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

Guoming Lai, François Margot, Nicola Secomandi
Tepper School of Business, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213-3890, USA, {guomingl, fmargot, ns7}@andrew.cmu.edu

Tepper Working Paper 2008-E38
September 2008

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 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; our lower bound is slower to compute than these heuristics but 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 [31], Puterman [26], Bertsekas [4]) is the natural approach to this valuation problem (Weston [35]), 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 [13, Chapter 8], Seppi [29], Eydeland and Wolyniec [14, Chapter 4], Geman [15, 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 [14, 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 [23, pp. 449-453], Eydeland and Wolyniec [14, pp. 351-367], Gray and Khandelwal [18, 17]), 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 2000 [13, Chapter 8], Seppi [29], Eydeland and Wolyniec [14, Chapter 5], Geman [15, 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.

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 [18] 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, 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 [18] 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

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

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 state-dependent basestock targets, to improve the efficiency of computing it.

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 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 and LPN heuristics are also very fast and capture significantly more value than the intrinsic value, but their suboptimality is large when compared to DUB, which is fairly tight and much tighter than PIUB. Our ADP lower bound exhibits less suboptimality but higher computational requirement than these two heuristics.

Moreover, reoptimization improves the valuation performance of all the policies. 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 the same. (For brevity, these results are not included in this paper, but are available upon request.) This suggests that our findings 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 [15]).

Our work is related to the literature on the valuation of commodity and energy real options (Trigeorgis [34], Smith and McCardle [30], Geman [15]). We are not the first ones to study the valuation of the real option to store natural gas. Thompson et al. [32] and Chen and Forsyth [12] 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. [16] and Barrera-Esteve et al. [3] propose to value natural gas storage as an extended swing option (Jaillet et al. [21]). 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 [25]). Carmona and Ludkovski [11] and Boogert and de Jong [7] apply such ADP methods to obtain a statistical approximation of the optimal value function of the storage valuation problem formulated as an MDP. Unlike these authors, although we use Monte Carlo simulation for policy evaluation, we do not employ statistical functional approximation to compute our ADP and RADP policies. Moreover, even if the models of Carmona and Ludkovski [11] 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 [27]. These are optimization-based models that compute heuristic control policies for intractable MDPs. Secomandi [27] 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 Andersen and Broadie [2] and Haugh and Kogan [20] on pricing American options. 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 [28], this is significantly different from the problems studied in the inventory management literature (Zipkin [36], Porteus 2002 [24]).

To the best of our knowledge, Secomandi [28] is the only other paper that compares optimal and heuristic approaches for natural gas storage valuation. This author also finds that a reoptimized deterministic model can perform very well in this context, 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 state-dependent basestock target characterization of the optimal policy of our ADP model extends that established by Secomandi [28]: 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 practice-based heuristics in §3. We present our ADP model in §4, and our computational implementation of this model 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 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 [14, 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 [23, p. 440], Eydeland and Wolyniec [14, 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 [14, 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 $\mathcal{I} := \{0, \ldots, N-1\}$. 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, $j \geqslant i$; $F(T_i, T_i)$ 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 $\mathbf{F}_N := 0$.

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 [14, Chapter 5] and Gray and Khandelwal [18, 17]). 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 maturities T_i and T_j are instantaneously correlated with constant correlation coefficient $\rho_{ij} \in [-1, 1]$, and $\rho_{ii} := 1$. This is the following N-factor model:

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

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

In our numerical experiments reported in §6, we use this model as representative of the high-dimensional forward models that, as discussed in §1, appear 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 [13, §8.6] discuss such a model. These types of models are related to the so called string and BGM models (Kennedy [22], Brace et al. [8]) used to value fixed income instruments (see also Eydeland and Wolyniec [14, 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 [28]. 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 sale or a purchase, 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

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

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

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

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

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.

The LP policy is based on spread option valuation and linear programming. A spread option is an option on the difference between two prices with a positive strike price (Carmona and Durrleman 2003 [10]). 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 injection and withdrawal marginal costs at time T_i. We refer to such an option as the i-j spread option. The time T_0 value of such an option is

$$S_0^{i,j}(\mathbf{F}_0) := \delta^i \mathbb{E}\left[\left\{\delta^{j-i}\alpha^W \tilde{F}_{i,j} - \left(\alpha^I \tilde{F}_{i,i} + \delta^{j-i}c^W + c^I\right)\right\}^+ |\mathbf{F}_0\right]. \tag{9}$$

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.

The LP policy works with portfolios of spread options $\{q_{i,j}, i \in \mathcal{I}, j \in \mathcal{I}, i < j\}$; 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, ..., $N-2$, and 1, 2, ..., $N-1$, respectively. 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 as an optimal solution to linear program (10)-(16) below. The decision variables in this linear program are the notional amounts in set $\{q_{i,j}, i \in \mathcal{I}, j \in \mathcal{I}, i < j\}$, and the inventory levels in set $\{x_i, i \in \mathcal{I} \cup \{N\}\}$ (these are not needed but simplify the formulation). This linear program, which only depends on the time T_0

information set $\mathcal{F}_0 := \{x_0, \mathbf{F}_0\}$, follows:

$$
\begin{aligned}
U_0^{LP}(\mathcal{F}_0) \quad := \quad & \max_{q,x} \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{I}, i<j} S_0^{i,j}(\mathbf{F}_0) q_{i,j} && (10) \\
& \text{s.t. } x_{i+1} = x_i + \sum_{j\in\mathcal{I}, j>i} q_{i,j} - \sum_{j\in\mathcal{I}, j<i} q_{j,i}, \ \forall i \in \mathcal{I} && (11) \\
& x_i \leqslant \overline{x}, \ \forall i \in \mathcal{I} \cup \{N\} && (12) \\
& \sum_{j\in\mathcal{I}, j>i} q_{i,j} \leqslant -C^I, \ \forall i \in \mathcal{I} \setminus \{N-1\} && (13) \\
& \sum_{i\in\mathcal{I}, i<j} q_{i,j} \leqslant C^W, \ \forall j \in \mathcal{I} \setminus \{0\} && (14) \\
& q_{i,j} \geqslant 0, \ \forall i, j \in \mathcal{I}, \ i < j && (15) \\
& x_i \geqslant 0, \ \forall i \in \mathcal{I} \cup \{N\}. && (16)
\end{aligned}
$$

The objective function (10) is the value of the portfolio of spread options. Constraint sets (11) and (12) express inventory balance and bounding conditions, respectively. Constraint sets (13)-(14) enforce capacity constraints. Constraint sets (15)-(16) pose nonnegativity conditions on the decision variables.

A version of this model is discussed by Eydeland and Wolyniec [14, pp. 362] and Gray and Khandelwal [17]. 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 2003 [10]). Once these coefficients are known, this model can be optimally solved very efficiently.

