Analysis and Enhancement of Practice-based Policies for the Real Option Management of Commodity Storage Assets

Nicola Secomandi
Tepper School of Business, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213-3890, USA
ns7@andrew.cmu.edu

Tepper Working Paper 2011-E11
June 2011; Revised: July 2012; January 2014

Abstract

The real option management of commodity storage assets is an important practical problem. Practitioners heuristically solve the resulting stochastic optimization model using the rolling intrinsic (RI) and rolling basket of spread options (RSO) policies. Combined with Monte Carlo simulation, these policies typically yield near optimal lower bound estimates on the value of storage. This paper provides novel structural and numerical support for the use of the RI and RSO policies, and enhances them by developing a simple and effective dual upper bound to be used in conjunction with these policies. Moreover, this work emphasizes the superiority of the RI policy over the RSO policy and proposes a variant of the RSO policy that, on the considered instances, slightly improves on the average performance of the RSO policy but yields a more substantial improvement when the suboptimality of this policy is more pronounced.

1. Introduction

Storable commodity industries include storage assets embedded in physical markets for the commodity and financial markets for commodity derivatives. These markets can be fairly competitive, as exemplified by the natural gas industries in North America and parts of Europe and the United Kingdom. In these markets, merchants rent storage capacity from the owners of storage facilities and use it to support intertemporal commodity trading. Merchants have adopted real option approaches to obtain the market value of storage assets and operating policies that support their trading activity (Maragos 2002).

In general, the optimal real option management of commodity storage assets gives rise to an intractable stochastic optimization model. This intractability is due to the continuous nature of commodity prices in models of their evolutions used in practice and the high dimensionality of the resulting Markov decision process (MDP). In particular, the use of a restricted class of low dimensional price evolution models, e.g., the one-factor model of Jaillet et al. (2004) and the two-factor model of Schwartz and Smith (2000), coupled with discretization of prices *does* yield a tractable MDP, because an optimal policy for this MDP can be obtained by numerically solving a low dimensional stochastic dynamic program (see, e.g., Manoliu 2004, Secomandi 2010, Parsons 2013, Wu et al. 2012). In contrast, the MDP that ensues when using general versions of these price evolution models, see, e.g., Clewlow and Strickland (2000, Chapter 8), is intractable, because its states include the entire commodity forward curve and this complication makes stochastic dynamic programming computationally infeasible. Practitioners thus use heuristics to manage these assets (Maragos 2002, Eydeland and Wolyniec 2003, pp. 351-367, Gray and Khandelwal 2004a,b, and Lai et al. 2010).

The rolling intrinsic (RI) and rolling basket of spread options (RSO) policies are two heuristics widespread in natural gas storage practice. These policies are based on the sequential *reoptimization* (“rolling”) of a dynamic program that models the deterministic version of the problem and a linear program whose objective function includes the values of options on futures price spreads, respectively (see Lai et al. 2010 and references therein). Software vendors, such as FEA (2007), KYOS (2009), and Lacima (2010), have included versions of these policies in their offerings.

Lai et al. (2010), Secomandi (2010), and Wu et al. (2012) provide numerical evidence for the near optimality of the RI and RSO policies in natural gas storage (Secomandi 2010 and Wu et al. 2012 focus on the RI policy). In particular, they bring to light the key role reoptimization has in making these policies near optimal. However, from a theoretical perspective the role played by reoptimization in the context of these policies is not well understood.

This paper provides structural support for the benefit of reoptimization when using the RI and RSO policies (the support is weaker for the RSO policy). These reoptimization findings complement the numerical work of Lai et al. (2010), Secomandi (2010), and Wu et al. (2012), and make precise the discussion in Gray and Khandelwal (2004a). A key step in this analysis is the reformulation of the deterministic dynamic program used by the RI policy as a linear program based on futures price spreads. Because replacing these price spreads with options on these spreads yields the linear program used by the RSO policy, this reformulation also provides a common perspective on the RI and RSO policies.

It is known that an optimal policy has a double basestock target structure (Secomandi 2010, Secomandi et al. 2012). The RI policy obviously has this structure, but it is not known whether the RSO policy also satisfies this property. This paper resolves this issue in the affirmative. This finding thus offers some structural justification also for the use of the RSO policy.

This paper conducts a numerical analysis of the RI and RSO policies based on extended and more up-to-date versions of the natural gas storage instances of Lai et al. (2010). This analysis confirms the known critical role of reoptimization for obtaining near optimal performance when using the RI and RSO policies. However, it also emphasizes the superiority of the RI policy over the RSO policy by pointing out that the RI policy performs substantially better than the RSO policy when the storage asset is fast, that is, the inventory adjustment capacity is equal to the maximum storage space. This case, not considered by Lai et al. (2010), tends to arise in practice when storing natural gas in a salt dome storage facility and trading natural gas on the liquid monthly bid-week spot market (Eydeland and Wolyniec 2003, p. 4).

Some practitioners might prefer the RSO policy to the RI policy, because hedging parameters (the so-called “Greeks”) can be directly obtained from a basket of spread options while they are not immediately available from the solution of the model that computes the intrinsic value of a commodity storage asset (TIMERA ENERGY 2013; however see Secomandi et al. 2012 for an approach to estimate the “deltas” of the value of a commodity storage asset when using a heuristic policy, such as the RI policy). This paper thus proposes a variant of the RSO policy based on reoptimizing a linear program that maximizes the average of a lower bound and an upper bound on the value of a basket of spread options policy, while the linear program used by the RSO policy optimizes only this lower bound. On the considered instances, this variant of the RSO policy yields a slight improvement on the average performance of the RSO policy but the improvement is more considerable when the suboptimality of this policy is more pronounced.

Combined with Monte Carlo simulation, the RI and (modified) RSO policies yield lower bound estimates on the value of storage. Lai et al. (2010) and Nadarajah et al. (2013) use the approach presented by Brown et al. (2010) to estimate dual upper bounds on this value, instantiating the so called dual penalties from the value functions of approximate dynamic programs (ADPs) that share the concavity in inventory of the optimal value function.

This paper proposes a simpler approach to obtain dual penalties based on the (optimal) value function of a simplified version of the problem: There are no frictions, that is, inventory adjustment costs and losses, and the storage asset is fast. This value function, which is linear in inventory, can be computed in essentially closed form using the exchange option formula of Margrabe (1978) when employing common commodity price evolution models. Using the value function of this restricted case yields two versions of dual penalties that can be used also in the general case: One version that reduces to spot and prompt month futures *price spreads* multiplied by the inventory level that results from performing an inventory trade, and another version that adds an exchange option based term to these price spread terms. The two resulting dual upper bounds are labeled as the DS and DEO upper bounds (here S and EO abbreviate spread and exchange option, respectively). The proposed dual upper bounds can be estimated by embedding within Monte Carlo simulation simple variants of the optimization models used to obtain the RSO and RSO policies. Hence, this approach is easy to use in conjunction with these policies, an appealing feature for both the users of these policies and the vendors of the commercial software that implements these policies.

The DS and DEO upper bounds are identical in theory, but their sample average Monte Carlo estimators can have different variances; e.g., the DEO sample average estimator has zero variance when the asset is fast and frictionless, which follows from Brown et al. (2010, Theorem 2.3), whereas the estimator of the DS does not. Moreover, these dual upper bounds are shown to be no weaker than the upper bound that corresponds to the value function function of the fast and frictionless asset (the EO upper bound).

Applied to the extended natural gas instances, the DS and DEO upper bounds are highly competitive with the best dual upper bound of Nadarajah et al. (2013), which also dominates the one of Lai et al. (2010). Given the same number of Monte Carlo samples, estimating the DS and DEO dual upper bounds is considerably faster than estimating, with comparable or improved precision, the best dual upper bound of Nadarajah et al. (2013). Moreover, the estimation of the DS upper bound is faster than the estimation of the DEO upper bound, which however is more precise. The estimated DS and DEO upper bounds are also considerably tighter than the computed EO upper bounds. The suggested enhancement of the RI and RSO policies thus has immediate practical relevance. The observed performance of the proposed dual upper bounds is remarkable given the simplicity of their dual penalties and the dismal performance of the EO upper bound.

The reoptimization analysis of this paper is related to the work of Secomandi (2008), but deals with a different context, hence considering dissimilar heuristics, and is self contained. The basestock target characterization of the RSO policy appears new. The theoretical analysis of the relationship between the EO upper bound and the DEO and DS upper bounds is similar to a result of Brown and Smith (2011) developed in the context of portfolio optimization with transaction costs, and its recent extension to more general settings by Brown and Smith (2013), but is self contained. Despite being less general than the approach of Brown and Smith (2011, 2013), the proposed dual upper bounds are simpler to implement, because, by construction, do not require the linearization step that is present in the gradient approach of these authors.

The real option literature on energy and commodity applications (Smith and McCardle 1999, Clewlow and Strickland 2000, Eydeland and Wolyniec 2003, Geman 2005) includes several papers on natural gas storage (Manoliu 2004, Chen and Forsyth 2007, Boogert and de Jong 2008, 2011/12, Thompson et al. 2009, Carmona and Ludkovski 2010, Lai et al. 2010, Secomandi 2010, 2011, Bjerksund et al. 2011, Lai et al. 2011, Secomandi et al. 2012, Nadarajah et al. 2013, Wu et al. 2012, Thompson 2012, Mazières and Boogert 2013, Parsons 2013, Ware 2013). This paper conducts a novel analysis of two heuristic policies commonly used in practice to manage commodity storage assets, proposes a variant of one of these policies, and provides a new approach for dual upper bound estimation.

This paper proceeds by formulating a stochastic optimization model of the real option management of commodity storage assets in §2. Section 3 introduces heuristic policies. Section 4 performs a theoretical analysis of the benefit of reoptimization and the structure of the RSO policy. Section 5 develops the DEO and DS upper bounds. Section 6 conducts a numerical analysis of the performance of the considered heuristic policies and these upper bounds, including the proposed variant of the RSO policy, which is introduced in this section. Section 7 concludes. Supporting material is included in Appendices A-C.

2. Stochastic Optimization Model

This section, in part based on Lai et al. (2010, §2), formulates the commodity storage asset management problem both as an MDP and a stochastic dynamic program, and presents the known structure of an optimal policy.

The set of futures contract maturity labels is $\mathcal{N} := \{0, \ldots, N-1\}$, with $N \geq 1$ an integer. This set is also the stage set. Commodity trading decisions are made at each of a finite number of times T_n, with $n \in \mathcal{N}$. An action $a \in \Re$ represents the change in the storage asset inventory level between two successive stages. A negative action corresponds to a buy-and-inject decision and gives rise to a negative cash flow; a positive action corresponds to a withdraw-and-sell decision ("inject" and "withdraw" are specific to natural gas storage and could be replaced with the more generic "increase" and "decrease," respectively). The zero action is the do-nothing decision.

The cash flow of a nonzero action taken in stage n occurs at time T_n, while the corresponding inventory adjustment is executed during the time interval in between times T_n and T_{n+1}. A buy-and-inject decision incurs both the purchase cost $\phi^I s$, where $\phi^I \geq 1$ models the inventory injection loss and $s \in \Re_+$ is the spot price, and the marginal injection cost c^I. A withdraw-and-sell decision earns the sale spot price $\phi^W s$, where $\phi^W \in (0, 1]$ models the inventory withdrawal loss, minus the marginal withdrawal cost c^W. Thus, the buy-and-inject spot price is $s^I := \phi^I s + c^I$ and the withdraw-and-sell spot price is $s^W := \phi^W s - c^W$, which is negative when $s \in (0, c^W/\phi^W)$. Given an action a and a spot price s, the per stage cash flow function $p(a, s)$ is equal to $s^I a$ if $a < 0$, 0 if $a = 0$, and $s^W a$ if $a > 0$.

The minimum inventory level is normalized to 0 and the maximum inventory level is $\overline{x} \in \Re_+$. The feasible inventory set is $\mathcal{X} := [0, \overline{x}]$. The injection and withdrawal capacity per stage are $C^I < 0$ and $C^W > 0$, respectively. Let $\cdot \wedge \cdot := \min(\cdot, \cdot)$ and $\cdot \vee \cdot := \max(\cdot, \cdot)$. Given the inventory level x, the feasible injection and withdrawal sets are $\mathcal{A}^I(x) := [C^I \vee (x - \overline{x}), 0]$ and $\mathcal{A}^W(x) := [0, x \wedge C^W]$, respectively, and the feasible action set is $\mathcal{A}(x) := \mathcal{A}^I(x) \cup \mathcal{A}^W(x)$.

The time T_n price of the maturity $T_m \geq T_n$ futures is $F_{n,m}$. The time T_n forward curve is $\boldsymbol{F}_n := (F_{n,m}, m \in \mathcal{N}, n \leq m)$. The time T_n forward curve exclusive of the spot price $s_n \equiv F_{n,n}$ is $\boldsymbol{F}'_n := (F_{n,m}, m \in \mathcal{N}, n < m)$. By convention $F_{n,N} := 0$, $\forall n \in \mathcal{N}$, $\boldsymbol{F}_N := 0$, and $\boldsymbol{F}'_N := 0$.

Denote by Π the set of feasible inventory trading policies. The decision rule of policy $\pi \in \Pi$ in stage n is $A_n^\pi(x, \boldsymbol{F}_n)$. Let x_n^π be the inventory level reached in stage n by such a policy π. Denote by δ the per stage risk free discount factor and by $\mathbb{E}$ risk neutral expectation (Shreve 2004, Chapter 5). The notation $\tilde{\cdot}$ indicates a random quantity. An optimal policy can be obtained by solving the following MDP:

$$\max_{\pi \in \Pi} \sum_{n \in \mathcal{N}} \delta^n \mathbb{E}[p(A_n^\pi(\tilde{x}_n^\pi, \tilde{\boldsymbol{F}}_n), \tilde{s}_n) | x_0, \boldsymbol{F}_0]. \tag{1}$$

Model (1) is now formulated as a stochastic dynamic program. Although this formulation is in general intractable (as implied by the work of Charnes et al. 1966), it is useful for the ensuing analysis to formulate this model and summarize some of its known structural results.

Denote by $V_n(x_n, \boldsymbol{F}_n)$ the optimal value function of this stochastic dynamic program in stage n and state $(x_n, \boldsymbol{F}_n)$, with $V_N(x_N, \boldsymbol{F}_N) := 0$. Define as $W_n(x, \boldsymbol{F}'_n) := \delta \mathbb{E}\left[V_{n+1}(x, \tilde{\boldsymbol{F}}_{n+1}) | \boldsymbol{F}'_n\right]$ the optimal continuation-value function for all $n \in \mathcal{N}$ and $(x, \boldsymbol{F}'_n) \in \mathcal{X} \times \Re_+^{N-n-1}$ (the risk neutral distribution of $\tilde{\boldsymbol{F}}_{n+1}$ only depends on $\boldsymbol{F}'_n$ by assumption). The Bellman equation of this stochastic dynamic program, for each stage $n \in \mathcal{N}$ and state $(x_n, \boldsymbol{F}_n) \in \mathcal{X} \times \Re_+^{N-n}$, is

$$V_n(x_n, \boldsymbol{F}_n) = \max_{a \in \mathcal{A}(x_n)} p(a, s_n) + W_n(x_n - a, \boldsymbol{F}'_n). \tag{2}$$

Property 1 summarizes known structural results about model (2). Part (b) of Property 1 depends on Assumption 1.

Assumption 1 (Lot size; Secomandi et al. 2012)**.** The capacity limits C^I and C^W and the maximum inventory level $\overline{x}$ are integer multiples of a positive real number. The largest common factor of C^I, C^W, and $\overline{x}$ is denoted by Q.

Property 1 (Basestock structure; Secomandi et al. 2012)**.** *(a) For every stage $n \in \mathcal{N}$ of model (2), the function $V_n(x_n, \boldsymbol{F}_n)$ is concave in inventory $x_n \in \mathcal{X}$ for each given forward curve $\boldsymbol{F}_n \in \Re_+^{N-n}$, and an optimal decision rule in this stage is characterized by two basestock targets, $\underline{b}_n(\boldsymbol{F}_n)$, $\overline{b}_n(\boldsymbol{F}_n) \in \mathcal{X}$, such that $\underline{b}_n(\boldsymbol{F}_n) \leq \overline{b}_n(\boldsymbol{F}_n)$ and returns $C^I \vee [x_n - \underline{b}_n(\boldsymbol{F}_n)]$ if $x_n \in [0, \underline{b}_n(\boldsymbol{F}_n))$, 0 if $x_n \in [\underline{b}_n(\boldsymbol{F}_n), \overline{b}_n(\boldsymbol{F}_n)]$, and $C^W \wedge [x_n - \overline{b}_n(\boldsymbol{F}_n)]$ if $x_n \in (\overline{b}_n(\boldsymbol{F}_n), \overline{x}]$. (b) Moreover, suppose Assumption 1 holds. For each given forward curve $\boldsymbol{F}_n \in \Re_+^{N-n}$, the function $V_n(x, \boldsymbol{F}_n)$ is piecewise linear continuous in inventory $x \in \mathcal{X}$ with break points in set $\mathcal{Q} := \{0, Q, 2Q, \ldots, \overline{x}\}$, and the basestock targets $\underline{b}_n(\boldsymbol{F}_n)$ and $\overline{b}_n(\boldsymbol{F}_n)$ can be taken to be in set $\mathcal{Q}$.*

3. Heuristic Policies

This section presents the I and SO models and policies (§§3.1-3.2) and the rolling versions of these policies that arise from the sequential reoptimization of these models (§3.3).

