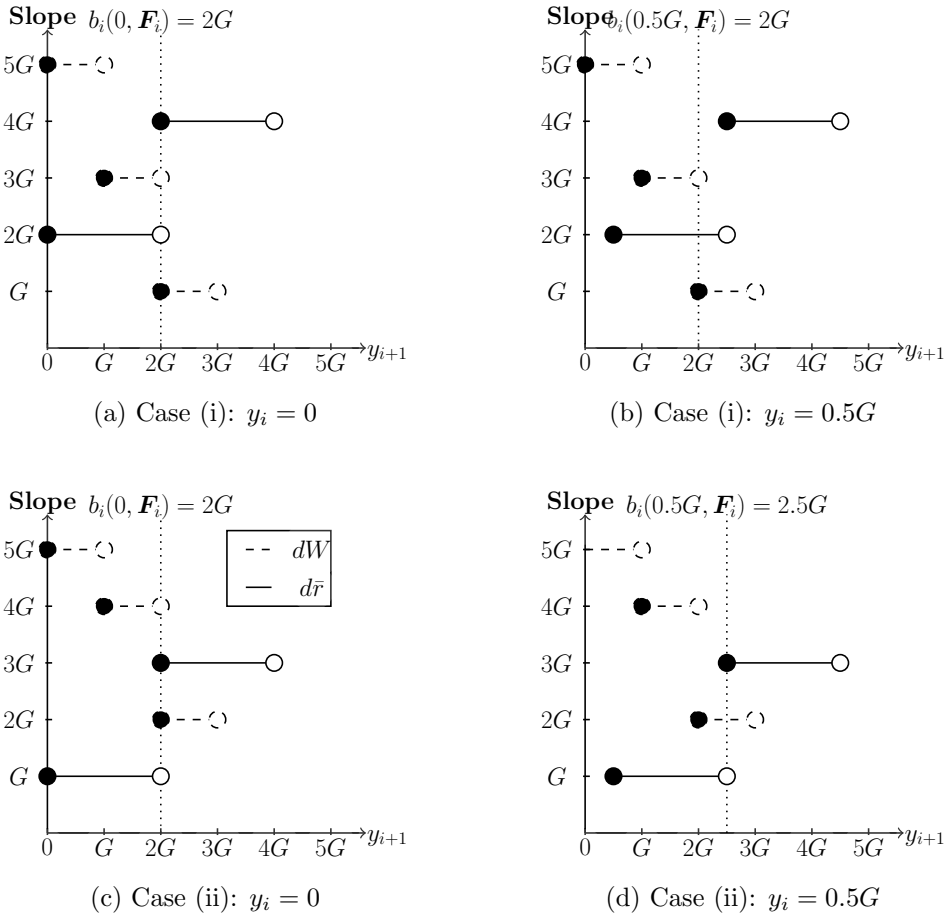
et al. 1993) imply that the optimal solution to the linear program (C.6)-(C.10) with $a = 0$ is integer. Let this optimal solution be $\hat{w}^*(0)$. Recall the definition of a^I on Page 84. Suppose that we increase the injection amount from 0 to ϵ such that $0 < \epsilon < \bar{G}$ and $\epsilon \leq a^I$, that is, $a = -\epsilon$. This corresponds to increasing the injection amount by less than 1 unit in (C.6)-(C.10). Note that $\hat{w}^*(0)$ defines a pseudo flow (see page 320 of Ahuja et al. 1993) for the problem with $a = -\epsilon$, that is, this solution violates mass balance by ϵ only at the nodes Source and ST^3. Since $\epsilon \leq a^I$ there exists a shortest path from the node Source to the node ST^3 in the residual network (see §9.1 in Ahuja et al. 1993). Augmenting by ϵ the flow along this path yields an optimal solution to (C.6)-(C.10) by Lemma 9.12 in Ahuja et al. (1993). This shortest path has capacity at least 1 since $\hat{w}^*(0)$ is an integral flow and all edge capacities in (C.6)-(C.10) are integral. Thus, when changing $a/\bar{G}$ between 0 and -1 an optimal solution to the resulting problem can be found by augmenting by the same amount the flow along the same shortest path. Consequently, the optimal solution value of the problem (C.6)-(C.10) is linear for values of $a/\bar{G}$ in between 0 and -1, which implies that the optimal solution to (C.1)-(C.5) is linear for $a \in [0, -\bar{G}]$. This argument can be repeated to prove an analogous result when the injection amount ϵ is between any two consecutive integers η and $\eta + 1$ such that $(\eta + 1)\bar{G} \leq a^I$. These arguments prove the claimed result for the injection case. Symmetric arguments can be used to show the claimed result for withdrawals. □

Proof of Proposition 10. By induction.

Stage $N - 1$.