Let $q_{i,j}^{LP}(\mathcal{F}_0)$ be an optimal portfolio, that is, it is part of an optimal solution to model (10)-(16). 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. Denote by $\mathcal{F}_i^{LP}$ the information available at time T_i when executing this policy, that is, this set includes all the relevant price and inventory related history, including the forward curve and inventory at time T_i. Given $\mathcal{F}_i^{LP}$, we define the following quantities:

$$
q_{i,j}^{LP,+}(\mathcal{F}_i^{LP}) := \begin{cases} 0 & \text{if } \delta^{j-i}\alpha^W F_{i,j} - \left(\alpha^I F_{i,i} + \delta^{j-i}c^W + c^I\right) \leqslant 0 \\ q_{i,j}^{LP}(\mathcal{F}_0) & \text{otherwise.} \end{cases} \tag{17}
$$

(Computation of these quantities does not require all the information available in $\mathcal{F}_i^{LP}$, but we find it notationally convenient to make them depend on $\mathcal{F}_i^{LP}$; we use similar notation throughout this paper.) The LP policy uses these quantities to obtain an action at time T_i given $\mathcal{F}_i^{LP}$ as follows:

$$
a_i^{LP}(\mathcal{F}_i^{LP}) := \sum_{j\in\mathcal{I}, j<i} q_{j,i}^{LP,+}(\mathcal{F}_i^{LP}) - \sum_{j\in\mathcal{I}, j>i} q_{i,j}^{LP,+}(\mathcal{F}_i^{LP}). \tag{18}
$$

We define the time T_0 value of the LP policy as

$$V_0^{LP}(\mathcal{F}_0) := \mathbb{E}\left[\sum_{i\in\mathcal{I}} r\left(a_i^{LP}(\tilde{\mathcal{F}}_i^{LP}), \tilde{s}_i\right) \mid \mathcal{F}_0\right], \tag{19}$$

which can be estimated using Monte Carlo simulation as discussed in §5.

Proposition 1 shows that the optimal objective function value of model (10)-(16) is no greater than the value of the LP policy, and that both of these values are lower bounds on $V_0(\mathcal{F}_0)$.

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

Proof. Constraint sets (11)-(16) are sufficient for the feasibility of the LP policy for model (7)-(8). To see this, notice that constraints (11) 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 the LP policy only withdraws and sells natural gas that was previously bought and injected. Moreover, constraint sets (13)-(14) 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)-(16) 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)-(16) that combines constraints (13)-(14) is sometimes used in practice. In this formulation, which we refer to as the reformulated linear program, constraints (13), for all $i \in \mathcal{I}\setminus\{0, N-1\}$, and (14), for all $j \in \mathcal{I}\setminus\{0, N-1\}$, are replaced with the following "net" constraints:

$$C^I \leqslant \sum_{j\in\mathcal{I}, j<i} q_{j,i} - \sum_{j\in\mathcal{I}, j>i} q_{i,j} \leqslant C^W, \ \forall i \in \mathcal{I} \setminus \{0, N-1\}. \tag{20}$$

The objective function of the reformulated linear program is obviously higher than $U_0^{LP}(\mathcal{F}_0)$. But, it may not yield a lower bound on the value of storage since constraints (20) 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 injects any amount of natural gas associated with the unique spread option that expires in this stage, provided that this option is in the money. 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 increasing 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).

As discussed by Gray and Khandelwal [17], it is typically possible to improve the performance of the LP and LPN policies by reoptimizing their associated linear programs at each maturity to take advantage of the price and inventory information that becomes available over time. (This is not obvious; we refer the reader to Secomandi [27] for an analysis of the merit of sequentially reoptimizing a math program to generate a feasible, but heuristic, sequence of actions to an MDP in inventory and revenue management contexts.) In this reoptimization mode, at time T_k one computes a new set of i-j spread option values, with $k \leqslant i < j$, as follows:

$$S_k^{i,j}(\mathbf{F}_k) := \delta^{i-k}\mathbb{E}\left[\left\{\delta^{j-i}\alpha^W \tilde{F}_{i,j} - \left(\alpha^I \tilde{F}_{i,i} + \delta^{j-i}c^W + c^I\right)\right\}^+ |\mathbf{F}_k\right]. \tag{21}$$

Let $1\{\cdot\}$ be the indicator function of event $\{\cdot\}$, which is equal to 1 if this event is true and 0 otherwise. The linear program solved at time T_k is

$$U_k^{LP}(x_k, \mathbf{F}_k) := \max_{y,q,x} s_k y_k + \sum_{i\in\mathcal{I}, i\geqslant k}\sum_{j\in\mathcal{I}, i<j} S_k^{i,j}(\mathbf{F}_k) q_{i,j} \tag{22}$$

$$\text{s.t. } x_{i+1} = x_i + \sum_{j\in\mathcal{I}, j>i} q_{i,j} - y_k 1\{i=k\} - \sum_{j\in\mathcal{I}, k\leqslant j<i} q_{j,i},\ \forall i\in\mathcal{I},\ i\geqslant k \tag{23}$$

$$x_i \leqslant \overline{x},\ \forall i \in \mathcal{I}\cup\{N\},\ i\geqslant k \tag{24}$$

$$\sum_{j\in\mathcal{I}, j>i} q_{i,j} \leqslant -C^I,\ \forall i\in\mathcal{I}\setminus\{N-1\},\ i\geqslant k \tag{25}$$

$$y_k \leqslant C^W \tag{26}$$

$$\sum_{i\in\mathcal{I}, k\leqslant i<j} q_{i,j} \leqslant C^W,\ \forall j\in\mathcal{I},\ j>k \tag{27}$$

$$y_k \geqslant 0 \tag{28}$$

$$q_{i,j} \geqslant 0,\ \forall i,j\in\mathcal{I},\ k\leqslant i<j \tag{29}$$

$$x_i \geqslant 0,\ \forall i\in\mathcal{I}\cup\{N\},\ i\geqslant k. \tag{30}$$

This model is very similar to model (10)-(16), but also uses the information available at time T_k and spot-sale decision variable y_k. This variable allows one to sell some or all of the inventory available at time T_k.

As now described, an optimal solution to this linear program can be used to determine a feasible policy for (7)-(8), which we call the reoptimized LP policy and denote by RLP (as mentioned in §1). Denote by $y_k^{RLP}(\mathcal{F}_k^{RLP})$ and $q_{k,j}^{RLP}(\mathcal{F}_k^{RLP})$ the optimal decision variables of linear program (22)-(30), where we explicitly indicate that they depend on the history $\mathcal{F}_i^{RLP}$ of RLP. An action of this policy at time T_k is defined as

$$a_k^{RLP}(\mathcal{F}_k^{RLP}) := y_k^{RLP}(\mathcal{F}_k^{RLP}) - \sum_{j\in\mathcal{I}, j>k} q_{k,j}^{RLP}(\mathcal{F}_k^{RLP}). \tag{31}$$

The value of this policy, denoted by $V_0^{RLP}(\mathcal{F}_0)$, follows:

$$V_0^{RLP}(\mathcal{F}_0) := \mathbb{E}\left[\sum_{i\in\mathcal{I}} r\left(a_i^{RLP}(\tilde{\mathcal{F}}_i^{RLP}), \tilde{s}_i\right) \mid \mathcal{F}_0\right]. \tag{32}$$

This value can be estimated by Monte Carlo simulation as discussed in §5. For brevity, we simply point out that reoptimization can also be applied to policy LPN to obtain policy RLPN; its value can be estimated in a similar fashion.

Another model of interest is the so called intrinsic value model, which computes the value of storage that can be attributed to seasonality (Secomandi [28]). This model is

$$U_N^I(x_N;\mathbf{F}_0) \;:=\; 0,\ \forall x \in \mathcal{X} \tag{33}$$

$$U_i^I(x_i;\mathbf{F}_0) \;:=\; \max_{a\in\mathcal{A}(x_i)} r(a, F_{0,i}) + \delta U_{i+1}^I(x_i - a;\mathbf{F}_0),\ \forall i \in \mathcal{I}, \forall x \in \mathcal{X}. \tag{34}$$

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. We denote the value of the I policy by $V_0^I(\mathcal{F}_0) \equiv U_0^I(x_0;\mathbf{F}_0)$.

