Indifference pricing of natural gas storage contracts.

Original Citation:

Löhndorf, Nils and Wozabal, David (2017) 
Indifference pricing of natural gas storage contracts. 
WU Vienna University of Economics and Business, Vienna. 
This version is available at: https://epub.wu.ac.at/5421/ 
Available in ePubWU: February 2017 
ePubWU, the institutional repository of the WU Vienna University of Economics and Business, is provided by the University Library and the IT-Services. The aim is to enable open access to the scholarly output of the WU.
Indifference pricing of natural gas storage contracts

Nils Löhndorf\textsuperscript{1}, David Wozabal\textsuperscript{2}

\textsuperscript{1}Vienna University of Economics and Business, Vienna, Austria
\textsuperscript{2}Technische Universität München, Munich, Germany
\{nils.loehndorf@wu.ac.at, david.wozabal@tum.de\}

Natural gas markets are incomplete due to physical limitations and low liquidity, but most valuation approaches for natural gas storage contracts assume a complete market. We propose an alternative approach based on indifference pricing which does not require this assumption but entails the solution of a high-dimensional stochastic-dynamic optimization problem under a risk measure. To solve this problem, we develop a method combining stochastic dual dynamic programming with a novel quantization method that approximates the continuous process of natural gas prices by a discrete scenario lattice. In a computational experiment, we demonstrate that our solution method can handle the high dimensionality of the optimization problem and that solutions are near-optimal. We then compare our approach with rolling intrinsic valuation, which is widely used in the industry, and show that the rolling intrinsic value is sub-optimal under market incompleteness, unless the decision-maker is perfectly risk-averse. We strengthen this result by conducting a backtest using historical data that compares both trading strategies. The results show that up to 40\% more profit can be made by using our indifference pricing approach.

\textit{Key words}: stochastic-dynamic programming, asset pricing, natural gas industry, Markov processes

\textit{History}:  

1. Introduction

Over the past decade natural gas has increasingly replaced other fossil fuels in the United States. The U.S. Energy Information Administration projects that this trend is likely to continue and that natural gas consumption will make up 35\% of total energy consumption by 2040 (\textsuperscript{EIA} 2016). This will not only increase trading activities in natural gas markets but also lead to a growing importance of storage facilities which are used to buffer variability in demand. Such storage facilities are scarce, as they require natural geological formations like depleted gas fields, salt domes, or aquifers. Due to its scarcity, storage capacity is an important asset in energy trading. Often, the ownership of gas and storage is separated, and a contract is used to transfer usage rights from storage owners to third parties. Pricing such storage contracts is therefore a
problem of practical importance in energy trading, and numerous commercial software products offer valuation tools (e.g., MSCI 2015, KYOS 2015, Lacima 2017).

The value of storage derives from buying and injecting gas at times of low prices and withdrawing and selling when prices are high. To do so, owners of storage trade on commodity exchanges, such as the New York Mercantile Exchange (NYMEX) in the United States, which hosts spot and futures markets for natural gas.

Existing approaches for gas storage valuation typically assume that gas markets are complete and that storage can be priced using replication. The idea of replication is that the value of a derivative equals the initial capital required to replicate its cash flow by an optimal trading strategy in the underlying (Karatzas and Shreve 1998). Such a strategy involves buying, holding, and selling the underlying as well as operating a cash account. The fundamental theorem of asset pricing states that, in a complete market, the optimum of the replication problem equals the expected discounted cash flow under a uniquely defined equivalent martingale measure. This approach is commonly referred to as risk-free pricing, as all market participants agree on one price, irrespective of their risk preferences.

The complete market assumption, however, is not tenable for gas storage valuation, because gas markets are rarely as liquid as financial markets, and the storage and transport of gas is subject to various physical limitations. In particular, the storage of gas is expensive, which contrasts the situation in financial markets, where storing the underlying is costless. Since the replication of any derivative requires the ownership of storage, the value of storage cannot be replicated, unless one owns it in the first place, which contradicts the idea of replication. Carmona and Ludkovski (2010) point out that pricing a storage contract should take market incompleteness into account and should be based on joint optimization of the portfolio of futures contracts and storage operation.