By Lemma 3, $\bar{r}_{N-1}(\cdot, \boldsymbol{s}_{N-1})$ is piecewise linear concave with slope changes at integer multiples of $\bar{G}$, and thus at integer multiples of G as well. Thus, $\max_{a \in \mathbb{R}} \bar{r}_{N-1}(a, \boldsymbol{s}_{N-1})$ has a maximizer a^*_{N-1} that is an integer multiple of G, where we suppress the dependence of this maximizer on $\boldsymbol{s}_{N-1}$. This maximizer is also finite, for every given $\boldsymbol{s}_{N-1}$, because, as discussed in §4.4.1, the linear program (4.13) is infeasible when $a \notin [-a^I, a^W]$. Notice that $a^*_{N-1} \geq 0$ since injecting ($a < 0$) incurs an additional cost compared to doing nothing ($a = 0$). If $a^*_{N-1} = 0$, then $b_{N-1}(y_{N-1}, \boldsymbol{F}_{N-1}) = y_{N-1}$ for all $y_{N-1} \in [0, \bar{y}]$, which is a linear function in y_{N-1} with slope equal to 1. Suppose that $a^*_{N-1} > 0$, that is, withdrawal is optimal. Then, it holds that $b_{N-1}(y_{N-1}, \boldsymbol{F}_{N-1}) = 0$ for all $y_{N-1} \in [0, \min\{a^*_{N-1}, \bar{y}\}]$ and $b_{N-1}(y_{N-1}, \boldsymbol{F}_{N-1}) = y_{N-1} - a^*_{N-1}$ for all $y_{N-1} \in (\min\{a^*_{N-1}, \bar{y}\}, \bar{y}]$. This target function is piecewise linear in y_{N-1} with possible slopes equal to 0 and 1. Thus, we have established the claimed structure of the target function at stage $N - 1$. This structure implies that $\underline{b}_{N-1}(\boldsymbol{F}_{N-1}) = 0$, while $\bar{b}_{N-1}(\boldsymbol{F}_{N-1}) = \bar{y}$ if $a^*_{N-1} = 0$ and $\bar{b}_{N-1}(\boldsymbol{F}_{N-1}) = 0$ if $a^*_{N-1} > 0$, which is consistent with our claimed partitioning of the feasible inventory set $\mathcal{Y}$. Finally, the target function structure also implies that the difference $y_{N-1} - b_{N-1}(y_{N-1}, \boldsymbol{F}_{N-1})$ is not increasing in y_{N-1} and the values returned by this function lie in the same region of the inventory partition that y_{N-1} belongs to.

By definition, $W_{N-1}(\cdot, \boldsymbol{F}_{N-1})$ is constant at zero, and thus trivially satisfies the claimed property. By Lemma 2, $V_{N-1}(\cdot, \boldsymbol{F}_{N-1})$ is concave. We proceed to prove that this function is piecewise linear concave with breakpoints at integer multiples of G. Recall from above that $a^*_{N-1} \equiv \operatorname{argmax}_{a \in \mathbb{R}} \bar{r}_{N-1}(\cdot, \boldsymbol{s}_{N-1})$. Define $\bar{a} := \min\{C^W, a^W, a^*_{N-1}, \bar{y}\}$, which is an integer multiple of G. For $y_{N-1} \in [0, \bar{a})$ it holds that $V_{N-1}(y_{N-1}, \boldsymbol{F}_{N-1}) = \bar{r}_{N-1}(y_{N-1}, \boldsymbol{s}_{N-1})$, and for $y_{N-1} \in [\bar{a}, \bar{y}]$ we have $V_{N-1}(y_{N-1}, \boldsymbol{F}_{N-1}) = \bar{r}_{N-1}(\bar{a}, \boldsymbol{s}_{N-1})$. The function $V_{N-1}(y_{N-1}, \boldsymbol{F}_{N-1})$



(a) Case (i): $y_i = 0$

(b) Case (i): $y_i = 0.5G$

(c) Case (ii): $y_i = 0$

(d) Case (ii): $y_i = 0.5G$

Figure C.2: Conceptual illustration of cases (i) and (ii) in the proof of Part (b) of Proposition 10.

inherits the slope of $\bar{r}_{N-1}(y_{N-1}, \boldsymbol{s}_{N-1})$ for $y_{N-1} \in [0, \bar{a}]$ and has a slope of zero for $y_{N-1} \in [\bar{a}, \bar{y}]$. Because the function $\bar{r}_{N-1}(\cdot, \boldsymbol{s}_{N-1})$ is piecewise linear concave with slope changes slope at integer multiples of G, so is $V_{N-1}(\cdot, \boldsymbol{F}_{N-1})$.

Induction hypothesis. Suppose that the value function $V_i(\cdot, \boldsymbol{F}_i)$ is piecewise linear concave with slope changes at integer multiples of G for stages $i' = i+1, \ldots, N-2$.

Stage i. Because $V_{i+1}(y_{i+1}, \boldsymbol{F}_{i+1})$ is finite (see the proof of Lemma 2), for a fixed $\boldsymbol{F}_{i+1}$, the induction hypothesis implies that $W_i(y_{i+1}, \boldsymbol{F}_i) \equiv \delta \mathbb{E}[V_{i+1}(y_{i+1}, \boldsymbol{F}_{i+1}) | \boldsymbol{F}_i]$ is a piecewise linear function with slope changes at integer multiples of G.

