An Approximate Dynamic Programming Approach to Benchmark Practice-based Heuristics for Natural Gas Storage Valuation

Guoming Lai[1], François Margot[2], Nicola Secomandi[2]
[1]McCombs School of Business, University of Texas at Austin, 1 University Station, B6000, GSB 3.136, Austin, TX 78712-1178, USA
[2]Tepper School of Business, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213-3890, USA
guoming.lai@mccombs.utexas.edu, {fmargot, ns7}@andrew.cmu.edu
Tepper Working Paper 2008-E38
Forthcoming in *Operations Research*
Area of review and editor: Financial Engineering, Mark N. Broadie
Submitted: September 2008; Revised: February 2009, May 2009; Accepted: July 2009
Keywords: Finance: Asset pricing, real options, storage valuation; Dynamic programming: heuristics, Markov, upper bounds; Industries: Petroleum/natural gas

Abstract

The valuation of the real option to store natural gas is a practically important problem that entails dynamic optimization of inventory trading decisions with capacity constraints in the face of uncertain natural gas price dynamics. Stochastic dynamic programming is a natural approach to this valuation problem, but it does not seem to be widely used in practice because it is at odds with the high-dimensional natural-gas price evolution models that are widespread among traders. According to the practice-based literature, practitioners typically value natural gas storage heuristically. The effectiveness of the heuristics discussed in this literature is currently unknown, because good upper bounds on the value of storage are not available. We develop a novel and tractable approximate dynamic programming method that coupled with Monte Carlo simulation computes lower and upper bounds on the value of storage, which we use to benchmark these heuristics on a set of realistic instances. We find that these heuristics are extremely fast to execute but significantly suboptimal as compared to our upper bound, which appears to be fairly tight and much tighter than a simpler perfect information upper bound; computing our lower bound takes more time than using these heuristics, but our lower bound substantially outperforms them in terms of valuation. Moreover, with periodic reoptimizations embedded in Monte Carlo simulation, the practice-based heuristics become nearly optimal, with one exception, at the expense of higher computational effort. Our lower bound with reoptimization is also nearly optimal, but exhibits a higher computational requirement than these heuristics. Besides natural gas storage, our results are potentially relevant for the valuation of the real option to store other commodities, such as metals, oil, and petroleum products.

1. Introduction

In North America, natural gas merchants manage contracts for the capacity of storage facilities as real options on natural gas prices, whose values derive from the intertemporal trading of natural gas allowed by storage. Organized markets such as the New York Mercantile Exchange (NYMEX) and the IntercontinentalExchange (ICE) trade natural gas related financial instruments, including futures and options on futures. Practitioners can use them to hedge and price natural gas storage contracts using risk-neutral valuation techniques. Valuing such a contract is difficult because it entails dynamic optimization of inventory trading decisions with capacity constraints in the face of uncertain natural gas price dynamics. This is an important problem that has received significant attention within the energy trading community.

Stochastic dynamic programming (Stokey and Lucas [36], Puterman [30], Bertsekas [4]) is the natural approach to this valuation problem (Weston [40]), but it is tractable only when a low-dimensional spot price model is employed to describe the stochastic evolution of the price of natural gas (Clewlow and Strickland [14, Chapter 8], Seppi [34], Eydeland and Wolyniec [16, Chapter 4], Geman [18, Chapter 3]). Natural gas storage traders do not appear to like this approach, since they doubt that the dynamics implied by such price models are consistent with the dynamics of the NYMEX or ICE financial instruments they trade to hedge price risk. For example, in discussing the viability of using stochastic dynamic programming for natural gas storage valuation, Eydeland and Wolyniec [16, p. 367] make the following observations:

> Great care must be taken when specifying and calibrating spot processes for the use in optimization, so that they are consistent with the hedging strategy to be pursued. Additionally, even for a given set of forward information, the critical surface may exhibit unstable behavior that renders it of limited use as a hedging tool.

Consequently, according to the practice-based literature (Maragos [27, pp. 449-453], Eydeland and Wolyniec [16, pp. 351-367], Gray and Khandelwal [21, 20]), the preferred modeling approach among natural gas storage traders seems to be modeling the full dynamics of the futures term structure using high-dimensional forward models (Clewlow and Strickland [14, Chapter 8], Seppi [34], Eydeland and Wolyniec [16, Chapter 5], Geman [18, Chapter 3]). This modeling choice overwhelms stochastic dynamic programming and makes the use of alternative valuation approaches necessary. Thus, according to this practice-based literature, practitioners typically value natural gas storage heuristically. Interestingly, commercial software products for the valuation of natural gas storage (see, e.g., FEA [17], KYOS [26]) include heuristic valuation models that can accommodate the stated high-dimensional forward models.

Two such heuristics combine linear programming and spread option valuation methods with or without periodic reoptimization embedded in Monte Carlo simulation; we label the four resulting heuristics LP, LPN, RLP, and RLPN, where LP abbreviates linear program, LPN indicates the so called LP with "net" capacity constraints, and the prefix R indicates reoptimization (Gray and Khandelwal [21] refer to the LP and RLP heuristics as the static and dynamic basket-of-spreads models). An additional heuristic is based on reoptimization of a deterministic dynamic program,

Table 1: Models and Policies Studied in this Paper.

	Policy	
Model	Without Reoptimization	With Reoptimization
I	I	RI
LP	LP	RLP
LPN	LPN	RLPN
ADP	ADP	RADP

which computes the intrinsic value of storage, within Monte Carlo simulation; we label this heuristic RI, where I abbreviates intrinsic and R stands for reoptimization (Gray and Khandelwal [21] refer to this heuristic as the rolling-intrinsic model). The effectiveness of these heuristics is currently unknown since good upper bounds on the value of storage are not available. Our objective in this paper is to assess it.

We contribute to the existing literature in two ways. First, we develop a novel and tractable approximate dynamic programming (ADP) method to value the real option to store a commodity, such as natural gas, which uses a high-dimensional model of the evolution of the forward curve. Our approach is based on transforming the intractable full information stochastic dynamic program that models the storage valuation problem into a tractable and approximate lower dimensional Markov decision process (MDP). We leverage our structural analysis of the optimal policy of this model, which exhibits two stage and price-state dependent basestock targets, to speed up its computation.

We use the optimal policy and value function of the ADP model within Monte Carlo simulation to compute both lower and upper bounds on the value of storage, that is, the optimal value function of the exact MDP in the initial stage and state. We compute the upper bound by applying the information relaxation and duality approach developed by Brown et al. [9]. We denote our upper bound by DUB, where D and UB abbreviate dual and upper bound, respectively. We also compute a perfect information upper bound, which we label PIUB, as a benchmark for DUB. Moreover, we compute another lower bound by reoptimizing our ADP model within Monte Carlo simulation. Thus, we obtain two lower bounds, the ADP and the RADP lower bounds. We point out that removing the price stochasticity from our ADP model yields the RI model, in which case the RADP and RI bounds coincide. Table 1 summarizes the models and policies that we analyze in this paper.

Second, we use DUB and the two ADP and RADP lower bounds to benchmark the practice-based heuristics on a set of realistic instances based on NYMEX price data and additional data available in the energy trading literature. We find that the intrinsic value of storage accounts for a relatively small amount of the total value of storage, but its computation is extremely fast. The LP heuristic is also very fast and captures significantly more value than the intrinsic value of storage, but its suboptimality is large when compared to DUB, which is fairly tight and much tighter than PIUB. The speed of the LPN heuristic is comparable to that of the LP heuristic, but its valuations are lower than those of the latter heuristic; sometimes they are also below the intrinsic value of storage. Our ADP lower bound exhibits less suboptimality, in most cases, but higher computational requirement than these heuristics.

Moreover, reoptimization improves the valuation performance of all the policies. In almost all the instances, the RLP, RI, and RADP lower bounds are all nearly optimal when compared to DUB. In contrast, the RLPN lower bound improves but remains substantially suboptimal. Of course, all these reoptimized lower bounds are more expensive to compute. In particular, the computational requirement of the RADP lower bound is higher than those of the other lower bounds. Overall, the RI heuristic strikes the best compromise between computational efficiency and valuation quality on our instances. DUB plays a critical role in making these assessments.

These results are important to natural gas storage traders as they provide scientific validation, support, and guidance for the use of heuristic storage valuation models in practice. Moreover, when we (artificially) eliminate the seasonality in the NYMEX natural gas forward curves that we use, we observe that these results remain substantially similar provided that reoptimization is used. (For brevity, these results are not included in this paper, but are available upon request.) This suggests that our findings related to the reoptimization case have potential relevance for the valuation of the real option to store other commodities, whose forward curves do not exhibit the pronounced seasonality of the natural gas forward curve, e.g., metals, oil, and petroleum products (Geman [18]).

Our work is related to the literature on the valuation of commodity and energy real options (Trigeorgis [39], Smith and McCardle [35], Geman [18]). We are not the first ones to study the valuation of the real option to store natural gas. Thompson et al. [37] and Chen and Forsyth [13] propose continuous-time stochastic control methods for this pricing problem, based on low-dimensional representations of the evolution of the spot price of natural gas. In contrast, we develop a discrete-time stochastic dynamic programming model that uses the high-dimensional representation of the evolution of the natural gas forward curve discussed in the practice-based literature.

Ghiuvea et al. [19] and Barrera-Esteve et al. [3] propose to value natural gas storage as an extended swing option (Jaillet et al. [24]). This approach employs low-dimensional models of the natural gas spot price evolution and imposes exogenous restrictions on the policy space. In contrast, our model uses a high-dimensional representation of the natural gas forward curve and does not impose arbitrary constraints on the trading decisions.

ADP methods typically rely on Monte Carlo simulation and statistical functional approximation to heuristically compute an approximation of the optimal value function of an MDP (Bertsekas and Tsitsiklis [5], Powell [29]). Carmona and Ludkovski [12] and Boogert and de Jong [7] apply such ADP methods to obtain a statistical approximation of the optimal value function of the MDP formulation of the storage valuation problem. Unlike these authors, we do not employ statistical functional approximation to compute our ADP and RADP policies, although we use Monte Carlo simulation for policy evaluation. Moreover, even if the models of Carmona and Ludkovski [12] and Boogert and de Jong [7] could in principle be used with high-dimensional forward price models, these authors do not explore this possibility.

The heuristics that we study in this paper can be interpreted as control algorithms, in the sense of Secomandi [32]. These are optimization-based models that compute heuristic control policies for intractable MDPs. Secomandi [32] analyzes control algorithms for inventory and revenue management, but does not consider the problem that we study.

Adelman [1] discusses math-programming-based ADP approaches that compute lower and upper bounds for problems different than ours. In contrast, DUB is based on the theory developed by Brown et al. [9], who generalize the work of Rogers [31], Andersen and Broadie [2], and Haugh and Kogan [23] on pricing American options (see also Davis and Karatzas [15]). The problem we study is significantly more difficult than the valuation of these options because it features an inventory component that is absent in these papers. Brown et al. [9] illustrate their theory using inventory management examples, but do not consider the problem that we focus on. As discussed by Secomandi [33], this is significantly different from the problems studied in the inventory management literature (Zipkin [41], Porteus [28]).

Gray and Khandelwal [21, 20], Carmona and Ludkovski [12], and Secomandi [33] compare different approaches for natural gas storage valuation. Different from our work, Gray and Khandelwal [21, 20] and Carmona and Ludkovski [12] do not develop an upper bound on the value of storage. Similar to our finding, Secomandi [33] finds that a reoptimized deterministic model can perform very well for natural gas storage valuation, but restricts his study to a one-factor mean-reverting spot price model, while we use a multi-factor forward curve model. Moreover, we benchmark other heuristics that he does not consider.

Our stage and price-state dependent basestock target characterization of the optimal policy of our ADP model extends that established by Secomandi [33]: in each stage, the basestock targets in his model depend only on one price state variable, whereas ours depend on two such variables.

The remainder of this paper is organized as follows. We present the natural gas storage valuation problem and an exact formulation of this problem in §2. We discuss the practice-based heuristics in §3, our ADP model and the upper bounds DUB and PIUB in §4, and our computational implementation of these models in §5. We analyze our numerical results in §6 and conclude in §7.

2. Valuation Problem and Exact Model

In this section, we describe the natural gas storage valuation problem and formulate an exact model of this problem. A natural gas storage contract gives a merchant the right to inject, store, and withdraw natural gas at a storage facility up to given limits during a finite time horizon. The injection and withdrawal capacity limits are expressed in million British thermal units (mmBtus) per unit of time, e.g., day. There are also limits on the minimum and maximum amounts of the natural gas inventory that the merchant can hold under such a contract. Some contracts specify injection and withdrawal capacities as functions of inventory, the so called ratchets (Eydeland and Wolyniec [16, pp. 353-354]). For simplicity, we do not deal with this aspect in this paper (see Remark 4 in §3). There are proportional charges and fuel losses associated with injections and withdrawals.

The wholesale natural gas market in North America features about one hundred geographically dispersed markets for the commodity. NYMEX and ICE trade financial contracts associated with about forty of these markets. The most liquid market is Henry Hub, Louisiana, which is the delivery location of the NYMEX natural gas futures contract. NYMEX also trades options on this contract. Moreover, NYMEX and ICE trade basis swaps, which are financially settled forward locational price differences relative to Henry Hub. Thus, these financial instruments make practical the risk-neutral valuation of natural-gas related cash flows.

The quantity of interest to us is the value of a given natural gas storage contract at the time of its inception. This value depends on how the natural gas price changes over time because a merchant uses this contract to support his trading activity in the natural gas commodity market as follows: buying natural gas and injecting it into the storage facility at a given point in time, storing it for some time, and withdrawing it out of the facility and selling it at a later point in time. Such a contract can be valued as the discounted risk-neutral expected value of the cash flows from optimally operating it during its tenure, while also respecting its operational constraints.

Of primary interest to traders is the value of the "forward" or "monthly-volatility" component of a storage contract (Maragos [27, p. 440], Eydeland and Wolyniec [16, p. 365]). The value of this component can be hedged by trading futures contracts, and corresponds to the value of the cash flows associated with making natural gas trading decisions on a monthly basis. Thus, we restrict attention to the valuation of such cash flows. We acknowledge that decisions under a storage contract can be made more frequently than a month, e.g., daily, so that this contract could generate cash flows more often than once per month. These cash flows could be replicated using instruments such as balance-of-the-month/week contracts and *Gas Daily* options (Eydeland and Wolyniec [16, p. 365]). Incorporating these cash flows in the valuation of a storage contract would clearly increase its value. However, given our focus in this paper, extending our work in this direction is a topic for further research.