In incomplete markets, there exists no unique equivalent martingale measure that can be used for risk-free valuation. Consequently, the market price of an asset and individual valuations of agents with different risk preferences may vary. To model the market price of storage, an actuarial value can be calculated as the expected cash flow under a martingale measure calibrated
to observed market prices. However, to determine the value an individual agent ascribes to storage, one has to determine the price at which the agent is indifferent between holding and selling the storage (Karatzas and Shreve 1998). This valuation is dependent on risk preferences and endowment and is referred to as *indifference pricing* (see Carmona 2009, for a comprehensive exposition of the subject).

We propose the first indifference pricing approach for natural gas storage valuation. The indifference value of storage depends on the discounted rewards that result from an optimal operational and trading strategy. Finding this strategy requires solving a high-dimensional stochastic-dynamic decision problem that takes into account the full dynamics of futures prices, the dynamic nature of storage and future contract balances, as well as physical constraints of the storage contract.

To position our contribution, we begin by discussing the extant literature on gas storage valuation. Two heuristic approaches popular among practitioners are *basket-of-spread options valuation* and *rolling intrinsic valuation*. Both approaches make simplifying assumptions to approximate the *extrinsic* value of storage. While the *intrinsic* value can be obtained by finding the optimal schedule of injections and withdrawals based on current futures prices, its *extrinsic* value captures the additional value of flexibility to adapt this schedule to price changes. The basket-of-spread options approach approximates the extrinsic value by the value of a portfolio of (time) spread options. The rolling intrinsic approach, by contrast, simulates the possible evolution of the futures curve over time and approximates the extrinsic value as the expected gains that result from changes in the schedule of injections and withdrawals. To simulate the dynamics of the futures curve, multifactor models, such as the multivariate Black model, are a popular choice (for further reference, see Eydeland and Wolyniec 2003, Gray and Khandelwal 2004b,a).

Further approaches from the literature are based on risk-free pricing. These approaches use spot-based models and ignore the dynamics of the futures curve (e.g. Boogert and de Jong 2008, Chen and Forsyth 2007, Thompson et al. 2009, Carmona and Ludkovski 2010, Secomandi 2010, Bjerksund et al. 2011, Boogert and de Jong 2012, Felix and Weber 2012). While spot
price models keep the underlying optimization problem low-dimensional, traders are suspicious of these approaches, as they are usually not consistent with observed futures prices \cite{Lai2010}.

\cite{Lai2010} and \cite{Nadarajah2015} account for futures price dynamics in a complete market setting and use approximate dynamic programming techniques \cite{Powell2011} that are based on low-dimensional relaxations of the otherwise high-dimensional stochastic-dynamic optimization problem. Both approaches yield no significant benefit over the rolling intrinsic solution.

Our indifference pricing approach avoids making simplifying assumptions and tackles the pricing problem in its full complexity. The contribution of our approach is three-fold: first, we propose a high-dimensional, joint optimization model for futures trading as well as storage operation that takes risk preferences into account. Second, as the optimization problem is subject to the curse of dimensionality, we propose an approximate dynamic programming approach to solve the problem. Third, to test whether accounting for the added complexity is worthwhile, we report the results of a comprehensive experiment based on historical data as well as simulations.

Our first contribution is a model for the indifference value of storage. In Section 2.2, we formulate the joint problem of futures trading and storage operation as a Markov decision process (MDP), whereby risk preferences are modeled using the nested conditional value-at-risk. In Section 2.3, we show that the optimal solution to this problem reduces to the intrinsic solution under perfect risk aversion and that, for all other cases, we have to find the optimal policy of the MDP to compute the indifference value.