3.1 The I Dynamic Program and Policy

The I dynamic program is derived from the stochastic dynamic program (2) by removing the uncertainty in the evolution of the forward curve. It is thus a deterministic dynamic program. Denote by $V_n^I(x_n, \boldsymbol{F}_0)$ the intrinsic value function in stage n and state x_n given $\boldsymbol{F}_0$. Define $V_N^I(x_N, \boldsymbol{F}_0) := 0$ for all $x \in \mathcal{X}$. The I dynamic program for all stages $n \in \mathcal{N}$ and states $x_n \in \mathcal{X}$ is

$$V_n^I(x_n, \boldsymbol{F}_0) = \max_{a \in \mathcal{A}(x_n)} p(a, F_{0,n}) + \delta V_{n+1}^I(x_n - a, \boldsymbol{F}_0). \tag{3}$$

Model (3) yields the value of storage due to seasonality at time T_0 given x_0 and $\boldsymbol{F}_0$, $V_0^I(x_0, \boldsymbol{F}_0)$ (Lai et al. 2010, §3.2). This value can be locked in at time T_0 by trading in the forward market at this time according to the optimal policy of model (3), that is, the I policy. This policy can be efficiently computed when Assumption 1 holds, because in this case the dynamic program (3) has a discrete state space. Moreover, the I policy satisfies Property 1 with the basestock targets in each stage n depending on $\boldsymbol{F}_0$ rather than $\boldsymbol{F}_n$, which facilitates the computation of this policy (see Secomandi 2010, Theorem 1, Lai et al. 2010, Theorem 1).

3.2 The SO Linear Program and Policy

To derive the SO linear program, it is useful to formulate the I dynamic program, (3), as a linear program. Define as $F_{0,n}^I := \phi^I F_{0,n} + c^I$ and $F_{0,n}^W := \phi^W F_{0,n} - c^W$ the time T_0 buy-and-inject and withdraw-and-sell, respectively, discounted futures prices for maturity T_n. Denote by u_n and w_n the buy-and-inject and withdraw-and-sell, respectively, decision variables for maturity T_n. The equivalent linear programming formulation of the I dynamic program, (3), is

$$\max \sum_{n \in \mathcal{N}} \delta^n F_{0,n}^W w_n - \sum_{n \in \mathcal{N}} \delta^n F_{0,n}^I u_n \tag{4}$$

$$\text{s.t.} \sum_{m=0}^{n-1} (u_m - w_m) \geq -x_0, \ \forall n \in \mathcal{N} \cup \{N\} \setminus \{0\}, \tag{5}$$

$$\sum_{m=0}^{n-1} (u_m - w_m) \leq \overline{x} - x_0, \ \forall n \in \mathcal{N} \cup \{N\} \setminus \{0\}, \tag{6}$$

$$u_n \leq -C^I, \ \forall n \in \mathcal{N}, \tag{7}$$

$$w_n \leq C^W, \ \forall n \in \mathcal{N}, \tag{8}$$

$$u_n \geq 0, \ \forall n \in \mathcal{N}, \tag{9}$$

$$w_n \geq 0, \ \forall n \in \mathcal{N}. \tag{10}$$

The objective function (4) maximizes the time T_0 value of the total cash flows collected between times T_0 and T_{N-1}. Constraints (5)-(6) impose minimum and maximum inventory restrictions, respectively. Constraints (7)-(8) enforce the injection and withdrawal capacity limits, respectively. Constraints (9)-(10) are the nonnegativity conditions on the decision variables. It is clear that one could set u_{N-1} equal to 0, as purchasing and injecting a positive amount of inventory at time T_{N-1} serves no use. The claimed equivalence of the I dynamic program, (3), and the linear program (4)-(10) holds because, as it is easy to verify, in the latter model simultaneous purchase-and-inject and withdraw-and-sell trades are suboptimal for every trading date.

In the linear program (4)-(10), pair a time T_m buy-and-inject trade with a time $T_n > T_m$ withdraw-and-sell trade, and denote by $q_{m,n}$ the corresponding notional amount. Also, denote by

z_n an amount of commodity withdrawn and sold at time T_n from the initial inventory x_0. It thus holds that

$$u_n = \sum_{m=n+1}^{N-1} q_{n,m}, \ \forall n \in \mathcal{N}, \tag{11}$$

$$w_n = \sum_{m=0}^{n-1} q_{m,n} + z_n, \ \forall n \in \mathcal{N}. \tag{12}$$

Substituting (11) and (12) into (5)-(10) and rearranging yields

$$\sum_{m=0}^{n-1} \sum_{l=n}^{N-1} q_{m,l} - \sum_{m=0}^{n-1} z_m \geq -x_0, \ \forall n \in \mathcal{N} \cup \{N\} \setminus \{0\}, \tag{13}$$

$$\sum_{m=0}^{n-1} \sum_{l=n}^{N-1} q_{m,l} - \sum_{m=0}^{n-1} z_m \leq \overline{x} - x_0, \ \forall \in \mathcal{N} \cup \{N\} \setminus \{0\}, \tag{14}$$

$$\sum_{m=n+1}^{N-1} q_{n,m} \leq -C^I, \ \forall n \in \mathcal{N}, \tag{15}$$

$$\sum_{m=0}^{n-1} q_{m,n} + z_n \leq C^W, \ \forall n \in \mathcal{N}, \tag{16}$$

$$q_{n,m} \geq 0, \ \forall n \in \mathcal{N} \setminus \{N-1\}, m \in \mathcal{N}, m > n, \tag{17}$$

$$z_n \geq 0, \ \forall n \in \mathcal{N}. \tag{18}$$

Define the vectors $\boldsymbol{q} := (q_{n,m}, n \in \mathcal{N} \setminus \{N-1\}, m \in \mathcal{N}, m > n)$ and $\boldsymbol{z} := (z_n, n \in \mathcal{N})$, and the polyhedron $\mathcal{P} := \left\{(\boldsymbol{q}, \boldsymbol{z}) \in \Re^{(N^2+N)/2} \text{ s.t. (13)-(18)}\right\}$. Using (11) and (12) to express (4) in terms of $\boldsymbol{q}$ and $\boldsymbol{z}$ yields the following I linear program:

$$\max_{(\boldsymbol{q},\boldsymbol{z}) \in \mathcal{P}} \sum_{n=0}^{N-1} \delta^n F_{0,n}^W z_n + \sum_{n=0}^{N-2} \sum_{m=n+1}^{N-1} \left(\delta^m F_{0,m}^W - \delta^n F_{0,n}^I\right) q_{n,m}. \tag{19}$$

The I linear program (19) is equivalent to the I linear program (4)-(10) at optimality (at optimality, because the latter linear program does not admit purchases with no subsequent sales).

Define the time T_l value of a spread option with payoff equal to the positive part of the spread $\delta^{n-m} F_{m,n}^W - s_m^I$ as $S_{l,m,n}(\boldsymbol{F}_l) := \delta^{m-l} \mathbb{E}\left[\left(\delta^{n-m} \tilde{F}_{m,n}^W - \tilde{s}_m^I\right)^+ \mid \boldsymbol{F}_l\right]$. Replacing $\delta^n F_{0,n}^W - \delta^m F_{0,m}^I$ with $S_{0,m,n}(\boldsymbol{F}_0)$ in (19) yields the SO linear program

$$U_0^{SO}(x_0, \boldsymbol{F}_0) := \max_{(\boldsymbol{q},\boldsymbol{z}) \in \mathcal{P}} \sum_{n=0}^{N-1} \delta^n F_{0,n}^W z_n + \sum_{n=0}^{N-2} \sum_{m=n+1}^{N-1} S_{0,n,m}(\boldsymbol{F}_0) q_{n,m}. \tag{20}$$

The SO policy is derived from an optimal solution to this linear program in a manner analogous to the description of the LP policy in Lai et al. (2010, §3.1). Specifically, the action performed in a given stage and state by this policy is the net of the total scheduled injections and withdrawals for this stage and state. The total scheduled injections are the ones corresponding to spread options that expire in the money in this stage and state. The total scheduled withdrawals are the ones associated with spot/forward sales for this stage and previously exercised spread options for which the withdrawal leg occurs in this stage.

With frictions, the quantity $U_0^{SO}(x_0, \boldsymbol{F}_0)$ is a lower bound on the value of the SO policy, denoted as $V_0^{SO}(x_0, \boldsymbol{F}_0)$:

$$U_0^{SO}(x_0, \boldsymbol{F}_0) \leq V_0^{SO}(x_0, \boldsymbol{F}_0). \tag{21}$$

This inequality follows from an easy extension of Proposition 1 in Lai et al. (2010). Intuitively, inequality (21) holds because the optimal objective function of the SO linear program "double counts" the costs and fuel losses of simultaneously injected and withdrawn amounts, whereas the SO policy nets out these amounts to obtain a single decision. When there are no frictions no double counting occurs and (21) holds as an equality.

Different from the spread option based linear program in Lai et al. (2010), the SO linear program (20) includes the forward sales for times T_1 through T_{N-1}, in addition to the time T_0 spot sale. This inclusion allows comparing the values of the I and SO policies in Lemma 2 in Appendix A.

The value of the SO policy can be estimated by Monte Carlo simulation of the forward curve, given a stochastic model thereof. With no frictions, the spread options in (20) reduce to exchange options, that is, spread options with zero strike price, and this value can be computed in essentially closed form via Margrabe (1978) exchange option formula when using common reduced form forward curve evolution models (e.g., the model (55)-(56) used in §6).

3.3 The RI and RSO Policies

The RI policy and the RSO policy, respectively, arise from using models (3) (or, equivalently, (4)-(10)) and (20) in a control algorithm sense with re-solving (reoptimization; Secomandi 2008). Specifically, in a given stage and state, the action of the RI policy is an optimal action for this stage and state determined by reformulating and reoptimizing model (3) accordingly. Or, equivalently, it can easily be obtained from the part of an optimal solution that pertains to this stage in the reformulated and reoptimized linear program (4)-(10). The action of the RSO policy is determined in an analogous manner by reformulating and reoptimizing the linear program (20) accordingly. The values of the RI and RSO policies, denoted by $V_0^{RI}(x_0, \boldsymbol{F}_0)$ and $V_0^{RSO}(x_0, \boldsymbol{F}_0)$, respectively, can be estimated within a Monte Carlo simulation of a stochastic model of the forward curve evolution.

4. Structural Analysis of Heuristic Policies

This section conducts a structural analysis of the heuristic policies discussed in §3, focusing on the benefit of reoptimization in §4.1 and the structure of the RSO policy in §4.2.

4.1 Benefit of Reoptimization

When the asset is fast ($-C^I$, $C^W \geq \overline{x}$) and there are no frictions ($\phi^I = \phi^W = 1$ and $c^I = c^W = 0$), it is easy to show that reoptimization of the I dynamic program yields an optimal policy. It is also easy to show that in this case the SO policy is optimal but reoptimization does not hurt, that is, the RSO is also optimal. Moreover, the numerical work of Lai et al. (2010), Secomandi (2010), and Wu et al. (2012) indicates the usefulness of reoptimization of the I and SO policies when the asset is slow and there are frictions. These considerations motivate studying whether reoptimization is provably beneficial when using the RI and RSO policies in the general case.

A sharp result about the benefit of reoptimization can be obtained for the RI policy: Proposition 1 shows that reoptimization of the I dynamic program, equivalently, the I linear programs (4)-(10) or (19), is beneficial.

Proposition 1 (RI policy and reoptimization)**.** $V_0^I(x_0, \boldsymbol{F}_0) \leq V_0^{RI}(x_0, \boldsymbol{F}_0)$.

Proof. Denote by $u_m(n)$ and $u_m(n)$ the optimal buy-and-inject and withdraw-and-sell decisions for stage m when the I linear program (19) is optimized in a given state at time $T_n \leq T_m$. Given state $(x_0, \boldsymbol{F}_0)$ in stage 0, it holds that

$$\begin{aligned}
V_0^I(x_0, \boldsymbol{F}_0) &= s_0^W w_0(0) - s_0^I u_0(0) + \sum_{n=1}^{N-1} \delta^n \left[F_{0,n}^W w_n(0) - F_{0,n}^I u_n(0)\right] \\
&= s_0^W w_0(0) - s_0^I u_0(0) + \delta \sum_{n=1}^{N-1} \delta^{n-1} \left(\mathbb{E}\left[\tilde{F}_{1,n}^W \mid \boldsymbol{F}_0\right] w_n(0) - \mathbb{E}\left[\tilde{F}_{1,n}^I \mid \boldsymbol{F}_0\right] u_n(0)\right) \\
&\leq s_0^W w_0(0) - s_0^I u_0(0) + \delta \mathbb{E}\left[\sum_{n=1}^{N-1} \delta^{n-1}\left(\tilde{F}_{1,n}^W \tilde{w}_n(1) - \tilde{F}_{1,n}^I \tilde{u}_n(1)\right) \mid x_0, \boldsymbol{F}_0\right] \\
&= s_0^W w_0(0) - s_0^I u_0(0) + \delta \mathbb{E}\left[V_1^I(x_0 + u_0(0) - w_0(0), \tilde{\boldsymbol{F}}_1) \mid x_0, \boldsymbol{F}_0\right], \qquad (22)
\end{aligned}$$

where the second equality holds by the martingale property of futures prices under the risk neutral measure (Shreve 2004, p. 244), and the inequality is true by optimality of $u_n(1)$ and $w_n(1)$ at time T_1. Given state $(x_n, \boldsymbol{F}_n)$ in stage n, it can be shown in a similar manner that

$$V_n^I(x_n, \boldsymbol{F}_n) \leq s_n^W w_n(n) - s_n^I u_n(n) + \delta \mathbb{E}\left[V_{n+1}^I(x_n + u_n(n) - w_n(n), \tilde{\boldsymbol{F}}_{n+1}) \mid x_n, \boldsymbol{F}_n\right]. \qquad (23)$$

Applying (23) with $n = 1$ and $x_1 = x_0 + u_0(0) - w_0(0)$ to (22) implies

$$\begin{aligned}
V_0^I(x_0, \boldsymbol{F}_0) &\leq s_0^W w_0(0) - s_0^I u_0(0) + \delta \mathbb{E}\left[\tilde{s}_1^W \tilde{w}_1(1) - \tilde{s}_1^I \tilde{u}_1(1) \mid x_0, \boldsymbol{F}_0\right] \\
&\quad + \delta \mathbb{E}\left[\delta \mathbb{E}\left[V_2^I\left(x_1 + \tilde{u}_1(1) - \tilde{w}_1(1), \tilde{\boldsymbol{F}}_2\right) \mid x_1, \tilde{\boldsymbol{F}}_1\right] \mid x_0, \boldsymbol{F}_0\right] \\
&= \sum_{n=0}^{1} \delta^n \mathbb{E}\left[\tilde{s}_n^W \tilde{w}_n(n) - \tilde{s}_n^I \tilde{u}_n(n) \mid x_0, \boldsymbol{F}_0\right] \\
&\quad + \delta^2 \mathbb{E}\left[V_2^I\left(x_0 + \sum_{n=0}^{1}(\tilde{u}_n(n) - \tilde{w}_n(n)), \tilde{\boldsymbol{F}}_2\right) \mid x_0, \boldsymbol{F}_0\right]. \qquad (24)
\end{aligned}$$

Repeated applications of (23) starting from (24) yield

$$V_0^I(x_0, \boldsymbol{F}_0) \leq \sum_{n=0}^{N-1} \delta^n \mathbb{E}\left[\tilde{s}_n^W \tilde{w}_n(n) - \tilde{s}_n^I \tilde{u}_n(n) \mid x_0, \boldsymbol{F}_0\right] \equiv V_0^{RI}(x_0, \boldsymbol{F}_0). \square$$

Proposition 1 is analogous to Proposition 2 in Secomandi (2008), who deals with inventory control and revenue management problems. To gain some intuition on Proposition 1, label the optimization of the I linear program in a given stage and state as the *current* optimization and the optimization of this model in a given state in the next stage as the *next* optimization. This intuition is as follows: (i) The optimal basis obtained in the current optimization remains feasible in every next optimization, after removing from this solution the part that was implemented in the previous stage, because the intrinsic policy is feasible and the constraint set of the I linear program does not depend on the forward curve; (ii) due to the martingale property of futures prices under the risk neutral measure (Shreve 2004, p. 244), the discounted value of the expectation, under this measure, of the objective function of the next optimization added to the payoff from implementing the intrinsic action from the current optimization is the objective function of the current optimization for every feasible solution to this optimization; (iii) updating the optimal basis in the next optimization cannot consequently yield a policy that is worse than the one corresponding to implementing the solution from the current optimization.

Compared to the RI policy, a weaker result about the benefit of reoptimization can be obtained for the RSO policy: Proposition 2 states that reoptimization can improve the value of the SO policy as seen by the SO linear program, $U_0^{SO}(x_0, \boldsymbol{F}_0)$, rather than the true value of this policy, $V_0^{SO}(x_0, \boldsymbol{F}_0)$, when there are frictions, but reoptimization of the SO linear program is beneficial when there are no frictions.