For a given y_i, note that $d\bar{r}(y_i - y_{i+1}, \boldsymbol{s}_i)/dy_{i+1}|_{y_{i+1}=y'_{i+1}} = -d\bar{r}(a, \boldsymbol{s}_i)/da|_{a=y_i-y'_{i+1}}$. Moreover, it holds that $d\bar{r}(a, \boldsymbol{s}_i)/da|_{a=y_i-y'_{i+1}}$ is a nondecreasing function of y'_{i+1} because the function $\bar{r}(\cdot, \boldsymbol{s}_i)$ is concave. Since $W_i(\cdot, \boldsymbol{F}_i)$ and $\bar{r}(\cdot, \boldsymbol{s}_i)$ are piecewise linear concave functions, we can state the optimality condition that determines $b_i(y_i, \boldsymbol{F}_i)$ as follows: $b_i(y_i, \boldsymbol{F}_i)$ is the smallest $y'_{i+1} \in \mathcal{Y}$ such that $d\bar{r}(a, \boldsymbol{s}_i)/da|_{a=y_i-y'_{i+1}} \geq dW_i(y_{i+1}, \boldsymbol{F}_i)/dy_{i+1}|_{y_{i+1}=y'_{i+1}}$.

Before using this optimality condition to formally prove the structure in y_i of the target function, we provide the intuition behind our proof. Observe that as y_i increases the slope of the continuation function $W_i(\cdot, \boldsymbol{F}_i)$ does not change but the slope of the reward function $\bar{r}_i(\cdot, \boldsymbol{s}_i)$ weakly decreases. In other words, $dW_i(\cdot, \boldsymbol{F}_i)/dy_{i+1}|_{y_{i+1}=y'_{i+1}}$ is a nonincreasing step function with changes at integer multiples of G that does not depend on y_i, whereas $d\bar{r}(a, \boldsymbol{s}_i)/da|_{a=y_i-y'_{i+1}}$ is a nondecreasing step function with changes at integer multiples of G that shifts to the right by the same proportion by which y_i is increased. Figure C.2 provides illustrative examples of this property. Our proof of the basestock target function structure relies on two possible types of optima. The first type of optimum, illustrated in panels (a) and (b) of Figure C.2 occurs when the slope of the reward function is "bracketed" above and below by the slopes of the continuation function as y_i is increased starting from a value that is an integer multiple of G. In this case, the target function is a constant until the reward function slope is no longer bracketed. The second type of optimum, illustrated in panels (c) and (d) of Figure C.2 occurs when the continuation function slope is bracketed between the reward function slopes as y_i is increased. In this case, the target function increases proportionately to the increase in y_i until the continuation function slope is no longer bracketed. We now make these intuitive arguments formal.

Let $dW(q)$ be the slope of the function $W_i(\cdot, \boldsymbol{F}_i)$ in the interval $[qG, (q+1)G)$ for $q = 0, 1, \ldots, (\bar{y}/G) - 1$. By Lemma 3, $\bar{r}_i(\cdot, \boldsymbol{s}_i)$ changes slope at integer multiples of $\bar{G}$, and thus at integer multiples of G as well. Define $d\bar{r}(q) := d\bar{r}(a, \boldsymbol{s}_i)/da|_{a=(q-1)\bar{G}}$ for $q = -(a^I/G)+1, -(a^I/G)+2, \ldots, a^W/G$. We define these slope functions to be right continuous at breakpoints, except for the right boundary point, where these functions are defined to be left continuous. Because the function $\bar{r}_i(-y_{i+1}, \boldsymbol{s}_i) + W_i(y_{i+1}, \boldsymbol{F}_i)$ changes slope in y_{i+1} at integer multiples of G, it holds that $b_i(0, \boldsymbol{F}_i) = \bar{q}G$ for some nonnegative integer $\bar{q} \leq \min\{a^I/G, \bar{y}/G\}$. Suppose that $0 < \bar{q} < \min\{a^I/G, \bar{y}/G\}$ (we discuss the boundary cases later). The optimality condition stated above implies that the following conditions must hold for $\bar{q}G$ to be optimal at $y_i = 0$:

$$d\bar{r}(\lfloor \bar{q}G/\bar{G} \rfloor) \geq dW(\bar{q}), \tag{C.11}$$

$$d\bar{r}(\lfloor (\bar{q}G - \epsilon)/\bar{G} \rfloor) < dW(\bar{q}-1), \text{for some } \epsilon > 0, \tag{C.12}$$

where $\lfloor \rfloor$ is the floor function. In addition, we could either have: (ai) $d\bar{r}(\lfloor (\bar{q}G - \epsilon)/\bar{G} \rfloor) \geq dW(\bar{q})$ (see Figure C.2(a) for an example) or (aii) $d\bar{r}(\lfloor (\bar{q}G - \epsilon)/\bar{G} \rfloor) < dW(\bar{q})$ (see Figure C.2(c) for an example). We now characterize an interval $[0, \eta G) \subseteq \mathcal{Y}$ by suitably defining a positive integer η such that the target function has slope 0 or 1 if case (i) or (ii) holds, respectively.