The intrinsic value model can also be useful for estimating the value of storage, that is, the sum of its intrinsic and extrinsic values, if used in a reoptimization fashion. This is the value of the RI policy, which is discussed by Maragos [23] and Gray and Khandelwal [17]. In this case, at time T_k one solves the following intrinsic model:

$$U_N^{RI}(x_N;\mathbf{F}_k) \;:=\; 0,\ \forall x \in \mathcal{X} \tag{35}$$

$$U_i^{RI}(x_i;\mathbf{F}_k) \;:=\; \max_{a\in\mathcal{A}(x_i)} r(a, s_i) + \delta U_{i+1}^{RI}(x_i - a;\mathbf{F}_k),\ \forall i \in \mathcal{I},\ i \geqslant k,\ \forall x \in \mathcal{X}. \tag{36}$$

A sequence of feasible actions can be constructed by taking the time T_k action to be

$$a_k^{RI}(\mathcal{F}_k^{RI}) \in \operatorname*{arg\,max}_{a\in\mathcal{A}(x_k)} r(a, s_k) + \delta U_{k+1}^{RI}(x_k - a;\mathbf{F}_k), \tag{37}$$

where $\mathcal{F}_k^{RI}$ is the history of the RI policy at time T_k. The value of this policy is

$$V_0^{RI}(\mathcal{F}_0) := \mathbb{E}\left[\sum_{i\in\mathcal{I}} r\left(a_i^{RI}(\tilde{\mathcal{F}}_i^{RI}), \tilde{s}_i\right) \mid \mathcal{F}_0\right], \tag{38}$$

which 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.

Our approach is to reduce the computationally intractable and exact model (7)-(8) to a computationally tractable, but approximate, model using information reduction. We accomplish this information reduction by considering a model with imperfect information. Specifically, the state of our ADP model in stage i includes the current inventory and the spot price; we compute the optimal value function of this model in this stage by conditioning on the possible values of the prompt (next) month futures price at time T_i, $F_{i,i+1}$, given the current spot price, s_i, *and* the prompt month futures price at time T_0, $F_{0,i+1}$. This allows us to heuristically reduce the intractable exact model with $N-i+1$ state variables in stage i to a tractable model with 2 such variables in each stage. The ADP model follows:

$$U_N^{ADP}(x_N, s_N) := 0, \ \forall x_N \in \mathcal{X} \tag{39}$$

$$U_i^{ADP}(x_i, s_i) = \mathbb{E}\left[u_i^{ADP}(x_i, s_i, \tilde{F}_{i,i+1}) \mid s_i, F_{0,i+1}\right], \ \forall i \in \mathcal{I}, \ (x_i, s_i) \in \mathcal{X} \times \Re_+ \tag{40}$$

$$u_i^{ADP}(x_i, s_i, F_{i,i+1}) := \max_{a \in \mathcal{A}(x_i)} r(a, s_i) + \delta \mathbb{E}\left[U_{i+1}^{ADP}(x_i - a, \tilde{s}_{i+1}) \mid F_{i,i+1}\right]. \tag{41}$$

The conditioning on the prompt month futures price given the spot price in stage i and the futures price for maturity T_{i+1} at time T_0 is done in (40) (by convention $F_{0,N} = F_{N-1,N} := 0$); this yields the ADP model value function $U_i^{ADP}(x_i, s_i)$ in stage i and state (x_i, s_i), which is the expected value of the auxiliary value function $u_i^{ADP}(x_i, s_i, F_{i,i+1})$ in this stage and extended state $(x_i, s_i, F_{i,i+1})$. The computation of this auxiliary value function in (41) also yields an optimal action in this stage and extended state as follows:

$$a_i^{ADP}(x_i, s_i, F_{i,i+1}) \in \arg\max_{a \in \mathcal{A}(x_i)} r(a, s_i) + \delta \mathbb{E}\left[U_{i+1}^{ADP}(x_i - a, \tilde{s}_{i+1}) \mid F_{i,i+1}\right]. \tag{42}$$

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 two or three, that is, $U_0^{ADP}(x_0, s_0) = V_0(x_0, \mathbf{F}_0)$ when $N = 2, 3$. This does not necessarily hold for $N \geqslant 4$, when $U_0^{ADP}(x_0, s_0)$ 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 (39)-(41) 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.

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

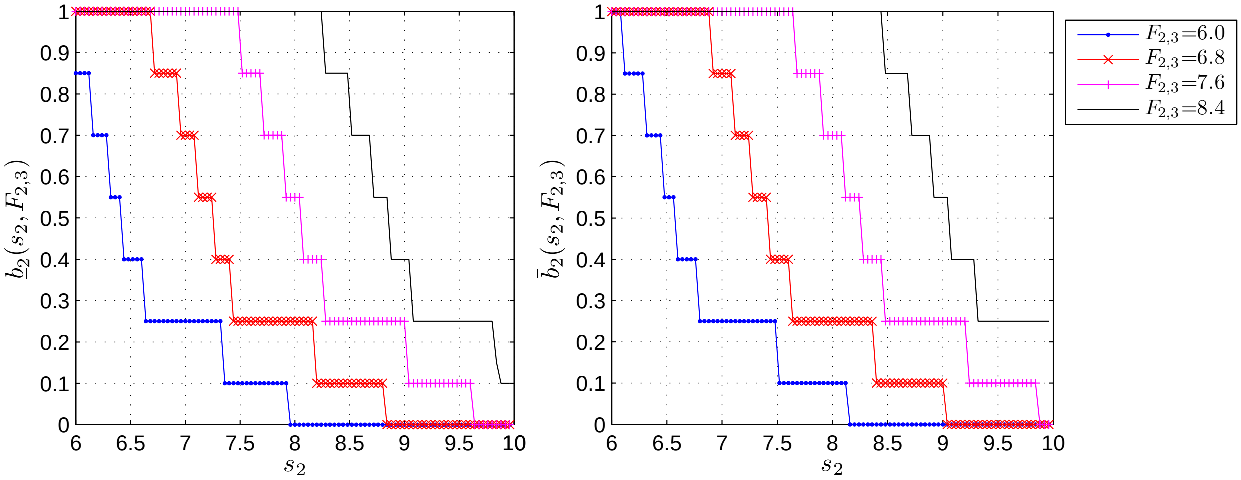

Figure 1: Illustration of the behaviors of the basestock targets with respect to the spot price and the prompt month futures price.

established in Theorem 1. This theorem characterizes the optimal policy of model ADP as having a 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, under an assumption on the conditional distribution of the spot price in the next stage given the prompt month futures price in the current stage. For example, this assumption is satisfied by multidimensional Black model (1)-(2). Figure 1 illustrates these behaviors for a given stage in one of the instances discussed in §6.

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*

$$a_i^{ADP}(x_i, s_i, F_{i,i+1}) = \begin{cases} C^I \vee (x_i - \underline{b}_i(s_i, F_{i,i+1})) & \text{if } x_i \in \mathcal{X}_i^I(s_i, F_{i,i+1}) \\ 0 & \text{if } x_i \in \mathcal{X}_i^{DN}(s_i, F_{i,i+1}) \\ C^W \wedge (x_i - \bar{b}_i(s_i, F_{i,i+1})) & \text{if } x_i \in \mathcal{X}_i^W(s_i, F_{i,i+1}) \end{cases}, \tag{43}$$

where

$$\mathcal{X}_i^I(s_i, F_{i,i+1}) := [0, \underline{b}_i(s_i, F_{i,i+1})) \tag{44}$$

$$\mathcal{X}_i^{DN}(s_i, F_{i,i+1}) := [\underline{b}_i(s_i, F_{i,i+1}), \bar{b}_i(s_i, F_{i,i+1})] \tag{45}$$

$$\mathcal{X}_i^W(s_i, F_{i,i+1}) := (\bar{b}_i(s_i, F_{i,i+1}), \bar{x}]. \tag{46}$$

Suppose 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, for each given $i \in \mathcal{I}$, (b) $U_i^{ADP}(x_i, s_i)$ is supermodular in $(x_i, s_i) \in \mathcal{X} \times \Re_+$, (c) $\underline{b}_i(s_i, F_{i,i+1})$ and $\overline{b}_i(s_i, F_{i,i+1})$ weakly decrease in s_i for given $F_{i,i+1}$, and (d) they weakly increase in $F_{i,i+1}$ for given s_i.