The contract tenure spans N futures maturities in set $\mathcal{I} := \{0, \dots, N-1\}$. Inventory trading decisions are made at each maturity time T_i with $i \in \mathcal{I}$. We use the standard notation $F(T_i, T_j)$ to denote the futures price at time T_i with maturity at time T_j, $\forall i, j \in \mathcal{I}$, $j \geqslant i$; $F(T_i, T_i)$ is the spot price at time T_i. For the most part we replace the notation $F(T_i, T_j)$ with the alternative notation $F_{i,j}$ to simplify the exposition: the former notation is useful when dealing with continuous time dynamics of futures prices, the latter simplifies the writing of discrete time dynamic programs. We define the forward curve at time T_i as $\mathbf{F}_i := (F_{i,j}, j \in \mathcal{I}, j \geqslant i)$, $\forall i \in \mathcal{I}$; by convention $\mathbf{F}_N := 0$. Notice that $\mathbf{F}_i$ includes the spot price at time T_i. We define the forward curve at time T_i without including the spot price as $\mathbf{F}'_i := (F_{i,j}, j \in \mathcal{I}, j > i)$, $\forall i \in \mathcal{I} \setminus \{N-1\}$; $\mathbf{F}'_{N-1} := 0$.

A multidimensional version of the celebrated Black [6] model of futures price evolution is a simple model of the dynamics of the forward curve that is discussed in the practice-based literature (Eydeland and Wolyniec [16, Chapter 5] and Gray and Khandelwal [21, 20]). In this model, the risk-neutral dynamics of the price of the natural gas futures associated with maturity T_i are described by a driftless geometric Brownian motion, with maturity-specific constant volatility $\sigma_i > 0$ and standard Brownian motion increment $dZ_i(t)$. Moreover, the standard Brownian motion increments corresponding to two different maturity times T_i and T_j are instantaneously correlated with constant correlation coefficient $\rho_{i,j} \in [-1, 1]$, and $\rho_{i,i} := 1$. This is the following N-factor model:

$$\frac{dF(t, T_i)}{F(t, T_i)} = \sigma_i dZ_i(t), \ \forall i \in \mathcal{I} \tag{1}$$

$$dZ_i(t) dZ_j(t) = \rho_{i,j} dt, \ \forall i, \ j \in \mathcal{I}, \ i \neq j. \tag{2}$$

In our numerical experiments reported in §6, we use this model as representative of the high-dimensional forward models that, as discussed in §1, seem to be employed in practice.

Remark 1 (Interpretation of multidimensional Black model (1)-(2))**.** It is easy to see that model (1)-(2) is equivalent to a model of the risk-neutral evolution of the forward curve where the dynamics of the price of each contract depend on N independent factors and N constant volatility functions, which together make the futures prices at a given point in time in the future correlated random variables. Clewlow and Strickland [14, §8.6] discuss such a model. These types of models are related to the so called string and BGM models (Kennedy [25], Brace et al. [8]) used to value fixed income instruments (see also Eydeland and Wolyniec [16, pp. 205-206]).

We now formulate the storage valuation problem as an MDP by modifying the one-factor spot-price periodic-review model of Secomandi [33]. Our formulation does not depend on the type of model used to represent the forward curve dynamics. In other words, it is not specific to model (1)-(2). For example, it is also relevant when one employs a different forward price model, e.g., a model of the type discussed in Remark 1, but possibly with less than N factors; in this case one can in fact obtain a simplified MDP formulation by replacing the forward curve in the state definition with fewer price-related state variables.

We denote an action by a. We use a monthly inventory review period, so that each review time corresponds to a futures price maturity. A positive action corresponds to a withdrawal followed by a sale, a negative action to a purchase followed by an injection, and zero is the do nothing action. The commercial part of an action taken at time T_i, that is, a purchase or a sale, occurs at time T_i, while the operational component of this action, that is, an injection or a withdrawal, is executed in between times T_i and T_{i+1}, that is, during a review period. This means that the natural gas purchased (respectively, sold) at time T_i is available (respectively, unavailable) in storage at time T_{i+1}.

We normalize the storage contract minimum inventory level to 0, and denote its maximum level as $\overline{x} \in \Re_+$. Thus, the set of feasible inventory levels is $\mathcal{X} := [0, \overline{x}]$. We denote the constant injection and withdrawal capacities in each review period as $C^I < 0$ and $C^W > 0$, respectively. Their absolute values express the maximum amount of inventory that can be injected into and withdrawn out of the facility in each review period. We define the feasible injection and withdrawal action sets, respectively, and the action set with feasible inventory x at any review time as

$$\mathcal{A}^I(x) := [C^I \vee (x - \overline{x}), 0] \tag{3}$$

$$\mathcal{A}^W(x) := [0, x \wedge C^W] \tag{4}$$

$$\mathcal{A}(x) := \mathcal{A}^I(x) \cup \mathcal{A}^W(x), \tag{5}$$

where $\cdot \wedge \cdot \equiv \min\{\cdot, \cdot\}$ and $\cdot \vee \cdot \equiv \max\{\cdot, \cdot\}$.

We denote the immediate reward associated with action a at time T_i as $r(a, s_i)$, where s_i is the spot price at time T_i, that is, $s_i \equiv F_{i,i}$. Let $\alpha^W \in (0,1]$ and $\alpha^I \geqslant 1$ be commodity price adjustment factors needed to model in kind fuel losses. Letting c^W and c^I be positive constant marginal withdrawal and injection costs, respectively, the immediate reward function is

$$r(a,s) \quad := \quad \begin{cases} \left(\alpha^I s + c^I\right) a & \text{if } a \in \Re_- \\ 0 & \text{if } a = 0 \\ \left(\alpha^W s - c^W\right) a & \text{if } a \in \Re_+ \end{cases}, \quad \forall s \in \Re_+. \tag{6}$$

In theory, the storage value under model (1)-(2) can be computed by optimally solving the following stochastic dynamic program:

$$V_N(x_N, \mathbf{F}_N) \quad := \quad 0, \ \forall x_N \in \mathcal{X} \tag{7}$$

$$V_i(x_i, \mathbf{F}_i) \quad = \quad \max_{a \in \mathcal{A}(x_i)} r(a, s_i) + \delta \mathbb{E}\left[V_{i+1}(x_i - a, \tilde{\mathbf{F}}_{i+1}) | \mathbf{F}'_i\right], \ \forall i \in \mathcal{I}, \ (x_i, \mathbf{F}_i) \in \mathcal{X} \times \Re_+^{N-i}; \tag{8}$$

here $V_i(x_i, \mathbf{F}_i)$ is the optimal value function in stage i and state $(x_i, \mathbf{F}_i)$ (in (7) we exploit our convention that $\mathbf{F}_N \equiv 0$); δ in (8) is the one review-period constant risk-free discount factor; and expectation $\mathbb{E}$ in (8) is taken with respect to the risk-neutral distribution of random vector $\tilde{\mathbf{F}}_{i+1}$ conditional on $\mathbf{F}_i$ (in the remainder we use $\tilde{\cdot}$ to denote a random entity). In practice, the number of maturities N associated with natural gas storage contracts is at least twelve, so that model (7)-(8) is computationally intractable because of its high-dimensional state space.

Remark 2 (Action set restriction in (8))**.** The maximization in (8) only allows one type of action, that is, only one of the buy-and-inject, do-nothing, and withdraw-and-sell actions is available. This is without loss of generality by the definition of the reward function (6), that is, performing separate buy-and-inject and withdraw-and-sell actions in the same stage is never optimal.

Remark 3 (Physical holding cost)**.** Model (7)-(8) does not feature any cost for physically holding inventory in between review times. To account for this cost component, we could let h denote such unit cost and add the term $-hx_i$ to the right hand side of (8). However, according to the practice-based literature natural gas storage contracts typically do not seem to include a holding cost. Thus, we proceed without modeling this cost component.

3. Practice-based Heuristics

In this section, we describe the practice-based policies LP, LPN, and I, and their reoptimization versions RLP, RLPN, and RI.

3.1 Models Based on Spread Options

The LP policy is based on spread option valuation and linear programming (as explained below, it also includes a spot sale of inventory at time T_0). A spread option is an option on the difference between two prices with a positive strike price (Carmona and Durrleman [11]). The LP policy uses spread options on the difference between futures prices $F_{i,j}$ and $F_{i,i}$, with $i < j$, adjusted for fuel, and strike price equal to the sum of the values of the time T_i injection and withdrawal marginal costs. We refer to such an option as the i-j spread option. Its time T_0 value is

$$S_0^{i,j}(F_{0,i}, F_{0,j}) := \delta^i \mathbb{E}\left[\left\{\delta^{j-i}\alpha^W \tilde{F}_{i,j} - \left(\alpha^I \tilde{F}_{i,i} + \delta^{j-i}c^W + c^I\right)\right\}^+ | F_{0,i}, F_{0,j}\right]. \tag{9}$$

This is the time T_0 value of injecting one unit of natural gas at time T_i and withdrawing it at time T_j provided that the value of this trade is nonnegative at time T_i ($\{\cdot\}^+ := \max\{\cdot, 0\}$).

The LP policy works with portfolios of spread options $\{q_{i,j}, i, j \in \mathcal{I}, i < j\}$; here $q_{i,j}$ is the notional amount of natural gas associated with spread option i-j. Such a portfolio includes notional amounts for spread options whose injections and withdrawals are associated with maturities 0, 1, $\ldots$, $N-2$, and 1, 2, $\ldots$, $N-1$, respectively. The LP policy also uses a spot sale y_0 that allows one to sell some or all of the inventory available at time T_0 (typically storage contracts do not entail a positive initial inventory, but this decision variable is useful in the reoptimization version of this linear program).

The initial step of the LP policy is to approximate the value of storage at time T_0 by constructing a portfolio of spread options and a spot sale as an optimal solution to linear program (10)-(18) below. The decision variables in this linear program are the notional amounts in set $\{q_{i,j}, i, j \in \mathcal{I}, i < j\}$; the inventory levels in set $\{x_i, i \in \mathcal{I} \setminus \{0\} \cup \{N\}\}$ (these are not needed but simplify the formulation); and the spot sale y_0. This linear program, which only depends on the time T_0 information set $\{x_0, \mathbf{F}_0\}$, follows

$$U_0^{LP}(x_0, \mathbf{F}_0) := \max_{y_0, q, x} s_0 y_0 + \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{I}, i<j} S_0^{i,j}(F_{0,i}, F_{0,j}) q_{i,j} \tag{10}$$

$$\text{s.t. } x_{i+1} = x_i + \sum_{j \in \mathcal{I}, j>i} q_{i,j} - y_0 1\{i = 0\} - \sum_{j \in \mathcal{I}, j<i} q_{j,i}, \ \forall i \in \mathcal{I} \tag{11}$$

$$x_i \leqslant \overline{x}, \ \forall i \in \mathcal{I} \setminus \{0\} \cup \{N\} \tag{12}$$

$$\sum_{j \in \mathcal{I}, j>i} q_{i,j} \leqslant -C^I, \ \forall i \in \mathcal{I} \setminus \{N-1\} \tag{13}$$

$$y_0 \leqslant C^W \tag{14}$$

$$\sum_{i \in \mathcal{I}, i<j} q_{i,j} \leqslant C^W, \ \forall j \in \mathcal{I} \setminus \{0\} \tag{15}$$

$$y_0 \geqslant 0 \tag{16}$$

$$q_{i,j} \geqslant 0,\ \forall i,\ j \in \mathcal{I},\ i < j \tag{17}$$

$$x_i \geqslant 0,\ \forall i \in \mathcal{I} \setminus \{0\} \cup \{N\}. \tag{18}$$

The objective function (10) is the value of the portfolio of spot sale and spread options. Constraint sets (11) and (12) express inventory balance and bounding conditions, respectively ($1\{\cdot\}$ in (11) is the indicator function of event $\{\cdot\}$, which is equal to 1 if this event is true and 0 otherwise). Constraint sets (13)-(15) enforce capacity constraints. Constraint sets (16)-(18) pose nonnegativity conditions on the decision variables.

A version of this model is discussed by Eydeland and Wolyniec [16, p. 362] and Gray and Khandelwal [20] (see also Byers [10]). There is no closed form formula for the coefficients in its objective function. However, they can be numerically computed or, alternatively, they can be approximated using closed form formulas, such as Kirk's formula that we use in §6 (Carmona and Durrleman [11]). Once these coefficients are known, this model can be optimally solved very efficiently.

Let $y_0^{LP}(x_0, \mathbf{F}_0)$ and $\{q_{i,j}^{LP}(x_0, \mathbf{F}_0), i, j \in \mathcal{I}, i < j\}$ be an optimal portfolio, that is, an optimal solution to model (10)-(18). This portfolio can be used to construct a feasible policy for model (7)-(8), which we call the LP policy. The ensuing description of this policy is based on our own understanding of how this could be done in practice.

We define the following quantities:

$$q_{i,j}^{LP,+}(F_{i,i}, F_{i,j}) := \begin{cases} 0 & \text{if } \delta^{j-i}\alpha^W F_{i,j} - \left(\alpha^I F_{i,i} + \delta^{j-i}c^W + c^I\right) \leqslant 0 \\ q_{i,j}^{LP}(x_0, \mathbf{F}_0) & \text{otherwise.} \end{cases} \tag{19}$$

These quantities depend on $\{x_0, \mathbf{F}_0\}$ but we suppress this dependence in our notation for ease of exposition. Given $\mathbf{F}^i := (\mathbf{F}_j)_{j=0}^{i}$, the sequence of forward curves observed up to and including time T_i, the LP policy uses the quantities defined by (19) and the optimal spot-sale $y_0^{LP}(x_0, \mathbf{F}_0)$ to obtain the following action at time T_i:

$$y_0^{LP}(x_0, \mathbf{F}_0)1\{i = 0\} + \sum_{j \in \mathcal{I}, j < i} q_{j,i}^{LP,+}(F_{j,j}, F_{j,i}) - \sum_{j \in \mathcal{I}, j > i} q_{i,j}^{LP,+}(F_{i,i}, F_{i,j}). \tag{20}$$

Notice that this action does not depend on the inventory level x_i; nevertheless the LP policy is feasible for model (7)-(8) as shown in Proposition 1. We denote the time T_0 value of the LP policy by $V_0^{LP}(x_0, \mathbf{F}_0)$, which can be estimated using Monte Carlo simulation as discussed in §5.