Case (i). Find the smallest positive integer η such that $d\bar{r}(\lfloor (\bar{q}-\eta)G/\bar{G} \rfloor) < dW(\bar{q})$. When such an η does not exist, set η equal to $\bar{q}/G$ (that is, we reach the left boundary of $\mathcal{Y}$). Therefore, for all $y_i \in [0, \eta G)$ it holds that $d\bar{r}(\lfloor (\bar{q}G - y_i)/\bar{G} \rfloor) \geq dW(\bar{q})$. Further, it holds that there exists an $\epsilon' > 0$ such that $d\bar{r}(\lfloor (\bar{q}G - y_i - \epsilon')/\bar{G} \rfloor) < dW(\bar{q}-1)$. This inequality follows from (C.12) and the concavity of $\bar{r}_i(\cdot, \boldsymbol{s}_i)$. Thus, we have $b_i(y_i, \boldsymbol{F}_i) = \bar{q}G$, $\forall y_i \in [0, \eta G]$. Panels (a) and (b) of Figure C.2 provide illustrative examples.

Case (ii). Find the smallest positive integer η such that $d\bar{r}(\lfloor (\bar{q}G - \epsilon)/\bar{G} \rfloor) \geq dW(\bar{q}+\eta)$.

When such an η does not exist, set η equal to $(\bar{y} - \bar{q}G)/G$ (that is, we reach the right boundary of $\mathcal{Y}$). Therefore, for all $y_i \in [0, \eta G)$, there exists an $\epsilon' > 0$ such that $d\bar{r}(\lfloor(\bar{q}G - \epsilon)/\bar{G}) < dW(\lfloor(\bar{q}G + y_i - \epsilon')/G\rfloor)$. Moreover, by (C.11) and the concavity of $W_i(\cdot, \boldsymbol{F}_i)$ we have $d\bar{r}(\lfloor \bar{q}G/\bar{G}\rfloor) \geq dW(\lfloor(\bar{q}G + y_i)/G\rfloor)$. Thus, it holds that $b_i(y_i, \boldsymbol{F}_i) = \bar{q}G + (y_i - \bar{q}G)$, $\forall y_i \in [0, \eta G]$. Panels (a) and (b) of Figure C.2 provide illustrative examples.

Irrespective of case (i) or (ii), $b_i(\eta G, \boldsymbol{F}_i) = \bar{q}G$. However, at $y_i = \eta G$ the type of optimum changes from the type of optimum at $y_i = 0$, that is, if case (i) is true at $y_i = 0$ then case (ii) holds at $y_i = \eta G$, and if case (ii) is true at $y_i = 0$ then case (i) holds at $y_i = \eta G$. We then repeat the procedure described above to identify a positive integer η' such that the target function either is constant or increases within the interval $[\eta G, \eta' G] \subset \mathcal{Y}$. This process is iterated until we reach the right boundary of $\mathcal{Y}$.

Now we consider the boundary cases. When $\bar{q} = 0$ a proof analogous to the interior case handled above, omitted for brevity, establishes the claimed result. When $\bar{q} = \min\{a^I/G, \bar{y}/G\}$, we have $b_i(y_i, \boldsymbol{F}_i) = \bar{q}G$, $\forall y_i \in \mathcal{Y}$. We have thus proved the claimed piecewise linearity of the target function at stage i.

We now show the partitioning of the feasible inventory set into the stated inject, do nothing, and withdraw regions. It is obvious that $-b_i(0, \boldsymbol{F}_i) \leq 0$ and $\bar{y} - b_i(\bar{y}, \boldsymbol{F}_i) \geq 0$. Our characterization of $b_i(y_i, \boldsymbol{F}_i)$ as a piecewise linear function of y_i with slopes 0 or 1 implies that (i) $b_i(y_i, \boldsymbol{F}_i)$ is continuous and nondecreasing in y_i, and (ii) $y_i - b_i(y_i, \boldsymbol{F}_i)$ is a nondecreasing function of y_i. The first property implies that the set $\{y_i | y_i = b_i(y_i, \boldsymbol{F}_i)\}$ is a nonempty closed interval. Thus, the functions $\underline{b}_i(\boldsymbol{F}_i)$ and $\overline{b}_i(\boldsymbol{F}_i)$ are well defined. The second property implies that an optimal storage action does not increase in y_i, which proves the partitioning of the inventory interval into the inject, do nothing, and withdraw regions and the target function returns values that lie in the region of this partition that y_i belongs to.