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 a proof in Secomandi [28]). The proof of the rest of part (a) extends a proof of Secomandi [28]. 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 [33], $U_{i+1}^{ADP}(x_i - a, s_{i+1})$ 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 [33], $\delta \mathbb{E}\left[U_{i+1}^{ADP}(x_i - a, \tilde{s}_{i+1}) \mid F_{i,i+1}\right]$ 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 [33] 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 $a_i^{ADP}(\mathcal{F}_i^{ADP})$ weakly increases in x_i for each given pair $(s_i, F_{i,i+1})$ by Theorem 2.8.2 in Topkis [33]. This implies the existence of the stated feasible inventory levels $\underline{b}_i(s_i, F_{i,i+1})$ and $\overline{b}_i(s_i, F_{i,i+1})$ and sets $\mathcal{X}_i^I(s_i, F_{i,i+1})$, $\mathcal{X}_i^{DN}(s_i, F_{i,i+1})$, and $\mathcal{X}_i^W(s_i, F_{i,i+1})$.

Pick a feasible extended state $(x_i, s_i, F_{i,i+1})$. In determining an optimal action in this extended state, 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

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

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

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

and any optimal solution to this problem does not depend on x_i. Similar arguments for the cases $x_i \in \mathcal{X}_i^{DN}(s_i, F_{i,i+1})$ and $x_i \in \mathcal{X}_i^W(s_i, F_{i,i+1})$ establish the claimed characterization of $a_i^{ADP}(\mathcal{F}_i^{ADP})$.

(b)-(d) 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 $U_i^{ADP}(x_i, s_i)$ is supermodular in $(x_i, s_i) \in \mathcal{X} \times \Re_+$ for all stages $i+1, \ldots, N-1$. Consider the determination of an optimal action in feasible extended state $(x_i, s_i, F_{i,i+1})$ in stage i. Suppose that $x_i \in \mathcal{X}_i^I(s_i, F_{i,i+1})$, so that the relevant optimization is

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

where $\mathcal{X}^F(x_i) := [0 \vee (x_i - C^W), (x_i - C^I) \wedge \overline{x}]$. Set $\mathcal{X}^F(x_i)$ is clearly a lattice. 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 [33] 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 $(s_i, F_{i,i+1})$, any optimal x_{i+1} increases in $F_{i,i+1}$ by Theorem 2.8.2 in Topkis [33].

Now focus on how an optimal action depends on the pair (x_i, s_i) for each given $F_{i,i+1}$. The relevant optimization is

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

It follows from Lemmas 2.6.1(a)-(b) and 2.6.2(b) and Corollary 2.6.2 in Topkis [33] that the objective function of this problem is supermodular in (x_i, s_i, a) for each given $F_{i,i+1}$. Thus, an optimal action in this problem increases in (x_i, s_i) for each given $F_{i,i+1}$ by Theorem 2.8.2 in Topkis [33]. Theorem 2.7.6 in Topkis [33] also implies that $u_i^{ADP}(x_i, s_i, F_{i,i+1})$ is supermodular in $(x_i, s_i) \in \mathcal{X}_i^I(s_i, F_{i,i+1}) \times \Re_+$ for each given $F_{i,i+1}$. Similar arguments when $x_i \in \mathcal{X}_i^{DN}(s_i, F_{i,i+1})$ and $x_i \in \mathcal{X}_i^W(s_i, F_{i,i+1})$ establish the claimed behavior of the basestock targets in stage i. This also implies that the function $u_i^{ADP}(x_i, s_i, F_{i,i+1})$ 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 [33] implies that $U_i^{ADP}(x_i, s_i)$ is supermodular in $(x_i, s_i) \in \mathcal{X} \times \Re_+$. Thus, parts (b)-(d) are true for every stage by the principle of mathematical induction. $\square$

The ADP model generates a feasible policy for the exact model (7)-(8). However, the value function of model (39)-(41) is not the value function of this policy evaluated under the full information available in model (7)-(8), which we call the ADP policy. 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 price and the forward curve in the initial stage. We denote an action of the ADP policy by $a_i^{ADP}(\mathcal{F}_i^{ADP})$, where $\mathcal{F}_i^{ADP}$ is the history generated by the ADP policy up to and including time T_i. We point out that $a_i^{ADP}(\mathcal{F}_i^{ADP}) \equiv a_i^{ADP}(x_i, s_i, F_{i,i+1})$, since $\mathcal{F}_i^{ADP}$ includes the information needed to compute $a_i^{ADP}(x_i, s_i, F_{i,i+1})$ in (42). We define the value of the ADP policy by

$$V_0^{ADP}(\mathcal{F}_0) := \mathbb{E}\left[\sum_{i \in \mathcal{I}} r\left(a_i^{ADP}(\tilde{\mathcal{F}}_i^{ADP}), \tilde{s}_i\right) \mid \mathcal{F}_0\right]. \tag{47}$$

This value can be estimated by Monte Carlo simulation as discussed in §5.

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 given the information available at that point. In other words, we reoptimize model ADP at each time T_k by using $\mathbf{F}_k$ rather than $\mathbf{F}_0$ as in (39)-(41). Specifically, we replace $F_{0,i+1}$ with $F_{k,i+1}$ in (40). The value of the resulting RADP policy, which we denote by $V_0^{RADP}(\mathcal{F}_0)$, is typically different, and in fact higher, than $V_0^{ADP}(\mathcal{F}_0)$. In other words, the RADP policy can typically take advantage of sequential reoptimization.

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

$$p_i^{ADP}(x_i, a; F_{i,i+1}) := U_{i+1}^{ADP}(x_i - a, s_{i+1}) - \mathbb{E}\left[U_{i+1}^{ADP}(x_i - a, \tilde{s}_{i+1}) \mid F_{i,i+1}\right], \ \forall i \in \mathcal{I}, \ (x_i, a) \in \mathcal{X} \times \mathcal{A}(x_i), \tag{48}$$

which are based on the optimal value function of the ADP model computed at time T_0. We denote by $\mathcal{G}$ the perfect price information set. This set includes a set of forward curves, one for each stage. Given this set, we solve the following DUB model:

$$U_N^{DUB}(x_N; \mathcal{G}) := 0, \ \forall x_N \in \mathcal{X} \tag{49}$$

$$U_i^{DUB}(x_i; \mathcal{G}) = \max_{a \in \mathcal{A}(x_i)} r(a, s_i) - p_i^{ADP}(x_i, a; F_{i,i+1}) + \delta U_{i+1}^{DUB}(x_i - a; \mathcal{G}), \forall i \in \mathcal{I}, \ x_i \in \mathcal{X}. \tag{50}$$

This is a perfect information model whose immediate rewards are penalized according to the penalty terms (48), which are not computed based on a perfect information assumption. An optimal action in the DUB model is

$$a_i^{DUB}(x_i; \mathcal{G}) \in \operatorname*{arg\,max}_{a \in \mathcal{A}(x_i)} r(a, s_i) - p_i^{ADP}(x_i, a; F_{i,i+1}) + \delta U_{i+1}^{DUB}(x_i - a; \mathcal{G}). \tag{51}$$

Given set $\mathcal{G}$, we can implement a sequence of such actions. We let set $\mathcal{G}_i^{DUB}$ include $\mathcal{G}$ and the inventory levels associated with implementing this action sequence up to and including stage i. Thus, we can write $a_i^{DUB}(\mathcal{G}_i^{DUB}) \equiv a_i^{DUB}(x_i; \mathcal{G})$. We obtain an upper bound on the value of storage in the manner stated in Proposition 2, which follows from Brown et al. [9].