Proposition 2 (RSO policy and reoptimization)**.** *(a)* $U_0^{SO}(x_0, \boldsymbol{F}_0) \leq V_0^{RSO}(x_0, \boldsymbol{F}_0)$*. (b) If there are no frictions then* $V_0^{SO}(x_0, \boldsymbol{F}_0) \leq V_0^{RSO}(x_0, \boldsymbol{F}_0)$*.*

Proof. Use the suffix (l) to denote an optimal solution to the SO linear program (20) obtained in a given state at time T_l. Without loss of generality, assume that $q_{m,n}(l)$ equals zero if $S_{l,m,n}(\boldsymbol{F}_l)$ equals zero. Given state $(x_0, \boldsymbol{F}_0)$ in stage 0, it thus holds that

$$
\begin{aligned}
U_0^{SO}(x_0, \boldsymbol{F}_0) \quad = \quad & s_0^W z_0(0) + \sum_{m=1}^{N-1} \delta^m F_{0,m}^W z_m(0) + \sum_{m=1}^{N-1} (\delta^m F_{0,m}^W - s_0^I) q_{0,m}(0) \\
& + \sum_{n=1}^{N-2} \sum_{m=n+1}^{N-1} S_{0,n,m}(\boldsymbol{F}_0) q_{n,m}(0) \\
= \quad & s_0^W z_0(0) - s_0^I \sum_{m=1}^{N-1} q_{0,m}(0) + \delta \sum_{m=1}^{N-1} \delta^{m-1} \mathbb{E}\left[\tilde{F}_{1,m}^W \mid \boldsymbol{F}_0\right] [z_m(0) + q_{0,m}(0)] \\
& + \delta \sum_{n=1}^{N-2} \sum_{m=n+1}^{N-1} \mathbb{E}\left[S_{1,n,m}(\tilde{\boldsymbol{F}}_1) \mid \boldsymbol{F}_0\right] q_{n,m}(0) \\
\leq \quad & s_0^W z_0(0) - s_0^I \sum_{m=1}^{N-1} q_{0,m}(0) + \delta \mathbb{E}\left[\sum_{m=1}^{N-1} \delta^{m-1} \tilde{F}_{1,m}^W \tilde{z}_m(1) \mid x_0, \boldsymbol{F}_0\right] \\
& + \delta \mathbb{E}\left[\sum_{n=1}^{N-2} \sum_{m=n+1}^{N-1} S_{1,n,m}(\tilde{\boldsymbol{F}}_1) \tilde{q}_{n,m}(1) \mid x_0, \boldsymbol{F}_0\right] \\
= \quad & s_0^W z_0(0) - s_0^I \sum_{m=1}^{N-1} q_{0,m}(0) \\
& + \delta \mathbb{E}\left[U_1^{SO}\left(x_0 + \sum_{m=1}^{N-1} q_{0,m}(0) - z_0(0), \tilde{\boldsymbol{F}}_1\right) \mid x_0, \boldsymbol{F}_0\right].
\end{aligned} \tag{25}
$$

Given state $(x_n, \boldsymbol{F}_n)$ in stage n, it can be shown in an analogous manner that

$$\begin{aligned} U_n^{SO}(x_n, \boldsymbol{F}_n) &\leq s_n^W z_n(n) - s_n^I \sum_{m=n+1}^{N-1} q_{n,m}(n) \\ &+ \delta \mathbb{E}\left[U_{n+1}^{SO}\left(x_n + \sum_{m=n+1}^{N-1} q_{n,m}(n) - z_n(n), \tilde{\boldsymbol{F}}_{n+1} \right) \mid x_n, \boldsymbol{F}_n \right]. \end{aligned} \tag{26}$$

Substituting (26) with $n = 1$ and $x_1 = x_0 + u_0(0) - w_0(0)$ into (25) gives

$$\begin{aligned} U_0^{SO}(x_0, \boldsymbol{F}_0) &\leq s_0^W z_0(0) - s_0^I \sum_{m=1}^{N-1} q_{0,m}(0) + \delta \mathbb{E}\left[\tilde{s}_1^W \tilde{z}_1(1) - \tilde{s}_1^I \sum_{m=2}^{N-1} \tilde{q}_{1,m}(1) \mid x_0, \boldsymbol{F}_0 \right] \\ &+ \delta \mathbb{E}\left[\delta \mathbb{E}\left[U_2^{SO}\left(x_1 + \sum_{m=2}^{N-1} \tilde{q}_{1,m}(1) - \tilde{z}_1(1), \tilde{\boldsymbol{F}}_2 \right) \mid x_1, \tilde{\boldsymbol{F}}_1 \right] \mid x_0, \boldsymbol{F}_0 \right] \\ &= \sum_{n=0}^{1} \delta^n \mathbb{E}\left[\tilde{s}_n^W \tilde{z}_n(n) - \tilde{s}_n^I \sum_{m=n+1}^{N-1} \tilde{q}_{n,m}(n) \mid x_0, \boldsymbol{F}_0 \right] \\ &+ \delta \mathbb{E}\left[U_2^{SO}\left(x_0 + \sum_{n=0}^{1} \sum_{m=n+1}^{N-1} \tilde{q}_{n,m}(n) - \sum_{n=0}^{1} \tilde{z}_n(n), \tilde{\boldsymbol{F}}_2 \right) \mid x_0, \boldsymbol{F}_0 \right]. \end{aligned} \tag{27}$$

Repeated applications of (26) starting from (27) yield part (a):

$$U_0^{SO}(x_0, \boldsymbol{F}_0) \leq \sum_{n=0}^{N-1} \delta^n \mathbb{E}\left[\tilde{s}_n^W \tilde{z}_n(n) - \tilde{s}_n^I \sum_{m=n+1}^{N-1} \tilde{q}_{n,m}(n) \mid x_0, \boldsymbol{F}_0 \right] \equiv V_0^{RSO}(x_0, \boldsymbol{F}_0).$$

If there are no frictions then $U_0^{SO}(x_0, \boldsymbol{F}_0) \equiv V_0^{SO}(x_0, \boldsymbol{F}_0)$ and part (b) follows from part (a). □

Proposition 2 is the analogue of Proposition 3 in Secomandi (2008). The weaker result on the benefit of reoptimization for the RSO policy than the RI policy is due to the fact that the optimal objective function of the I linear program is the value of the I policy while the optimal objective function of the SO linear program is a lower bound on the value of the SO policy when there are frictions, as discussed on page 8. In contrast, with no frictions the optimal objective function of the SO linear program is the value of the SO policy, as also discussed on page 8, and reoptimization is provably beneficial for the RSO policy.

Moreover, as stated in Lemma 2 in Appendix A, the value of the I policy is no larger than the value of the SO policy seen by the SO linear program. Thus, similar to the RI policy, the RSO policy is guaranteed to perform at least as well as the I policy. Proposition 6 in Appendix A establishes that both the RI and RSO policies have finite optimality gaps. In this sense, these reoptimization policies cannot perform catastrophically, which is a rather conservative statement in light of their excellent numerical performance documented by Lai et al. (2010) and the numerical results discussed in §6. The worst case in which the value of the optimal policy is positive and the values of both these reoptimization policies is zero (because the value of the I policy cannot be negative) can occur in pathological cases, such as the one discussed in Example 1 in Appendix A.

Ensuring that reoptimization of the SO policy is not harmful in the presence of frictions would require optimizing this policy using its *exact* evaluation. This optimization is more involved than solving a linear program, because with frictions the value of the SO policy is nonlinear in the notional amounts that define this policy. This nonlinearity arises because the SO policy nets out the injections and withdrawals corresponding to a given stage, and modeling this netting requires using indicator functions that depend on the spread option and forward sale notional amounts.

4.2 Basestock Target Structure

It is clear that the rolling intrinsic policy has the basestock target structure presented in Property 1. As pointed out at the beginning of §4.1, the RSO policy is optimal, even without reoptimization, in the case of a fast asset with no frictions. Hence, the RSO policy has this structure in this case. It is less clear whether the RSO policy also has this structure in general. Proposition 3 shows that this policy indeed has this structure in the general case. Although this result only provides weak support for the use of the RSO policy, it is reassuring that this policy shares the same structure of an optimal policy.

Proposition 3 (RSO policy and basestock target structure)**.** *The RSO policy has a double basestock target structure analogous to the one stated in Property 1.*

Proof. Without loss of generality, the claimed result is proved only for $n = 0$. Consider the linear program (20) and, without loss of optimality, relax each spread option value $S_{0,0,n}(\boldsymbol{F}_0)$ to the difference $\delta^n F^W_{0,n} - s^I_0$. The resulting linear program, which emphasizes the stage 0 decision variables z_0 and $q_{0,m}$'s and the stage 0 constraints (31)-(32), is

$$\max s^W_0 z_0 - s^I_0 \sum_{m=1}^{N-1} q_{0,m} + \sum_{n=1}^{N-1} \delta^n F^W_{0,n}(z_n + q_{0,n}) + \sum_{n=1}^{N-2} \sum_{m=n+1}^{N-1} S_{0,n,m}(\boldsymbol{F}_0) q_{n,m} \tag{28}$$

$$\text{s.t.} \sum_{m=1}^{n-1} \sum_{l=n}^{N-1} q_{m,l} - \sum_{m=1}^{n-1} z_m \geq -x_0 + z_0 - \sum_{l=n+1}^{N-1} q_{0,l}, \ \forall n \in \mathcal{N} \cup \{N\} \setminus \{0\}, \tag{29}$$

$$\sum_{m=1}^{n-1} \sum_{l=n}^{N-1} q_{m,l} - \sum_{m=1}^{n-1} z_m \leq \overline{x} - x_0 + z_0 - \sum_{l=n+1}^{N-1} q_{0,l}, \ \forall n \in \mathcal{N} \cup \{N\} \setminus \{0\}, \tag{30}$$

$$\sum_{m=1}^{N-1} q_{0,m} \leq -C^I, \tag{31}$$

$$z_0 \leq C^W, \tag{32}$$

$$\sum_{m=n+1}^{N-1} q_{n,m} \leq -C^I, \ \forall n \in \mathcal{N} \setminus \{0\}, \tag{33}$$

$$\sum_{m=0}^{n-1} q_{m,n} + z_n \leq C^W, \ \forall n \in \mathcal{N} \setminus \{0\}, \tag{34}$$

$$q_{n,m} \geq 0, \ \forall n \in \mathcal{N} \setminus \{N-1\}, m \in \mathcal{N}, m > n, \tag{35}$$

$$z_n \geq 0, \ \forall n \in \mathcal{N}. \tag{36}$$

Given $(y, \boldsymbol{F}_0) \in \mathcal{X} \times \Re^N_+$, define the basket of spread options continuation-value function as

$$W^{SO}_0(y, \boldsymbol{F}_0) \ := \ \max \sum_{n=1}^{N-1} \delta^n F^W_{0,n} z_n + \sum_{n=1}^{N-2} \sum_{m=n+1}^{N-1} S_{0,n,m}(\boldsymbol{F}_0) q_{n,m}$$

$$
\begin{aligned}
\text{s.t. } & \sum_{m=1}^{n-1}\sum_{l=n}^{N-1} q_{m,l} - \sum_{m=1}^{n-1} z_m \geq -y,\ \forall n \in \mathcal{N} \cup \{N\} \setminus \{0\}, \\
& \sum_{m=1}^{n-1}\sum_{l=n}^{N-1} q_{m,l} - \sum_{m=1}^{n-1} z_m \leq \overline{x} - y,\ \forall n \in \mathcal{N} \cup \{N\} \setminus \{0\}, \\
& \sum_{m=n+1}^{N-1} q_{n,m} \leq -C^I,\ \forall n \in \mathcal{N} \setminus \{0\}, \\
& \sum_{m=0}^{n-1} q_{m,n} + z_n \leq C^W,\ \forall n \in \mathcal{N} \setminus \{0\}, \\
& q_{n,m} \geq 0,\ \forall n \in \mathcal{N} \setminus \{0, N-1\}, m \in \mathcal{N}, m > n, \\
& z_n \geq 0,\ \forall n \in \mathcal{N}.
\end{aligned}
$$

The function $W_0^{SO}(\cdot, \boldsymbol{F}_0)$ is concave for each given $\boldsymbol{F}_0$ (Bertsimas and Tsitsiklis 1997, §5.2). Use this function to define the math program

$$\max s_0^W \zeta_0 - s_0^I u_0 + W_0^{SO}(x_0 + u_0 - \zeta_0, \boldsymbol{F}_0) \tag{37}$$

$$\text{s.t. } u_0 - \zeta_0 \leq \overline{x} - x_0, \tag{38}$$

$$u_0 - \zeta_0 \geq -x_0, \tag{39}$$

$$u_0 - \zeta_0 \leq -C^I, \tag{40}$$

$$\zeta_0 \leq C^W, \tag{41}$$

$$u_0 \geq 0, \tag{42}$$

$$\zeta_0 \geq 0, \tag{43}$$

where the decision variables u_0 and ζ_0 are the amounts of inventory bought-and-injected and withdrawn-and-sold in stage 0, respectively. It is easy to verify that for this math program it is never optimal to simultaneously purchase-and-inject and withdraw-and-sell in stage 0. The linear program (28)-(36) and the math program (37)-(43) share the same optimal objective function value. Indeed, if it is optimal to purchase and inject some amount of commodity in stage 0 for the math program (37)-(43), that is, $u_0 > 0$ in an optimal solution to this math program, then this amount of commodity is entirely sold in later stages in the linear program corresponding to $W_0^{SO}(x_0 + u_0, \boldsymbol{F}_0)$, that is, $\sum_{m=1}^{n-1} z_m = x_0 + u_0$ in an optimal solution to this linear program (if this were not the case, then optimality of $u_0 > 0$ for the math program (37)-(43) would be contradicted). Moreover, there exist optimal solutions $(\boldsymbol{q}^*, \boldsymbol{z}^*)$ and $\{u_0^*, \zeta_0^*\}$ to the linear program (28)-(36) and the math program (37)-(43), respectively, that satisfy $z_0^* = \zeta_0^*$ and $\sum_{m=1}^{N-1} q_{0,m}^* = u_0^*$. Label this equivalence property as EP.

The orthogonality at optimality of the decision variables of the math program (37)-(43) implies that this math program is equivalent to the following math program:

$$\max_{a \in \mathcal{A}(x_0)} p(a, s_0) + W_0^{SO}(x_0 - a, \boldsymbol{F}_0). \tag{44}$$

Define $\underline{b}_0^{SO}$ and $\overline{b}_0^{SO}$ as the smallest and largest optimal solutions to the optimization models $\max_{y \in \mathcal{X}} W_0^{SO}(y, \boldsymbol{F}_0) - s_0^I y$ and $\max_{y \in \mathcal{X}} W_0^{SO}(y, \boldsymbol{F}_0) - s_0^W y$, respectively. The concavity of $W_0^{SO}(\cdot, \boldsymbol{F}_0)$

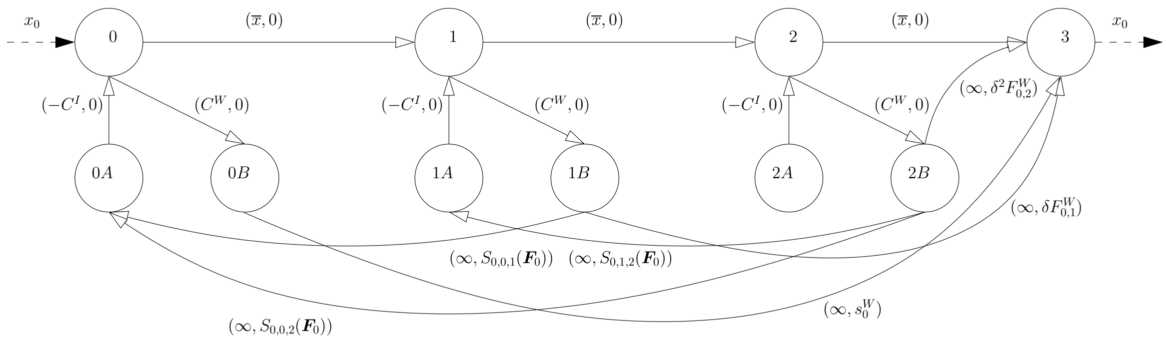

Figure 1: The graph $G := (\overline{\mathcal{N}}, \mathcal{E})$ for $N = 3$; the dashed arrows into and out of nodes 0 and 3 indicate the initial inventory supply and demand, respectively, the edges are labeled with (upper bound, gain) pairs on the flow variables, and the lower bounds on all the flow variables are zero.

implies that $\underline{b}_0^{SO}$ and $\bar{b}_0^{SO}$ define a basestock target structure for the math program (44) (see Secomandi 2010, Theorem 1, Lai et al. 2010, Theorem 1): $\underline{b}_0^{SO} \leq \bar{b}_0^{SO}$ and an optimal solution a_0^* to this math program satisfies $a_0^* = C^I \vee [x_0 - \underline{b}_0^{SO}]$ if $x_0 \in [0, \underline{b}_0^{SO})$, $a_0^* = 0$ if $x_0 \in [\underline{b}_0^{SO}, \bar{b}_0^{SO}]$, and $a_0^* = C^W \wedge [x_0 - \bar{b}_0^{SO}]$ if $x_0 \in (\bar{b}_0^{SO}, \overline{x}]$. The equivalence at optimality between the math programs (37)-(43) and (44), the EP property, and the equivalence between the linear programs (20) and (28)-(36) imply that there exists an optimal solution $(\boldsymbol{q}^*, \boldsymbol{z}^*)$ to the linear program (20) that is consistent with this basestock target structure. That is, this solution satisfies $\sum_{m=1}^{N-1} q^*_{0,m} = C^I \vee [x_0 - \underline{b}_0^{SO}]$ if $x_0 \in [0, \underline{b}_0^{SO})$, $z_0^* = \sum_{m=1}^{N-1} q^*_{0,m} = 0$ if $x_0 \in [\underline{b}_0^{SO}, \bar{b}_0^{SO}]$, and $z_0^* = C^W \wedge [x_0 - \bar{b}_0^{SO}]$ if $x_0 \in (\bar{b}_0^{SO}, \overline{x}]$.

Suppose now that Assumption 1 holds. The remaining part of this proof relies on constructing the graph $G := (\overline{\mathcal{N}}, \mathcal{E})$ with node set $\overline{\mathcal{N}}$ and edge set $\mathcal{E}$ and a network flow model on G. Figure 1 illustrates this graph and network flow model for $N = 3$. The node set $\overline{\mathcal{N}}$ includes four sets of nodes, arranged in two layers: $\overline{\mathcal{N}} := \mathcal{N} \cup \{N\} \cup \mathcal{N}^A \cup \mathcal{N}^B$. The first layer consists of the stage set $\mathcal{N} \equiv \{0, \ldots, N-1\}$ and the sink node $\{N\}$. The second layer includes A and B versions of the stage set $\mathcal{N}$: $\mathcal{N}^A := \{0A, \ldots, (N-1)A\}$ and $\mathcal{N}^B := \{0B, \ldots, (N-1)B\}$. Each edge $(n, m) \in \mathcal{E}$ from node n to node m is directed and there are no self directed edges. Denote by $\mathcal{E}(n)$ the set of edges that are incident to node $n \in \overline{\mathcal{N}}$. The edge set is thus $\mathcal{E} := \cup_{n \in \mathcal{N}} \mathcal{E}(n)$.