Our second contribution addresses the curse of dimensionality. In Section 3, we introduce \emph{approximate dual dynamic programming} (ADDP) as a unifying framework that integrates scenario lattices with \cite{Pereira1991}'s stochastic dual dynamic programming (SDDP) algorithm to solve a general class of discrete-time, continuous-state, risk-averse MDPs. The SDDP algorithm has proven to be successful for tackling stochastic-dynamic optimization problems with many decision epochs. However, conventional SDDP is limited to stagewise independent randomness. It does not seem to be widely known that this limitation can be overcome by
using scenario lattices. Discretizing a continuous Markov process to a lattice has been first pro-
posed in Bally and Pagès (2003) and applied to SDDP independently in Bonnans et al. (2012) and Löhndorf et al. (2013). Bally and Pagès (2003) propose an approach to construct scenario lattices for Brownian motions using optimal quantization. We extend their work and propose an alternative construction method that combines quantization with ideas from moment matching (Høyland et al. 2003). In Section 4.3 we demonstrate that this combination is crucial to determine the indifference value of storage. In Section 3.3 we show how the nested CVaR can be integrated into SDDP by adapting the approach of Philpott et al. (2013) to scenario lattices.

Our third contribution concerns a comparison of the rolling intrinsic solution and indifference pricing. Lai et al. (2010) derive a dual upper bound for the value of storage in a complete market setting and demonstrate that the rolling intrinsic value is near-optimal. Secomandi (2015) provides additional theoretical support for this finding. In Section 4.4 we demonstrate that the result of Lai et al. (2010) does not hold in incomplete markets, since the optimal risk-neutral policy yields higher profits than the rolling intrinsic policy, which we show to be the optimal policy under perfect risk aversion. In Section 4.5 we compare both approaches in a rolling horizon, out-of-sample backtest using historical market data of natural gas futures prices from NYMEX Henry Hub. We find that indifference pricing leads to up to 40% higher profits than the rolling intrinsic solution when managing a storage contract over time, which provides additional evidence for the importance of considering market incompleteness in practical applications.

Our results provide energy traders with a new approach to jointly optimize futures trading and storage management, which is an attractive alternative to rolling intrinsic valuation. In particular, the model allows traders to choose a statistical model that matches the empirical realities of gas markets and to choose the level of risk exposure that matches their preferences. Furthermore, as the method is quite general, we believe that the proposed solution approach is of interest to researchers working on similar high-dimensional stochastic-dynamic optimization problems.
2. Model Formulation

2.1. Assumptions

We consider the problem of a price-taking energy trader who manages a gas storage contract over time, which grants the buyer the right to inject, withdraw, and store gas over a finite time horizon. The injection, withdrawal, and storage capacity is limited and defined by the contract. We assume that injection and withdrawal limits are independent of storage level, i.e., ratchets are ignored as is common with storage contracts.

To decide about buying and selling storage capacity, a trader has to price the contract at the time of inception. The utility of the storage is derived from the distribution of future discounted rewards (cash flow) arising from buying and selling futures contracts while fulfilling matured contracts. In line with the literature on gas storage valuation, we assume that rewards are generated in monthly increments (Eydeland and Wolyniec 2003, Lai et al. 2010).

In contrast to the extant literature, we use an indifference pricing approach to deal with the storage valuation problem in incomplete gas markets. Let us therefore introduce the concept of indifference pricing and how it differs from the more common risk-free pricing.

To avoid notational clutter, we define \([I] = \{1, \ldots, I\}\) for \(I \in \mathbb{N}\). Denote \(F_t, t \in [T]\), as the random futures prices defined on the measure space \((\Omega, \mathcal{F}, P)\), with \(T\) as the number of discrete time periods. The fundamental theorem of asset pricing states that, in a complete market, there exists a unique equivalent (risk-neutral) martingale measure \(Q\) on \((\Omega, \mathcal{F})\) that can be used to price derivatives. More specifically, let \(C_t(F_t, \pi_t)\) be the discounted immediate reward in \(t\) dependent on the policy \(\pi = (\pi_1, \ldots, \pi_T)\), where \(\pi_t\) maps the history \(F^t = (F_1, \ldots, F_t)\) to a decision in \(t\). Then, the risk-free value of storage is given by

\[
\max_\pi E_Q \left( \sum_{t=1}^{T} C_t(F_t, \pi_t(F^t)) \right),
\]

i.e., the maximal expected discounted rewards under the risk-neutral measure \(Q\). In an incomplete market, no unique risk-neutral measure \(Q\) exists and we use the physical measure, \(P\), which represents the actual real-world probabilities of the evolution of the futures curve. In this setting, we can no longer assume that pricing is independent of risk preferences and, hence,
have to replace the expectation in (1) by a more general probability functional, $R$, to model a decision-maker’s risk-aversion. Consequently, the indifference value of storage is given by

$$\max_{\pi} R_P \left( C_1(F_1, \pi_1(F^1)), \ldots, C_T(F_T, \pi_T(F^T)) \right),$$