Proposition 2 (DUB). *Define*

$$V_0^{DUB}(\mathcal{F}_0) := \mathbb{E}\left[\sum_{i \in \mathcal{I}} r\left(a_i^{DUB}(\tilde{\mathcal{G}}_i^{DUB}), \tilde{s}_i\right) \mid \mathcal{F}_0\right]. \tag{52}$$

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

The upper bound $V_0^{DUB}(\mathcal{F}_0)$ 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 (48) equal to zero and proceeding analogously to the computation of upper bound (52) yields the PIUB model, which does not depend on the value function of the ADP model. We denote the optimal value function of the PIUB model in the initial stage by $V_0^{PIUB}(\mathcal{F}_0)$. It is clear that this is an upper bound on the value of storage (this also follows from Brown et al. [9]). This upper bound provides a benchmark for the performance of upper bound $V_0^{DUB}(\mathcal{F}_0)$. Although we do not have a structural result that compares $V_0^{DUB}(\mathcal{F}_0)$ and $V_0^{PIUB}(\mathcal{F}_0)$, we expect the latter upper bound to be looser than the former. Our numerical results in §6 support this statement.

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 relevant state variables as well as of the distribution functions of the relevant random variables.

Price evolution and discretization. The forward curve evolves in continuous time according to the multidimensional Black model (1)-(2). In the ADP model, we use a discretized simplification of its dynamics. This is done as follows. For all $i \in \mathcal{I} \setminus \{N-1\}$, we construct a 3-dimensional binomial tree (Haugh [19, §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 $G_{i,i+1}(\cdot \mid F_{0,i}, F_{0,i+1})$ 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 $G_{i,i+1}(\cdot \mid F_{0,i}, F_{0,i+1})$, 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 probability mass function $L_{i+1}(\cdot \mid F_{i,i+1})$ for 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 $F(t, N-1)$ starting from $F_{0,N-1}$.

Computing and evaluating the policies. In solving the ADP model, we use a regular discretization with 100 points of the feasible inventory set $\mathcal{X}$ in each stage $i \in \mathcal{I}$. As given in (39), 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 auxiliary 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 (41) using $L_{i+1}(\cdot \mid F_{i,i+1})$. Here, typically, some rounding operation is 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 (41).

In the determination of an optimal action in (41), 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, we can easily compute an optimal action and the value of the auxiliary 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 (40) 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 2003 [10]). We solve the linear programs associated with these policies using the Clp linear solver of COIN-OR (www.coin-or.org).

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 $N-1$. Starting with a given 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 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 13 in Online Appendix A 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 14 in Online Appendix A 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 employ market data to imply out correlations between futures prices for different maturi-

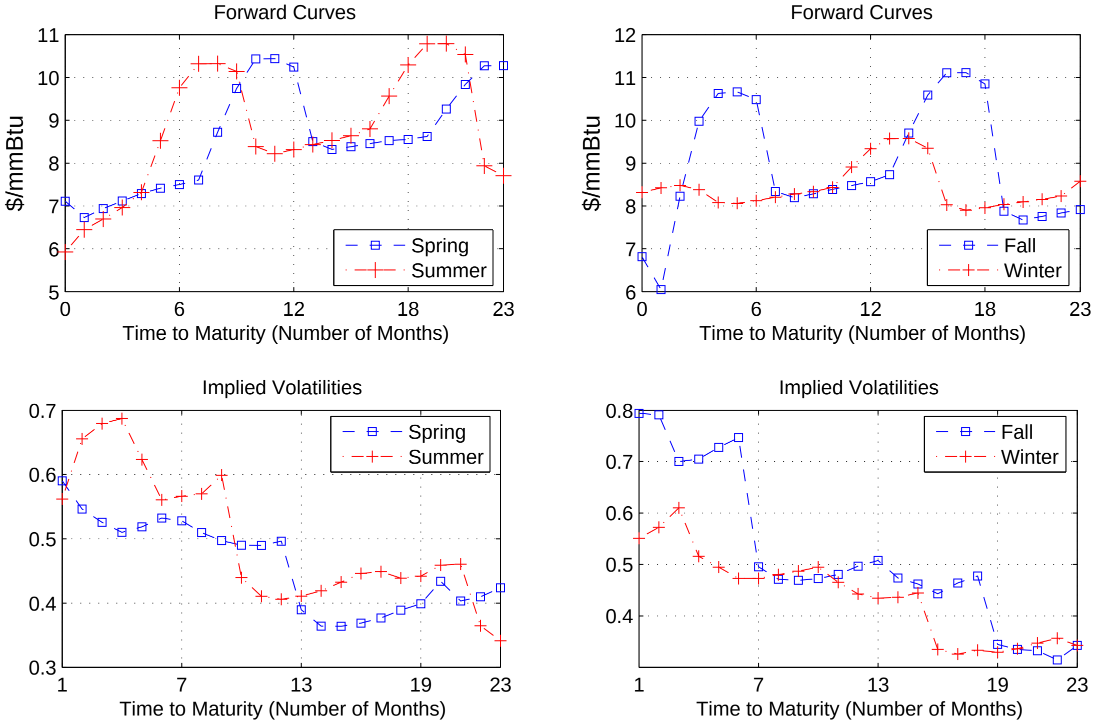

Figure 2: NYMEX natural gas forward curves (top panels) and implied volatilities (bottom panels) on 3/1/2006 (Spring), 6/1/2006 (Summer), 8/31/2006 (Fall), and 12/1/2006 (Winter).

ties. Instead, we construct two correlation matrices, labeled A and B, by modifying and extending a matrix of natural gas futures price correlations reported by Eydeland and Wolyniec [14, p. 102]. We modify and extend it, respectively, because this matrix is not positive definite and it covers ten maturities, whereas in our numerical experiments we employ more than ten maturities. We do this by mimicking the decreasing pattern of the correlation coefficients of the original matrix. Specifically, we create correlation matrices A and B by setting their respective i-j elements, denoted by ρ_{ij}^A and ρ_{ij}^B, $i, j = 1, \ldots, 23$, as follows:

$$\rho_{ij}^A = 1 - 0.02|i-j| \tag{53}$$

$$\rho_{ij}^B = \begin{cases} 0.88 - 0.02|i-j| & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}. \tag{54}$$

Figure 3 illustrates the two patterns that represent these two matrices. (Tables 15-16 in Online Appendix A partially display them.) The correlation matrix B entails a weaker correlation between future prices than matrix A.

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.

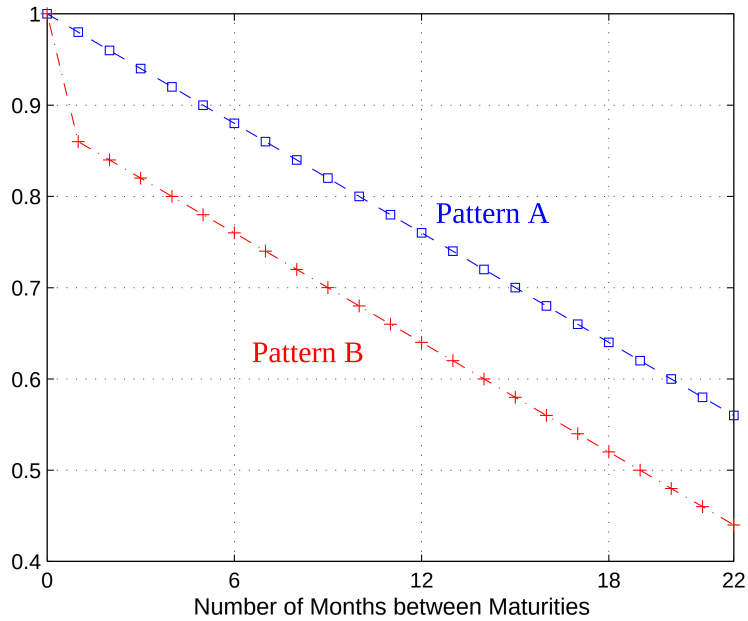

Figure 3: Patterns used to build correlation matrices A and B.