Proposition 1 shows that the optimal objective function value of model (10)-(18) is no greater than the value of the LP policy, and that both of these values are lower bounds on $V_0(x_0, \mathbf{F}_0)$.

Proposition 1 (LP policy value)**.** *It holds that* $U_0^{LP}(x_0, \mathbf{F}_0) \leqslant V_0^{LP}(x_0, \mathbf{F}_0) \leqslant V_0(x_0, \mathbf{F}_0)$.

Proof. Constraint sets (11)-(18) are sufficient for the feasibility of the LP policy for model (7)-(8). To see this, notice that constraints (12)-(18) enforce feasibility (in model (7)-(8)) of the inventory levels generated by the LP policy, when this policy exercises all the spread options that define it. If this policy exercises a strictly smaller number of these spread options, then its resulting inventory levels would be strictly smaller, but always nonnegative, than they would be otherwise. This is true because, aside from stage 0, the LP policy only withdraws and sells natural gas that was previously bought and injected. Moreover, constraint sets (13)-(15) make sure that each action of the LP policy respects the capacity limits C^I and C^W. The claimed result follows from observing that the objective function (10) underestimates the value of the LP policy, because it "double counts" the injection or withdrawal costs of the LP policy, and that this policy is feasible but not necessarily optimal for model (7)-(8). □

Remark 4 (Minimum inventory requirements or ratcheted capacity)**.** If one were to impose requirements on the minimum inventory level at given maturity times, or if the injection and/or the withdrawal capacities were ratcheted, model (10)-(18) would have to be reformulated as an integer linear program. In this case, Proposition 1 would not necessarily hold. For brevity and simplicity of exposition, we do not discuss these aspects in this paper.

Conversations with practitioners reveal that a version of model (10)-(18) that combines constraints (13)-(15) is sometimes used in practice. In this formulation, which we refer to as the reformulated linear program, constraints (13) for all $i \in \mathcal{I} \backslash \{N-1\}$, (14), and (15) for all $j \in \mathcal{I} \backslash \{0, N-1\}$, are replaced with the following "net" constraints:

$$C^I \leqslant y_0 - \sum_{j \in \mathcal{I}, j>0} q_{0,j} \leqslant C^W \tag{21}$$

$$C^I \leqslant \sum_{j \in \mathcal{I}, j<i} q_{j,i} - \sum_{j \in \mathcal{I}, j>i} q_{i,j} \leqslant C^W, \ \forall i \in \mathcal{I} \setminus \{0, N-1\}. \tag{22}$$

The objective function of the reformulated linear program is obviously higher than $U_0^{LP}(x_0, \mathbf{F}_0)$. But it may not yield a lower bound to the value of storage since constraints (21)-(22) do not ensure that an optimal portfolio composition in the reformulated linear program also gives rise to a feasible policy for model (7)-(8). It is however possible to use an optimal solution to this linear program to derive such a feasible policy, as now described. The following scheme reflects our own understanding of how one might do this, and defines what we call the LPN policy.

In the initial stage, 0, the LPN policy subtracts from the optimal value of the spot sale the sum of all the optimal notional amounts corresponding to spread options that expire in the money in this stage; withdraws or injects an amount of natural gas equal to the resulting difference according to its sign; and does nothing otherwise. In each other stage i, excluding the last one $(N-1)$, this policy subtracts from the total amount of natural gas to be withdrawn during this stage, which is associated with spread options that expired in the money in earlier stages, the total amount of natural gas to be injected during this stage, which is associated with spread options that expire in the money at time T_i. Call this the net action. There are two possibilities.

(1) The net action is a valid action as defined by (3)-(4), given the current inventory level, that is, it satisfies the injection and withdrawal capacity constraints and leads to a feasible inventory level in the next stage; in this case, the LPN policy executes the net action.

(2) The net action violates one of the capacity constraints (3)-(4).

(2a) If the net action is negative, some spread options that expire in the money at time T_i cannot be operationally exercised, that is, the constraint being violated is (3). In this case, the LPN policy sorts these options in decreasing order of their values at this time, and sequentially executes as many options as possible starting from the most valuable one, without violating injection constraint (3); it then discards the rest of these options from further consideration in any of the remaining stages (partial exercise of an option is possible here, in which case the LPN policy only discards the remaining notional amount of this option).

(2b) If the net action is positive, then some natural gas associated with options that were executed during an earlier stage cannot be withdrawn and must be carried forward, that is, the constraint being violated is (4). The LPN policy withdraws and sells as much natural gas as possible and adds the residual amount of natural gas to a pool of inventory that will be considered as a withdrawal in the computation of the net action in each remaining stage, irrespective of price, until it is completely withdrawn and sold in one of these stages.

In the last stage $(N-1)$, the LPN policy withdraws and sells as much as possible of any remaining inventory (as this was natural gas associated with spread options that were exercised in some earlier stage).

The time T_0 value of the LPN policy can be estimated using Monte Carlo simulation as discussed in §5.

3.2 Model Based on Forward Contracts

Another model of interest is the so called intrinsic value model, which computes the value of storage that can be attributed to seasonality, as expressed by the forward curve in the initial stage (Secomandi [33]). This model is

$$U_N^I(x_N; \mathbf{F}_0) := 0, \ \forall x_N \in \mathcal{X} \tag{23}$$

$$U_i^I(x_i; \mathbf{F}_0) := \max_{a \in \mathcal{A}(x_i)} r(a, F_{0,i}) + \delta U_{i+1}^I(x_i - a; \mathbf{F}_0), \ \forall i \in \mathcal{I}, \forall x_i \in \mathcal{X}. \tag{24}$$

This model computes an optimal policy that only considers the information available at time T_0. This is the I policy, which corresponds to a sequence of purchases-and-injections or withdrawals-and-sales, one for each stage, determined based on the information available at the initial time. The cash flows associated with this policy can be secured at time T_0 by transacting in the forward market for natural gas at this time. The time T_0 value of the I policy is $U_0^I(x_0; \mathbf{F}_0)$.

3.3 Models Based on Reoptimization

It is typically possible to improve the performance of the LP, LPN, and I policies by reoptimizing their associated linear and dynamic programs at each maturity to take advantage of the price and inventory information that becomes available over time, implementing the action pertaining to the maturity when the reoptimization is performed, and repeating this process up to and including the last maturity; reoptimization of the LP and I models is discussed by Gray and Khandelwal [20], that of the I model by Maragos [27]. For brevity, we do not provide the details of this process. The time T_0 values of the reoptimization versions of the LP, LPN, and I policies, that is, the RLP, RLPN, and RI policies, are clearly lower bounds on the value of storage and can be estimated by Monte Carlo simulation as discussed in §5.

4. ADP Model

In this section, we discuss our ADP model, some structural results for this model, and how we use it to compute lower and upper bounds on the value of storage.

4.1 ADP Policy

Our approach to reduce the computationally intractable and exact model (7)-(8) to a computationally tractable and approximate model uses information *and* value function approximations, which allow us to reduce the high dimensionality of model (7)-(8) and compute an approximate and feasible policy for this model.

We introduce the ADP model by reformulating the exact model. To this aim, define the forward curve at time T_i excluding the spot and prompt month futures prices as $\mathbf{F}''_i := (F_{i,j}, j \in \mathcal{I}, j > i+1)$, $\forall i \in \mathcal{I} \setminus \{N-2, N-1\}$; $\mathbf{F}''_{N-2} := 0$. Also define the function

$$V'_i(x_i, s_i, \mathbf{F}''_{i-1}) := \mathbb{E}\left[V_i(x_i, s_i, \tilde{\mathbf{F}}'_i) \mid s_i, \mathbf{F}''_{i-1}\right], \ \forall i \in \mathcal{I} \setminus \{0\}, \ (x_i, s_i, \mathbf{F}_i) \in \mathcal{X} \times \Re_+^{N-i}. \tag{25}$$

Thus, expression (8) in stage $i \in \mathcal{I} \setminus \{N-1\}$ and state $(x_i, \mathbf{F}_i)$ can be reformulated as

$$\begin{aligned}
V_i(x_i, \mathbf{F}_i) &= \max_{a \in \mathcal{A}(x_i)} r(a, s_i) + \delta \mathbb{E}\left[V_{i+1}(x_i - a, \tilde{\mathbf{F}}_{i+1}) | \mathbf{F}'_i\right] \\
&= \max_{a \in \mathcal{A}(x_i)} r(a, s_i) + \delta \mathbb{E}\left[\underbrace{\mathbb{E}\left[V_{i+1}(x_i - a, s_{i+1}, \tilde{\mathbf{F}}'_{i+1}) | \tilde{s}_{i+1} = s_{i+1}, \mathbf{F}''_i\right]}_{V'_{i+1}(x_i - a, \tilde{s}_{i+1}, \mathbf{F}''_i)} | F_{i,i+1}\right] \\
&= \max_{a \in \mathcal{A}(x_i)} r(a, s_i) + \delta \mathbb{E}\left[V'_{i+1}(x_i - a, \tilde{s}_{i+1}, \mathbf{F}''_i) | F_{i,i+1}\right]. \qquad (26)
\end{aligned}$$

The maximization in (26) is computationally intractable. Thus, we make the following approximations to make it tractable:

1. *Information:* We replace $\mathbf{F}''_i$ in (26) with $(F_{0,i+2}, \ldots, F_{0,N-1})$, which is information known at time T_0. This effectively reduces the dimensionality of maximization (26) because the third argument of the function $V'_{i+1}(\cdot, \cdot, \cdot)$ becomes equal to the "constant" $(F_{0,i+2}, \ldots, F_{0,N-1})$, that is, a quantity known at time T_0.

2. *Value function:* We replace the function $V'_{i+1}(\cdot, \cdot, (F_{0,i+2}, \ldots, F_{0,N-1}))$ with the function $U^{ADP}_{i+1}(\cdot, \cdot)$, which we specify below.

With these approximations, the maximization in (26) becomes

$$\max_{a \in \mathcal{A}(x_i)} r(a, s_i) + \delta \mathbb{E}\left[U^{ADP}_{i+1}(x_i - a, \tilde{s}_{i+1}) \mid F_{i,i+1}\right], \tag{27}$$

which is tractable. To complete this argument, we need to specify the function $U^{ADP}_{i+1}(\cdot, \cdot)$. We do so by introducing the approximate value function

$$\begin{aligned}
u^{ADP}_i(x_i, s_i, F_{i,i+1}) &:= \max_{a \in \mathcal{A}(x_i)} r(a, s_i) + \delta \mathbb{E}\left[U^{ADP}_{i+1}(x_i - a, \tilde{s}_{i+1}) \mid F_{i,i+1}\right], \\
&\quad\ \forall i \in \mathcal{I}, \ (x_i, s_i, F_{i,i+1}) \in \mathcal{X} \times \Re^2_+, \qquad (28)
\end{aligned}$$

letting

$$U^{ADP}_N(x_N, s_N) := 0, \ \forall x_N \in \mathcal{X}, \tag{29}$$

where we exploit the fact that $s_N \equiv 0$, and finally leveraging (25) to define

$$U_i^{ADP}(x_i, s_i) := \mathbb{E}\left[u_i^{ADP}(x_i, s_i, \tilde{F}_{i,i+1}) \mid s_i, F_{0,i+1}\right], \ \forall i \in \mathcal{I}, \ (x_i, s_i) \in \mathcal{X} \times \Re_+, \tag{30}$$

with $F_{0,N} = F_{N-1,N} := 0$. Expressions (28)-(30) define our ADP model.

The ADP model can be used to generate a feasible policy for the exact model (7)-(8) through the maximization in the right-hand side of (28). We call this policy the ADP policy. We denote by $a_i^{ADP}(x_i, s_i, F_{i,i+1})$ the action of this policy in stage i and state $(x_i, s_i, F_{i,i+1})$; if set

$$\arg\max_{a \in \mathcal{A}(x_i)} r(a, s_i) + \delta \mathbb{E}\left[U_{i+1}^{ADP}(x_i - a, \tilde{s}_{i+1}) \mid F_{i,i+1}\right], \tag{31}$$

is not a singleton, we set $a_i^{ADP}(x_i, s_i, F_{i,i+1})$ equal to the element in this set with the smallest absolute value.

We denote the value of the ADP policy by $U_0^{ADP}(x_0, \mathbf{F}_0)$. This value can be estimated by Monte Carlo simulation as discussed in §5. Notice that we evaluate the value of the ADP policy in this manner because the approximate value function computed by the ADP model (28)-(30) is not necessarily equal to the value function of the ADP policy when this policy is evaluated under the full information available in model (7)-(8). In other words, when implementing the ADP policy one has access to all the relevant price information, that is, the entire forward curve at a given time; when computing this policy in the ADP model only partial information is available, that is, the current spot and prompt month futures prices at a given time and the forward curve in the initial stage.

The following result holds because the ADP policy is feasible for the exact model.

Proposition 2 (ADP policy value). *It holds that* $V_0^{ADP}(x_0, \mathbf{F}_0) \leqslant V_0(x_0, \mathbf{F}_0)$.

Remark 5 (ADP model and exact model)**.** It can be verified that the ADP model computes the exact value of storage in stage 0 when the number of maturities is equal to one, two, or three, that is, $U_0^{ADP}(x_0, \mathbf{F}_0) = V_0(x_0, \mathbf{F}_0)$ when $N = 1$, 2, 3. This does not necessarily hold for $N \geqslant 4$, when $U_0^{ADP}(x_0, \mathbf{F}_0)$ is not necessarily smaller or larger than $V_0(x_0, \mathbf{F}_0)$.