When y_i is within an interval $[qG, (q+1)G]$ the target function is either (i) equal to a constant, which implies that an optimal storage action increases at rate 1 in y_i and the value function $V_i(\cdot, \boldsymbol{F}_i)$ inherits the slope of $\bar{r}_i(\cdot, \boldsymbol{s}_i)$ or (ii) increases with slope 1, the optimal action is a constant, and the value function $V_i(\cdot, \boldsymbol{F}_i)$ inherits the slope of the continuation function $W_i(\cdot, \boldsymbol{F}_i)$. Since both $\bar{r}_i(\cdot, \boldsymbol{s}_i)$ and $W_i(\cdot, \boldsymbol{F}_i)$ are piecewise linear functions with slope changes at multiples of G, the claimed result follows.

The claimed properties hold for all the stages by the principle of mathematical induction. The decision rule (4.17) tries to change the current storage inventory level to the basestock target function value while accounting for the storage injection and withdrawal capacities. □

Proof of Proposition 11. Let π^* be an optimal policy to SDP (4.15). The value of this policy is $V_0(x_0, \boldsymbol{F}_0)$. Let $\pi^{ST,*}$ and $\pi^{TR,*}$ be the storage and transport policies that make up π^*. The inequality (4.21) follows because $\pi^{ST,*}$ and $\pi^{TR,*}$ are feasible, but not necessarily optimal, policies to (4.19) and (4.20), respectively. □

References

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

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

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

D. Adelman and D. Klabjan. Computing near optimal policies in generalized joint replenishment. *INFORMS Journal on Computing*, 24(1), 2012.

D. Adelman and A. Mersereau. Dynamic capacity allocation to customers who remember past service. *Management Science*, 59(3):592–612, 2013.

R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, New Jersey, USA, 1993.

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

Ø. Arvesen, V. Medbø, S. E. Fleten, A. Tomasgard, and S. Westgaard. Linepack storage valuation under price uncertainty. *Energy*, 52(1):155–164, 2013.

C. H. Bannister and R. J. Kaye. A rapid method for optimization of linear systems with storage. *Operations Research*, 39(2):220–232, 1991.

A. Barbieri and M. B. Garman. Putting a price on swings. *Energy and Power Risk Management*, 1(6):17–19, 1996.

C. Bender. Dual pricing of multi-exercise options under volume constraints. *Finance and Stochastics*, 15(1):1–26, 2011.

P. B. Bertsekas. *Dynamic Programming and Optimal Control*, volume 2. Athena Scientific, Belmont, MA, USA, 3rd edition edition, 2007.

D. Bertsimas and J.N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, MA, USA, 1997.

N. Bhat, V. F. Farias, and C. C. Moallemi. Non-parametric approximate dynamic programming via the kernel method. Working paper, Columbia Univ., 2012.

J. Birge. Quasi-convex stochastic dynamic programming. Working paper, Univ. of Chicago, 2011.

P. Bjerksund and G. Stensland. Closed form spread option valuation. *Quantitative Finance*, Forthcoming, 2011.

T. Björk. *Arbitrage Theory in Continuous Time*. Oxford University Press, Oxford, Oxfordshire, UK, 2004.

F. Black. The pricing of commodity contracts. *Journal of Financial Economics*, 3(1-2): 167–179, 1976.

C. Blanco, D. Soronow, and P. Stefiszyn. Multi-factor models for forward curve analysis: An introduction to principal component analysis. *Commodities Now*, (June):76–78, 2002.

A. Boogert and C. De Jong. Gas storage valuation using a Monte Carlo method. *The Journal of Derivatives*, 15(3):81–98, 2008.

A. Boogert and C. De Jong. Gas storage valuation using a multifactor price process. *The Journal of Energy Markets*, 4(4):29–52, 2011/12.

A. Boogert and D. Mazières. A radial basis function approach to gas storage valuation. Working paper, Birkbeck University of London, 2011.

O. Boyabatli. Supply management in multi-product firms with fixed proportions technology. Working paper, Singapore Management Univ., 2011.

O. Boyabatli, P. R. Kleindorfer, and S. R. Koontz. Integrating long-term and short-term contracting in beef supply chains. *Management Science*, 57(10):1771–1787, 2011.

J. Breslin, L. Clewlow, T. Elbert, C. Kwok, C. Strickland, and D. van der Zee. Gas storage: Rolling instrinsic valuation. *Energy Risk*, January:61–65, 2009.

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

D. B. Brown and J. E. Smith. Information relaxations, duality, and convex stochastic dynamic programs. Working paper, Duke Univ., 2013.

D. B. Brown, J. E. Smith, and P. Sun. Information relaxations and duality in stochastic dynamic programs. *Operations Research*, 58(4):1–17, 2010.

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

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

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

A. Charnes, J. Drèze, and M. Miller. Decision and horizon rules for stochastic planning problems: A linear example. *Econometrica*, 34(2):307–330, 1966.

Z. Chen and P.A. Forsyth. A semi-Lagrangian approach for natural gas storage valuation and optimal operation. *SIAM Journal on Scientific Computing*, 30(1):339–368, 2007.

L. Clewlow and C. Strickland. *Energy Derivatives: Pricing and Risk Management*. Lacima, London, UK, 2000.

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

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

J. T. Cuddington and Z. Wang. Assessing the degree of spot market integration for U.S. natural gas: Evidence from daily price data. *Journal of Regulatory Economics*, 29(2): 195–210, 2006.