(2)

where $R_P$ is a dynamic risk measure dependent on the physical measure $P$ (Ruszczyński 2010).

For what follows, we disregard the measure theoretic subtleties above and directly work with the random variables $F_t$ and their distributions under $P$.

The physical measure, $P$, is generally not a martingale and therefore might lead to positive expected gains from trading in the futures market. To ensure that pricing is not distorted by excessive trading, we enforce that all futures contracts yield a physically implementable policy under the storage’s operational constraints at all times. Following [Lai et al. (2010)], we assume that each physical injection and/or withdrawal incurs a marginal cost in addition to an in-kind fuel loss and that storing natural gas incurs no holding cost.

As is common in the literature, e.g., Clewlow and Strickland (2000), Eydeland and Wolyniec (2003), Lai et al. (2010), the dynamics of the futures curve is assumed to follow a multivariate geometric Brownian motion (MGBM). At time $t \in [T]$, let $F_t = (F_{tt}, \ldots, F_{tT})$ be the vector of futures prices, with $F_{tt}$ being the spot price and $F_{tj}$ being the futures price with maturity in period $j = t+1, \ldots, T$. Then, the price process is given by the following stochastic differential equations,

$$\frac{dF_{tj}}{F_{tj}} = \mu_j dt + \sigma_j dZ_j, \quad dZ_jdZ_k = \rho_{jk}, \quad 1 \leq j, k \leq T,$$

(3)

where $dZ_j$ with $j \in [T]$ are correlated increments of standard Brownian motions. Note that, in contrast to Lai et al. (2010), we allow for non-zero drifts $\mu_j \neq 0$, resulting in processes that are not martingales. In our problem, we discretize this process to monthly time increments.

We model the risk preferences in (2) by the nested conditional value-at-risk (CVaR) (e.g., see Ruszczyński and Shapiro 2006). For a random variable $X$ with distribution function $F_X$, let $\text{CVaR}_\alpha(X) = \alpha^{-1} \int_0^\alpha F_X^{-1}(t)dt$ and define $\rho_{\alpha,\lambda}(X) = \lambda \text{CVaR}_\alpha(X) + (1 - \lambda)E(X)$ for $0 \leq \lambda \leq 1$ and $0 < \alpha < 1$. For a sequence of random variables $X_1, \ldots, X_T$ the nested CVaR is defined as

$$X_1 + \rho_{\alpha,\lambda}(X_2 + \rho_{\alpha,\lambda}(X_3 + \cdots)),$$

(4)
i.e., by a convex combination of expectation and CVaR that recursively includes other convex combinations of expectation and CVaR. Unlike the terminal CVaR, which measures the risk of the distribution of total discounted rewards, the nested CVaR is time-consistent, which means that past losses and gains do not affect the policy (Shapiro 2009).

### 2.2. Markov Decision Process

We model the decision problem outlined above as a finite horizon, continuous-state, discrete-time Markov decision process (MDP). The state of the MDP comprises the resource state, $R_t$, which includes state variables that the decision-maker can influence, and the environmental state, $F_t$, which evolves independently from the decisions. The resource state, $R_t$, includes all futures contracts positions, $f_{tj}$, with $j \geq t$, as well as the gas storage level, $s_t$, while the environmental state consists of the futures prices.