Remark 6 (Other ADP models)**.** It is possible to formulate models similar to (28)-(30) by removing fewer price-related state variables from the state definition of the exact model (7)-(8). For example, one may keep in the state both the spot price and the prompt month futures price and condition on the futures price two months in the future. However, this choice yields a model that is computationally harder to solve to optimality.

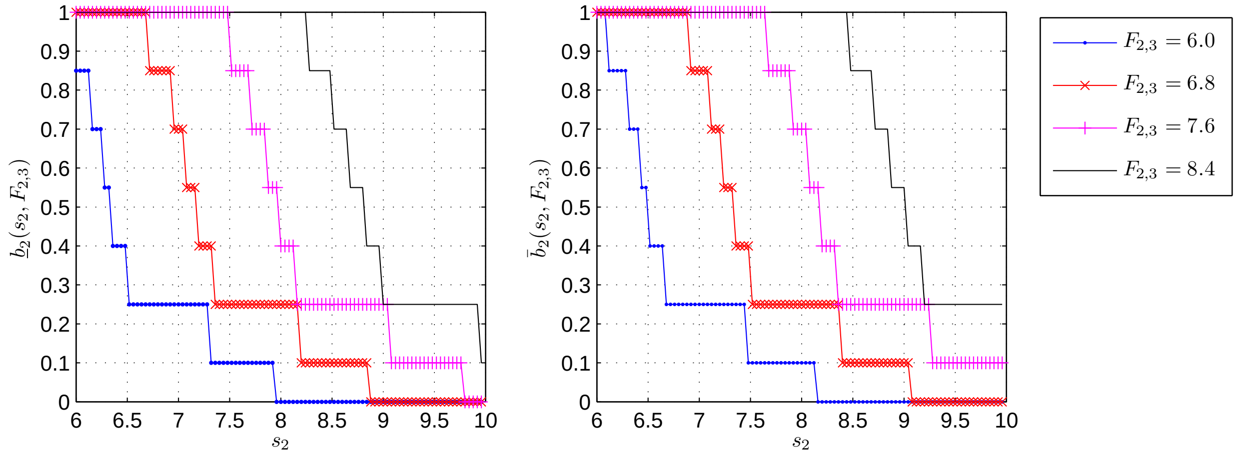

Figure 1: Illustration of the behaviors of the basestock targets with respect to the spot price and the prompt month futures price in one of the instances discussed in §6.

4.2 ADP policy Structure

We discuss our computational implementation of the ADP model in §5. This implementation takes advantage of the properties of the optimal value function and policy of the ADP model established in Theorem 1. This theorem characterizes the optimal policy of model ADP as having a stage and price-state dependent basestock target structure; that is, in each stage there exist two critical inventory levels, which depend on the available price information, such that it is optimal to buy-and-inject-up (respectively, withdraw-and-sell-down) to get as close as possible to the lower (respectively, higher) critical level from any inventory level below (respectively, above) such level. This structure makes the computation of an optimal policy more efficient. This theorem also establishes the behavior of these targets in the price related information, in one case under the assumption that the distribution of the spot price in the next stage conditional on the prompt month futures price in the current stage, denoted by $\Phi(s_{i+1}|F_{i,i+1})$, stochastically increases in the latter quantity; that is, this distribution satisfies the property that $1 - \Phi(s_{i+1}|F_{i,i+1})$ increases in $F_{i,i+1}$ for each given s_{i+1} (see, e.g., Lemma 3.9.1(b) in Topkis [38]). For example, this assumption is satisfied by the multidimensional Black model (1)-(2). Figure 1 illustrates the stated behaviors of the basestock targets for a given stage in one of the instances discussed in §6. Theorem 1 relies on the theory of supermodular functions (Topkis [38]); a real valued function f defined on a lattice $\mathcal{Z}$ is supermodular on $\mathcal{Z}$ if $f(z') + f(z'') \leqslant f(z' \wedge z'') + f(z' \vee z'')$ for all z' and $z'' \in \mathcal{Z}$, with $z' \wedge z'' := (z'_1 \wedge z''_1, \dots, z'_n \wedge z''_n)$ and $z' \vee z'' := (z'_1 \vee z''_1, \dots, z'_n \vee z''_n)$, where n is the dimension of z' and z'', that is, $z' = (z'_1, \dots, z'_n)$ and $z'' = (z''_1, \dots, z''_n)$.

Theorem 1 (Optimal policy for ADP model)**.** *(a) The function $U_i^{ADP}(x_i, s_i)$ is concave in x_i for* *each given s_i in each stage i, and the optimal policy for model ADP in each stage i features two basestock targets, $\underline{b}_i(s_i, F_{i,i+1})$, $\bar{b}_i(s_i, F_{i,i+1}) \in \mathcal{X}$, such that $\underline{b}_i(s_i, F_{i,i+1}) \leqslant \bar{b}_i(s_i, F_{i,i+1})$ and*

$$a_i^{ADP}(x_i, s_i, F_{i,i+1}) = \begin{cases} C^I \vee (x_i - \underline{b}_i(s_i, F_{i,i+1})) & \text{if } x_i \in \mathcal{X}_i^I(s_i, F_{i,i+1}) \\ 0 & \text{if } x_i \in \mathcal{X}_i^{DN}(s_i, F_{i,i+1}) \\ C^W \wedge (x_i - \bar{b}_i(s_i, F_{i,i+1})) & \text{if } x_i \in \mathcal{X}_i^W(s_i, F_{i,i+1}) \end{cases}, \tag{32}$$

where

$$\mathcal{X}_i^I(s_i, F_{i,i+1}) := [0, \underline{b}_i(s_i, F_{i,i+1})) \tag{33}$$

$$\mathcal{X}_i^{DN}(s_i, F_{i,i+1}) := [\underline{b}_i(s_i, F_{i,i+1}), \bar{b}_i(s_i, F_{i,i+1})] \tag{34}$$

$$\mathcal{X}_i^W(s_i, F_{i,i+1}) := (\bar{b}_i(s_i, F_{i,i+1}), \bar{x}]. \tag{35}$$

Moreover, (b) $U_i^{ADP}(x_i, s_i)$ is supermodular in $(x_i, s_i) \in \mathcal{X} \times \Re_+$ for all $i \in \mathcal{I}$; (c) $\underline{b}_i(s_i, F_{i,i+1})$ and $\bar{b}_i(s_i, F_{i,i+1})$ decrease in s_i for given $F_{i,i+1}$ and for all $i \in \mathcal{I}$; and (d) under the assumption that the distribution of random variable $\tilde{s}_{i+1}$ conditional on $F_{i,i+1}$ stochastically increases in $F_{i,i+1}$ for all $i \in \mathcal{I} \setminus \{N-1\}$, $\underline{b}_i(s_i, F_{i,i+1})$ and $\bar{b}_i(s_i, F_{i,i+1})$ increase in $F_{i,i+1}$ for given s_i and for all $i \in \mathcal{I} \setminus \{N-1\}$.

Proof. (a) The proof of the concavity of $U_i^{ADP}(x_i, s_i)$ in x_i for each given s_i in each stage i is omitted for brevity (it is a simple adaptation of the proof of Proposition 1 in Secomandi [33]). For the proof of the rest of this part, focus on stage i. Define the feasible inventory and action set $\mathcal{B} := \{(x, a) : x \in \mathcal{X}, a \in \mathcal{A}(x)\}$, which is both convex and a lattice. By the concavity of $U_i^{ADP}(x_i, s_i)$ in x_i for each given s_i and Lemma 2.6.2(b) in Topkis [38], $U_{i+1}^{ADP}(x_i - a, s_{i+1})$ is supermodular in $(x_i, a) \in \mathcal{B}$ for each given s_{i+1}. Thus, by Lemma 2.6.1(a) and Corollary 2.6.2 in Topkis [38], $\delta\mathbb{E}\left[U_{i+1}^{ADP}(x_i - a, \tilde{s}_{i+1}) \mid F_{i,i+1}\right]$ is supermodular in $(x_i, a) \in \mathcal{B}$ for each given $F_{i,i+1}$. Since $r(a, s_i)$ is trivially supermodular in $(x_i, a) \in \mathcal{B}$ for each given s_i, it follows from Lemma 2.6.1(b) in Topkis [38] that $r(a, s_i) + \delta\mathbb{E}\left[U_{i+1}^{ADP}(x_i - a, \tilde{s}_{i+1}) \mid F_{i,i+1}\right]$ is supermodular in $(x_i, a) \in \mathcal{B}$ for each given pair $(s_i, F_{i,i+1})$. Thus, any optimal action $a_i^{ADP}(x_i, s_i, F_{i,i+1})$ increases in x_i for each given pair $(s_i, F_{i,i+1})$ by Theorem 2.8.2 in Topkis [38]. This implies the existence of two feasible inventory levels $\underline{b}_i(s_i, F_{i,i+1}) \leqslant \bar{b}_i(s_i, F_{i,i+1})$ that depend on the pair $(s_i, F_{i,i+1})$ and partition the feasible inventory set into the stated sets $\mathcal{X}_i^I(s_i, F_{i,i+1})$, $\mathcal{X}_i^{DN}(s_i, F_{i,i+1})$, and $\mathcal{X}_i^W(s_i, F_{i,i+1})$, where it is respectively optimal to buy and inject, do nothing, and withdraw and sell. In particular, if $x_i \in \mathcal{X}_i^{DN}(s_i, F_{i,i+1})$, then the relevant part of (32) holds. Otherwise, in determining an optimal action in the extended state $(x_i, s_i, F_{i,i+1})$, relax the injection and withdrawal limits C^I and C^W on this action and let $x_{i+1} = x_i - a$. Thus, the relevant maximization is

$$\max_{x_{i+1} \in \mathcal{X}} r(x_i - x_{i+1}, s_i) + \delta\mathbb{E}\left[U_{i+1}^{ADP}(x_{i+1}, \tilde{s}_{i+1}) \mid F_{i,i+1}\right]. \tag{36}$$

Suppose that $x_i \in \mathcal{X}_i^I(s_i, F_{i,i+1})$. Then, maximization (36) can be written as

$$\max_{x_{i+1}\in\mathcal{X}} -\left(\alpha^I s_i + c^I\right) x_{i+1} + \delta\mathbb{E}\left[U_{i+1}^{ADP}(x_{i+1}, \tilde{s}_{i+1}) \mid F_{i,i+1}\right] + \left(\alpha^I s_i + c^I\right) x_i, \tag{37}$$

which shows that any optimal solution to this problem does not depend on x_i. Hence, $\underline{b}_i(s_i, F_{i,i+1})$ is an optimal solution to (37) and the pertinent part of (32) holds. Similar arguments establish the validity of (32) when $x_i \in \mathcal{X}_i^W(s_i, F_{i,i+1})$.

(b)-(c) By reverse induction on i. The function $U_N^{ADP}(x_N, s_N)$ is trivially supermodular in (x_N, s_N) on the lattice $\mathcal{X}\times\Re_+$. Let the induction hypothesis be that $U_j^{ADP}(x_j, s_j)$ is supermodular in $(x_j, s_j) \in \mathcal{X}\times\Re_+$ for all stages $j = i+1, \ldots, N-1$. Consider the determination of an optimal action in stage i in feasible extended state $(x_i, s_i, F_{i,i+1})$. The relevant optimization is

$$\max_{a\in\mathcal{A}(x_i)} r(a, s_i) + \delta\mathbb{E}\left[U_{i+1}^{ADP}(x_i - a, \tilde{s}_{i+1}) \mid F_{i,i+1}\right].$$

It follows from Lemmas 2.6.1(a)-(b) and 2.6.2(b) and Corollary 2.6.2 in Topkis [38] that the objective function of this problem is supermodular in $(x_i, a, s_i) \in \mathcal{B}\times\Re_+$ for each given $F_{i,i+1}$. Theorem 2.8.2 in Topkis [38] implies that $a_i^{ADP}(x_i, s_i, F_{i,i+1})$ increases in (x_i, s_i) for each given $F_{i,i+1}$, which implies that $\underline{b}_i(s_i, F_{i,i+1})$ and $\bar{b}_i(s_i, F_{i,i+1})$ decrease in s_i for each given $F_{i,i+1}$. Moreover, Theorem 2.7.6 in Topkis [38] implies that $u_i^{ADP}(x_i, s_i, F_{i,i+1})$ is supermodular in $(x_i, s_i) \in \mathcal{X}\times\Re_+$ for each given $F_{i,i+1}$. Corollary 2.6.2 in Topkis [38] implies that $U_i^{ADP}(x_i, s_i)$ is supermodular in $(x_i, s_i) \in \mathcal{X}\times\Re_+$. Thus, parts (b)-(c) are true for every stage by the principle of mathematical induction.

(d) Consider stage $i \in \mathcal{I}\setminus\{N-1\}$ and pick a feasible extended state $(x_i, s_i, F_{i,i+1})$. Suppose that $x_i \in \mathcal{X}_i^I(s_i, F_{i,i+1})$, so that the relevant optimization is (37). Recall the assumption that the distribution of random variable $\tilde{s}_{i+1}$ conditional on $F_{i,i+1}$ stochastically increases in $F_{i,i+1}$ for all $i \in \mathcal{I}\setminus\{N-1\}$. Then, Theorem 3.10.1 and Lemma 2.6.1(a) in Topkis [38] imply that $\delta\mathbb{E}\left[U_{i+1}^{ADP}(x_{i+1}, \tilde{s}_{i+1}) \mid F_{i,i+1}\right]$ is supermodular in $(x_{i+1}, F_{i,i+1}) \in \mathcal{X}\times\Re_+$. The term $-(\alpha^I s_i + c^I)x_{i+1}$ is trivially supermodular in $(x_{i+1}, F_{i,i+1}) \in \mathcal{X}\times\Re_+$. Thus, given s_i, any optimal x_{i+1}, including $\underline{b}_i(s_i, F_{i,i+1})$, increases in $F_{i,i+1}$ by Theorem 2.8.2 in Topkis [38]. The stated behavior of $\bar{b}_i(s_i, F_{i,i+1})$ in $F_{i,i+1}$ given s_i can be established in a similar manner. □

4.3 RADP Policy

The ADP policy is computed at time T_0, that is, the ADP model is solved only at this time. We also consider a reoptimization version of the ADP policy, that is, the RADP policy, which we obtain by re-solving the ADP model in each stage after the initial stage given the information available at that time. In other words, we reoptimize model ADP at each time T_j, $j \in \mathcal{I}\setminus\{0\}$, by using $\mathbf{F}_j$

rather than $\mathbf{F}_0$ as in (28)-(30). Specifically, we replace $F_{0,i+1}$ with $F_{j,i+1}$ in (30). The time T_0 value of the resulting RADP policy is typically different, and in fact higher, than that of the ADP policy. In other words, the RADP policy can typically take advantage of sequential reoptimization. Since the RADP policy is feasible for the exact model, its time T_0 value is a lower bound on $V_0(x_0, \mathbf{F}_0)$.