C. A. Dahl and T. K. Matson. Evolution of the U.S. natural gas industry in response to changes in transaction costs. *Land Economics*, 74(3):390–408, 1998.

D. P. de Farias and B. Van Roy. The linear programming approach to approximate dynamic programming. *Operations Research*, 51(6):850–865, 2003.

D. P. de Farias and B. Van Roy. On constraint sampling for the linear programming approach to approximate dynamic programming. *Mathematics of Operations Research*, 29(3):462–478, 2004.

C. de Jong, A. Boogert, and C. Clancy. A decade of rough storage trading results in the uk nbp gas market. Technical report, KYOS Energy Consulting, 2010.

A. De Vany and W. D. Walls. Pipeline access and market integration in the natural gas industry: Evidence from cointegration tests. *The Energy Journal*, 14(4):1–19, 1993.

S. Deng, B. Johnson, and A. Sogomonian. Exotic electricity options and the valuation of electricity generation and transmission assets. *Decision Support Systems*, 30(3):383–392, 2001.

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

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

J. Detemple. *American-style Derivatives: Valuation and Computation*. Taylor & Francis, London, UK, 2006.

S.K. Devalkar, R. Anupindi, and A. 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, New Jersey, USA, 1994.

EIA. Natural gas 1996: Issues and trends. Technical report, Energy Information Agency (EIA), 1996.

EIA. Anual Energy Outlook 2013. Technical report, Energy Information Agency (EIA), 2013.

A. Eydeland and K. Wolyniec. *Energy and Power Risk Management*. John Wiley and Sons Inc., Hoboken, NJ, USA, 2003.

V. F. Farias and B. Van Roy. An approximate dynamic programming approach to network revenue management. Working paper, Stanford Univ., 2007.

J. P. Favennec. *Refinery Operation and Management*. Editions Technip, Paris, France, 2001.

FEA. StoragePLUS features. Technical report, Financial Engineering Associates (FEA), http://www.fea.com/products/energy/storage_plus/features.html, 2013.

B.J. Felix and C. Weber. Gas storage valuation applying numerically constructed recombining trees. *European Journal of Operational Research*, 216(1):178–187, 2012.

N. Friedman and B. Philbin. New York spot natural-gas prices rise to record as cold approaches. *Wall Street Journal*, January 6 2014.

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

P. Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer, New York, NY, USA, 2004.

P. Glasserman and B. Yu. Simulation for American options: Regression now or regression later. In H. Niederreiter, editor, *Monte Carlo and Quasi-Monte Carlo Methods 2002*, pages 213–226. Springer-Verlag, Berlin, Germany, 2004.

J. Gray and P. Khandelwal. Realistic gas storage models II: Trading strategies. *Commodities Now*, September:1–5, 2004.

J. Gray and P. Khandelwal. Towards a realistic gas storage model. *Commodities Now*, pages 1–4, June 2004.

Inc. Gurobi Optimization. Gurobi optimizer reference manual version 5.0. Houston, Texas: Gurobi Optimization, 2012.

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

L. G. Gyurko, B. Hambly, and J. H. Witte. Monte Carlo methods via a dual approach for some discrete time stochastic control problems. *ArXiv e-prints*, 2011.

E. G. Haug. *The Complete Guide to Option Pricing Formulas*. McGraw-Hill, NY, USA, second edition, 2006.

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

M. B. Haugh and L. Kogan. Duality theory and approximate dynamic programming for pricing American options and portfolio optimization. In John R. Birge and Vadim Linetsky, editors, *Financial Engineering*, volume 15 of *Handbooks in Operations Research and Management Science*, pages 925 – 948. Elsevier, 2007.

D. P. Heyman and M. J. Sobel. *Stochastic Models in Operations Research: Stochastic Optimization*, volume 2. Dover, Murray Hill, New Jersey, 2003.

J.B. Hiriart-Urruty and C. Lemaréchal. *Fundamentals of Convex Analysis*. Springer-Verlag, Berlin, Germany, 2001.

T. S. Y. Ho and S. B. Lee. Term structure movements and pricing interest rate contingent claims. *Journal of Finance*, 41(5):1011–1029, 1986.

IEA. World energy outlook 2012. Technical report, International Energy Agency (IEA), 2012.

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

INGAA. Natural gas pipeline and storage infrastructure projections through 2030. Technical report, Interstate Natural Gas Association of America (INGAA), 2009.

INGAA. North American midstream infrastructure through 2035: Capitalizing on our energy abundance. Technical report, Interstate Natural Gas Association of America (INGAA), 2014.

P. Jaillet, E.I. Ronn, and S. Tompaidis. Valuation of commodity-based swing options. *Management Science*, 50(7):909–921, 2004.

J. H. Kim and W. B. Powell. Optimal energy commitments with storage and intermittent supply. *Operations research*, 59(6):1347–1360, 2011.

D. Klabjan and D. Adelman. An infinite-dimensional linear programming algorithm for deterministic semi-Markov decision processes on borel spaces. *Mathematics of Operations Research*, 32(3):528–550, 2007.