The decision policy, $\pi = \{\pi_1, \ldots, \pi_T\}$, includes all trading and operational decisions subject to the state-dependent feasible set $\Pi_t(R_t)$. Starting with a deterministic state $(R_1, F_1)$, we define the dynamic programing equations of the problem as

$$V_t(F_t, R_t) = \max_{\pi_t \in \Pi_t(R_t)} \left\{ C_t(F_t, \pi_t) + \rho_{\alpha, \lambda} \left[ V_{t+1}(F_{t+1}, R_{t+1}(\pi_t)) | F_t \right] \right\}, \ t \in [T],$$

with $V_{T+1} \equiv 0$. Therefore, an optimal policy, $\pi^*$, maximizes the sum of immediate reward and $\rho_{\alpha, \lambda}$ of the value that results from future storage operations and trading.

To specify the feasible set, $\Pi_t(R_t)$, denote $i_{tj}$ and $w_{tj}$ as the injection and withdrawal decisions in period $t$ needed to close futures contract position $f_{tj}$ that matures in periods $j$. Physical injection and withdrawal incurs marginal costs of $c^i$ and $c^w$, respectively, as well as in-kind losses $d^i$ and $d^w$. Futures buying and selling decisions are denoted by $x_{tj}$ and $y_{tj}$, respectively.

Accordingly, the immediate reward is given by

$$C_t(F_t, \pi_t) = \sum_{j=t}^{T} \gamma_j F_{tj}(y_{tj} - x_{tj}) - \gamma_t (c^i i_{tj} + c^w w_{tj}),$$

where $\gamma_t$ is the discount factor for period $t$. Note that in order to avoid transaction costs being charged more than once, we only account for injection and withdrawal costs incurred in period $t$ but not for planned future injections and withdrawals.
When a futures contract matures, it must be either fulfilled physically through storage operation or cleared in the spot market, i.e.,

\[ f_{tt} + x_{tt} - y_{tt} = i_{tt} - w_{tt}, \ t \in [T]. \] (7)

The physical storage balance is given by

\[ s_{t+1,t+1} = s_{tt} + i_{tt} - w_{tt}, \ t \in [T]. \] (8)

To enforce the assumption that futures contract positions yield an implementable schedule, all tradable contracts that mature in some future period must be balanced with future injections and withdrawals at all times,

\[ f_{tj} = i_{tj} - w_{tj}, \ t \in [T - 1], \ j = t + 1, \ldots, T. \] (9)

Moreover, all contracts are subject to balance constraints

\[ f_{t+1,j} = f_{tj} + x_{tj} - y_{tj}, \ t \in [T - 1], \ j = t + 1, \ldots, T, \] (10)

which track changes in futures contract positions resulting from purchases or sales in \( t \). Injections and withdrawals for \( j > t \) are subject to storage balance

\[ s_{t,j+1} = s_{tj} + i_{tj} - w_{tj}, \ t \in [T - 1], \ j = t, \ldots, T, \] (11)

where \( s_{tj} \) is the storage level in period \( j \) as projected in period \( t \) if all trades are physically balanced by storage operations.

Storage operational limits are given by the following set of constraints,

\[ \bar{i} \leq i_{tj} \leq \bar{i}, \ t \in [T], \ j = t, \ldots, T, \] (12)

\[ \bar{w} \leq w_{tj} \leq \bar{w}, \ t \in [T], \ j = t, \ldots, T, \] (13)

\[ \underline{s} \leq s_{tj} \leq \bar{s}, \ t \in [T], \ j = t, \ldots, T. \] (14)

At the end of period \( t \), the physical storage state as well as the futures contract position yields the final resource state

\[ R_{t+1} = \{s_{t+1,t+1}\} \times \{f_{t+1,j}\}_{j=t+1}^{T}, \ t \in [T]. \] (15)
Note that the resource state’s dimensionality decreases in $t$ because one futures contract matures in each period.

**Remark 1.** The above constraints define the feasible sets $\Pi_t(R_t)$, $t \in [T]$, for the policy $\pi$. Since all constraints as well as the objective function are linear, the maximization problem at stage $T$ is linear. The resource state $R_T$ appears only on the right-hand side of the constraints, such that the objective value and therefore the value function $V_T$ is a concave function in $R_T$. As concavity is preserved by $\rho_{\alpha,\lambda}$, the value function in $T - 1$ is also concave in $R_T$. By backward induction, it follows that all value functions $V_t$ are concave functions in $R_t$ and all problems in (5) are convex optimization problems.