4.4 Upper Bounds

We also use the ADP model to compute an upper bound on the value of storage. Following Brown et al. [9], we define the penalty terms

$$\begin{aligned} p_i^{ADP}(x_i, a, s_{i+1}, F_{i,i+1}) \quad &:= \quad U_{i+1}^{ADP}(x_i - a, s_{i+1}) - \mathbb{E}\left[U_{i+1}^{ADP}(x_i - a, \tilde{s}_{i+1}) \mid F_{i,i+1}\right], \\ &\quad\quad \forall i \in \mathcal{I},\ (x_i, a) \in \mathcal{X} \times \mathcal{A}(x_i), \end{aligned} \tag{38}$$

which are based on the optimal value function of the ADP model computed at time T_0. We denote by P_0 a sequence of pairs of spot and prompt-month future prices for maturities 0 through $N-1$; that is, $P_0 := ((s_i, F_{i,i+1}))_{i=0}^{N-1}$. Given P_0, we solve the following DUB model:

$$U_N^{DUB}(x_N; P_0) \quad := \quad 0,\ \forall x_N \in \mathcal{X} \tag{39}$$

$$\begin{aligned} U_i^{DUB}(x_i; P_0) \quad &= \quad \max_{a \in \mathcal{A}(x_i)} r(a, s_i) - p_i^{ADP}(x_i, a, s_{i+1}, F_{i,i+1}) + \delta U_{i+1}^{DUB}(x_i - a; P_0), \\ &\quad\quad \forall i \in \mathcal{I},\ x_i \in \mathcal{X}. \end{aligned} \tag{40}$$

This is a perfect price information model whose immediate rewards are penalized according to the penalty terms (38). We obtain an upper bound on the value of storage in the manner stated in Proposition 3, which follows from Brown et al. [9].

Proposition 3 (DUB)**.** *Define*

$$V_0^{DUB}(x_0, \mathbf{F}_0) := \mathbb{E}\left[U_0^{DUB}(x_0; \tilde{P}_0) \mid \mathbf{F}_0\right]. \tag{41}$$

It holds that $V_0(x_0, \mathbf{F}_0) \leqslant V_0^{DUB}(x_0, \mathbf{F}_0)$.

The upper bound $V_0^{DUB}(x_0, \mathbf{F}_0)$ can be estimated by Monte Carlo simulation (see §5).

It is not necessary to use the DUB model to obtain an upper bound on the value of storage. For example, setting the penalty terms defined in (38) equal to zero and proceeding analogously to the computation of upper bound (41) yields the perfect information upper bound on the value of storage, labeled PIUB (this follows from Brown et al. [9]). This upper bound provides a benchmark for the performance of upper bound $V_0^{DUB}(x_0, \mathbf{F}_0)$.

5. Numerical Implementation

In this section, we describe a software implementation of the models and policies discussed in §§3-4. In particular, the I and ADP models have continuous state spaces. Thus, we introduce discretizations of the domains of the state variables and of the distribution functions of the relevant random variables.

5.1 Price Evolution and Discretization

The forward curve evolves in continuous time according to the multidimensional Black model (1)-(2). In computing, but not evaluating, the ADP policy, we use a discretized simplification of the dynamics of the forward curve. This is done as follows.

For all $i \in \mathcal{I}\setminus\{N-1\}$, we construct a 3-dimensional binomial tree (Haugh [22, §3.3]) to represent the evolution of the pair $(F(t,T_i), F(t,T_{i+1}))$, starting with the pair $(F_{0,i}, F_{0,i+1})$ and ending with a probability mass function $G_{i,i+1}(\cdot,\cdot \mid F_{0,i}, F_{0,i+1})$ for the random pair $(\tilde{s}_i, \tilde{F}_{i,i+1})$ conditional on the initial pair. We generate this tree using a given number m_i of time discretization steps. From $G_{i,i+1}(\cdot,\cdot \mid F_{0,i}, F_{0,i+1})$, for each value s_i with positive probability of occurring, we extract the probability mass function $H_{i,i+1}(\cdot \mid s_i, F_{0,i+1})$ of random variable $\tilde{F}_{i,i+1}$ conditional on the pair $(s_i, F_{0,i+1})$. For each value $F_{i,i+1}$ having a positive probability in $H_{i,i+1}(\cdot \mid s_i, F_{0,i+1})$, we use a 2-dimensional binomial tree evolving $F(t,T_{i+1})$ starting from $F_{i,i+1}$ to obtain the probability mass function $L_{i+1}(\cdot \mid F_{i,i+1})$ of the random variable $\tilde{s}_{i+1}$ given $F_{i,i+1}$. Here, we use a number m of time discretization steps. For each $i \in \mathcal{I} \setminus \{N-1\}$, the discretizations of the domains of s_i, $F_{i,i+1}$, and s_{i+1} are byproducts of how we construct these probability mass functions.

We generate the discretization of s_{N-1} separately, using the values with positive probability in the probability mass function $L_{N-1}(\cdot \mid F_{0,N-1})$, which we obtain from a 2-dimensional binomial tree for the evolution of $F(t,T_{N-1})$ starting from $F_{0,N-1}$.

5.2 Computing and Evaluating the Policies

In solving the ADP model, we use a regular discretization of the feasible inventory set $\mathcal{X}$ in each stage $i \in \mathcal{I}$. Assume that there exists a number Q with the property that it is the largest number such that the maximum inventory level, $\overline{x}$, the injection capacity, C^I, and the withdrawal capacity, C^W, are all integer multiples of Q. The regular discretization uses a number $1 + \overline{x}/Q$ of equally spaced points (see Secomandi [33, §3.3]). In our experiments discussed in §6, $Q = 0.05$ and $\overline{x} = 1$, so that the inventory discretization includes 21 points. As given in (29), we set $U_N^{ADP}(x_N, s_N)$

equal to 0 for all the discrete values of x_N (recall our convention that $s_N \equiv 0$). For each $i \in \mathcal{I}$, when computing $U_i^{ADP}(x_i, s_i)$ for discretized state (x_i, s_i), we can thus assume that the values of $U_{i+1}^{ADP}(x_{i+1}, s_{i+1})$ are known for all values of x_{i+1} and s_{i+1} in their respective discretizations.

Given a discretized state (x_i, s_i), the computation of $U_i^{ADP}(x_i, s_i)$ starts with computing the approximate value function $u_i^{ADP}(x_i, s_i, F_{i,i+1})$ for all the $F_{i,i+1}$ values having a positive probability in $H_{i,i+1}(\cdot \mid s_i, F_{0,i+1})$. The possible actions in $\mathcal{A}(x_i)$ are further restricted to actions a such that $x_i - a$ is a point of the discretization of set $\mathcal{X}$. We compute the expectation over $\tilde{s}_{i+1}$ in (28) using $L_{i+1}(\cdot \mid F_{i,i+1})$. Here, some rounding operation is typically required. Suppose that $\overline{s}_{i+1}$ has a positive probability in $L_{i+1}(\cdot \mid F_{i,i+1})$. If $\overline{s}_{i+1}$ is not in the discretization of s_{i+1} used in stage $i+1$, it is rounded to the nearest value in this discretization and its corresponding entry in the table that stores $U_{i+1}^{ADP}(\cdot, \cdot)$ is used in (28).

In the determination of an optimal action in (28), we apply the basestock policy established in Theorem 1. In particular, given the pair $(s_i, F_{i,i+1})$, we first find optimal actions for inventory levels 0 and $\overline{x}$ with the withdrawal and injection limits removed. This gives us the basestock targets corresponding to $(s_i, F_{i,i+1})$. Then, by leveraging (32), we can easily compute an optimal action and the value of the approximate value function for all inventory levels when these limits are enforced. We store these quantities in tables. Finally, we compute the expectation over $\tilde{F}_{i,i+1}$ in (30) using $H_{i,i+1}(\cdot \mid s_i, F_{0,i+1})$.

The I policy is trivial to compute. Given the forward curve $\mathbf{F}_0$ and the same regular discretization of inventory set $\mathcal{X}$ that we use to solve model ADP, we solve a deterministic dynamic program with storage and withdrawal/injection capacity constraints. The key issue for the LP and LPN policies is the valuation of spread options. We use Kirk's approximation for this purpose (Carmona and Durrleman [11]).

We apply Monte Carlo simulation to evaluate the various policies. We generate a pool of sample paths that evolve the forward curve from stage 0 to stage $N-1$ by sampling from the continuous time and space stochastic process that describes its evolution. Starting with a given feasible inventory level x_0, it is straightforward to compute the average performances of the I, LP, and LPN policies on these sample paths. (For consistency, we also evaluate the performance of the I policy by simulation.) For the ADP model, at time T_i we round the realized pair of prices $(s_i, F_{i,i+1})$, first by rounding s_i to the nearest value $\overline{s}_i$ in its discretization and then by rounding $F_{i,i+1}$ to the nearest value $\overline{F}_{i,i+1}$ that has positive probability in $H_{i,i+1}(\cdot \mid \overline{s}_i, F_{0,i+1})$. We have an inventory x_i, and the triplet $(x_i, \overline{s}_i, \overline{F}_{i,i+1})$ allows us to access its corresponding optimal action in the table that stores the optimal actions of the ADP model.

The implementation of the reoptimization versions of all the policies and of DUB is straightforward. For brevity, we do not discuss it here.

6. Numerical Results

In this section, we numerically analyze the performance of the models and policies presented in §§3-4 on a set of realistic benchmark instances. Before discussing our numerical results, we describe how we created these instances.

6.1 Instances

We generate the benchmark instances by combining real data and data reported in the energy trading literature. We use four sets of prices from the NYMEX natural gas price database, each corresponding to the information available at the close of trading on the following four days: 3/1/2006 (Spring), 6/1/2006 (Summer), 8/31/2006 (Fall), and 12/1/2006 (Winter). (The reason why we choose 8/31/2006 instead of 9/1/2006 to represent Fall is explained below.) The choice of these days allows us to generate instances with trading information typical of the four seasons of the year. For each selected trading day, we consider the Henry Hub spot price and the futures prices of the first 23 maturities (recall that Henry Hub is the delivery location of the NYMEX natural gas futures contract). This gives us four forward curves, each consisting of 24 prices. The two top panels of Figure 2 illustrate these forward curves. (Table 10 in the Online Appendix lists this data.) These panels clearly show the pronounced seasonality in the natural gas forward curve.

We use prices of NYMEX call options on natural gas futures to calibrate the market implied volatilities of the 23 futures prices on each of the four considered trading days. We do not have access to the prices of the options traded in September 2006, but we have these prices for 8/31/2006. This is why we choose this date, as opposed to 9/1/2006, to represent Fall. The two bottom panels of Figure 2 illustrate the implied volatilities on each of the four trading days. (Table 11 in the Online Appendix tabulates this data.) These panels show that these volatilities "tend" to decrease with increasing maturity, which is as expected, but also bring to light what appear to be seasonal patterns that somewhat mirror those displayed by the forward curves.

We do not have access to calendar spread options on NYMEX futures prices. Thus, we are unable to imply out correlations between futures prices for different maturities. Instead, we construct a historical correlation matrix based on the futures prices of the first 23 maturities on each trading day from 1/2/1997 to 12/14/2006. Tables 12-14 in the Online Appendix display the part of that matrix above the main diagonal.

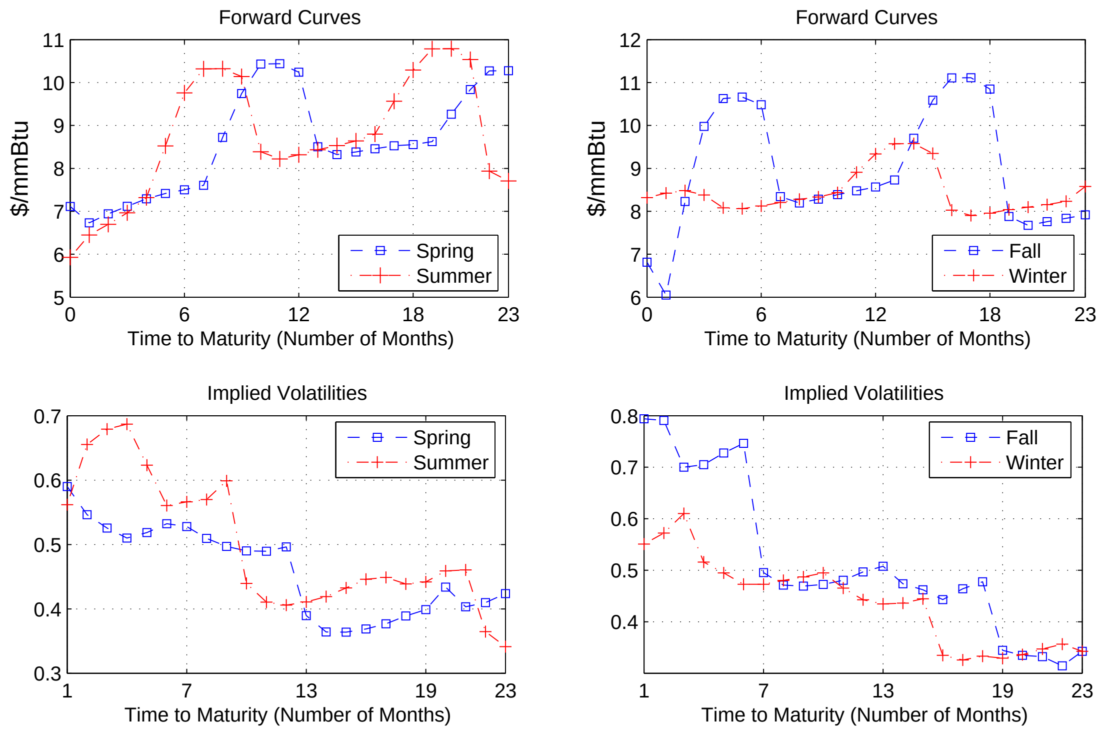

Figure 2: NYMEX natural gas forward curves (top panels) and implied volatilities (bottom panels) on 3/1/2006 (Spring), 6/1/2006 (Summer), 8/31/2006 (Fall), and 12/1/2006 (Winter).