Table 2: Capacity Pairs (Fractions of Maximum Inventory).

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

We normalize the maximum inventory $\overline{x}$ to one mmBtu. We employ the three pairs of injection and withdrawal capacities relative to the maximum inventory shown in Table 2. The first pair roughly reflects the capacities in Example 8.11 in Eydeland and Wolyniec [14, pp. 355]. We obtain the other two by multiplying the first one by 2 and 3, respectively. Following Maragos [23], 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.
- Correlation matrix: A or B.

We consider twenty-four 12-stage instances, labeled 12-Sp-1-A to 12-Wi-3-B, and twenty-four 24-stage instances, labeled 24-Sp-1-A to 24-Wi-3-B. Thus, we generate a total of forty-eight 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. Compared to their respective average cpu times on the 24-stage instances, on the 12-stage instances, without reoptimization those of the ADP policy are roughly half, while those of the other three policies are roughly one third; with reoptimization those of all the policies are roughly equal to their original cpu times divided by four.

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. We typically obtain standard errors within the range of 0.03 to 0.08. For brevity we do not report individual standard errors in the discussion to follow.

DUB. We compare the value of all the policies to the upper bound of Proposition 2, 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 $m = 20$. Table 3 reports the values of DUB for values of m_i ranging from 5 to 500, as well as the value of the perfect information upper bound, PIUB. Setting $m_i = 500$ yields the tightest DUB. This is why this table displays the values of DUB for $m_i < 500$ and PIUB as percentages of the values of DUB obtained for $m_i = 500$. This table also summarizes the average performance of the DUB values on the instances corresponding to the A and B correlation matrices, as well as on all such instances. We report analogous averages in all the remaining valuation-performance related tables. 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 $m_i = 5$. Thus, we use DUB for $m_i = 500$ in all the ensuing comparisons.

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 8. Computing

Table 3: DUB Values for Different Values of Discretization Parameter m_i and PIUB Value. The Values of DUB for $m_i < 500$ and PIUB are Percentages of the DUB Value for $m_i = 500$.

	DUB						
	Number of Discretization Steps						
Instance	500	5	20	50	100	250	PIUB
24-Sp-1-A	4.8597	117.3951	108.9803	103.6052	101.5328	100.4299	136.7362
24-Sp-2-A	6.3765	125.0207	111.3580	104.5720	101.9989	100.5672	147.5956
24-Sp-3-A	7.1410	130.1143	112.6000	105.1153	102.2574	100.6510	153.6276
24-Su-1-A	5.3927	116.0302	107.7635	103.4773	101.3516	100.4423	133.2845
24-Su-2-A	7.4107	121.9929	109.2521	104.0263	101.6116	100.5519	141.3435
24-Su-3-A	8.2542	127.0490	110.6281	104.5416	101.8559	100.6259	147.9958
24-Fa-1-A	4.8115	119.4119	108.4070	103.5282	101.7001	100.4610	141.0485
24-Fa-2-A	7.5764	122.7128	109.0622	103.7526	101.8353	100.5058	143.1092
24-Fa-3-A	9.0684	125.6313	109.6941	104.0090	101.9530	100.5213	146.0791
24-Wi-1-A	2.5363	138.6797	117.0081	107.1225	103.1183	100.9506	178.2721
24-Wi-2-A	3.6098	147.4832	119.1341	107.9490	103.5135	101.0327	189.0644
24-Wi-3-A	4.2642	152.8792	120.2492	108.4069	103.7374	101.0921	194.7019
Average (A)		126.9943	111.2168	104.6794	102.0644	100.6111	150.6702
24-Sp-1-B	6.8384	106.8490	102.7431	101.0431	100.4163	100.1012	117.2297
24-Sp-2-B	10.5846	108.6673	103.1593	101.0950	100.4696	100.1143	118.1150
24-Sp-3-B	13.1787	108.9463	103.1968	101.1101	100.4727	100.1169	118.0010
24-Su-1-B	7.4926	107.0825	102.6566	101.0720	100.3865	100.0613	115.7813
24-Su-2-B	11.7759	108.6439	103.1369	101.2118	100.4484	100.0577	116.8463
24-Su-3-B	14.5190	109.1783	103.2950	101.2556	100.4663	100.0585	117.5005
24-Fa-1-B	7.0097	108.8208	102.9453	101.0856	100.5361	100.1201	118.1262
24-Fa-2-B	12.0093	109.1188	103.1118	101.1491	100.5571	100.1191	117.1908
24-Fa-3-B	15.3828	109.1511	103.1106	101.1877	100.5701	100.1027	117.1107
24-Wi-1-B	4.5961	113.8916	104.4581	101.6390	100.6780	100.1651	126.7187
24-Wi-2-B	7.6264	114.9441	104.7131	101.7464	100.6744	100.1726	126.1635
24-Wi-3-B	9.9324	114.3963	104.5487	101.7015	100.6552	100.1750	124.9547
Average (B)		109.7358	103.3609	101.2511	100.5207	100.1083	118.8880
Average (All)		116.1367	106.2746	102.5226	101.0932	100.2948	130.6755

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

	DUB						
	Number of Discretization Steps						
Statistic	5	20	50	100	250	500	PIUB
Maximum	34.4298	51.9761	73.9538	121.2697	337.9875	935.2750	6.8260
Minimum	31.9032	48.6596	69.6804	111.9547	304.2407	827.3310	6.6200
Average	32.9677	49.9142	71.4621	116.7544	320.1910	886.0233	6.7008
Standard Deviation	0.8446	0.8321	1.1635	2.3396	8.5010	31.0186	0.0528

Table 5: Effect of Different Values 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-A	74.9542	84.8436	87.0225	88.1479	88.7986	88.8998
24-Sp-2-A	70.9053	85.3660	89.0011	90.8484	91.7887	92.0618
24-Sp-3-A	69.1337	86.1615	90.6532	92.7021	93.8193	94.2290
24-Su-1-A	75.3752	85.6374	88.0375	88.9957	89.5455	89.7317
24-Su-2-A	74.3231	87.5127	91.4341	92.5480	93.4750	93.8253
24-Su-3-A	72.5418	87.8180	92.3855	93.9700	95.0917	95.5464
24-Fa-1-A	73.6310	84.9267	88.0018	89.1061	89.8354	89.9783
24-Fa-2-A	74.5367	87.5327	91.2329	92.8244	93.5925	93.9226
24-Fa-3-A	73.8842	88.6809	92.6638	94.5513	95.4584	95.8435
24-Wi-1-A	44.5204	66.0512	71.8864	74.0360	75.7089	76.0283
24-Wi-2-A	45.4141	71.5653	79.1093	81.6651	84.0007	84.5481
24-Wi-3-A	46.0761	74.5238	82.7220	86.0706	88.4079	89.0798
Average (A)	69.1212	84.5256	88.7539	90.4625	91.5344	91.8827
24-Sp-1-B	73.3681	79.5229	80.5386	81.0022	81.2131	81.1559
24-Sp-2-B	76.1658	83.7753	85.1703	85.7417	86.0580	86.0877
24-Sp-3-B	78.4584	86.5837	88.2887	88.8016	89.1954	89.2786
24-Su-1-B	75.1365	81.4186	82.4629	82.9436	83.1361	83.1865
24-Su-2-B	78.2888	86.1021	87.5755	88.1640	88.3678	88.4765
24-Su-3-B	79.4993	88.1714	89.8877	90.4987	90.8396	90.9718
24-Fa-1-B	74.7436	81.3160	82.6530	83.0299	83.4180	83.5278
24-Fa-2-B	79.1492	86.5038	88.1009	88.4148	88.9178	88.9652
24-Fa-3-B	81.0204	88.9467	90.6155	91.0120	91.4853	91.5412
24-Wi-1-B	64.2854	72.5121	74.2490	74.8733	75.1267	75.2564
24-Wi-2-B	70.1384	79.5569	81.5535	82.3365	82.6424	82.7343
24-Wi-3-B	73.8850	83.8727	86.0039	86.7355	87.0959	87.2065
Average (B)	76.6129	84.5493	86.1440	86.6732	87.0195	87.0968
Average (All)	73.8344	84.5405	87.1120	88.0786	88.6940	88.8714

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 3% to 5% of this bound on most instances. Thus, the valuation performance of each policy as a percentages 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.