The following notation is useful to introduce the edges in set $\mathcal{E}$. Denote by nA the node in set $\mathcal{N}^A$ obtained by concatenating the node $n \in \mathcal{N}$ and the label A: $nA := n \oplus A$, where $\oplus$ indicates concatenation. Given the node $m \in \mathcal{N}^A$, the corresponding node in set $\mathcal{N}$ is obtained by deconcatenating the label A from m. That is, if $m = nA$ then $m \ominus A = n$, with $\ominus$ indicating deconcatenation. Analogous notation applies to related nodes in sets $\mathcal{N}$ and $\mathcal{N}^B$.

The edge sets of the nodes in the first layer are $\mathcal{E}(0) := \{(0,1), (0A, 0), (0, 0B)\}$, $\mathcal{E}(n) := \{(n-1, n), (n, n+1), (nA, n), (n, nB)\}$, $\forall n \in \mathcal{N} \setminus \{0\}$, and $\mathcal{E}(N) := \{(N-1, N)\} \cup \{\bigcup_{n \in \mathcal{N}^B} (n, N)\}$. The edge set of each node n in the A part of the second layer is

$$\mathcal{E}(n) := \{(n, n \ominus A)\} \cup \left\{ \bigcup_{m \in \mathcal{N}^B, m \ominus B > n \ominus A} \{(m, n)\} \right\}.$$

The analogous set of each node n in the B part of the second layer is

$$\mathcal{E}(n) := \{(n \ominus B, n), (n, N)\} \cup \left\{ \bigcup_{m \in \mathcal{N}^A, m \ominus A < n \ominus B} \{(n, m)\} \right\}.$$

Denote by $f_{n,m}$ the flow on edge $(n, m) \in \mathcal{E}$. Relative to the decision variables used in (11)-(12), the flow on arc (n, m) corresponds to the variable u_n if $n \in \mathcal{N}^A$ and $m = n \ominus A$; w_n if $n \in \mathcal{N}$ and $m = n \oplus B$; $q_{m,n}$ if $n \in \mathcal{N}^B$, $m \in \mathcal{N}^A$, and $(n \ominus B) > (m \ominus A)$; and z_n if $n \in \mathcal{N}^B$ and $m = N$. The flow on each edge $(n, n+1)$, with $n \in \mathcal{N}$, does not map to any decision variable in (11)-(12) and represents the inventory level at time T_{n+1}.

The upper bound $\overline{f}_{n,m}$ on the flow $f_{n,m}$ is $\overline{x}$ if the arc (n, m) connect nodes in the first layer, that is, if $n \in \mathcal{N}$ and $m = n + 1$; $-C^I$ if the edge (n, m) is outgoing from an A node, that is, if $n \in \mathcal{N}^A$ and $m = n \ominus A$; C^W if the edge (n, m) is incoming to a B node, that is, if $n \in \mathcal{N}$ and $m = n \oplus B$; and ∞ if the arc (n, m) is outgoing from a B node, that is, if $n \in \mathcal{N}^B$ and $m \in \{N\} \cup \mathcal{N}^A$. The lower bound on each flow is 0.

The unit gain $g_{n,m}$ on each flow $f_{n,m}$ from the B node n to the A node m, with $n \ominus B > m \ominus A$, is the spread option value $S_{0,m,n}(\boldsymbol{F}_0)$. The unit gain on each flow $f_{n,m}$ from the B node n to the sink node N is the discounted withdraw-and-sell futures price $\delta^n F^W_{0,n}$. The unit gains of all other flows are equal to 0.

Denote as $\overline{\mathcal{N}}^{\downarrow}(n) := \{m \in \overline{\mathcal{N}}, (m, n) \in \mathcal{E}\}$ the subset of nodes in set $\overline{\mathcal{N}}$ that are the origin of edges that are incoming to node n, and as $\overline{\mathcal{N}}^{\uparrow}(n) := \{m \in \overline{\mathcal{N}}, (n, m) \in \mathcal{E}\}$ the subset of nodes in set $\overline{\mathcal{N}}$ that are the destination of edges that are outgoing from node n. Consider the maximum gain network flow model

$$\max \sum_{(n,m) \in \mathcal{E}} g_{n,m} f_{n,m} \tag{45}$$

$$\text{s.t. } x_0 1\{n = 0\} + \sum_{m \in \overline{\mathcal{N}}^{\downarrow}(n)} f_{m,n} = \sum_{m \in \overline{\mathcal{N}}^{\uparrow}(n)} f_{n,m} + x_0 1\{n = N\}, \ \forall n \in \overline{\mathcal{N}}, \tag{46}$$

$$0 \leq f_{n,m} \leq \overline{f}_{n,m}, \ \forall (n, m) \in \mathcal{E}. \tag{47}$$

The objective function (45) maximizes the total gain earned by the flow vector $(f_{n,m}, (n, m) \in \mathcal{E})$. The inequalities (46) are flow balance constraints, augmented with the supply and demand of the initial inventory x_0 for nodes 0 and N, respectively. Constraints (47) place lower and upper bounds on the flow variables.

Define the vectors $\boldsymbol{u} := (u_n, n \in \mathcal{N})$ and $\boldsymbol{w} := (w_n, n \in \mathcal{N})$ and the polyhedron $\mathcal{P}'' := \left\{(\boldsymbol{u}, \boldsymbol{w}) \in \Re^{2N}, (\boldsymbol{q}, \boldsymbol{z}) \in \Re^{(N^2+N)/2} \text{ s.t. (11)-(12), (13)-(18)}\right\}$. By construction of the graph G, the network flow model (45)-(47) is equivalent to the following *extended* SO linear program:

$$\max_{(\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{q}, \boldsymbol{z}) \in \mathcal{P}''} \sum_{n=0}^{N-1} \delta^n F^W_{0,n} z_n + \sum_{n=0}^{N-2} \sum_{m=n+1}^{N-1} S_{0,n,m}(\boldsymbol{F}_0) q_{n,m}.$$

Moreover, this linear program is equivalent to the SO linear program (20) at optimality. If the initial inventory x_0, the capacity limits C^I and C^W, and the maximum inventory level $\overline{x}$ are integer multiples of the lot size Q, then an optimal solution to the network flow model (45)-(47) also is integer multiple of Q (Bertsimas and Tsitsiklis 1997, Chapter 7), and so is an optimal solution to the SO linear program. In particular, this occurs when $x_0 = 0$ or $x_0 = \overline{x}$. It follows that the basestock targets $\underline{b}_0^{SO}$ and $\overline{b}_0^{SO}$ can be taken to be integer multiples of Q. $\square$

5. Dual Upper Bounds

Upper bounds on the value of storage are important to benchmark the performance of heuristic policies. Subsection 5.1 briefly introduces dual upper bounds (Brown et al. 2010) on the value of storage, which can be efficiently estimated by Monte Carlo simulation provided feasible dual penalties are available and the resulting sample optimization model can be easily solved. Lai et al. (2010) and Nadarajah et al. (2013) estimate such bounds by instantiating the dual penalties using the value functions of ADPs, which must be solved numerically. Subsection 5.2 investigates a simpler approach: The dual penalties are determined using the optimal value function of the tractable case in which the storage asset is fast and there are no frictions, which, as discussed later, is typically available in essentially closed form.

5.1 Dual Upper Bounding Approach

Dual upper bounds are based on dual sample path optimizations that penalize knowledge of future information. Let $\boldsymbol{G} := (\boldsymbol{F}_n)_{n\in\mathcal{N}}$ be a sample path of forward curves from stage 0 through stage $N-1$. The n-th element of $\boldsymbol{G}$ is $\boldsymbol{F}_n(\boldsymbol{G})$, and $\boldsymbol{F}'_n(\boldsymbol{G})$ is interpreted accordingly. Suppose that a function $U_n(x, \boldsymbol{F}_n)$ defined on $\mathcal{N} \times \mathcal{X} \times \Re^{N-n}$ is available. Typically (Lai et al. 2010 and Nadarajah et al. 2013), this function is interpreted as an approximation of the optimal value function in stage n and state $(x, \boldsymbol{F}_n)$ of stochastic dynamic program (2). Consider feasible action a in stage n and modified state $(x_n, \boldsymbol{F}'_n(\boldsymbol{G}))$ – the modification is the use of $\boldsymbol{F}'_n(\boldsymbol{G})$ in lieu of $\boldsymbol{F}_n(\boldsymbol{G})$) given the sample path $\boldsymbol{G}$. Define the dual penalty corresponding to performing this action in this stage and modified state given this sample path as

$$P_n(a, x, \boldsymbol{G}) := \delta \left\{ U_{n+1}(x-a, \boldsymbol{F}_{n+1}(\boldsymbol{G})) - \mathbb{E}[U_{n+1}(x-a, \tilde{\boldsymbol{F}}_{n+1}) \mid \boldsymbol{F}'_n(\boldsymbol{G})] \right\}. \tag{48}$$

Intuitively, expression (48) defines, approximately, the additional value of knowing future information included in $\boldsymbol{G}$: The first term on the right hand side of (48) is the approximate value of having an amount of inventory equal to $x-a$ in stage $n+1$ given knowledge of the forward curve $\boldsymbol{F}_{n+1}(\boldsymbol{G})$; the second term is the approximate value of this inventory level in this stage given knowledge of the forward curve $\boldsymbol{F}'_n$, rather than $\boldsymbol{F}_{n+1}$; the difference between these two terms is then the approximate value of knowing $\boldsymbol{F}_{n+1}$ when performing action a in stage n and modified state $(x, \boldsymbol{F}_n)$. The dual penalty defined in (48) is feasible (Brown et al. 2010) because $\mathbb{E}[P_n(a, x, \boldsymbol{F}_n, \boldsymbol{F}) \mid \boldsymbol{F}'_n] = 0$.

The dual penalties (48) are used in the following dual (D) dynamic program, the value function of which is $V_n^D(x_n; \boldsymbol{G})$:

$$V_n^D(x_n; \boldsymbol{G}) = \max_{a\in\mathcal{A}(x_n)} p(a, s_n(\boldsymbol{G})) - P_n(a, x_n, \boldsymbol{G}) + \delta V_{n+1}^D(x_n - a; \boldsymbol{G}), \tag{49}$$

for all stages $n \in \mathcal{N}$ and states $x_n \in \mathcal{X}$, with $V_N^D(x_N; \boldsymbol{G}) := 0$ for all $x_N \in \mathcal{X}$.

A dual upper bound is $\mathbb{E}[V_0^D(x_0; \tilde{\boldsymbol{G}}) \mid \boldsymbol{F}_0]$. This bound can be easily estimated by Monte Carlo simulation, that is, by solving a collection of dual dynamic programs (49), one for each sample path $\boldsymbol{G}$, provided that each such dynamic program can be efficiently solved.

5.2 Simple Dual Upper Bounds

It would be desirable if the optimization models used to obtain the RI and RSO policies could be *easily* modified to make them suitable for dual upper bound estimation. When these policies are computed by solving linear programs, such a modification requires using penalties that are linear in the next stage inventory level – linear penalties, for short. Linear penalties are also relevant when using the I dynamic program to obtain the RI, because their use entails minimal change to this model. Charnes et al. (1966) demonstrate that the optimal value function of a fast storage asset is linear in inventory. Using this value function would yield linear penalties, but computing this function is intractable when there are frictions. It is now shown how to obtain linear penalties from the tractable optimal value function of the fast and frictionless storage asset.

Consider the sequence of actions $(a_n)_{n\in\mathcal{N}}$. If there are no frictions, the total discounted value of this sequence is

$$\sum_{n\in\mathcal{N}}\delta^n s_n a_n = \sum_{n\in\mathcal{N}}\delta^n s_n(x_n - x_{n+1}) = s_0x_0 + \sum_{n\in\mathcal{N}\setminus\{N-1\}}\delta^n(\delta s_{n+1} - s_n)x_{n+1} - \delta^{N-1}s_{N-1}x_N.$$

Further, if the storage asset is fast any feasible inventory level can be reached in the next stage starting from any feasible inventory level in the current stage. Model (1) can thus be equivalently expressed as choosing a set of inventory random variables $\{\tilde{x}_{n+1},\ n\in\mathcal{N}\}$ as follows

$$\max_{\tilde{x}}\sum_{n\in\mathcal{N}\setminus\{N-1\}}\delta^n\mathbb{E}[(\delta\tilde{s}_{n+1}-\tilde{s}_n)\tilde{x}_{n+1}\mid \boldsymbol{F}_0] - \delta^{N-1}\mathbb{E}[\tilde{s}_{N-1}\tilde{x}_N\mid\boldsymbol{F}_0]\text{ s.t. }\tilde{x}_{n+1}\in\mathcal{X},\ \forall n\in\mathcal{N}. \quad (50)$$

Since $\tilde{x}_{n+1}$ depends on information available at time T_n and $F_{n,n+1} = \mathbb{E}\left[\tilde{s}_{n+1}\mid F_{n,n+1}\right]$ (Shreve 2004, p. 244), it follows that

$$\mathbb{E}[(\delta\tilde{s}_{n+1}-\tilde{s}_n)\tilde{x}_{n+1}\mid\boldsymbol{F}_0] = \mathbb{E}[\mathbb{E}[(\delta\tilde{s}_{n+1}-\tilde{s}_n)\tilde{x}_{n+1}\mid\boldsymbol{F}_n]\mid\boldsymbol{F}_0] = \mathbb{E}[(\delta\tilde{F}_{n,n+1}-\tilde{s}_n)\tilde{x}_{n+1}\mid\boldsymbol{F}_0].$$

Hence, model (50) can be rewritten as

$$\max_{\tilde{x}}\sum_{n\in\mathcal{N}\setminus\{N-1\}}\delta^n\mathbb{E}[(\delta\tilde{F}_{n,n+1}-\tilde{s}_n)\tilde{x}_{n+1}\mid\boldsymbol{F}_0] - \delta^{N-1}\mathbb{E}[\tilde{s}_{N-1}\tilde{x}_N\mid\boldsymbol{F}_0]\text{ s.t. }\tilde{x}_{n+1}\in\mathcal{X},\ \forall n\in\mathcal{N}.$$

An optimal solution to this model is $\tilde{x}^*_{n+1} = \overline{x}1\{\delta\tilde{F}_{n,n+1}-\tilde{s}_n > 0\}$ for all $n\in\mathcal{N}\setminus\{N-1\}$ and $\tilde{x}^*_N = 0$. This solution can be interpreted as determining in stage n the inventory level to reach in stage $n+1$, that is, x_{n+1}, contingent on the sign of the price spread $\delta F_{n,n+1} - s_n$. Implementing this solution is thus equivalent to optimally exercising a portfolio of exchange options, with the payoff of the exchange option for stage n being $\overline{x}(\delta F_{n,n+1} - s_n)^+$. This analysis yields Proposition 4. The value function of the fast and frictionless asset is denoted as $V_n^{EO}(x_n, \boldsymbol{F}_n)$.

Proposition 4 (Fast storage asset with no frictions)**.** *If the storage asset is fast and there are no frictions an optimal decision rule in stage $n\in\mathcal{N}$ is $x-\overline{x}$ if $\delta F_{n,n+1} > s_n$, 0 if $\delta F_{n,n+1} = s_n$, and x if $\delta F_{n,n+1} < s_n$. If $x_0\in\{0,\overline{x}\}$ then the optimal policy defined by these decision rules only visits states with inventory component in this set, that is, $x_n\in\{0,\overline{x}\}$, $\forall n\in\mathcal{N}\cup\{N\}$. Moreover, the optimal value function in stage n and state $(x_n,\boldsymbol{F}_n)$ of the resulting stochastic dynamic program (2) is $V_n^{EO}(x_n,\boldsymbol{F}_n) = s_nx_n + \overline{x}\sum_{m=n}^{N-2}\delta^{m-n}\mathbb{E}[(\delta\tilde{F}_{m,m+1}-\tilde{s}_m)^+\mid\boldsymbol{F}_n]$.*

The optimal value function established in Proposition 4 can be used to define valid linear dual penalties as follows:

$$\begin{aligned} P_n^{EO}(a,x,\boldsymbol{G}) &:= \delta\left\{V_{n+1}^{EO}(x-a,\boldsymbol{F}_{n+1}(\boldsymbol{G})) - \mathbb{E}[V_{n+1}^{EO}(x-a,\tilde{\boldsymbol{F}}_{n+1}) \mid \boldsymbol{F}_n'(\boldsymbol{G})]\right\} \\ &= \delta[s_{n+1}(\boldsymbol{G}) - F_{n,n+1}(\boldsymbol{G})](x-a) + \text{Constant}_n(\boldsymbol{G}), \end{aligned} \tag{51}$$

where the equality follows from the martingale property of futures prices under the risk neutral measure (Shreve 2004, p. 244), and the $\text{Constant}_n(\boldsymbol{G})$ term is defined as

$$\overline{x}\sum_{m=n+1}^{N-2}\delta^{m-n}\left\{\mathbb{E}[(\delta\tilde{F}_{m,m+1}-\tilde{s}_m)^+ \mid \boldsymbol{F}_{n+1}(\boldsymbol{G})] - \mathbb{E}[(\delta\tilde{F}_{m,m+1}-\tilde{s}_m)^+ \mid \boldsymbol{F}_n(\boldsymbol{G})]\right\}. \tag{52}$$

Let $V_n^{DEO}(\cdot;\boldsymbol{G})$ be the stage n dual value function obtained by solving (49) using the penalties (51). The DEO upper bound is the dual upper bound $\mathbb{E}[V_0^{DEO}(x_0;\tilde{\boldsymbol{G}}) \mid \boldsymbol{F}_0]$. It is clear that the optimal value function of the fast and frictionless storage asset is an upper bound on the optimal value function of the slow storage asset, with or without frictions. Denote this bound by EO. Proposition 5 relates the $V_n^{DEO}(\cdot;\boldsymbol{G})$ and $V_n^{EO}(\cdot,\boldsymbol{F}_n(\boldsymbol{G}))$ value functions, and the EO and DEO upper bounds.