Table 2: Absolute Values of the Capacity Pairs (mmBtu/Month).

Number	Injection	Withdrawal
1	0.15	0.30
2	0.30	0.60
3	0.45	0.90

The one-year treasury rates on the four selected dates, as reported by the U.S. Department of Treasury, are 4.74%, 5.05%, 5.01%, and 4.87%, respectively. We use them as risk-free interest rates to generate the monthly discount factors used in our experiments.

We normalize the maximum inventory $\overline{x}$ to one mmBtu. We employ the three pairs of injection and withdrawal capacities shown in Table 2. The first pair roughly reflects the capacities in Example 8.11 in Eydeland and Wolyniec [16, p. 355]. We obtain the other two by multiplying this pair by 2 and 3, respectively. Following Maragos [27], we set the injection and withdrawal costs to $0.02 and $0.01 per mmBtu, respectively, and the injection and withdrawal fuel coefficients to 1.01 and 0.99, respectively.

The distinguishing features of our benchmark instances are the number of months in the contract tenure (number of stages), the season corresponding to the initial stage (forward curve and volatilities), the withdrawal/injection limits, and the correlation matrix. The label of an instance encodes this information in the following order:

- Number of stages: 12 or 24;
- Season: one of Sp, Su, Fa, and Wi, which abbreviate Spring, Summer, Fall, and Winter, respectively;
- Injection and withdrawal pair number: 1, 2, or 3.

We consider twelve 12-stage instances, labeled 12-Sp-1 to 12-Wi-3, and twelve 24-stage instances, labeled 24-Sp-1 to 24-Wi-3. Thus, we generate a total of twenty-four instances. For brevity, the ensuing discussion focuses on the 24-stage instances, because the valuation results pertaining to the 12-stage instances are very similar to those of the 24-stage instances. However, the Cpu times required to compute and evaluate the various policies are significantly smaller on the 12-stage instances than the 24-stage instances. When reoptimization is used, the average Cpu time of each of the heuristics is approximately divided by four. Without reoptimization, the average Cpu time for the ADP based heuristics is divided approximately by two, while for the other three heuristics it is approximately divided by four.

The machine used for the computations is a 64 bits `Monarch Empro 4-Way Tower Server` with four `AMD Opteron 852 2.6GHz` processors, each with eight `DDR-400` SDRAM of 2 `GB` and running `Linux Fedora 9`. The compiler is `g++ version 4.3.0 20080428 (Red Hat 4.3.0-8)`. Results are obtained using only one processor. We solve the linear programs associated with the LP and LPN policies using the Clp linear solver of COIN-OR (www.coin-or.org).

6.2 Results

We evaluate all the policies and the two upper bounds by starting with zero initial inventory and by using $10,000$ futures price sample paths.

DUB. We compare the value of all the policies to the upper bound of Proposition 3, DUB. This bound depends on the discretization used to solve the ADP model, that is, the values of the parameters m_i and m discussed in §5. We set $m = 20$. We examine the values of DUB for values of m_i ranging from 5 to 500. Setting $m_i = 500$ yields the tightest DUB. Thus, for convenience of comparison, we report all other values as percentages of the DUB values obtained for $m_i = 500$ (Table 15 in the Online Appendix displays the values of DUB for $m_i = 500$). Table 3 reports the values of DUB for m_i ranging from 5 to 250, as well as the value of the perfect information upper

Table 3: DUB Values for Several Values of Discretization Parameter m_i and PIUB Value as Percentages of the DUB Value for $m_i = 500$.

	DUB					
	Number of Discretization Steps					
Instance	5	20	50	100	250	PIUB
24-Sp-1	108.06	112.41	105.64	102.86	100.74	147.09
24-Sp-2	125.32	116.76	107.54	103.72	100.90	164.13
24-Sp-3	137.35	119.54	108.72	104.26	101.03	175.02
24-Su-1	107.22	111.45	104.74	102.58	100.62	142.77
24-Su-2	121.20	114.36	106.08	103.24	100.85	155.12
24-Su-3	132.00	116.97	107.22	103.76	100.99	165.98
24-Fa-1	110.64	112.61	105.43	102.88	100.69	153.01
24-Fa-2	122.72	114.42	106.30	103.31	100.81	158.18
24-Fa-3	130.93	116.06	107.01	103.63	100.92	164.02
24-Wi-1	141.54	130.92	113.80	106.76	101.69	224.37
24-Wi-2	167.14	136.05	115.88	107.72	101.96	245.04
24-Wi-3	184.94	139.31	117.22	108.25	102.13	258.36

Table 4: Statistics on the Cpu Seconds Needed to Compute DUB for Several Values of Discretization Parameter m_i and PIUB.

	DUB						
	Number of Discretization Steps						
Statistic	5	20	50	100	250	500	PIUB
Maximum	11.06	15.71	24.96	41.96	123.33	382.45	4.46
Minimum	10.32	14.44	22.48	37.55	107.21	320.64	2.30
Average	10.59	14.89	23.38	39.43	115.42	353.16	3.02
Standard Deviation	0.23	0.37	0.70	1.25	4.85	21.02	1.00

bound, PIUB. It should be clear from Table 3 that increasing m_i to larger values would likely improve the resulting value of DUB only marginally. Moreover, PIUB is much weaker than DUB, even for relatively small values of m_i such as 5 and 20. Note that the average values of DUB are not always monotonically decreasing as m_i increases, although this is the usual pattern. For example, in Table 3 for the instances 24-Sp-1, 24-Su-1, and 24-Fa-1 the values obtained with $m_i = 20$ are worse than those for $m_i = 5$.

The standard errors for the values displayed in Table 3 are typically around 0.60% of the corresponding DUB values for $m_i = 500$. The average standard error ranges from 0.49% (for $m_i = 500$) to 0.71% (for $m_i = 5$) of DUBs. The standard errors for PIUBs are between 1.07% and 1.89% with an average of 1.32%.

Table 4 reports some statistics of the Cpu times needed to compute the various upper bounds. This can be done in a manageable amount of time. Obtaining tighter DUBs requires more Cpu time. For example, the small improvement in the quality of DUB obtained by increasing m_i from 100 to 500 is obtained at the cost of multiplying the Cpu time by roughly a factor of 9. Computing PIUB is much faster than computing DUB, but, as noted above, the latter bound is significantly better than the former.

We will see later in this section that DUB for $m_i = 500$ is close to the optimal value of storage, as we are able to produce a feasible policy whose value is within 1% to 6% of this bound on most instances. Thus, the valuation performance of each policy as a percentage of DUB for $m_i = 500$ can be taken as a fairly accurate estimate of the fraction of the value of storage that can be captured by each policy. All subsequent performance tables use this reporting format. Moreover, the standard errors discussed below are percentages of the relevant DUB values for $m_i = 500$.

ADP policy and discretization. Before comparing the performances of the different policies, we emphasize that increasing the value of m_i also improves the performance of the ADP policy. This is shown in Table 5, which reports the performance of this policy for values of m_i ranging from 5 to 500. Improved valuations require larger Cpu times, as shown in Table 6. The valuation results for m_i equal to 100 and 500 are fairly close, but their associated Cpu times are in a ratio of 35. In the following comparative analysis we use the ADP policy obtained for $m_i = 500$.

The standard errors for the values reported in Table 5 are typically around 1%, with a minimum of 0.86% and a maximum of 1.63%. The average standard errors range from 1.10% (for $m_i = 5$) to 1.18% (for $m_i = 500$). The largest standard errors occur on the Winter instances with a spread from 1.24% to 1.63%.

No reoptimization. We now compare the performance of the ADP, I, LP, and LPN policies,

Table 5: Effect of Discretization Parameter m_i on the Quality of the ADP Policy without Reoptimization (Percent of the DUB Value for $m_i = 500$).

	Number of Discretization Steps					
Instance	5	20	50	100	250	500
24-Sp-1	82.83	88.52	91.37	92.49	92.89	93.11
24-Sp-2	74.38	84.71	90.09	92.42	93.48	94.01
24-Sp-3	70.66	83.06	89.76	93.13	94.59	95.12
24-Su-1	83.19	89.58	92.29	93.31	93.74	93.88
24-Su-2	77.64	87.53	92.22	94.48	95.48	96.14
24-Su-3	73.80	86.06	91.92	94.74	96.39	96.88
24-Fa-1	80.82	88.86	92.18	93.58	94.29	94.64
24-Fa-2	76.08	87.28	92.22	94.39	95.82	96.45
24-Fa-3	73.85	86.58	92.31	94.91	96.56	97.37
24-Wi-1	41.81	63.45	72.40	76.55	78.89	80.17
24-Wi-2	36.76	62.84	75.10	80.89	83.96	86.15
24-Wi-3	33.35	62.56	76.71	83.32	86.69	89.10

Table 6: Statistics on the Cpu Seconds Needed to Compute the ADP Policy for Several Values of Discretization Parameter m_i.

	Number of Discretization Steps					
Statistic	5	20	50	100	250	500
Maximum	0.44	0.57	1.38	4.27	28.35	146.23
Minimum	0.42	0.56	1.33	4.07	27.21	135.29
Average	0.43	0.56	1.35	4.15	27.68	140.01
Standard Deviation	0.003	0.003	0.019	0.069	0.383	4.536

Table 7: Statistics on the Cpu Seconds Needed to Compute the Four Policies without Reoptimization.

	Policy			
Statistic	ADP	I	LP	LPN
Maximum	146.23	0.53	0.64	0.69
Minimum	135.29	0.40	0.46	0.47
Average	140.01	0.46	0.54	0.56
Standard Deviation	4.54	0.05	0.07	0.08

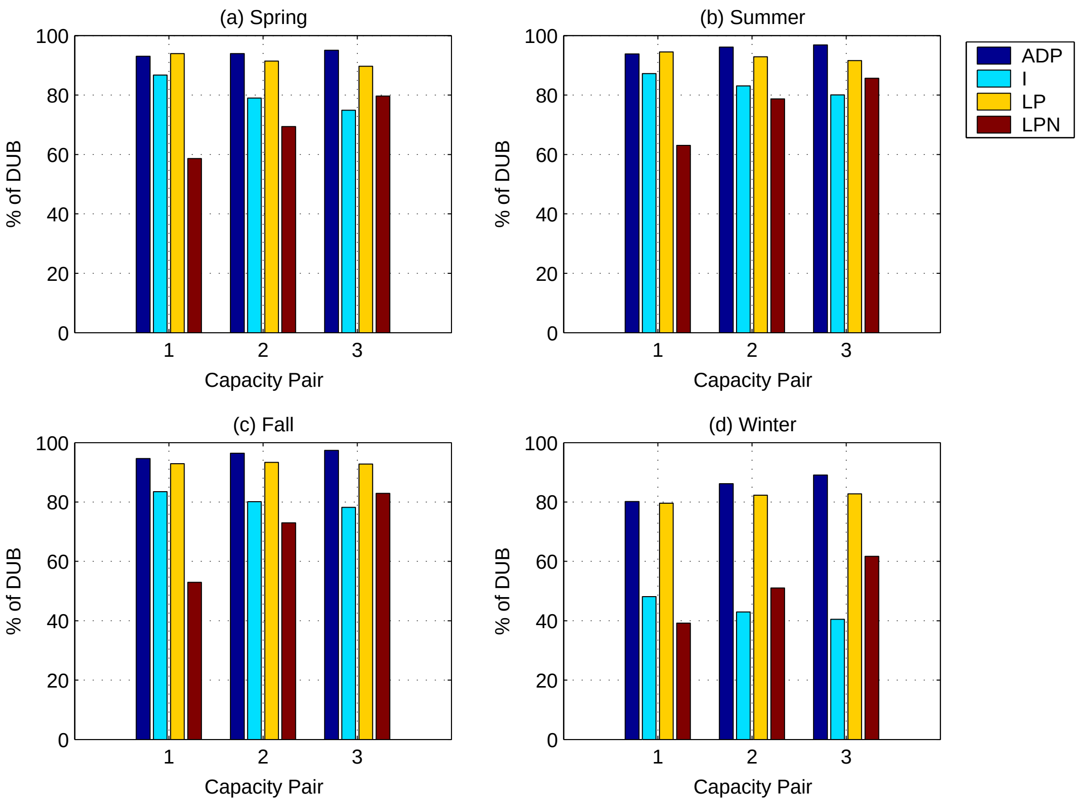

Figure 3: Valuation Performance of the Four Policies without Reoptimization (Percent of the DUB Value for $m_i = 500$).

that is, the policies that do not use reoptimization. Recall that the I policy computes the intrinsic value of storage, that is, that part of the storage value that can be attributed to the seasonality in the natural gas forward curve, rather than its volatility. Thus, its valuation performance should be interpreted accordingly. In other words, even if we include the I policy in our comparisons, we do so with the understanding that this policy is likely to yield lower valuations than the others.

Figure 3 reports the valuation performance of the four policies on the 24-stage instances (the detailed values are presented in Table 16 in the Online Appendix). Overall, the ADP and LP policies perform consistently better than the I and LPN policies. The performance of the latter two may reach as low as 40% of the DUB value. The ADP policy performs better than the other policies on most of the instances except on the instances 24-Sp-1 and 24-Su-1, where the LP policy outperforms it.

The standard errors on the values of the policies without reoptimization are typically around 1% with a minimum of 0.66% and a maximum of 1.63%. We obtain average standard errors of 1.18%, 1.16%, 1.12% and 0.84% for the ADP, I, LP and LPN policies, respectively.