In line with Powell (2011), let us define the post decision value in $t + 1$ as

$$
\hat{V}_t(F_t, R_{t+1}) = \rho_{\alpha,\lambda} \left[ V_{t+1}(F_{t+1}, R_{t+1}) | F_t \right].
$$

(16)

Since $R \mapsto \hat{V}_t(F_t, R)$ is concave, it can be approximated from above by the minimum of a set of $L$ supporting hyperplanes, $\hat{V}_t$, which is defined by scalars $a_l(F_t, \hat{R}_{l,t+1})$ and slope vectors $b_l(F_t, \hat{R}_{l,t+1})$, whereby $\hat{R}_{l,t+1}$ denotes a feasible post-decision resource state, i.e.,

$$
\hat{V}_t(F_t, R_{t+1}) = \min_l \left\{ a_l(F_t, \hat{R}_{l,t+1}) + b_l(F_t, \hat{R}_{l,t+1})^\top (R_{t+1} - \hat{R}_{l,t+1}), \quad l = 1, \ldots, L \right\}, \quad t \in [T - 1].
$$

(17)

Combining (6) with (17), (5) can be approximated as

$$
V_t(F_t, R_t) \approx \max \sum_{j=t}^{T} \gamma_j F_j (y_{ij} - x_{ij}) - \gamma_i (c^{i tt} + c^{w} w_{tt}) + v_t
$$

s.t. $v_t \leq a_l(F_t, \hat{R}_{l,t+1}) + b_l(F_t, \hat{R}_{l,t+1})^\top (R_{t+1} - \hat{R}_{l,t+1}), \quad l \in [L], \quad t \in [T - 1], \quad (7), (8), (9), (10), (11), (12), (13), (14),$

$$
 x_{ij}, y_{ij}, i_{ij}, w_{ij}, s_{ij} \geq 0, \quad t \in [T - 1], j = t, \ldots, T,$$

$$
 f_{t+1,j} \in \mathbb{R}, \quad t \in [T - 1], j = t, \ldots, T - 1,$$

$$
 v_t \in \mathbb{R}, \quad t \in [T - 1].
$$

(18)

### 2.3. Special Cases

Depending on parameters $\alpha$ and $\lambda$, the optimal policy has the optimal expectation maximizing policy as well as the rolling intrinsic solution as special cases. The first case is trivial. By setting $\lambda = 0$, $\rho_{\alpha,\lambda}$ reduces to the expected value and we obtain a policy that maximizes expected discounted rewards.
To relate our policy to the rolling intrinsic solution, we consider the case of $\lambda = 1$, i.e., the decision-maker maximizes only the CVaR. By additionally setting $\alpha = 0$, we obtain

$$\text{CVaR}_0(X) = \text{ess inf}(X).$$

Under the nested CVaR objective with $\alpha = 0$ and $\lambda = 1$, the optimization problem becomes

$$\max_{\pi} \rho_{\alpha,\lambda}(C(F_1,\pi_1), \ldots, C(F_T,\pi_T)) = \max_{\pi} \sum_{t=1}^{T} \text{ess inf} C(F_t,\pi_t). \quad (19)$$

**Proposition 1.**

1. The optimal policy for (19) is the rolling intrinsic policy. In particular, the optimal first-stage objective value equals the intrinsic value of storage.

2. The optimal policy for $\rho_{\alpha,1}$ converges to the rolling intrinsic policy as $\alpha \to 0$.

**Proof.** In $t = 1$, the rolling intrinsic policy maximizes the first-stage reward, $C_1(F_1,\pi_1)$, subject to storage constraints, irrespective of possible gains that might arise because of random price changes in the future. Let these decisions be denoted by $\pi_1^*$. Any policy with lower first-stage reward must earn a positive amount through price changes in later periods almost surely to compensate for the difference if it is to be optimal for (19).

Therefore, it is enough to show that for every decision $\bar{\pi}_1$, there is a positive probability that the value of associated futures contracts decreases in subsequent periods.