Proposition 5 (DEO and EO Value Functions and Upper Bounds)**.** *(a) For each given $\boldsymbol{G}\in\Re_+^N$, it holds that $V_n^{DEO}(x_n;\boldsymbol{G}) \le V_n^{EO}(x_n,\boldsymbol{F}_n(\boldsymbol{G}))$, $\forall(n,x_n)\in\mathcal{N}\times\mathcal{X}$. (b) $\mathbb{E}[V_0^{DEO}(x_0;\tilde{\boldsymbol{G}}) \mid \boldsymbol{F}_0] \le V_0^{EO}(x_0,\boldsymbol{F}_0)$.*

Proof. (a) The claimed property holds in stage $N-1$ and state x_N because

$$V_{N-1}^{DEO}(x_{N-1};\boldsymbol{G}) \equiv V_{N-1}(x_{N-1},\boldsymbol{F}_{N-1}(\boldsymbol{G})) \le V_{N-1}^{EO}(x_{N-1},\boldsymbol{F}_{N-1}(\boldsymbol{G})).$$

Suppose it is also true in every state in stages $n+1$ through $N-2$. In stage n and state x_n it holds that

$$\begin{aligned} V_n^{DEO}(x_n;\boldsymbol{G}) &= \max_{a\in\mathcal{A}(x_n)} p(a,s_n(\boldsymbol{G})) - P_n^{EO}(a,x_n,\boldsymbol{G}) + \delta V_{n+1}^{DEO}(x_n-a;\boldsymbol{G}) \\ &= \max_{a\in\mathcal{A}(x_n)} p(a,s_n(\boldsymbol{G})) - \delta V_{n+1}^{EO}(x-a,\boldsymbol{F}_{n+1}(\boldsymbol{G})) \\ &\qquad + \delta\mathbb{E}[V_{n+1}^{EO}(x-a,\tilde{\boldsymbol{F}}_{n+1}) \mid \boldsymbol{F}_n'(\boldsymbol{G})] + \delta V_{n+1}^{DEO}(x_n-a;\boldsymbol{G}) \\ &\le \max_{a\in\mathcal{A}(x_n)} p(a,s_n(\boldsymbol{G})) - \delta V_{n+1}^{EO}(x-a,\boldsymbol{F}_{n+1}(\boldsymbol{G})) \\ &\qquad + \delta\mathbb{E}[V_{n+1}^{EO}(x-a,\tilde{\boldsymbol{F}}_{n+1}) \mid \boldsymbol{F}_n'(\boldsymbol{G})] + \delta V_{n+1}^{EO}(x_n-a,\boldsymbol{F}_{n+1}(\boldsymbol{G})) \\ &= \max_{a\in\mathcal{A}(x_n)} p(a,s_n(\boldsymbol{G})) + \delta\mathbb{E}[V_{n+1}^{EO}(x-a,\tilde{\boldsymbol{F}}_{n+1}) \mid \boldsymbol{F}_n'(\boldsymbol{G})] \\ &= V_n^{EO}(x_n,\boldsymbol{F}_n(\boldsymbol{G})), \end{aligned}$$

where the inequality follows from the induction hypothesis. The claimed property is thus true in every stage and state by the principle of mathematical induction.

(b) Part (a) implies $\mathbb{E}[V_0^{DEO}(x_0;\tilde{\boldsymbol{G}}) \mid \boldsymbol{F}_0] \le \mathbb{E}[V_0^{EO}(x_0,\boldsymbol{F}_0(\tilde{\boldsymbol{G}})) \mid \boldsymbol{F}_0] \equiv V_0^{EO}(x_0,\boldsymbol{F}_0)$. □

Proposition 5 establishes that the dual upper bound $V_n^{DEO}(x_n;\boldsymbol{G})$ is no weaker than the upper bound $V_n^{EO}(x_n,\boldsymbol{F}_n(\boldsymbol{G}))$ in every stage n and state $(x_n,\boldsymbol{F}_n(\boldsymbol{G}))$, and, in particular, the DEO bound is no worse than the EO bound. The relative performance of the DEO and EO bounds is numerically quantified in §6. In the context of dynamic portfolio optimization with transaction costs, Brown and Smith (2011)[Proposition 4.2(3)] obtain a result similar to Proposition 5 for a dual upper bound obtained from gradient-based linear dual penalties associated with an approximate model that ignores transaction costs – Brown and Smith (2013, Proposition 2.2(iii)) generalize this result beyond this application. In contrast, the penalties used here are by construction linear in the inventory change, because the value function of the fast and frictionless storage asset is linear in inventory. Thus, the proposed approach is simpler but less general than the approach of Brown and Smith (2011, 2013).

Estimating the DEO bound requires being able to compute the terms (52). These terms can be easily evaluated by the exchange option formula of Margrabe (1978) when using common reduced form models of the forward curve evolution, e.g., the model (55)-(56) used in §6. However, an easier to implement approach is to simplify the dual penalties (51) by dropping these terms. The resulting penalties, denoted as $P_n^S(a,x,\boldsymbol{G})$, reduce to discounted spreads between the spot price in the next stage and the current prompt futures price multiplied by the next stage inventory level (hence the superscript "S" on $P_n^S(a,x,\boldsymbol{G})$), and are thus available in closed form:

$$P_n^S(a,x,\boldsymbol{G}) := \delta[s_{n+1}(\boldsymbol{G}) - F_{n,n+1}(\boldsymbol{G})](x-a). \tag{53}$$

These penalties are feasible because futures prices are martingales under the risk neutral measure (Shreve 2004, p. 244). Moreover, because the terms (52) have zero mean given $\boldsymbol{F}_0$, use of these penalties leads to a bound that is equal to DEO, which is now shown more formally.

Denote as $V_n^{DS}(x_n;\boldsymbol{G})$ the dual value function corresponding to using the penalties (53) in (49). The DS upper bound is $\mathbb{E}[V_0^{DS}(x_0;\tilde{\boldsymbol{G}}) \mid \boldsymbol{F}_0]$. Given that $P_n^{EO}(a,x,\boldsymbol{G}) = P_n^S(a,x,\boldsymbol{G}) + \text{Constant}_n(\boldsymbol{G})$, it is easy to show that

$$\begin{aligned} V_n^{DEO}(x_n;\boldsymbol{G}) &= V_n^{DS}(x_n;\boldsymbol{G}) - \sum_{m=n}^{N-2} \delta^{m-n}\text{Constant}_m(\boldsymbol{G}) \\ &= V_n^{DS}(x_n;\boldsymbol{G}) \\ &\quad - \sum_{m=n}^{N-2} \delta^{m-n}\left\{\mathbb{E}[(\delta\tilde{F}_{m,m+1} - \tilde{s}_m)^+ \mid \boldsymbol{F}_m(\boldsymbol{G})] - \mathbb{E}[(\delta\tilde{F}_{m,m+1} - \tilde{s}_m)^+ \mid \boldsymbol{F}_n(\boldsymbol{G})]\right\}, \end{aligned} \tag{54}$$

where the second equality follows from simple algebra. The second term in (54) has zero mean given $\boldsymbol{F}_0$. Thus, the DS and DEO bounds coincide and the DS bound is no weaker than the EO bound. At later stages, $n > 0$, it may not be true that the dual value function $V_n^{DS}(\cdot;\tilde{\boldsymbol{G}})$ is no larger than the value function of the fast and frictionless asset $V_n^{EO}(\cdot,\boldsymbol{F}_n(\tilde{\boldsymbol{G}}))$, which contrasts the property established in part (a) of Proposition 5 for the dual value function $V_n^{DEO}(\cdot;\tilde{\boldsymbol{G}})$. However, Corollary 1 shows that the dual value function $V_n^{DS}(\cdot;\tilde{\boldsymbol{G}})$ satisfies an "average" version of this property.

Corollary 1 (DS and EO Value Functions)**.** *For each given $\boldsymbol{G} \in \Re_+^N$, it holds that*

$$\mathbb{E}\left[V_n^{DS}(x_n;\tilde{\boldsymbol{G}}) \mid \boldsymbol{F}_n(\boldsymbol{G})\right] \le V_n^{EO}(x_n,\boldsymbol{F}_n(\boldsymbol{G})),\ \forall (n,x_n) \in \mathcal{N} \times \mathcal{X}.$$

Proof. It follows from (54), part (a) of Proposition 5, and the expression for $V_n^{EO}(x_n, \boldsymbol{F}_n)$ given in Proposition 4 that $V_n^{DS}(x_n; \tilde{\boldsymbol{G}}) \leq s_n(\boldsymbol{G})x_n + \sum_{m=n}^{N-2} \delta^{m-n}\mathbb{E}[(\delta\tilde{F}_{m,m+1} - \tilde{s}_m)^+ \mid \boldsymbol{F}_m(\boldsymbol{G})]$, from which the claimed result ensues immediately. □

Although it is simpler to estimate the DS bound than the DEO bound, the sample average estimator of the DEO bound has zero variance when the asset is fast and frictionless, because in this case the penalties are ideal (Brown et al. 2010, Theorem 2.3), while the analogous estimator of the DS bound does not have this property. In the general case, it is however unclear if the sample average estimator of the DEO bound might be more precise (have smaller variance) than the analogous estimator of the DS bound. This issue is investigated numerically in §6.

Obviously, under Assumption 1 the dual dynamic program (49) can be efficiently solved with each dual penalty $P_n(a, x, \boldsymbol{G})$ set equal to $P_n^{EO}(a, x, \boldsymbol{G})$ or $P_n^{S}(a, x, \boldsymbol{G})$. However, because the latter penalties are linear, the resulting dual dynamic programs can be equivalently reformulated as linear programs. These linear programs are similar to the I linear program (4)-(10), but are defined with respect to the sample path $\boldsymbol{G}$ and include additional terms in their respective objective functions. In particular, the linear program corresponding to the penalties (53) is

$$\max \sum_{n\in\mathcal{N}} \delta^n[s_n^W(\boldsymbol{G}) + s_{n+1}(\boldsymbol{G}) - F_{n,n+1}(\boldsymbol{G})]w_n - \sum_{n\in\mathcal{N}} \delta^n[s_n^I + s_{n+1}(\boldsymbol{G}) - F_{n,n+1}(\boldsymbol{G})]u_n$$
$$- \sum_{n\in\mathcal{N}} \delta^n[s_{n+1}(\boldsymbol{G}) - F_{n,n+1}(\boldsymbol{G})]x_n \text{ s.t. (5)-(10).}$$

The linear program associated with the penalties (51) differs from this one only because the second term in the difference (54) instantiated with $m = 0$ is subtracted from its objective function. The equivalence (at optimality) between the I linear programs (4)-(10) and (19) implies that the linear program (19) can be similarly modified, also using (11) and (12), to obtain two linear programs for estimating the DS and DEO bounds. Current implementations of the RI and RSO policies, in particular commercial ones (FEA 2007, KYOS 2009, Lacima 2010), can thus be easily adapted to estimate these dual upper bounds.

When the storage asset is fast and frictionless, the DEO and DS upper bounds, in addition to being tight, coincide with the dual upper bounds of Lai et al. (2010) and Nadarajah et al. (2013), because in this case the approximate value function computed by the ADPs of these authors has the same slope of the exact value function with respect to inventory (ignoring price discretization error when estimating their upper bounds). When there are frictions and the storage asset is fast, the optimal value function is linear in inventory (Charnes et al. 1966) and all these upper bounds are based on linear penalties, because the ADPs of Lai et al. (2010) and Nadarajah et al. (2013) yield approximate value functions that are linear in inventory, but the penalties used to obtain the DEO and DS upper bounds ignore frictions. If the storage asset is slow, with or without frictions, the optimal value function is piecewise linear concave in inventory (see Property 1 in §2) and the dual upper bounds of Lai et al. (2010) and Nadarajah et al. (2013) rely on a value function approximation that satisfies this property, while the DEO and DS upper bounds continue to use linear penalties. These considerations suggest that the dual upper bounds of these authors should be stronger than the DEO and DS upper bounds when the storage asset is slow or there are frictions.

As mentioned at the beginning of this section, the dual upper bounds of Lai et al. (2010) and Nadarajah et al. (2013) hinge on approximate value functions obtained by numerically solving

ADPs, which can be computationally expensive. Specifically, these value functions are encoded using look-up tables (grids) that in a given stage depend on the inventory level and spot price, as well as the prompt month futures price for the best bound in Nadarajah et al. (2013). Estimating these bounds requires accessing these look-up tables multiple times to compute dual penalties when solving (49). In contrast, the dual penalties used by the DS and DEO upper bounds are available in closed form and essentially closed form, respectively. Thus, for a given number of forward curve Monte Carlo samples, estimating the DS upper bound when using (49) for dual optimization is faster than estimating the upper bounds of Lai et al. (2010) and Nadarajah et al. (2013). Since the exchange option terms used to obtain the DEO upper bound do not depend on the inventory level, they can be computed once for every sample path used to estimate this bound, in particular *before* performing the dual optimization for a given sample path. It seems reasonable to assume that this precomputation incurs a smaller overhead than accessing look-up tables when solving (49). Under this assumption, and given a number of forward curves Monte Carlo samples, estimating the DEO upper bound by using (49) for dual optimization is faster than estimating the upper bounds of Lai et al. (2010) and Nadarajah et al. (2013).

Section 6 numerically compares the quality of and computational performance of estimating the DEO and DS upper bounds and the best dual upper bound of Nadarajah et al. (2013), because the numerical results of these authors indicate that this dual upper bound outperforms the one of Lai et al. (2010).

6. Numerical Study

This section presents a set of new natural gas storage instances in §6.1. In §6.2 it applies to these instances the policies presented in §3 and the upper bounds developed in §5, and it also introduces and investigates a variant of the RSO policy motivated by the somewhat inferior performance of the RSO policy on some fast storage asset instances.

6.1 Instances

The instances used in this study extend the natural gas storage instances of Lai et al. (2010). There are forty-eight instances obtained by considering three values for the number of stages, N, four valuation dates in different seasons, and four capacity pairs. Each instance is labeled according to the N-Season-Capacity pattern.

The stages correspond to futures price maturities. Natural gas futures have monthly maturities. Inventory trading decisions are thus made on a monthly basis. The possible values for the number of stages are 24, 36, and 48 (Lai et al. 2010 consider 12 and 24 monthly stages).

The valuation dates correspond to the following four dates in the Spring, Summer, Fall, and Winter seasons, abbreviated to Sp, Su, Fa, and Wi, respectively: 3/1/2012, 6/1/2012, 9/4/2012, and 12/3/2012. These dates are the first trading days in March, June, December, and September 2012, and are the analogues of the 2006 dates considered by Lai et al. (2010), with the exception that 8/31/2006 is replaced with 9/4/2012 (Lai et al. 2010 used 8/31/2006 instead of 9/1/2006 because they did not have prices for options on natural gas futures for 9/1/2006, but such prices are not used here). The risk free discount factor depends on the valuation date and is computed using the one year treasury rates reported by the U.S. Department of Treasury observed on the valuation dates for these instances: 0.18%, 0.17%, 0.16%, and 0.18% for the Sp, Su, Fa, and Wi instances, respectively.

The four injection and withdrawal capacity pairs (C^I, C^W) are $(-0.15, 0.30)$, $(-0.30, 0.60)$, $(-0.45, 0.90)$, and $(-1.00, 1.00)$, and are labeled as capacity pairs 1, 2, 3, and 4, respectively (Lai et al. 2010 do not consider capacity pair 4). The storage asset thus becomes faster when the value of the capacity pair label increases, and it is fast when this label is equal to 4. As in Lai et al. (2010), the maximum inventory, $\overline{x}$, is normalized to 1 and the initial inventory, x_0, is 0. Assumption 1 is thus satisfied in all these instances with lot size, Q, equal to 0.05, 0.10, 0.05, and 1.00 for capacity pairs 1, 2, 3, and 4, respectively. Hence, the feasible inventory set $\mathcal{X}$ can be optimally discretized using 21, 11, 21, and 2 values, respectively, for these capacity pairs. Following Lai et al. (2010), the injection and withdrawal marginal costs, c^I and c^W, are \$0.02 and \$0.01 per unit, respectively, and the injection and withdrawal fuel coefficients, ϕ^I and ϕ^W, are 1.01 and 0.99, respectively.

As in Lai et al. (2010), the forward curve evolves according to the extended Black model:

$$dF(t, T_n)/F(t, T_n) = \sigma_n dZ_n(t), \ \forall n \in \mathcal{N} \setminus \{0\}, \ t \in [0, T_n), \tag{55}$$

$$dZ_n(t) dZ_m(t) = \rho_{n,m} dt, \ \forall n, m \in \mathcal{N} \setminus \{0\}, \ t \in [0, T_n \wedge T_m), \tag{56}$$

where $F(t, T_n)$ is the time t price of the futures contract with maturity on date T_n ($F(t, T_n) \equiv F_{n,m}$), σ_n and $dZ_n(t)$ are the volatility and standard Brownian motion increment, respectively, corresponding to this price, and $\rho_{n,m}$ is the instantaneous correlation between $dZ_m(t)$ and $dZ_m(t)$. The volatility and correlation parameters of this model are estimated using a principal component analysis of daily futures price returns observed between 2003 and 2012 (Clewlow and Strickland 2000, §8.6). The closing NYMEX natural gas forward curves on the four valuation dates are used as the time 0 forward curves for each of these dates. These forward curves and the estimates of the parameters of model (55)-(55) are available upon request.