To discuss how the valuation performance of a policy depends on the injection/withdrawal capacities, we define the *range* of valuation performances for a policy to be the difference between its minimum and maximum valuation performance figures on each of the three instances that differ only in their injection/withdrawal capacities. For example, the range of the ADP policy on instances 24-Sp-1, 24-Sp-2, and 24-Sp-3 is $(95.12-93.11)\% = 2.01\%$. The LP policy is the least sensitive with a rough average range of 3%, whereas the ranges of the ADP, I, and LPN policies are about 4%, 8%, and 24%, respectively. It appears that the ADP and LPN policies are able to capture a larger share of the value of storage for instances with higher injection/withdrawal capacities, while the intrinsic value becomes smaller when the injection/withdrawal capacities increase. In particular, the values obtained by the LP policy do not show a monotone pattern as the injection/withdrawal capacities vary.

The statistics on the Cpu times needed to compute and evaluate the different policies are reported in Table 7. The ADP policy requires on average much more Cpu time than the other policies, but, as pointed out in comments on Table 6, this number can be significantly reduced by using a coarser discretization without significantly affecting the valuation performance of this policy. The fastest policy to compute and evaluate is I, but the computational requirements of both the LP and LPN policies are very small.

Reoptimization. Figure 4 reports the valuation performance of the RADP, RI, RLP, and RLPN policies (the detailed values are provided in Table 17 in the Online Appendix). Notice that

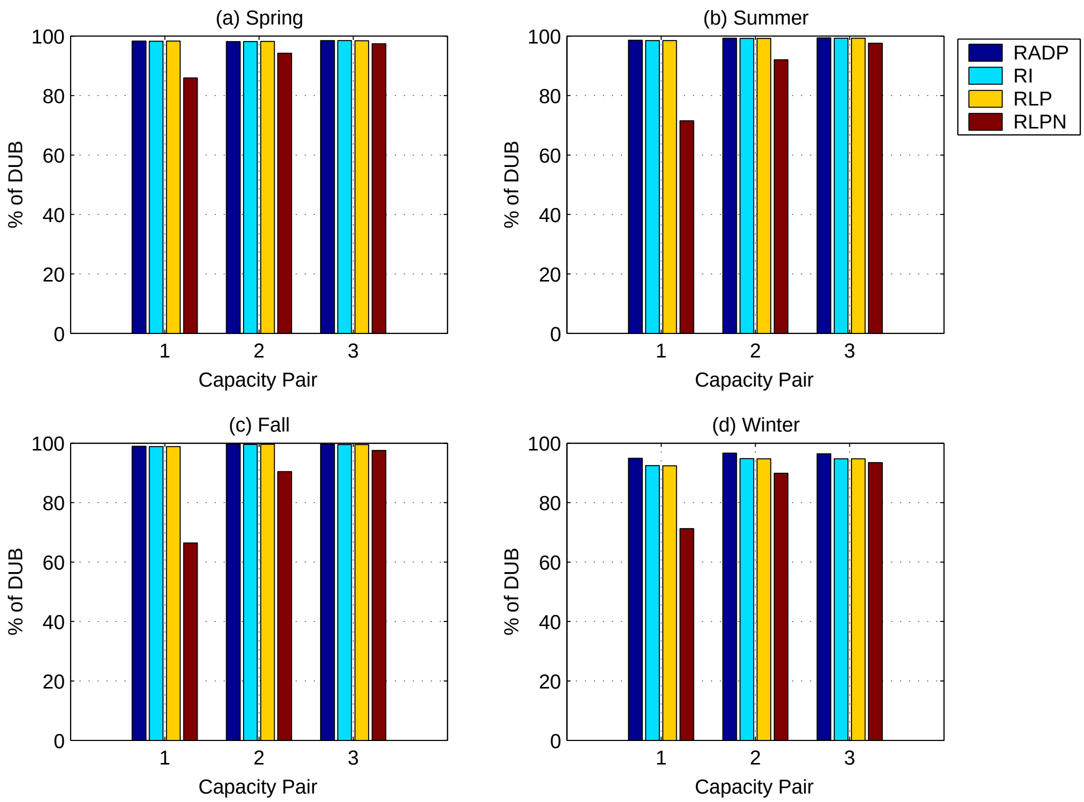

Figure 4: Valuation Performance of the Four Policies with Reoptimization (Percent of the DUB Value for $m_i = 500$).

Table 8: Statistics on the Cpu Seconds Needed to Compute the Four Policies with Reoptimization.

	Policy			
Statistic	RADP	RI	RLP	RLPN
Maximum	634.22	24.81	296.12	249.21
Minimum	570.66	23.20	262.88	233.84
Average	595.73	23.90	278.74	240.94
Standard Deviation	20.85	0.47	12.26	4.58

we compute the RADP policy using $m_i = 5$ and $m = 5$ to keep its computational requirement at a manageable level. The RADP policy captures at least 96% of the value of storage on all the instances, except on 24-Wi-1 for which this figure is 94.91%. Moreover, the RADP policy performs mostly better than the other policies, with the exception of 24-Sp-2, where it is marginally outperformed by RLP. The performances of RADP, RI and RLP are all very insensitive to changes in injection/withdrawal capacities with average ranges less than 1%, whereas the average range of the RLPN policy is a whooping 23%.

Similar to the case without reoptimization, the standard errors of the policies with reoptimization are around 1%. We obtain average standard errors of 1.19%, 1.17%, 1.17% and 1.16% for the RADP, RI, RLP, and RLPN policies, respectively.

It is noteworthy that the gap between DUB and the value of the RADP policy is substantially higher in the Winter instances than in the instances associated with the other seasons (see panel (d) of Figure 4). This may be due to DUB being looser or to the performance of the RADP policy being worse on the Winter instances compared to the other instances. Thus, we also evaluate the RADP policy with m_i set equal to 10 and 20. Unfortunately, the value of the RADP policy in these cases does not appreciably change relative to the case of $m_i = 5$, and we are unable to shed any light on this issue.

The statistics on the Cpu times needed to compute the four reoptimization policies are reported in Table 8. The RADP policy needs on average substantially more Cpu time than the other policies, whose computational requirements, however, increase markedly relative to the no-reoptimization case. The policy that can be evaluated the fastest is RI; this takes roughly one tenth of the time needed to evaluate the RLP and RLPN policies, which in turn take half the time required to evaluate the RADP policy.

No simulation. We now discuss how the "raw" values of the four optimization models used to compute the various policies compare to DUB. That is, we focus on the values generated by optimization of the four models at time T_0 without simulating their corresponding policies. In contrast, all the valuation results discussed so far entail simulation to evaluate the quality of the policy obtained from each model. Figure 5 reports these raw values (the detailed values are provided in Table 18 in the Online Appendix). The ADP model underestimates the value of storage for this set of experiments. This underestimation is much more marked for models I and LP. Unlike these three models, model LPN may overestimate or underestimate the value of storage. Recall from §3 that the raw values of the I and LP models are lower bounds on the optimal value of storage, while those of the ADP and LPN models may not be. These results also imply that it seems difficult for

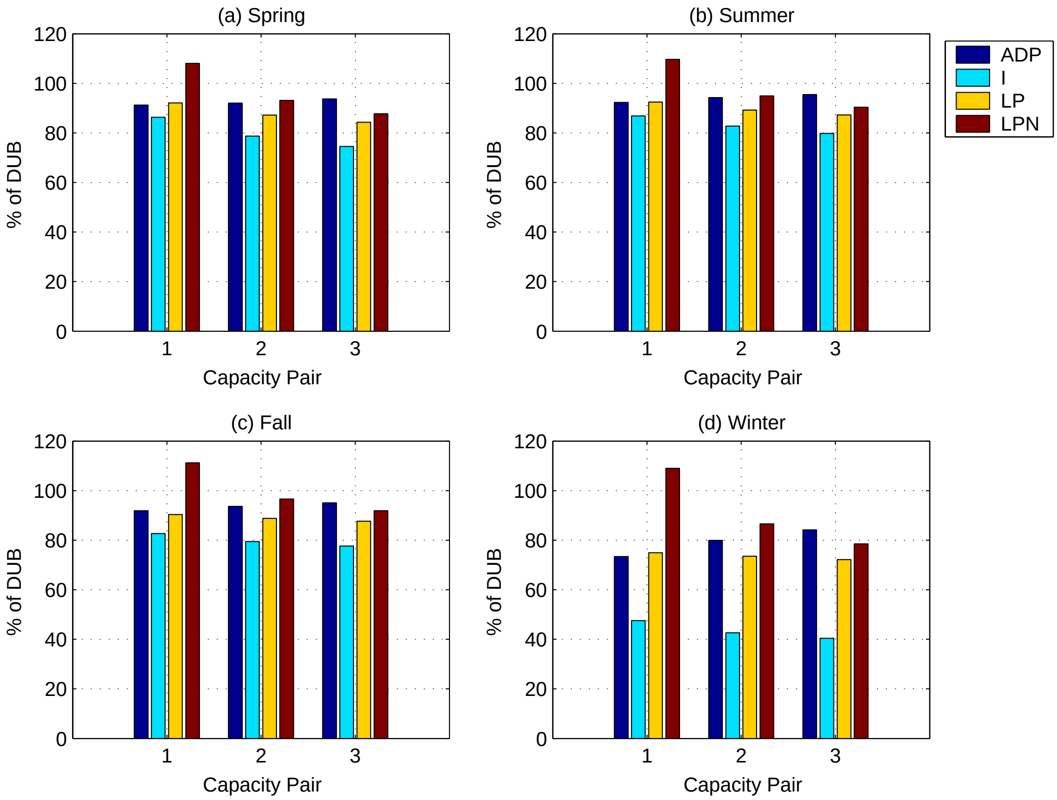

Figure 5: "Raw" Values of the Four Models without Simulation (Percent of the DUB Value for $m_i = 500$).

any of these models to consistently produce near optimal storage valuations without reoptimization and simulation.

Table 9 displays some statistics of the Cpu times needed to solve the four models: solving the I, LP, and LPN models is remarkably fast, whereas solving the ADP model requires considerably more time.

Summary. Our computational results bring to light the value of combining reoptimization and Monte Carlo simulation for natural gas storage valuation. This approach allows one to compute a near optimal policy in a rather simple and relatively fast fashion, for example, by sequentially

Table 9: Statistics on the Cpu Seconds Needed to Compute the "Raw" Values of the Four Models.

	Model			
Statistic	ADP	I	LP	LPN
Maximum	145.5830	0.0010	0.0050	0.0030
Minimum	134.6100	0.0001	0.0030	0.0010
Average	139.4726	0.0005	0.0038	0.0021
Standard Deviation	4.4848	0.0005	0.0007	0.0005

reoptimizing model I, which is a deterministic and dynamic model that can be optimized very efficiently. Reoptimizing model ADP, which is a stochastic and dynamic model, yields a slightly better policy at the expense of significantly higher Cpu requirements (recall that model I is a special case of model ADP, that is, I is the deterministic version of ADP). Reoptimizing model LP, which is a stochastic and static model, also generates a very good policy and is faster than reoptimizing model ADP. Moreover, although beneficial, reoptimization applied to model LPN does not yield a near optimal policy. Hence, the choice of which model to sequentially reoptimize is important in order to obtain near optimal valuation of natural gas storage contracts by sequential reoptimization coupled with Monte Carlo simulation.

7. Conclusions

The valuation of the real option to store natural gas is an important problem in practice. Exact valuation of this real option using the multidimensional models of the evolution of the natural gas forward curve that seem to be used in practice is an intractable problem. Thus, practitioners typically value storage using heuristics. In this paper, we develop a novel approach to benchmark a set of such heuristics. Unlike these heuristics or methods that are available in the extant literature, our approach yields both lower and upper bounds on the value of storage using the multidimensional representation of the dynamics of the natural gas forward curve that seem to be used in practice.

These bounds allow us to assess the effectiveness of the stated heuristics on a set of realistic instances. Our upper bound appears to be fairly tight. We find that the practice-based policies that we analyze are very fast to compute but also significantly suboptimal, and are dominated by our policy. When employed in a reoptimization fashion within Monte Carlo simulation, the valuation performances of all but one of these policies become nearly optimal. The price to be paid for this improvement is a significantly higher computational burden. Our own policy with reoptimization is very competitive with these reoptimization-based policies, but requires more time to compute. Overall, sequential reoptimization within Monte Carlo simulation of a deterministic model that computes the intrinsic value of storage strikes the best balance between valuation quality and computational requirements.

These results have immediate relevance for natural gas storage traders. Although not discussed in this paper, our findings with reoptimization remain substantially similar when we (artificially) remove the seasonality from the natural gar forward curves that we employ in our instances. This suggests that our results in the reoptimization case have potential relevance for traders involved in the valuation of the real option to store other commodities, whose forward curves do not exhibit the marked seasonality of the natural gas forward curve.

Our results imply that improving the computational efficiency of valuation algorithms that use periodic reoptimizations of optimization models embedded within Monte Carlo simulation is an interesting area for further research and applications in the valuation of commodity storage real options. Moreover, as explained in §2, our focus in this paper is on the value of the "monthly-volatility" component of a storage contract; that is, when trading decisions are made on a monthly basis. Extending our work to the case when trading decisions are made more frequently, such as in Carmona and Ludkovski [12] and Boogert and de Jong [7], would also be of interest.

Acknowledgments

We thank Alan Scheller-Wolf for his initial suggestion to explore information reduction as a way to make the full information stochastic dynamic program tractable, and Duane Seppi for useful discussions regarding the framing of our research. The useful and constructive comments of an associate editor and two referees led to an improved version of this paper and are gratefully acknowledged. This work was supported by a grant from the Center for Analytical Research in Technology at the Carnegie Mellon Tepper School of Business. The second author was also supported by ONR grant N00014-03-1-0188.

References

[1] D. Adelman. Math programming approaches to approximate dynamic programming. Tutorial, INFORMS Annual Meeting, Pittsburgh, PA, USA, 2006.

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

[3] C. Barrera-Esteve, F. Bergeret, C. Dossal, E. Gobet, A. Meziou, R. Munos, and D. Reboul-Salze. Numerical methods for the pricing of swing options: A stochastic control approach. *Methodology And Computing In Applied Probability*, 8:517–540, 2006.

[4] D. P. Bertsekas. *Dynamic Programming and Optimal Control – Volume I.* Athena Scientific, Belmont, MA, USA, third edition, 2005.

[5] D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming.* Athena Scientific, Belmont, MA, USA, 1996.