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

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

	Number of Discretization Steps					
Statistic	5	20	50	100	250	500
Maximum	0.4509	0.7809	2.6716	9.0886	58.0302	259.5719
Minimum	0.4409	0.7599	2.5456	8.6537	54.9446	246.5089
Average	0.4448	0.7676	2.5949	8.8364	56.1950	251.7668
Standard Deviation	0.0020	0.0063	0.0368	0.1492	0.9428	3.8898

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 25. In the following comparative analysis we use the ADP policy obtained for $m_i = 500$.

No reoptimization. We now compare the performance of the ADP, I, LP, and LPN policies, 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 much lower valuations than the others.

Table 7 reports the valuation performance of the four policies on the 24-stage instances. Overall, on average the ADP, LP, and LPN policies are no more than 12%, 21%, and 30% suboptimal, respectively, and the intrinsic value of storage accounts for roughly at least 46% of its overall value. It is worthwhile to point out that the ADP policy performs better than the other policies on most of the instances except on the two instances 24-Sp-1-A and 24-Su-1-A, where the LP policy outperforms it. The ranking of these policies implied by their overall average valuation performances persists when one restricts attention to their average performances on the A and B instances: the ADP policy outperforms the LP policy, this policy performs better than the LPN policy, which in turn outperforms the I policy. However, the performance gaps between policies vary on these two sets of instances. For example, the valuation performances of the ADP and LP policies are much closer on the A instances than on the B instances. Moreover, the intrinsic value of storage is much smaller relative to the valuation of the other three policies on the B instances than on the A instances.

The valuation performance of the same policy on the A and B instances is also different. The ADP policy captures 4% less of the value of storage on average on the B instances than on the A instances. Both the I and LP policies are more sensitive to the change in correlation pattern from A to B, with these two policies losing roughly 25% and 14% of their values, respectively, on the B

Table 7: Valuation Performance of Different Policies without Reoptimization (Percent of the DUB Value for $m_i = 500$).

	Policy			
Instance	ADP	I	LP	LPN
24-Sp-1-A	88.8998	75.8412	89.8356	41.9483
24-Sp-2-A	92.0618	66.0404	88.3087	61.0629
24-Sp-3-A	94.2290	60.6551	87.9891	73.4078
24-Su-1-A	89.7317	76.9759	89.9225	63.4591
24-Su-2-A	93.8253	70.9391	90.1915	79.5999
24-Su-3-A	95.5464	66.3829	90.0094	86.2830
24-Fa-1-A	89.9783	72.5243	88.1506	56.6188
24-Fa-2-A	93.9226	68.0729	90.0192	75.3341
24-Fa-3-A	95.8435	65.1235	90.4921	84.5419
24-Wi-1-A	76.0283	35.0970	73.7955	45.0298
24-Wi-2-A	84.5481	30.2395	78.8480	62.0403
24-Wi-3-A	89.0798	27.7113	81.2765	72.6505
Average (A)	91.8827	62.8694	87.9272	70.4165
24-Sp-1-B	81.1559	53.7550	74.3954	51.7797
24-Sp-2-B	86.0877	39.6010	74.7495	66.4230
24-Sp-3-B	89.2786	32.6767	75.0192	76.7502
24-Su-1-B	83.1865	55.4023	75.8511	54.1480
24-Su-2-B	88.4765	44.6067	76.7641	68.3701
24-Su-3-B	90.9718	37.6708	77.2581	78.4331
24-Fa-1-B	83.5278	49.8640	74.0365	65.2256
24-Fa-2-B	88.9652	42.9989	76.1628	76.8748
24-Fa-3-B	91.5412	38.4787	77.1518	83.3749
24-Wi-1-B	75.2564	19.3243	59.3449	55.4773
24-Wi-2-B	82.7343	14.2546	64.1636	66.8480
24-Wi-3-B	87.2065	11.8242	68.1079	75.6047
Average (B)	87.0968	37.0165	73.9295	71.0862
Average (All)	88.8714	46.6024	79.1197	70.8379

Table 8: Statistics on the Cpu Seconds Needed to Compute the Different Policies without Reoptimization.

	Policy			
Statistic	ADP	I	LP	LPN
Maximum	259.5719	0.5329	0.6709	0.6979
Minimum	246.5089	0.4049	0.4589	0.4679
Average	251.7668	0.4483	0.5318	0.5553
Standard Deviation	3.8898	0.0440	0.0756	0.0847

instances. Surprisingly, the valuation performance of the LPN policy improves on the B instances, even though the improvement is marginal.

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-A, 24-Sp-2-A, and 24-Sp-3-A is $(94.2290-88.8998)\% = 5.3292\%$. 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 8%, 12%, and 25%, respectively. It appears that the ADP, LP, and LPN policies are able to capture a larger share of the value of storage for instances with higher injection/withdrawal capacities. In contrast, the intrinsic value becomes smaller when the injection/withdrawal capacities increase.

Table 8 reports statistics of the cpu times needed to compute and evaluate the different policies. 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. Table 9 reports the valuation performance of the RADP, RI, RLP, and RLPN policies. Notice that 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-A (94.3848%), 24-Sp-1-B (95.7427%), 24-Fa-1-B (95.8931%), and 24-Wi-1-B (92.9425%). Moreover, the RADP policy is the best policy on all the instances, with the exception of 24-Fa-3-A, where it is marginally outperformed by RI and RLP. The ranking of the four policies based on their average performances on the A, B, and all the instances is as follows: RADP is marginally better than RI and RLP, which in turn perform significantly better than RLPN. In addition, compared to the A instances, on average the four policies capture roughly 1% less of the value of storage on the B instances.

RADP is the least sensitive policy to changes in injection/withdrawal capacities, with roughly an average range of 2.4%, whereas the ranges of the RI and RLP policies are 2.8% and 2.7%, respectively, and the range of the RLPN policy is a whooping 26.1%.

Table 10 reports summary statistics of the cpu times required to evaluate the four policies. 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

Table 9: Valuation Performance of Different Policies with Reoptimization (Percent of the DUB Value for $m_i = 500$).