6.2 Results

The estimated lower and upper bounds discussed in the ensuing analysis are expressed as percentages relative to the estimated values of the UB2 dual upper bound proposed by Nadarajah et al. (2013), which in the numerical investigation of these authors outperforms all the other upper bounds they consider, including the upper bound of Lai et al. (2010). These ratios are referred to as *percentage qualities*. The values of the estimated UB2 upper bounds are available in Table 9 in Appendix B.

Tables 1, 2, and 3 report the percentage quality of the I, RI, SO, and RSO policies, and the EO, DEO, DS, and UB2 upper bounds for the 24, 36, and 48 stage instances, respectively – the DEO1 and DEO2 upper bounds in these tables are two versions of the DEO upper bound estimated with different number of samples, as now explained. The lower bounds corresponding to the RI, SO, and RSO policies and the DEO2, DS, and UB2 upper bounds are evaluated using 10,000 Monte Carlo forward curve samples; the DEO1 upper bound is estimated using 1,000 such samples. The I policy lower bounds and the EO upper bounds are computed exactly, the latter using the Margrabe (1978) formula. Moreover, the EO upper bounds do not depend on the capacity pair that defines an instance. However, since the UB2 upper bound does depend on the capacity pair, the ratios of the EO upper bounds and the UB2 upper bound estimates vary when the value of the capacity pair label changes. A percentage quality of a policy above 100 is due to Monte Carlo sampling error.

Table 1: Percentage quality of the computed lower and upper bounds on the 24 stage instances.

	Lower Bound				Upper Bound			
Instance	I	RI	SO	RSO	EO	DEO1	DEO2	DS
24-Sp-1	86.91	100.58	91.43	99.57	171.51	101.81	102.03	101.88
24-Sp-2	80.82	100.29	85.91	98.08	141.74	101.23	101.58	101.46
24-Sp-3	78.38	100.04	83.29	97.38	130.65	100.87	101.17	101.06
24-Sp-4	77.07	99.80	80.20	96.12	120.71	100.52	100.70	100.60
24-Su-1	62.05	97.92	82.98	96.09	242.23	101.75	101.94	101.65
24-Su-2	62.98	98.27	81.53	96.08	173.57	100.64	100.83	100.62
24-Su-3	63.51	98.35	79.91	96.09	147.60	99.99	100.21	100.04
24-Su-4	63.23	98.41	72.95	95.00	127.97	99.43	99.58	99.42
24-Fa-1	62.28	98.12	83.22	96.30	242.23	101.97	102.16	101.87
24-Fa-2	63.17	98.47	80.91	96.31	173.57	100.82	101.01	100.81
24-Fa-3	63.68	98.54	80.07	96.31	147.60	100.16	100.38	100.20
24-Fa-4	63.38	98.59	73.14	95.19	127.97	99.59	99.73	99.58
24-Wi-1	50.41	97.71	77.76	95.70	235.67	104.24	104.44	104.30
24-Wi-2	48.19	97.69	78.88	94.82	173.97	103.37	103.48	103.38
24-Wi-3	48.10	98.17	71.91	93.86	153.48	103.18	103.22	103.13
24-Wi-4	45.49	98.60	56.47	87.19	136.45	100.80	100.61	100.53

Table 2: Percentage quality of the computed lower and upper bounds on the 36 stage instances.

	Lower Bound				Upper Bound			
Instance	I	RI	SO	RSO	EO	DEO1	DEO2	DS
36-Sp-1	76.41	100.63	88.27	98.72	179.67	101.07	101.99	101.92
36-Sp-2	71.17	100.70	81.25	97.55	147.31	101.10	101.29	101.23
36-Sp-3	68.83	100.65	80.84	96.90	135.03	100.93	100.90	100.84
36-Sp-4	66.75	100.53	79.73	94.93	123.89	100.72	100.43	100.38
36-Su-1	76.10	99.01	87.11	97.08	190.85	99.85	100.40	100.32
36-Su-2	74.95	99.08	83.16	96.35	143.99	99.79	99.87	99.81
36-Su-3	71.90	98.91	83.45	95.62	132.65	99.70	99.64	99.59
36-Su-4	68.67	98.83	82.90	93.98	123.07	99.67	99.48	99.43
36-Fa-1	57.23	99.20	81.58	96.49	236.75	99.74	101.15	100.97
36-Fa-2	57.53	99.46	77.17	95.48	173.44	99.65	100.08	99.95
36-Fa-3	57.24	99.45	79.17	95.44	149.04	99.47	99.60	99.49
36-Fa-4	55.64	99.48	75.37	94.49	129.28	99.10	99.06	98.96
36-Wi-1	48.06	98.86	80.15	95.97	233.07	101.80	103.15	102.90
36-Wi-2	45.71	99.15	75.59	94.96	172.45	101.88	102.17	101.99
36-Wi-3	44.87	99.58	73.62	94.44	152.30	101.78	101.91	101.74
36-Wi-4	41.87	99.69	63.55	89.42	134.97	100.05	100.07	99.92

Table 3: Percentage quality of the computed lower and upper bounds on the 48 stage instances.

	Lower Bound				Upper Bound			
Instance	I	RI	SO	RSO	EO	DEO1	DEO2	DS
48-Sp-1	67.37	98.85	85.16	95.76	184.14	100.80	100.45	100.73
48-Sp-2	62.90	99.27	81.34	95.07	149.47	100.46	100.17	100.40
48-Sp-3	60.82	99.34	80.39	94.75	136.69	100.19	99.92	100.13
48-Sp-4	58.71	99.34	78.36	93.27	125.12	99.73	99.56	99.75
48-Su-1	69.51	98.59	84.86	95.64	195.63	100.72	100.43	100.59
48-Su-2	67.92	99.30	82.89	95.73	147.23	100.28	100.08	100.20
48-Su-3	64.98	99.34	82.92	95.34	134.69	100.11	99.91	100.02
48-Su-4	61.70	99.09	82.34	93.59	124.30	99.81	99.73	99.83
48-Fa-1	53.20	97.87	80.13	94.20	230.47	100.92	100.86	101.20
48-Fa-2	53.03	98.84	79.20	94.05	170.26	100.21	100.27	100.52
48-Fa-3	52.65	99.12	80.07	94.41	147.89	99.83	99.94	100.16
48-Fa-4	50.80	99.11	77.38	93.75	129.34	99.40	99.51	99.70
48-Wi-1	45.69	97.94	80.42	94.02	228.99	101.99	101.67	102.14
48-Wi-2	43.46	98.60	77.81	93.84	169.23	101.48	101.16	101.51
48-Wi-3	42.62	99.06	75.98	93.36	149.77	101.32	100.96	101.27
48-Wi-4	39.84	99.34	68.49	89.92	133.50	99.79	99.60	99.88

Specifically, the ranges of the standard errors of the estimated RI, SO, RSO, DEO1, DEO2, DS, and UB2 bounds are 0.006-0.014, 0.006-0.014, 0.007-0.015, 0.007-0.013, 0.002-0.004, 0.003-0.008, and 0.002-0.008, respectively. To facilitate comparison of the displayed percentage qualities, the ranges of these standard errors expressed as percentages of their respective UB2 estimates are 0.50-1.13%, 0.46-1.08%, 0.52-1.20%, 0.39-1.48%, 0.12-0.45%, 0.26-0.64%, and 0.23-0.43%.

Tables 4-6 report the Cpu seconds required to estimate the various lower and upper bounds on the considered instances, excluding the I lower bound, the EO lower bound, and the RMSO lower bounds (the Cpu time of each RMSO policy is essentially the same as the Cpu time of the RSO policy). The models are implemented in C++ and compiled using the g++ 4.8.2 20131017 (Red Hat 4.8.2-1) compiler. The SO linear program is implemented by evaluating the spread options that appear in its objective function using the closed form lower bounding approach proposed by Bjerksund and Stensland (2011), and is solved using Gurobi 5.0 (Gurobi Optimization 2012) with a single thread. The RI policy is obtained by reoptimization of the I dynamic program. The estimation of the DEO and DS upper bounds is based on solving (49) for dual optimization. The UB2 upper bound is estimated using the code of Nadarajah et al. (2013). The reported Cpu times are obtained on a 64 bits PowerEdge R515 with twelve AMD Opteron 4176 2.4GHz processors, of which only one is used, with 64GB of memory, and the Linux Fedora 19 operating system.

Lower bounds. The lower bound results displayed in Tables 1-3 are largely consistent with the ones reported by Lai et al. (2010). Reoptimization of the I and SO policies is critical to obtain near optimal policies. The respective ranges of the percentage quality of the I and SO lower bounds are 39.84-86.91 and 56.47-91.43. The ratios of the grand averages of the estimated I and SO lower bounds, respectively, and the grand average of the estimated UB2 upper bounds are 60.58% and 79.53%. These percentage quality ranges and grand average ratios for the RI and

Table 4: Cpu seconds required to estimate the bounds on the 24 stage instances.

	Lower Bound			Upper Bound				UB2	
Instance	RI	SO	RSO	DEO1	DEO2	DS	UB2	DUB	ADP
24-Sp-1	15.89	0.36	147.79	0.47	4.33	1.65	195.12	83.12	112.00
24-Sp-2	7.66	0.36	135.73	0.39	3.68	1.00	146.07	52.67	93.40
24-Sp-3	33.24	0.36	127.80	0.63	6.05	3.41	240.62	129.04	111.58
24-Sp-4	1.09	0.35	100.62	0.34	3.09	0.41	106.47	30.22	76.25
24-Su-1	15.93	0.36	141.68	0.47	4.35	1.64	193.95	81.77	112.18
24-Su-2	7.65	0.36	123.34	0.40	3.71	1.00	146.88	53.31	93.57
24-Su-3	33.17	0.36	118.33	0.65	6.06	3.42	240.36	128.86	111.50
24-Su-4	1.10	0.35	93.31	0.33	3.11	0.41	107.05	30.52	76.53
24-Fa-1	15.92	0.38	140.87	0.47	4.34	1.64	194.34	82.21	112.13
24-Fa-2	7.65	0.35	122.50	0.40	3.72	1.00	145.84	52.40	93.44
24-Fa-3	33.19	0.36	117.39	0.64	6.05	3.42	242.32	130.34	111.98
24-Fa-4	1.10	0.38	94.21	0.33	3.15	0.41	107.84	30.78	77.06
24-Wi-1	15.92	0.37	143.52	0.45	4.30	1.66	193.57	81.42	112.15
24-Wi-2	7.66	0.36	131.71	0.39	3.69	1.00	146.55	52.79	93.76
24-Wi-3	33.26	0.36	125.03	0.63	6.03	3.41	242.89	131.17	111.72
24-Wi-4	1.10	0.36	98.25	0.34	3.08	0.40	106.77	30.19	76.58

Table 5: Cpu seconds required to estimate the bounds on the 36 stage instances.

	Lower Bound			Upper Bound				UB2	
Instance	RI	SO	RSO	DEO1	DEO2	DS	UB2	DUB	ADP
36-Sp-1	42.20	0.83	528.41	1.03	9.28	2.94	287.33	130.00	157.33
36-Sp-2	18.58	0.81	454.26	0.92	8.25	1.83	211.92	81.02	130.90
36-Sp-3	80.88	0.81	429.42	1.29	11.87	5.62	362.55	203.89	158.66
36-Sp-4	2.65	0.83	334.67	0.83	7.35	0.90	155.24	47.42	107.82
36-Su-1	39.09	0.82	501.73	1.03	9.30	2.95	287.55	128.68	158.87
36-Su-2	18.58	0.82	435.07	0.92	8.21	1.82	212.79	81.58	131.21
36-Su-3	80.71	0.81	403.60	1.30	11.89	5.66	361.32	203.14	158.18
36-Su-4	2.63	0.82	315.50	0.84	7.28	0.89	155.06	47.39	107.67
36-Fa-1	39.03	0.82	484.31	1.02	9.28	2.97	287.65	128.67	158.98
36-Fa-2	18.63	0.82	415.78	0.93	8.20	1.83	211.96	81.59	130.37
36-Fa-3	80.64	0.82	388.44	1.29	11.87	5.68	362.26	204.38	157.88
36-Fa-4	2.63	0.82	309.31	0.82	7.26	0.91	154.74	47.15	107.59
36-Wi-1	39.00	0.83	488.19	1.00	9.32	3.00	286.13	128.12	158.01
36-Wi-2	18.55	0.82	438.33	0.92	8.22	1.85	212.17	81.00	131.17
36-Wi-3	80.78	0.83	406.05	1.29	11.89	5.61	360.96	203.15	157.81
36-Wi-4	2.63	0.83	326.77	0.82	7.41	0.91	154.96	47.22	107.74

Table 6: Cpu seconds required to estimate the bounds on the 48 stage instances.

	Lower Bound			Upper Bound				UB2	
Instance	RI	SO	RSO	DEO1	DEO2	DS	UB2	DUB	ADP
48-Sp-1	74.67	1.54	1247.08	1.90	16.26	4.59	372.05	177.17	194.88
48-Sp-2	35.05	1.55	1087.49	1.75	14.72	3.04	274.56	112.63	161.93
48-Sp-3	149.02	1.55	1008.89	2.28	19.75	8.31	470.08	277.61	192.47
48-Sp-4	4.73	1.55	810.51	1.62	13.39	1.68	197.78	65.01	132.77
48-Su-1	74.74	1.53	1224.59	1.80	16.33	4.58	368.57	175.89	192.68
48-Su-2	35.05	1.55	1061.52	1.68	14.80	3.01	271.99	111.72	160.27
48-Su-3	148.59	1.54	984.97	2.27	19.83	8.26	473.14	280.39	192.75
48-Su-4	4.73	1.53	794.08	1.61	13.38	1.70	197.93	65.05	132.88
48-Fa-1	74.78	1.54	1152.86	1.91	16.28	4.46	368.09	175.20	192.89
48-Fa-2	35.12	1.55	1029.99	1.74	14.69	3.02	272.92	111.62	161.30
48-Fa-3	148.71	1.55	940.63	2.27	19.84	8.25	470.09	276.97	193.12
48-Fa-4	4.77	1.54	761.82	1.62	13.44	1.68	196.07	64.04	132.03
48-Wi-1	74.67	1.54	1156.20	1.90	16.27	4.58	369.91	176.60	193.31
48-Wi-2	35.03	1.55	1047.87	1.76	14.78	3.03	270.65	110.15	160.50
48-Wi-3	148.58	1.54	1002.66	2.25	19.77	8.27	468.42	275.88	192.54
48-Wi-4	4.72	1.55	788.28	1.64	13.49	1.66	197.92	64.88	133.04

RSO lower bounds, respectively, are 97.69-100.70 and 87.19-99.57 and 99.19% and 94.92%. The two reoptimization-based policies are thus both near optimal. However, the RI policy exhibits less variability in its percentage quality across instances than the RSO policy, outperforms this policy by 4.51% on average (this value is the grand average percentage improvement of the estimated RI lower bound relative to the estimated RSO lower bound), and also dominates it on all the instances.

There are also three notable exceptions to the near optimality of the RSO policy. These exceptions occur on the 24-Wi-4, 36-Wi-4, and 48-Wi-4 instances, for which the percentage quality of the RSO policy is 87.19%, 89.42%, and 89.92%, respectively. In contrast, these figures are 98.60%, 99.69%, and 99.34% for the RI policy. These instances correspond to a fast storage asset. Indeed, for each given combination of number of stages and season, the RSO policy achieves its worst performance when the storage asset is fast (capacity pair equal to 4), while the RI policy is near optimal even on these instances. The RSO policy thus struggles when the storage asset is fast, even thought both the SO and RSO policies are optimal when the storage asset is fast *and* frictionless, as mentioned at the beginning of §4.1, and the frictions are small for these instances.

Compared to a slow storage asset, the *optimal* management of a fast storage asset is simpler, in the sense that if it is optimal to trade, in a given stage and for a given forward curve, then the storage asset should either be filled up or emptied (Charnes et al. 1966), while this is not generally the case for a slow storage asset (Secomandi 2010, Example 3). Hence, starting from an empty storage asset in the initial stage, as in the instances considered here, an optimally managed fast storage asset is either empty or full at the start of every subsequent stage and for every forward curve in this stage. By Proposition 3, the same is true of a fast storage asset managed under the RSO policy. However, in this case a mistake can be more costly than in the case of a slow storage asset, because (i) the effect of using a wrong action is proportional to the difference between the (absolute values of the) sizes of the wrong action and the optimal action, and (ii) this difference is

Table 7: Performance of the best RMSO policy; each triple includes the percentage quality of this policy, the difference between the percentage quality of this policy and the one of the RSO policy, and the value of λ corresponding to the best RMSO policy.