[6] F. Black. The pricing of commodity contracts. *Journal of Financial Economics*, 3:167–179, 1976.

[7] A. Boogert and C. de Jong. Gas storage valuation using a Monte Carlo method. *Journal of Derivatives*, 15:81–98, 2008.

[8] A. Brace, D. Gatarek, and M. Musiela. The market model of interest rate dynamics. *Mathematical Finance*, 7:127–147, 1997.

[9] D. B. Brown, J. E. Smith, and P. Sung. Information relaxation and duality in stochastic dynamic programs. Working Paper, Fuqua School of Business, Durham, NC, USA, 2008.

[10] J. W. Byers. Commodity storage valuation: A linear optimization based on traded instruments. *Energy Economics*, 28:275–287, 2006.

[11] R. Carmona and V. Durrleman. Pricing and hedging spread options. *Society for Industrial and Applied Mathematics Review*, 45:627–685, 2003.

[12] R. Carmona and M. Ludkovski. Valuation of energy storage: An optimal switching approach. Working Paper, Princeton University, Princeton, NJ, USA, 2007.

[13] Z. Chen and P. A. Forsyth. A semi-Lagrangian approach for natural gas storage valuation and optimal operation. *SIAM Journal on Scientific Computing*, 30:339–368, 2007.

[14] L. Clewlow and C. Strickland. *Energy Derivatives: Pricing and Risk Management*. Lacima Publications, London, UK, 2000.

[15] M. Davis and I. Karatzas. A deterministic approach to optimal stopping, with applications. In P. Kelly, editor, *Probability, Statistics and Optimization: A Tribute to Peter Whittle*, pages 255–466. John Wiley & Sons, Chichester, UK, 1994.

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

[17] FEA. @ENERGY/Storage – Pricing, hedging and optimization of gas storage contracts and facilities. http://www.fea.com/products/pdf/fea_energy_storage.pdf, 2007.

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

[19] C. S. Ghiuvea, J. P. Lehoczky, and D. Seppi. Pricing swing and other generalized American options. Working Paper, Carnegie Mellon University, Pittsburgh, PA, USA, 2003.

[20] J. Gray and P. Khandelwal. Realistic gas storage models II: Trading strategies. *Commodities Now*, pages 1–5, September 2004.

[21] J. Gray and P. Khandelwal. Towards a realistic gas storage model. *Commodities Now*, pages 1–4, June 2004.

[22] E. G. Haugh. *The Complete Guide to Option Pricing Formulas.* McGraw-Hill, New York, NY, USA, 1998.

[23] M. B. Haugh and L. Kogan. Pricing American options: A duality approach. *Operations Research*, 52:258–270, 2004.

[24] P. Jaillet, E. I. Ronn, and S. Tompaidis. Valuation of commodity-based swing options. *Management Science*, 50:909–921, 2004.

[25] D. P. Kennedy. The term structure of interest rates as a Gaussian random field. *Mathematical Finance*, 4:247–258, 1994.

[26] KYOS. MayStore-Gas storage valuation – Optimal trading decisions in forward and spot markets. http://www.kyos.com/upload/Modeling/MayStore_Brochure_2009.pdf, 2009.

[27] 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.

[28] E. L. Porteus. *Foundations of Stochastic Inventory Theory.* Stanford Business Books, Stanford, CA, USA, 2002.

[29] W. B. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality.* Wiley-Interscience, Hoboken, NJ, USA, 2007.

[30] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* John Wiley & Sons, New York, NY, USA, 1994.

[31] L. C. G. Rogers. Monte Carlo valuation of American options. *Mathematical Finance*, 12:271–286, 2002.

[32] N. Secomandi. An analysis of the control-algorithm re-solving issue in inventory and revenue management. *Manufacturing & Service Operations Management*, 10:468–483, 2008.

[33] N. Secomandi. Optimal commodity trading with a capacitated storage asset. *Management Science*, Forthcoming, 2009.

[34] D. J. Seppi. Risk-neutral stochastic processes for commodity derivative pricing: An introduction and survey. In E. Ronn, editor, *Real Options and Energy Management Using Options Methodology to Enhance Capital Budgeting Decisions*, pages 3–60. Risk Publications, London, UK, 2002.

[35] J. E. Smith and K. F. McCardle. Options in the real world: Lessons learned in evaluating oil and gas investments. *Operations Research*, 47:1–15, 1999.

[36] N. L. Stokey and R. E. Lucas. *Recursive Methods in Economic Dynamics.* Harvard University Press, Cambridge, MA, USA, 1989. With E. C. Prescott.

[37] M. Thompson, M. Davison, and H. Rasmussen. Natural gas storage valuation and optimization: A real options application. *Naval Research Logistics*, 56:226–238, 2009.

[38] D. M. Topkis. *Supermodularity and Complementarity.* Princeton University Press, Princeton, NJ, USA, 1998.

[39] L. Trigeorgis. *Real Options: Managerial Flexibility and Strategy in Resource Allocation.* The MIT Press, Cambridge, MA, USA, 1996.

[40] T. Weston. Applying stochastic dynamic programming to the valuation of gas storage and generation assets. In E. Ronn, editor, *Real Options and Energy Management Using Options Methodology to Enhance Capital Budgeting Decisions*, pages 183–216. Risk Publications, London, UK, 2002.

[41] P. H. Zipkin. *Foundations of Inventory Management.* McGraw-Hill, Boston, MA, USA, 2000.

Online Appendix

The tables reported in this Online Appendix include the details of the market data used in the numerical experiments and the valuation results. In particular:

- Tables 10-11: The two top and bottom panels of Figure 2, respectively;
- Tables 12-14 display the correlation matrix;
- Table 15 reports the DUB values for discretization parameter $m_i = 500$;
- Tables 16-18 are associated with Figures 3-5, respectively.

Table 10: Forward Curves ($/mmBtu).

Months to Maturity	Spring	Summer	Fall	Winter
0	7.112	5.925	6.816	8.318
1	6.733	6.448	6.048	8.422
2	6.940	6.698	8.228	8.480
3	7.115	6.963	9.978	8.380
4	7.290	7.318	10.628	8.080
5	7.417	8.523	10.663	8.062
6	7.502	9.758	10.483	8.125
7	7.607	10.318	8.343	8.207
8	8.722	10.323	8.193	8.278
9	9.742	10.138	8.283	8.333
10	10.432	8.388	8.388	8.433
11	10.437	8.218	8.478	8.908
12	10.242	8.315	8.568	9.338
13	8.502	8.435	8.728	9.573
14	8.322	8.530	9.703	9.578
15	8.382	8.640	10.588	9.348
16	8.457	8.800	11.108	8.028
17	8.527	9.565	11.113	7.903
18	8.552	10.290	10.848	7.958
19	8.622	10.785	7.878	8.038
20	9.262	10.790	7.673	8.098
21	9.837	10.535	7.758	8.153
22	10.272	7.935	7.838	8.233
23	10.277	7.705	7.918	8.578

Table 11: Implied Volatilities.

Months to Maturity	Spring	Summer	Fall	Winter
1	0.59	0.56	0.79	0.55
2	0.55	0.66	0.79	0.57
3	0.53	0.68	0.70	0.61
4	0.51	0.69	0.70	0.52
5	0.52	0.62	0.73	0.49
6	0.53	0.56	0.75	0.47
7	0.53	0.57	0.50	0.47
8	0.51	0.57	0.47	0.48
9	0.50	0.60	0.47	0.49
10	0.49	0.44	0.47	0.49
11	0.49	0.41	0.48	0.47
12	0.50	0.41	0.50	0.44
13	0.39	0.41	0.51	0.43
14	0.36	0.42	0.47	0.44
15	0.36	0.43	0.46	0.44
16	0.37	0.45	0.44	0.33
17	0.38	0.45	0.46	0.33
18	0.39	0.44	0.48	0.33
19	0.40	0.44	0.34	0.33
20	0.43	0.46	0.33	0.34
21	0.40	0.46	0.33	0.35
22	0.41	0.36	0.31	0.36
23	0.42	0.34	0.34	0.34

Table 12: Correlation Matrix: Submatrix 1.

	Maturity											
Maturity	1	2	3	4	5	6	7	8	9	10	11	12
1	1	0.958	0.933	0.910	0.887	0.871	0.858	0.843	0.832	0.819	0.806	0.795
2		1	0.983	0.959	0.935	0.919	0.906	0.889	0.876	0.860	0.846	0.833
3			1	0.982	0.962	0.946	0.934	0.918	0.905	0.890	0.876	0.863
4				1	0.990	0.975	0.961	0.947	0.935	0.921	0.907	0.895
5					1	0.991	0.978	0.965	0.955	0.942	0.929	0.918
6						1	0.992	0.981	0.971	0.960	0.948	0.936
7							1	0.992	0.985	0.974	0.963	0.952
8								1	0.996	0.986	0.976	0.966
9									1	0.994	0.984	0.975
10										1	0.994	0.986
11											1	0.995
12												1

Table 13: Correlation Matrix: Submatrix 2.

	Maturity										
Maturity	13	14	15	16	17	18	19	20	21	22	23
1	0.784	0.768	0.749	0.727	0.708	0.696	0.681	0.668	0.654	0.643	0.634
2	0.822	0.809	0.790	0.768	0.750	0.737	0.720	0.706	0.693	0.680	0.671
3	0.852	0.841	0.825	0.804	0.787	0.773	0.754	0.739	0.724	0.712	0.703
4	0.884	0.873	0.861	0.847	0.831	0.817	0.796	0.780	0.767	0.755	0.746
5	0.907	0.896	0.884	0.872	0.859	0.846	0.825	0.809	0.796	0.784	0.775
6	0.926	0.915	0.904	0.891	0.880	0.869	0.848	0.833	0.820	0.807	0.798
7	0.942	0.930	0.916	0.903	0.891	0.881	0.870	0.856	0.843	0.830	0.820
8	0.956	0.944	0.931	0.918	0.907	0.897	0.887	0.876	0.864	0.851	0.841
9	0.967	0.955	0.942	0.929	0.919	0.909	0.899	0.889	0.879	0.867	0.858
10	0.978	0.967	0.955	0.943	0.932	0.923	0.913	0.903	0.894	0.885	0.876
11	0.988	0.979	0.967	0.955	0.946	0.936	0.928	0.918	0.909	0.901	0.893
12	0.995	0.986	0.976	0.964	0.954	0.946	0.937	0.928	0.919	0.911	0.904

Table 14: Correlation Matrix: Submatrix 3.

	Maturity										
Maturity	13	14	15	16	17	18	19	20	21	22	23
13	1	0.994	0.984	0.973	0.964	0.955	0.947	0.939	0.930	0.922	0.915
14		1	0.993	0.984	0.975	0.967	0.955	0.946	0.937	0.930	0.923
15			1	0.994	0.986	0.978	0.962	0.953	0.945	0.938	0.931
16				1	0.995	0.987	0.971	0.963	0.955	0.947	0.939
17					1	0.995	0.980	0.972	0.965	0.957	0.949
18						1	0.987	0.980	0.972	0.964	0.957
19							1	0.995	0.988	0.981	0.973
20								1	0.996	0.988	0.980
21									1	0.994	0.987
22										1	0.994
23											1

Table 15: DUB Values ($/mmBtu) for Discretization Parameter $m_i = 500$.

Instances	DUB
24-Sp-1	4.25
24-Sp-2	5.34
24-Sp-3	5.79
24-Su-1	4.75
24-Su-2	6.33
24-Su-3	6.84
24-Fa-1	4.19
24-Fa-2	6.45
24-Fa-3	7.56
24-Wi-1	1.85
24-Wi-2	2.55
24-Wi-3	2.92

Table 16: Valuation Performance of the Four Policies without Reoptimization (Percent of the DUB Value for $m_i = 500$).

	Policy			
Instance	ADP	I	LP	LPN
24-Sp-1	93.11	86.86	94.04	58.79
24-Sp-2	94.01	79.16	91.52	71.28
24-Sp-3	95.12	75.06	89.80	79.71
24-Su-1	93.88	87.26	94.55	63.06
24-Su-2	96.14	83.10	92.93	78.72
24-Su-3	96.88	80.07	91.62	85.69
24-Fa-1	94.64	83.45	92.87	52.95
24-Fa-2	96.45	80.12	93.37	72.98
24-Fa-3	97.37	78.18	92.76	82.89
24-Wi-1	80.17	48.14	79.62	39.38
24-Wi-2	86.15	42.96	82.29	51.15
24-Wi-3	89.10	40.48	82.77	61.74

Table 17: Valuation Performance of the Four Policies with Reoptimization (Percent of the DUB Value for $m_i = 500$).

	Policy			
Instance	RADP	RI	RLP	RLPN
24-Sp-1	98.37	98.29	98.37	85.93
24-Sp-2	98.20	98.20	98.23	94.25
24-Sp-3	98.52	98.51	98.44	97.47
24-Su-1	98.61	98.51	98.52	71.51
24-Su-2	99.31	99.24	99.26	92.04
24-Su-3	99.39	99.32	99.30	97.61
24-Fa-1	98.96	98.84	98.84	66.45
24-Fa-2	99.67	99.59	99.61	90.41
24-Fa-3	99.63	99.51	99.53	97.54
24-Wi-1	94.91	92.45	92.39	71.27
24-Wi-2	96.69	94.80	94.76	89.87
24-Wi-3	96.44	94.73	94.73	93.46

Table 18: "Raw" Values of the Four Models without Simulation (Percent of the DUB Value for $m_i = 500$).

Instance	Model ADP	I	LP	LPN
24-Sp-1	91.31	86.45	92.13	107.95
24-Sp-2	92.10	78.86	87.26	93.14
24-Sp-3	93.73	74.72	84.32	87.74
24-Su-1	92.31	86.86	92.45	109.67
24-Su-2	94.26	82.74	89.22	94.94
24-Su-3	95.45	79.79	87.24	90.33
24-Fa-1	91.88	82.71	90.32	111.23
24-Fa-2	93.63	79.48	88.81	96.59
24-Fa-3	95.05	77.64	87.68	91.90
24-Wi-1	73.38	47.54	74.94	108.98
24-Wi-2	79.90	42.62	73.53	86.58
24-Wi-3	84.19	40.41	72.18	78.49