P. R. Kleindorfer and D. J. Wu. Integrating long-and short-term contracting via business-to-business exchanges for capital-intensive industries. *Management Science*, 49(11):1597–1615, 2003.

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

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.

G. Levy. *Computational Finance: Numerical Methods for Pricing Financial Instruments*. Butterworth-Heinemann, Oxford, UK, 2004.

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

S. Lund, J. Manyika, S. Nyquist, L. Mendonca, and S. Ramaswamy. Game changers: Five opportunities for US growth and renewal. Technical report, McKinsey Global Institute, July 2013.

A. S. Manne. Linear programming and sequential decisions. *Management Science*, 60(3): 259–267, 1960.

S. Maragos. Valuation of the operational flexibility of natural gas storage reservoirs. In E. Ronn, editor, *Real Options and Energy Management Using Options Methodology to Enhance Capital Budgeting Decisions*, pages 431–456. Risk Publications, London, UK, 2002.

W. Margrabe. The value of an option to exchange one asset for another. *The Journal of Finance*, 33(1):177–186, 1978.

R. E. Markland. Analyzing multi-commodity distribution networks having milling-in-transit features. *Management Science*, 21(12):1405–1416, 1975.

R. E. Markland and R. J. Newett. Production-distribution planning in a large scale commodity processing network. *Decision Sciences*, 7(4):579–594, 1976.

V. Marmer, D. Shapiro, and P. MacAvoy. Bottlenecks in regional markets for natural gas transmission services. *Energy Economics*, 29(1):37–45, 2007.

J. E. Mason. *Markov Decision Processes and Approximate Dynamic Programming Methods for Optimal Treatment Design*. PhD thesis, North Carolina State University, 2012.

N. Meinshausen and B. M. Hambly. Monte Carlo methods for the valuation of multiple-exercise options. *Mathematical Finance*, 14(4):557–583, 2004.

C. C. Moallemi, S. Kumar, and B. Van Roy. Approximate and data-driven dynamic programming for queuing networks. Working paper, Stanford Univ., 2008.

J. R. Morrison and P. R. Kumar. New linear program performance bounds for queuing networks. *Journal of Optimization Theory and Applications*, 100(3):575–597, 1999.

S. Nadarajah, N. Secomandi, G. Sowers, and J. Wassick. Real option management of hydrocarbon cracking operations. *Optimization and Analytics in the Oil and Gas Industry*, Forthcoming, 2013.

J. Nascimento and W. Powell. An optimal approximate dynamic programming algorithm for concave, scalar storage problems with vector-valued controls. *IEEE Transactions on Automatic Control*, Forthcoming, 2013a.

J. Nascimento and W.B. Powell. An optimal approximate dynamic programming algorithm for concave, scalar storage problems with vector-valued controls. *IEEE Transactions on Automatic Control*, 58(12):2995–3010, 2013b.

Juliana M. Nascimento and Warren B. Powell. An optimal approximate dynamic programming algorithm for the lagged asset acquisition problem. *Mathematics of Operations Research*, 34(1):210–237, 2009.

B. O'Donoghue, Y. Wang, and S. Boyd. Min-max approximate dynamic programming. *Proceedings IEEE Multi-Conference on Systems and Control*, pages 424–431, 2011.

B. O'Donoghue, Y. Wang, and S. Boyd. Iterated approximate value functions. *Proceedings European Control Conference*, pages 3882–3888, 2013.

M. Petrik. Approximate dynamic programming by minimizing distributionally robust bounds. *ArXiv e-prints*, 2012.

M. Petrik and S. Zilberstein. Constraint relaxation in approximate linear programs. In *Proceedings of the Twenty-Sixth International Conference on Machine Learning*, pages 809–816, Montreal, Canada, 2009.

R. S. Pindyck. Volatility and commodity price dynamics. *Journal of Futures Markets*, 24 (11):1029–1047, 2004.

Pipeline Knowledge & Development. The interstate natural gas transmission system: Scale, physical complexity and business model. Technical report, The Interstate Natural Gas Association of America (INGAA), 2010. http://www.ingaa.org/11885/Reports/10724.aspx.

W. B. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality, 2nd Edition.* John Wiley & Sons, Hoboken, New Jersey, USA, 2011.

W. B. Powell, A. George, H. Simão, W. Scott, A. Lamont, and J. Stewart. Smart: a stochastic multiscale model for the analysis of energy resources, technology, and policy. *INFORMS Journal on Computing*, 24(4):665–682, 2012a.

W. B. Powell, A. George, H. Simo, W. Scott, A. Lamont, and J. Stewart. Smart: A stochastic multiscale model for the analysis of energy resources, technology, and policy. *INFORMS Journal on Computing*, 24(4):665–682, 2012b.

M.L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1994.

S. Resnick. *A Probability Path*. Birkhäuser, Boston, MA, USA, 1999.

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

F. Rømo, A. Tomasgard, L. Hellemo, M. Fodstad, B. H. Eidesen, and B. Pedersen. Optimizing the Norwegian natural gas production and transport. *Interfaces*, 39(1):46–56, 2009.