Instance		Instance		Instance	
24-Sp-1	$(99.63, 0.05, 0.8)$	36-Sp-1	$(98.85, 0.13, 0.7)$	48-Sp-1	$(95.98, 0.21, 0.7)$
24-Sp-2	$(98.26, 0.18, 0.8)$	36-Sp-2	$(97.78, 0.24, 0.7)$	48-Sp-2	$(95.58, 0.51, 0.6)$
24-Sp-3	$(97.62, 0.24, 0.6)$	36-Sp-3	$(97.24, 0.34, 0.7)$	48-Sp-3	$(95.25, 0.50, 0.6)$
24-Sp-4	$(96.68, 0.57, 0.5)$	36-Sp-4	$(95.94, 1.01, 0.4)$	48-Sp-4	$(94.45, 1.18, 0.5)$
24-Su-1	$(96.14, 0.05, 0.9)$	36-Su-1	$(97.21, 0.14, 0.7)$	48-Su-1	$(95.75, 0.12, 0.8)$
24-Su-2	$(96.08, 0.00, 0.9)$	36-Su-2	$(96.56, 0.21, 0.8)$	48-Su-2	$(95.93, 0.20, 0.6)$
24-Su-3	$(96.17, 0.08, 0.9)$	36-Su-3	$(95.91, 0.29, 0.6)$	48-Su-3	$(95.63, 0.29, 0.6)$
24-Su-4	$(95.07, 0.08, 0.8)$	36-Su-4	$(94.63, 0.65, 0.5)$	48-Su-4	$(94.22, 0.63, 0.5)$
24-Fa-1	$(96.35, 0.06, 0.9)$	36-Fa-1	$(96.64, 0.15, 0.7)$	48-Fa-1	$(94.56, 0.36, 0.7)$
24-Fa-2	$(96.31, 0.00, 1.0)$	36-Fa-2	$(95.65, 0.17, 0.8)$	48-Fa-2	$(94.46, 0.40, 0.7)$
24-Fa-3	$(96.36, 0.05, 0.9)$	36-Fa-3	$(95.62, 0.18, 0.7)$	48-Fa-3	$(94.79, 0.38, 0.7)$
24-Fa-4	$(95.28, 0.10, 0.9)$	36-Fa-4	$(94.88, 0.38, 0.6)$	48-Fa-4	$(94.37, 0.62, 0.6)$
24-Wi-1	$(95.70, 0.00, 1.0)$	36-Wi-1	$(96.12, 0.15, 0.8)$	48-Wi-1	$(94.20, 0.18, 0.7)$
24-Wi-2	$(94.87, 0.05, 0.9)$	36-Wi-2	$(95.34, 0.38, 0.7)$	48-Wi-2	$(94.31, 0.47, 0.6)$
24-Wi-3	$(93.96, 0.10, 0.7)$	36-Wi-3	$(94.84, 0.39, 0.7)$	48-Wi-3	$(94.00, 0.64, 0.6)$
24-Wi-4	$(91.86, 4.66, 0.5)$	36-Wi-4	$(92.84, 3.42, 0.4)$	48-Wi-4	$(92.63, 2.72, 0.5)$

likely to be larger with a fast storage asset than with a slow storage asset due the more stringent capacity constraints in the latter case. These arguments and the numerical results discussed earlier suggest that the RSO policy makes more mistakes than the RI policy and is penalized more heavily for making these mistakes when the storage asset is fast.

This finding is new relative to the work of Lai et al. (2010), whose instances do not include a fast storage asset. It is also important to point out that there are two differences between the version of the SO linear program used here and the version considered by Lai et al. (2010): (i) The valuation of the spread options by the formula of Bjerksund and Stensland (2011) here rather than the approximation of Kirk (Carmona and Durrleman 2003) in Lai et al. (2010), and (ii) the presence of forward sales for all the possible maturities in the objective of the SO linear program rather than only the spot sale. These differences do not seem to cause appreciable performance differences between the resulting policies. Applied to the instances of Lai et al. (2010), the versions of the SO and RSO policies used here yield lower bounds that vary between 97.27% and 101.87% of the ones reported by Lai et al. (2010), and the ratios of the grand averages of the SO and RSO lower bounds estimated here and the ones estimated by Lai et al. (2010) are 99.13% and 99.43%, respectively. Thus, the identified brittleness of the RSO policy does not appear attributable to the implementation of the SO linear program used in this paper.

A reason why the RSO policy might make more mistakes than the RI policy is that in the presence of frictions the SO linear program optimizes a *lower bound* on the value of a given SO policy (see (21) on page 8). As discussed at the end of §4.1, in this case exact optimization of this policy would be more involved than solving a linear program. A simpler alternative, investigated below, is to replace the objective function of the SO linear program with an average of a lower bound and an upper bound on the value of the SO policy, in the hope that this average might be closer to the true value of this policy. Natural choices here are the objective function of the SO

Table 8: Summary of the performance of the RMSO policies.

λ											
0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	Best
Minimum Percentage Quality											
85.49	85.54	88.83	90.78	91.53	91.86	91.79	87.69	87.66	87.44	87.19	91.86
Maximum Percentage Quality											
98.85	99.02	99.18	99.32	99.40	99.49	99.57	99.57	99.63	99.62	99.57	99.63
Grand Average (Percent of UB2 Grand Average)											
92.36	92.60	93.87	94.60	95.12	95.36	95.40	95.22	95.17	95.06	94.92	95.47
Number of Times Optimal											
0	0	0	0	2	6	10	14	7	7	2	

model itself for the lower bound and the objective function that results from ignoring frictions for the upper bound, the latter of which reduces to a collection of exchange options. Specifically, define as $E_{l,m,n}(\boldsymbol{F}_l) := \delta^{m-l}\mathbb{E}\left[\left(\delta^{n-m}\tilde{F}_{m,n} - \tilde{s}_m\right)^+ \mid \boldsymbol{F}_l\right]$ the time T_l value of the exchange option with payoff equal to the positive part of the difference $\delta^{n-m}F_{m,n} - s_m$. Denoting by $\lambda \in [0,1]$ the weight given to each spread option term (one could use other averaging schemes, e.g., different weights for different options), the proposed weighted-average objective function is

$$\sum_{n=0}^{N-1} \delta^n F_{0,n}^W z_n + \sum_{n=0}^{N-2} \sum_{m=n+1}^{N-1} \left[\lambda S_{0,n,m}(\boldsymbol{F}_0) + (1-\lambda) E_{0,n,m}(\boldsymbol{F}_0)\right] q_{n,m}. \tag{57}$$

The resulting linear program, labeled as the MSO linear program, is analogous to the SO linear program but maximizes (57). In particular, the MSO and SO linear programs coincide when λ equals 1. Sequential reformulation and reoptimization of the MSO model yields the RMSO policy.

Table 7 displays the percentage quality of the *best* RMSO policy for a set of RMSO policies corresponding to values of λ ranging from 0.0 to 1.0 in increments of 0.1, the difference between the percentage quality of this policy and the one of the RSO policy, as well the value of λ corresponding to the best RMSO policy for all the considered instances. The percentage quality of the best RMSO policy varies between 91.86 and 99.63, whereas, as previously discussed, the percentage quality range of the RSO policy is 87.19-99.57. Thus, the best RMSO policy appears to be effective when the RSO policy struggles. Indeed, while the differences between the percentage qualities of the best RMSO policy and of the RSO policy are generally small, less than 1.00, they are equal to 4.66, 3.42, and 2.72 on the 24-Wi-4, 36-Wi-4, and 48-Wi-4 instances, respectively, on which the RSO policy achieves its worst performance. There is no single value of λ that yields the best RMSO policy across all the considered instances. Overall, the grand average of the estimated best RMSO policy lower bounds expressed as a ratio of the grand average of the estimated UB2 upper bounds is 95.47%, which is only 0.58% larger than the analogous grand average for the RSO policy (94.92%).

Table 8 summarizes the performance of each of the considered RMSO policies – Tables 10-12 in Appendix C report the percentage quality of these policies on all the instances – and of the best RMSO policy. Much of the improvement on the RSO policy achieved by the best RMSO policy can be obtained by the RMSO policy corresponding to λ equal to 0.6: Its percentage quality varies between 91.79 and 99.57, the grand average of its estimated lower bounds is 95.40% of the grand average of the estimated UB2 upper bounds, which corresponds to a 0.51% improvement on the grand average of the estimated RSO lower bounds, and is the best RMSO policy ten times, whereas the RSO policy, which is the RMSO policy with λ equal to 1.0, is optimal only twice. Despite the small overall improvement brought about by using this RMSO policy relative to the RSO policy, using this RMSO policy seems important to avoid the occasional poor performance exhibited by the RSO policy. However, even the best RMSO policy is outperformed by the RI policy.

Computing and evaluating the I policy is extremely fast (always taking less than 0.00001 Cpu seconds), because it amounts to solving the deterministic I dynamic program. As shown in Tables 4-6, evaluating the SO policy is more expensive, taking about 0.91 Cpu seconds on average with a range of 0.35-1.55 Cpu seconds, because it also involves Monte Carlo simulation. Reoptimization has its price: Evaluating the RI and RSO policies, respectively, takes 35.58 and 515.03 Cpu seconds on average, with respective ranges equal to 1.09-149.02 and 93.31-1247.08. The slower evaluation of the RSO policy compared to the RI policy is due to the repeated computation of the spread option values in the objective function of the SO model, (20), and the fact that the RI policy is implemented by reoptimizing the I deterministic dynamic program rather than the I linear program (19), which is analogous to the SO model. These comparisons are consistent with the ones of Lai et al. (2010). As mentioned earlier, the Cpu times of the RMSO policies are not included here because they are comparable to the ones of the RSO policy.

The reported Cpu seconds obviously increase when the number of stages becomes larger. For a given number of stages and a given capacity pair label, these Cpu times are insensitive to the season. Fixing the number of stages and the season, the Cpu requirement of the RI policy becomes larger when the value of the capacity pair label changes from 4 to 2, then from 2 to 1, and finally from 1 to 3. These increases are due to the corresponding changes in the lot size Q and, hence, in the number of values used to discretize the set $\mathcal{X}$ and each set $\mathcal{A}(x_n)$ (see the I model, (3)). Moreover, even though the capacity pair labels 1 and 3 share the same lot size, their respective Cpu times differ because the I dynamic program is solved by considering only inventory levels that can be reached in a given stage, and this number is smaller when the value of the capacity pair label equals 1 rather than 3. Given the number of stages and the season, the Cpu times of the SO policy are rather insensitive to the value of capacity pair label, while the ones of the RSO policy decrease when the storage asset becomes faster (that is, the value of the capacity pair label changes from 1 to 4).

Upper bounds. As discussed in §5.2, the DEO and DS upper bounds are identical in theory, but their estimators have different precisions (e.g., only the DEO upper bound estimator has zero variance when the storage asset is fast and frictionless), and estimating the DS upper bound requires less computational effort than estimating the DS upper bound when using the same number of forward curve Monte Carlo samples. As shown in Tables 1-3, the estimated DEO and DS upper bounds are fairly similar: The percentage quality ranges of the estimated DEO1, DEO2, and DS upper bounds are 99.10-104.24, 99.06-104.44, and 98.96-104.30, respectively, and their respective grand averages reported as percentages of the estimated UB2 upper bound grand average are 100.48%, 100.52%, and 100.55%. However, the DEO1, DEO2, and DS upper bound estimates differ in their precisions and computational requirements (the latter are reported in Tables 4-6). The DEO2 upper bound estimates are more precise than the DS upper bound estimates: Recall that the standard errors of the DEO2 and DS upper bound estimates vary between 0.002 and 0.004 and 0.003 and 0.008, respectively, and that these bounds are estimated using the same number of samples (10,000). Thus, including the zero mean intercept terms (52) in the definition of the dual penalties appears useful from a precision perspective. However, the Cpu seconds needed to estimate the DEO2 upper bounds exceed the ones required to estimate the DS upper bounds by about one order of magnitude: These Cpu seconds vary between 3.08 and 13.49 and 0.40 and 1.66, respectively. Reducing from 10,000 to 1,000 the number of samples used to estimate the DEO upper bounds, that is, estimating the DEO1 upper bounds instead of the DEO2 upper bounds, yields substantial computational savings: The Cpu seconds needed to estimate the DEO1 upper bounds range from 0.33 to 1.64. However, the DEO1 upper bound estimates are less precise than both the DEO2 and DS upper bound estimates, with standard errors varying between 0.007 and 0.013. Estimating the DEO1 upper bounds is also faster than estimating the DS upper bounds, but the DS upper bounds estimates are more precise than the DEO1 upper bound estimates.

As established in §5.2, the DEO and DS upper bounds cannot be worse than the EO upper bound. Moreover, as discussed at the end of §5.2, it is reasonable to expect that the UB2 upper bound should outperform the DEO and DS upper bounds. Tables 1-3 show that the percentage qualities of the estimated DEO and DS upper bounds are drastically better than the percentage quality of the EO upper bounds. Specifically, the ranges of the percentage qualities for the DEO1, DEO2, DS, and EO upper bounds are 99.10-104.24, 99.06-104.44, 98.96-104.30, and 120.71-242.23, and their respective grand averages expressed as percentages of the UB2 upper bound grand average are 100.48%, 100.52%, 100.55%, and 153.86%. These figures also indicate that the DEO and DS upper bounds are competitive with the UB2 upper bound.

As discussed in §5.2, it is reasonable to expect that estimating the DEO and DS upper bounds should be faster than estimating the UB2 upper bound. Tables 4-6 indicate that the estimation of the two DEO and the DS upper bounds is substantially faster than the estimation of the UB2 upper bound across all the considered instances. The Cpu times required to compute the UB2 estimates are one to three orders of magnitude larger than the Cpu times needed to estimate the other dual upper bounds: Estimating the UB2 upper bound takes between 1.77 and 7.89 Cpu *minutes*, with an average of 4.19 Cpu minutes, while the estimation effort of the DEO and DS upper bounds is always on the order of a few seconds or fractions thereof. Tables 4-6 also separate the Cpu seconds required to estimate the UB2 upper bound between the CPU seconds incurred for solving the ADP and for executing the dual upper bound (DUB) Monte Carlo simulation. Both phases contribute in a substantial manner to the overall Cpu time required to estimate the UB2 upper bound.

As expected, the Cpu seconds required to estimate the various dual upper bounds increase when the number of stages becomes larger. Given the number of stages and a capacity pair label, these times are fairly insensitive to the season. Fixing the number of stages and the season, these times depend on the value of the capacity pair label in a manner similar to how the Cpu times of the RI policy depend on it (changing the value of the lot side Q affects the difficulty of solving the dual optimization (49) and the I dynamic program in a similar manner).

These results suggest that it appears useful to enhance currently implementations of the RI or RSO policies, in particular commercial software (FEA 2007, KYOS 2009, Lacima 2010), with the estimation of the DEO or DS upper bounds, by making minimal changes to the optimization models used to obtain these policies. In particular, the DS upper bound seems to achieve a reasonable compromise between precision and computational effort.

7. Conclusions

This paper investigates the real option management of commodity storage assets. Practitioners solve the resulting stochastic optimization model heuristically using the RI and RSO policies, which are based on the sequential reoptimization of optimization models. Combined with Monte Carlo simulation, these policies can be used to estimate near optimal lower bounds on the value of storage. The numerical and structural analysis of this paper provides novel support for the use of these policies. This research also points out the superior performance of the RI policy relative to the RSO policy. It thus proposes a variant of the RSO policy that, on the considered instances, exhibits slightly improved average performance compared to the RSO policy but achieves more considerable improvement when the suboptimality of this policy is larger. This paper also introduces effective dual upper bounds that can be estimated with minimal modifications of the math programs used to obtain these policies. Compared to a known dual upper bound, these upper bounds display similar accuracy and comparable or improved precision but can be estimated with substantially less computational effort. They thus have immediate practical relevance.

A. Optimality Gaps

Proposition 6 shows that the suboptimality of the I, and SO, RI, and RSO policies is finite. Lemmas 1-4 are used to establish Proposition 6.

Lemma 1 is immediate and states that the value of an optimal policy when then storage asset is fast and there are no frictions, $V_0^{EO}(x_0, \boldsymbol{F}_0)$ (see Proposition 4 in §5.2), is an upper bound on the value of any feasible policy in the general case.

Lemma 1 (Upper bound)**.** $V_0^{\pi}(x_0, \boldsymbol{F}_0) \leq V_0^{EO}(x_0, \boldsymbol{F}_0)$, $\forall \pi \in \Pi$.

Lemma 2 establishes that the value of the SO policy as seen by the SO linear program is no smaller than the value of the I policy.

Lemma 2 (Comparison of the I and SO models)**.** $V_0^{I}(x_0, \boldsymbol{F}_0) \leq U_0^{SO}(x_0, \boldsymbol{F}_0)$.

Proof. Jensen's inequality, the properties of a futures price under the risk neutral measure (Shreve 2004, p. 244), and the inequality $(\cdot)^+ \geq \cdot$ imply

$$\begin{aligned} S_{m,n}(\boldsymbol{F}_0) = \delta^m \mathbb{E}\left[\left(\delta^{n-m}\tilde{F}^W_{m,n} - \tilde{s}^I_m\right)^+ \mid \boldsymbol{F}_0\right] &\geq \delta^m \left(\mathbb{E}\left[\delta^{n-m}\tilde{F}^W_{m,n} - \tilde{s}^I_m \mid \boldsymbol{F}_0\right]\right)^+ \\ &= \left(\delta^n F^W_{0,n} - \delta^m F^I_{0,m}\right)^+ \\ &\geq \left(\delta^n F^W_{0,n} - \delta^m F^I_{0,m}\right). \end{aligned} \quad (58)$$

The claimed result follows from (58) and the observation that the I linear program (19) and the SO linear program, (20), share the same constraint set. □

Lemma 3 compares the values of the I and SO policies. It follows from Lemma 2 and inequality (21).

Lemma 3 (Comparison of the I and SO policies)**.** $V_0^{I}(x_0, \boldsymbol{F}_0) \leq V_0^{SO}(x_0, \boldsymbol{F}_0)$.

Lemma 4 bounds the optimality gaps of the I, SO, RSO, and RSO policies.

Lemma 4 (Bound on optimality gaps)**.** *For all* $\pi \in \{I, SO, RI, RSO\}$ *it holds that* $V_0(x_0, \boldsymbol{F}_0) - V_0^{\pi}(x_0, \boldsymbol{F}_0) \leq V_0^{EO}(x_0, \boldsymbol{F}_0) - V_0^{I}(x_0, \boldsymbol{F}_0)$.