Instance	Policy			
	RADP	RI	RLP	RLPN
24-Sp-1-A	98.1283	97.9532	98.0819	84.1250
24-Sp-2-A	98.6214	98.5379	98.5612	93.7341
24-Sp-3-A	99.0584	98.9865	98.9563	97.8605
24-Su-1-A	98.0466	97.9287	97.9821	70.9507
24-Su-2-A	99.0588	99.0345	99.0489	91.3886
24-Su-3-A	99.2635	99.2570	99.2420	97.5410
24-Fa-1-A	98.0214	97.9963	97.9981	65.7532
24-Fa-2-A	99.2581	99.1847	99.1987	90.3281
24-Fa-3-A	99.4782	99.4825	99.4930	97.6554
24-Wi-1-A	94.3848	92.4686	92.5703	69.8354
24-Wi-2-A	96.7929	95.8081	95.7208	89.9249
24-Wi-3-A	97.4626	96.5665	96.4828	95.0783
Average (A)	98.5313	98.3117	98.3204	89.2468
24-Sp-1-B	95.7427	95.4351	95.6071	78.8791
24-Sp-2-B	97.2937	97.0411	97.2025	91.4707
24-Sp-3-B	98.2328	98.0933	98.1573	96.9954
24-Su-1-B	96.1216	95.8790	95.9545	69.8644
24-Su-2-B	98.0222	97.8702	97.9110	90.4271
24-Su-3-B	98.7871	98.7100	98.6900	97.2250
24-Fa-1-B	95.8931	95.6266	95.7730	64.5791
24-Fa-2-B	98.0432	97.8708	97.9083	89.5015
24-Fa-3-B	98.8130	98.7363	98.7206	97.1631
24-Wi-1-B	92.9425	90.5840	90.7256	63.0987
24-Wi-2-B	96.3672	94.6914	94.8088	88.4249
24-Wi-3-B	97.7601	96.4240	96.4470	96.1962
Average (B)	97.4872	97.0458	97.1077	88.6969
Average (All)	97.8743	97.5152	97.5574	88.9008

Table 10: Statistics on the Cpu Seconds Needed to Compute the Different Policies with Reoptimization.

Statistic	Policy			
	RADP	RI	RLP	RLPN
Maximum	1493.3400	77.9432	299.4240	246.3460
Minimum	1396.6400	74.4247	260.2010	229.8540
Average	1436.9396	75.7675	278.5315	239.1764
Standard Deviation	33.6591	0.9209	11.4705	3.9614

Table 11: “Raw” Values of Different Models.

Instance	Model ADP	I	LP	LPN
24-Sp-1-A	87.9214	75.4455	87.3844	123.6068
24-Sp-2-A	91.6286	65.8096	83.7874	102.2805
24-Sp-3-A	94.2054	60.3996	81.9617	93.5256
24-Su-1-A	89.0709	76.5631	87.9446	126.9817
24-Su-2-A	93.3715	70.6234	86.1682	103.5973
24-Su-3-A	95.2498	66.1533	84.7448	95.4696
24-Fa-1-A	89.0830	72.0304	85.8594	132.2520
24-Fa-2-A	93.2162	67.6262	85.6901	106.1467
24-Fa-3-A	95.2694	64.7308	85.1014	96.9512
24-Wi-1-A	75.4652	34.7317	70.4607	140.7547
24-Wi-2-A	84.0647	30.0456	72.8241	107.1115
24-Wi-3-A	88.8430	27.7043	73.6397	93.7250
Average (A)	91.3762	62.5516	83.3942	106.9337
24-Sp-1-B	100.2373	53.5398	72.8440	201.9940
24-Sp-2-B	100.3314	39.5861	72.1232	142.6351
24-Sp-3-B	99.8384	32.6781	72.0294	117.7895
24-Su-1-B	101.1027	55.1055	74.2714	205.3183
24-Su-2-B	101.0029	44.4443	74.1690	142.0520
24-Su-3-B	100.4346	37.6089	74.0127	117.5480
24-Fa-1-B	101.8168	49.4419	72.3231	211.3072
24-Fa-2-B	101.2232	42.6638	73.6322	142.9084
24-Fa-3-B	100.4212	38.1600	74.2180	117.0073
24-Wi-1-B	101.8033	19.1664	57.5127	239.0505
24-Wi-2-B	101.8248	14.2213	61.5826	157.1782
24-Wi-3-B	100.7893	11.8941	64.9076	123.8814
Average (B)	100.7725	36.8607	71.3270	147.8873
Average (All)	97.2885	46.3866	75.8014	132.7022

policy that can be evaluated the fastest is RI; this takes roughly four times less time than evaluating the RLP and RLPN policies, which in turn takes five to six times less time than evaluating 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. Table 11 reports these raw values. Based on the overall average, the ADP model seems to slightly underestimate the value of storage. This underestimation is much more marked for models I and LP. Unlike these three models, model LPN grossly overestimates the value of storage. Recall from §3 that the values of the I and LP models are lower bounds

Table 12: Statistics on the Cpu Seconds Needed to Compute the "Raw" Values of Different Models.

	Model			
Statistic	ADP	I	LP	LPN
Maximum	258.6550	0.0020	0.0050	0.0030
Minimum	245.6140	0.0000	0.0020	0.0020
Average	250.8532	0.0010	0.0038	0.0024
Standard Deviation	3.8848	0.0006	0.0008	0.0005

on the optimal value of storage. Table 11 suggests that model LPN yields values that are higher than the optimal value of storage, while those of model ADP are sometimes higher and sometimes lower than this value. These results imply that although model ADP generates values that are at least 97% of the optimal storage value on average, it seems difficult for this model to consistently produce near optimal storage valuations.

Table 12 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 considerable 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 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 appear 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 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 is slower. 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 efficiency.

These results have immediate relevance for natural gas storage traders. Although not discussed in this paper, our findings remain substantially similar when we remove the seasonality from the natural gar forward curves that we employ in our instances. This suggests that our results 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.

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. 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, 2000.

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

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

[12] 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.

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

[14] A. Eydeland and K. Wolyniec. *Energy and Power Risk Management.* John Wiley & Sons, Inc., Hoboken, NJ, USA, 2003.

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

[16] 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.

[17] J. Gray and P. Khandelwal. Realistic gas storage models II: Trading strategies. *Commodities Now*, September 2004.

[18] J. Gray and P. Khandelwal. Towards a realistic gas storage model. *Commodities Now*, June 2004.

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

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

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

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

[23] 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.

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

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

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

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

[28] N. Secomandi. Optimal commodity trading with a capacitated storage asset. Tepper Working Paper 2005-E59, Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA, USA, 2008.

[29] 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.

[30] 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.

[31] N. L. Stokey and R. E. Lucas. *Recursive Methods in Economic Dynamics.* Harvard University Press, 1989.

[32] M. Thompson, M. Davison, and H. Rasmussen. Natural gas storage valuation and optimization: A real options application. University of Western Ontario, Department of Applied Mathematics, London, Ontario, Canada, 2003.

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

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

[35] 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.

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

Online Appendix

A. Detailed Market Data Used in the Numerical Experiments

The tables reported in this section include the details of the market data used in the numerical experiments. Tables 13-16 are associated with Figures 2-3 as follows:

- Tables 13-14: The two top and bottom panels of Figure 2, respectively,
- Tables 15-16: Figure 3.

Table 13: 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 14: 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 15: Partial Correlation Matrix A.

	Maturity					
Maturity	1	2	3	...	22	23
1	1.00	0.98	0.96	...	0.58	0.56
2	0.98	1.00	0.98	...	0.60	0.58
3	0.96	0.98	1.00	...	0.62	0.60
...	...	...	...	...	...	...
22	0.58	0.60	0.62	...	1.00	0.98
23	0.56	0.58	0.60	...	0.98	1.00

Table 16: Partial Correlation Matrix B.

	Maturity					
Maturity	1	2	3	...	22	23
1	1.00	0.86	0.84	...	0.46	0.44
2	0.86	1.00	0.86	...	0.48	0.46
3	0.84	0.86	1.00	...	0.50	0.48
...	...	...	...	...	...	...
22	0.46	0.48	0.50	...	1.00	0.86
23	0.44	0.46	0.48	...	0.86	1.00