M. Rubinstein. Return to OZ. *Risk Magazine*, 1994.

A. J. Schaefer, M. D. Bailey, S. M. Shechter, and M. S. Roberts. Modeling medical treatment using Markov decision processes. *Operations Research and Health Care*, pages 593–612, 2005.

E. Schwartz and J. E. Smith. Short-term variations and long-term dynamics in commodity prices. *Management Science*, 46(7):893–911, 2000.

P. J. Schweitzer and A. Seidmann. Generalized polynomial approximations in Markovian decision processes. *Journal of Mathematical Analysis and Applications*, 110(2):568–582, 1985.

W. R. Scott, W. B. Powell, and S. Moazeni. Least squares policy iteration with instrumental variables vs. direct policy search: Comparison against optimal benchmarks using energy storage. Working paper, Princeton Univ., 2014.

N. Secomandi. An analysis of the control-algorithm re-solving issue in inventory and revenue management. *Manufacturing & Service Operations Management*, 10(3):468–483, 2008.

N. Secomandi. On the pricing of natural gas pipeline capacity. *Manufacturing & Service Operations Management*, 12(3):393–408, 2010.

N. Secomandi. Optimal commodity trading with a capacitated storage asset. *Management Science*, 56(3):1090–1049, 2010.

N. Secomandi. The role of price spreads and reoptimization in the real option management of commodity storage assets. Working paper, Carnegie Mellon Univ., 2012.

N. Secomandi. Analysis and enhancement of practice-based policies for the real option management of commodity storage assets. Working paper, Carnegie Mellon Univ., 2014.

N. Secomandi and D. Seppi. *Real Options and Merchant Operations of Energy and Other Commodities*. Foundations and Trends in Technology, Information and Operations Management, 2014. forthcoming.

N. Secomandi and M. X. Wang. A computational approach to the real option management of network contracts for natural gas pipeline transport capacity. *Manufacturing & Service Operations Management*, 14(3):441–454, 2012.

N. Secomandi, G. Lai, F. Margot, A. Scheller-Wolf, and D. Seppi. Merchant commodity storage and term structure model error. Working paper, Carnegie Mellon Univ., 2012.

D. Seppi. Risk-neutral stochastic processes for commodity derivative pricing: An introduction and survey. In Ehud I. Ronn, editor, *Real Options and Energy Management*. Risk Books, 2002.

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

H. P. Simo, J. Day, A. P. George, T. Gifford, J. Nienow, and W. B. Powell. An approximate dynamic programming algorithm for large-scale fleet management: A case application. *Transportation Science*, 43(2):178–197, 2008.

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

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

R. Smith. Can gas undo nuclear power? *Wall Street Journal*, January 30 2013.

M. Thompson. Natural gas storage valuation, optimization, market and credit risk management. Working paper, Queens Univ., 2012.

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

M. Thompson, M. Davison, and H. Rasmussen. Natural gas storage valuation and optimization: A real options application. *Naval Research Logistics*, 56(3):226–238, 2009.

D.M. Topkis. *Supermodularity and Complementarity*. Princeton University Press, Princeton, New Jersey, USA, 1998.

M. A. Trick and S. E. Zin. Spline approximations to value functions. *Macroeconomic Dynamics*, 1(1):255–277, 1997.

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

J. N. Tsitsiklis and B. Van Roy. Feature-based methods for large scale dynamic programming. *Machine Learning*, 22(1-3):59–94, 1996.

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

Vladimir Naumovich Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, New York, USA, 2000.

M. H. Veatch. Approximate linear programming for networks: Average cost bounds. Working paper, Gordon College, 2010.

S. Veeraraghavan and A. Scheller-Wolf. Now or later: A simple policy for effective dual sourcing in capacitated systems. *Operations Research*, 56(4):850–864, 2008.

P. Veronesi. *Fixed Income Securities: Valuation, Risk, and Risk Management*. John Wiley and Sons Inc., Hoboken, NJ, USA, 2010.

Y. Wang and S. Boyd. Approximate dynamic programming via iterated bellman inequalities. *Working paper, Stanford Univ.*, 2010.

S.J. Wright. *Primal-dual interior-point methods*. Society for Industrial and Applied Mathematics, Pennsylvania, USA, 1997.

O.Q. Wu, D.D. Wang, and Z. Qin. Seasonal energy storage operations with limited flexibility: The price adjusted rolling intrinsic policy. *Manufacturing & Service Operations Management*, 14(3):455–471, 2012.

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

Y. Zhou, A. Scheller-Wolf, N. Secomandi, and S. Smith. Managing wind-based electricity generation in the presence of storage and transmission capacity. Working paper, Carnegie Mellon Univ., 2013a.

Y. Zhou, A. Scheller-Wolf, N. Secomandi, and S. Smith. Is it more valuable to store or destroy electricity surpluses? Working paper, Carnegie Mellon Univ., 2013b.