Proof. The claimed property holds by Lemma 1 for the I policy; by Lemma 1 and Proposition 1 for the RI policy; by Lemmas 1 and 3 for the SO policy; and by Lemma 1, part (a) of Proposition 2, and Lemma 2 for the RSO policy. □

Proposition 6 (Finite optimality gaps)**.** *$V_0(x_0, \boldsymbol{F}_0) - V_0^\pi(x_0, \boldsymbol{F}_0) < \infty$, $\forall \pi \in \{I, SO, RI, RSO\}$.*

Proof. It holds that $V_0^I(x_0, \boldsymbol{F}_0) \leq V_0^{EO}(x_0, \boldsymbol{F}_0)$ (by Lemma 1) and

$$\begin{aligned} V_0^{EO}(x_0, \boldsymbol{F}_0) &\equiv s_0 x_0 + \sum_{n \in \mathcal{N} \setminus \{0\}} \delta^{n-1} \mathbb{E}[(\delta \tilde{F}_{n-1,n} - \tilde{s}_{n-1})^+ \mid \boldsymbol{F}_0] \overline{x} \\ &< s_0 x_0 + \sum_{n \in \mathcal{N} \setminus \{0\}} \delta^{n-1} \mathbb{E}[\delta \tilde{F}_{n-1,n} \mid \boldsymbol{F}_0] \overline{x} \\ &= s_0 x_0 + \sum_{n \in \mathcal{N} \setminus \{0\}} \delta^n \tilde{F}_{0,n} \overline{x} \\ &< \infty. \end{aligned}$$

Since $V_0^I(x_0, \boldsymbol{F}_0) \geq 0$, it then follows that

$$V_0^{EO}(x_0, \boldsymbol{F}_0) - V_0^I(x_0, \boldsymbol{F}_0) < \infty. \tag{59}$$

Lemma 4 and inequality (59) imply the claimed result. □ □

Proposition 6 implies that in the worst case the value of an optimal policy is positive and the value of any one of the considered heuristic policies is zero (it is clear that this value cannot be negative). Example 1 illustrates a pathological situation in which this occurs for all these policies, and hence reoptimization fails to be useful.

Example 1 (Worst suboptimality)**.** Consider a two stage setting; that is, $N = 2$. Let the initial inventory be zero: $x_0 = 0$. The storage asset is fast but there are frictions. The optimal value function in stage 1 is $V_1(x_1, s_1) = \left(s_1^W\right)^+ x_1$. The optimal value function in stage 0 is the positive part of the spread between the value of a call option and the buy-and-inject spot price in stage 0 multiplied by the maximum inventory: $V_0(0, \boldsymbol{F}_0) = \left(\delta \mathbb{E}\left[\left(\tilde{s}_1^W\right)^+ \mid F_{0,1}\right] - s_0^I\right)^+ \overline{x}$. Suppose that the random variable $\tilde{s}_1$ can take only two values, s_1^1 and s_1^2, with $s_1^1 > s_1^2$. Then, the risk neutral probability $\mathbb{P}(\tilde{s}_1 = s_1^1 | F_{0,1})$ is equal to $(F_{0,1} - s_1^2)/(s_1^1 - s_1^2)$ (Hull 2012, p. 368). Let $F_{0,1} = 0.9$, $s_1^1 = 1.5$, $s_1^2 = 0.05$, $s_0 = 0.5$, $\delta = 0.99$, $\phi^I = 1.1$, $\phi^W = 0.8$, $c^I = 0.1$, $c^W = 0.07$, and $\overline{x} = 1$ (the relevant units of measure are suppressed). It holds that $\delta \mathbb{E}\left[\left(\tilde{s}_1^W\right)^+ \mid F_{0,1}\right] = 0.6558$ and $s_0^I = 0.65$, so that $\delta \mathbb{E}\left[\left(\tilde{s}_1^W\right)^+ \mid F_{0,1}\right] - s_0^I = 0.0058 = V_0(0, \boldsymbol{F}_0)$. In stage 0 all the considered heuristic policies buy and inject at cost s_0^I one unit of inventory when $\delta F_{0,1}^W > s_0^I$. It holds that $\delta F_{0,1}^W = 0.6435$ and $s_0^I = 0.65$, so that $1\left\{\delta F_{0,1}^W > s_0^I\right\} = 0$, where $1\{\cdot\}$ is the indicator function. The value of all these policies is zero because no inventory is injected in stage 0 and no sale can thus occur in stage 1.

B. Estimated UB2 Upper Bounds

Table 9 reports the UB2 upper bound estimates for all the considered instances.

C. Percentage Quality of the RMSO Policies

Tables 10, 11, and 12 display the percentage quality of the considered RMSO policies on the 24, 36, and 48 stage instances, respectively.

Acknowledgments

I thank François Margot, Selvaprabu Nadarajah, and Alan Scheller-Wolf for their feedback on an earlier draft of this paper. In particular, I thank François Margot for suggesting the network flow formulation used in the proof of part (b) of Proposition 3, and Selvaprabu Nadarajah for useful discussions on this research and his help in obtaining the numerical results discussed in §6. Part of this research was supported by NSF grant CMMI-1129163.

Table 9: UB2 upper bound estimates.

Instance		Instance		Instance	
24-Sp-1	1.046470	36-Sp-1	1.414960	48-Sp-1	1.812340
24-Sp-2	1.266210	36-Sp-2	1.725810	48-Sp-2	2.232650
24-Sp-3	1.373770	36-Sp-3	1.882700	48-Sp-3	2.441470
24-Sp-4	1.486890	36-Sp-4	2.052060	48-Sp-4	2.667130
24-Su-1	0.645116	36-Su-1	1.354070	48-Su-1	1.701020
24-Su-2	0.900279	36-Su-2	1.794810	48-Su-2	2.260220
24-Su-3	1.058720	36-Su-3	1.948160	48-Su-3	2.470660
24-Su-4	1.221140	36-Su-4	2.099880	48-Su-4	2.677110
24-Fa-1	0.645116	36-Fa-1	0.977331	48-Fa-1	1.329260
24-Fa-2	0.900279	36-Fa-2	1.334090	48-Fa-2	1.799360
24-Fa-3	1.058720	36-Fa-3	1.552420	48-Fa-3	2.071470
24-Fa-4	1.221140	36-Fa-4	1.789790	48-Fa-4	2.368650
24-Wi-1	0.580790	36-Wi-1	0.908653	48-Wi-1	1.265300
24-Wi-2	0.786777	36-Wi-2	1.228060	48-Wi-2	1.712090
24-Wi-3	0.891848	36-Wi-3	1.390490	48-Wi-3	1.934620
24-Wi-4	1.003130	36-Wi-4	1.569060	48-Wi-4	2.170330

Table 10: Percentage quality of the RMSO policies on the 24 stage instances.

	λ										
Instance	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	1.00
24-Sp-1	98.85	99.02	99.18	99.32	99.40	99.49	99.57	99.57	99.63	99.62	99.57
24-Sp-2	96.39	96.71	97.24	97.62	97.93	98.14	98.20	98.19	98.26	98.14	98.08
24-Sp-3	94.84	95.24	96.23	96.81	97.19	97.47	97.62	97.61	97.61	97.50	97.38
24-Sp-4	91.78	92.14	94.81	95.91	96.50	96.68	96.54	96.54	96.34	96.25	96.12
24-Su-1	95.18	95.31	95.49	95.71	95.78	96.01	96.04	96.07	96.12	96.14	96.09
24-Su-2	93.41	93.69	94.21	94.74	95.27	95.73	95.89	95.93	96.05	96.08	96.08
24-Su-3	91.93	92.32	93.69	94.56	95.17	95.71	95.89	96.04	96.12	96.17	96.09
24-Su-4	87.64	87.83	91.08	93.01	94.08	94.68	94.82	94.92	95.07	95.06	95.00
24-Fa-1	95.38	95.51	95.70	95.90	96.03	96.24	96.26	96.29	96.34	96.35	96.30
24-Fa-2	93.64	93.92	94.41	94.96	95.48	95.97	96.11	96.15	96.27	96.28	96.31
24-Fa-3	92.14	92.54	93.88	94.78	95.36	95.94	96.10	96.22	96.36	96.36	96.31
24-Fa-4	87.79	88.05	91.32	93.19	94.21	94.86	95.00	95.13	95.25	95.28	95.19
24-Wi-1	94.24	94.48	94.64	94.89	95.08	95.36	95.39	95.58	95.61	95.66	95.70
24-Wi-2	91.74	92.16	92.75	93.38	94.05	94.44	94.57	94.73	94.80	94.87	94.82
24-Wi-3	89.53	90.14	91.54	92.40	93.23	93.67	93.82	93.96	93.87	93.89	93.86
24-Wi-4	85.49	85.54	88.83	90.78	91.53	91.86	91.79	87.69	87.66	87.44	87.19

Table 11: Percentage quality of the RMSO policies on the 36 stage instances.

	λ										
Instance	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	1.00
36-Sp-1	98.09	98.26	98.40	98.54	98.62	98.75	98.82	98.85	98.83	98.81	98.72
36-Sp-2	95.94	96.24	96.70	97.08	97.41	97.66	97.77	97.78	97.74	97.66	97.55
36-Sp-3	94.53	94.91	95.90	96.44	96.90	97.16	97.23	97.24	97.16	97.04	96.90
36-Sp-4	91.22	91.42	94.08	95.27	95.94	95.87	95.70	95.59	95.43	95.20	94.93
36-Su-1	96.46	96.64	96.78	96.93	97.05	97.15	97.17	97.21	97.20	97.13	97.08
36-Su-2	95.04	95.32	95.65	95.95	96.24	96.42	96.51	96.55	96.56	96.41	96.35
36-Su-3	93.61	93.91	94.70	95.21	95.60	95.81	95.91	95.85	95.82	95.69	95.62
36-Su-4	90.86	90.81	93.00	94.05	94.55	94.63	94.57	94.42	94.29	94.18	93.98
36-Fa-1	95.77	95.96	96.13	96.28	96.45	96.55	96.61	96.64	96.57	96.56	96.49
36-Fa-2	93.33	93.73	94.16	94.59	95.15	95.47	95.60	95.59	95.65	95.55	95.48
36-Fa-3	92.06	92.50	93.59	94.41	95.12	95.42	95.56	95.62	95.59	95.53	95.44
36-Fa-4	88.15	88.31	91.47	93.33	94.43	94.76	94.88	94.86	94.82	94.68	94.49
36-Wi-1	95.01	95.19	95.46	95.67	95.82	95.99	96.05	96.08	96.12	96.04	95.97
36-Wi-2	92.68	93.12	93.78	94.31	94.79	95.19	95.31	95.34	95.25	95.11	94.96
36-Wi-3	90.89	91.35	92.67	93.48	94.20	94.62	94.77	94.84	94.66	94.54	94.44
36-Wi-4	87.59	87.63	90.67	91.74	92.84	92.81	92.81	90.22	90.06	89.85	89.42

Table 12: Percentage quality of the RMSO policies on the 48 stage instances.

	λ										
Instance	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	1.00
48-Sp-1	95.25	95.42	95.57	95.74	95.81	95.91	95.94	95.98	95.95	95.89	95.76
48-Sp-2	93.85	94.18	94.59	95.01	95.36	95.54	95.58	95.53	95.42	95.23	95.07
48-Sp-3	92.53	92.92	93.84	94.54	95.01	95.22	95.25	95.24	95.14	94.97	94.75
48-Sp-4	89.77	89.95	92.47	93.72	94.30	94.45	94.36	94.18	93.90	93.58	93.27
48-Su-1	95.05	95.19	95.36	95.49	95.57	95.65	95.68	95.73	95.75	95.73	95.64
48-Su-2	94.31	94.58	95.00	95.35	95.68	95.89	95.93	95.92	95.89	95.83	95.73
48-Su-3	93.11	93.41	94.34	94.89	95.37	95.57	95.63	95.62	95.59	95.48	95.34
48-Su-4	90.29	90.20	92.56	93.50	94.08	94.22	94.12	93.91	93.80	93.73	93.59
48-Fa-1	93.75	93.89	94.06	94.24	94.39	94.49	94.53	94.56	94.49	94.38	94.20
48-Fa-2	92.52	92.84	93.29	93.64	94.11	94.38	94.43	94.46	94.41	94.23	94.05
48-Fa-3	91.56	91.96	93.07	93.75	94.38	94.70	94.75	94.79	94.71	94.56	94.41
48-Fa-4	88.43	88.36	91.15	92.79	93.96	94.29	94.37	94.32	94.22	93.98	93.75
48-Wi-1	93.35	93.56	93.71	93.87	93.98	94.10	94.20	94.20	94.16	94.11	94.02
48-Wi-2	92.15	92.54	93.05	93.53	94.03	94.24	94.31	94.26	94.19	94.00	93.84
48-Wi-3	90.56	91.06	92.18	92.99	93.68	93.96	94.00	93.84	93.79	93.60	93.36
48-Wi-4	87.30	87.20	90.02	91.37	92.47	92.63	92.43	90.51	90.42	90.23	89.92

References

Bertsimas, D., J. N. Tsitsiklis. 1997. *Introduction to Linear Optimization*. Athena Scientific, Belmont, MA, USA.

Bjerksund, P., G. Stensland. 2011. Closed form spread option valuation. *Quantitative Finance* **Forthcoming**.

Bjerksund, P., G. Stensland, F. Vagstad. 2011. Gas storage valuation: Price modeling v. optimization models. *The Energy Journal* **32**(1) 203–227.

Boogert, A., C. de Jong. 2008. Gas storage valuation using a Monte Carlo method. *Journal of Derivatives* **15**(3) 81–98.

Boogert, A., C. de Jong. 2011/12. Gas storage valuation using a multifactor price process. *Journal of Energy Markets* **4**(4) 29–52.

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

Brown, D. B., J. E. Smith. 2013. Information relaxations, duality, and convex stochastic dynamic programs. Working Paper, Fuqua School of Business, Durham, NC, USA.

Brown, D. B., J. E. Smith, P. Sung. 2010. Information relaxation and duality in stochastic dynamic programs. *Operations Research* **58**(4) 1–17.

Carmona, R., V. Durrleman. 2003. Pricing and hedging spread options. *Society for Industrial and Applied Mathematics Review* **45**(4) 627–685.

Carmona, R., M. Ludkovski. 2010. Valuation of energy storage: An optimal switching approach. *Quantitative Finance* **10**(4) 359–374.

Charnes, A., J. Drèze, M. Miller. 1966. Decision and horizon rules for stochastic planning problems: A linear example. *Econometrica* **34**(2) 307–330.

Chen, Z., P. A. Forsyth. 2007. A semi-Lagrangian approach for natural gas storage valuation and optimal operation. *SIAM Journal on Scientific Computing* **30**(1) 339–368.

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

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

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

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

Gray, J., P. Khandelwal. 2004a. Realistic gas storage models II: Trading strategies. *Commodities Now* (September) 1–5.

Gray, J., P. Khandelwal. 2004b. Towards a realistic gas storage model. *Commodities Now* (June) 1–4.

Hull, J. C. 2012. *Options, Futures, and Other Derivatives Securities*. Eight ed. Prentice Hall, Englewood Cliffs, NJ, USA.

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

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

Lacima. 2010. Lacima analytics – Valuation & optimisation suite. http://www.lacimagroup.com/uploads/LacimaAnalyticsbrochure.pdf.

Lai, G., F. Margot, N. Secomandi. 2010. An approximate dynamic programming approach to benchmark practice-based heuristics for natural gas storage valuation. *Operations Research* **58**(3) 564–582.

Lai, G., M. X. Wang, S. Kekre, A. Scheller-Wolf, N. Secomandi. 2011. Valuation of storage at a liquefied natural gas terminal. *Operations Research* **59**(3) 1–15.

Manoliu, M. 2004. Storage options valuation using multilevel trees and calendar spreads. *International Journal of Theoretical and Applied Finance* **7**(4) 425–464.

Maragos, S. 2002. Valuation of the operational flexibility of natural gas storage reservoirs. E. Ronn, ed., *Real Options and Energy Management Using Options Methodology to Enhance Capital Budgeting Decisions*. Risk Publications, London, UK, 431–456.

Margrabe, W. 1978. The value of an option to exchange one asset for another. *Journal of Finance* **33**(1) 177–186.

Mazières, D., A. Boogert. 2013. A radial basis function approach to gas storage valuation. *Journal of Energy Markets* **6**(2) 19–50.

Nadarajah, S., F. Margot, N. Secomandi. 2013. Approximate dynamic programs for natural gas storage valuation based on approximate linear programming relaxations. Working Paper 2011-E5, Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA, USA.

Parsons, C. 2013. Quantifying natural gas storage optionality: A two-factor tree model. *Journal of Energy Markets* **6**(1) 95–124.

Schwartz, E. S., J. E. Smith. 2000. Short-term variations and long-term dynamics in commodity prices. *Management Science* **46** 893–911.

Secomandi, N. 2008. An analysis of the control-algorithm re-solving issue in inventory and revenue management. *Manufacturing & Service Operations Management* **10**(3) 468–483.

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

Secomandi, N. 2011. An opportunity cost view of basestock optimality for the warehouse problem. P. Kouvelis, O. Boyabatli, L. Dong, R. Li, eds., *Handbook of Integrated Risk Management in Global Supply Chains*. John Wiley & Sons, Inc., Hoboken, NJ, USA, 447–461.

Secomandi, N., G. Lai, F. Margot, A. Scheller-Wolf, D. J. Seppi. 2012. Merchant commodity storage and term structure model error. Working Paper 2010-E71, Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA, USA.

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

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

Thompson, M. 2012. Natural gas storage valuation, optimization, market and credit risk management. Working Paper, Queen's School of Business, Queen's University, Kingston, Ontario, CA.

Thompson, M., M. Davison, H. Rasmussen. 2009. Natural gas storage valuation and optimization: A real options application. *Naval Research Logistics* **56**(3) 226–238.

TIMERA ENERGY. 2013. Cracking gas storage and swing valuation. http://www.timera-energy.com/uk-gas/cracking-gas-storage-and-swing-valuation/.

Ware, A. 2013. Accurate semi-lagrangian time stepping for stochastic optimal control problems with application to the valuation of natural gas storage. *SIAM Journal on Financial Mathematics* **4** 427–451.

Wu, O., D. Wang, Z. Qin. 2012. Seasonal energy storage operations with limited flexibility: The price-adjusted rolling intrinsic policy. *Manufacturing & Service Operations Management* **14**(3) 455–471.