In particular, the set of prices in $t = 2$, that do not cause the value of the futures positions, $\bar{f}_{2,t}$, to increase, contains

$$\mathcal{H} = \{ F_2 = (F_{2,2}, \ldots, F_{2,T}) : \bar{f}_{2,t}(F_{2,t} - F_{1,t}) \leq 0, \ \forall \ t = 2, \ldots, T \}. \quad (20)$$

$\mathcal{H}$ is an intersection of half-spaces in $\mathbb{R}^{T-1}$ that intersect the positive orthant. As prices under (3) are log-normally distributed, the probability of $\mathcal{H}$ is positive. This concludes the proof of 1.

To prove 2, note that $\text{CVaR}_\alpha$ converges to $\text{CVaR}_0$ as $\alpha \to 0$. Therefore, it follows that $\rho_{\alpha,1} \xrightarrow{\alpha \to 0} \rho_{0,1}$ and consequently from Shapiro et al. [2009], Theorem 7.27, that $\rho_{\alpha,1} \xrightarrow{epi} \rho_{0,1}$, as $\alpha \to 0$.

Considering the above and the fact that the decisions are bounded, it follows that the decisions as well as objective values converge (see Shapiro et al. [2009], Proposition 7.26). \[\square\]
Remark 2. The indifference value is the optimal objective value of the MDP, which is not the same as the expected discounted rewards from following the optimal policy, unless the agent is risk-neutral. By Proposition 1, this implies that the indifference value is the intrinsic value for any agent for whom the rolling intrinsic policy is optimal. This implication contrasts the classical interpretation of rolling intrinsic valuation which defines the value of storage as the expected discounted reward from following the rolling intrinsic policy. However, this presumes that the decision-maker is risk-neutral which contradicts Proposition 1. Therefore, the rolling intrinsic value cannot be used for indifference pricing in our setting.

3. Method

The problem defined in the last section is a discrete-time, continuous-state, risk-averse MDP. As it is generally not possible to solve such problems exactly, we use approximate dynamic programming (Powell 2011) to approximate the optimal policy of the continuous-state problem, which is equivalent to approximating the value functions of the MDP.

The problem from Section 2.2 divides the state space of the MDP into an environmental state, $F_t$, which is exogenously given and random, and a resource state, $R_t$, which is determined by the decision process. This separation enables us to approximate the value functions in two steps: first, we search for an optimal set of representative discrete states for the environmental state, which we organize in a scenario lattice. To accomplish this goal, we propose a novel method that combines optimal quantization with moment matching.

Second, we use a version of SDDP that approximates the value function at each node of the lattice by a concave, piecewise-linear function of the resource state. This contrasts conventional SDDP, as it allows the data process to be Markovian, rather than requiring stagewise independence.

We refer to this two-stage procedure as approximate dual dynamic programming (ADDP) to emphasize that we construct an approximate solution for a continuous-state MDP. A policy for the continuous-state MDP can be obtained by using the piecewise-linear value function associated with the discrete state closest to a given (continuous) environmental state to make
a decision. The result is an approximate value function that is piecewise-constant in the environment state and piecewise-linear in the resource state.

This procedure has several advantages. A scenario lattice can be constructed by a computationally inexpensive stochastic gradient algorithm. Furthermore, it eliminates the need to discretize the high-dimensional resource state. Instead, the procedure exploits the fact that for a given environmental state of the MDP, i.e., a node on the lattice, the value function is piecewise-linear and concave in the resource state.

### 3.1. Scenario Lattices

Since the stochastic process defined in (3) is Markovian, the conditional distribution of future prices in $t + 1$ at stage $t$ does not depend on the entire price history but only on prices in $t$. If we discretize (3) to a scenario tree, many branches of the tree would have identical sub trees which can be combined without loss of information. We refer to such a recombining scenario tree as scenario lattice, in line with the terminology often used in mathematical finance.

More formally, a lattice is a graph organized in a finite number of layers. Each layer is associated with a discrete point in time and contains a finite number of nodes. Successive layers are connected by arcs. A node represents a possible state of the stochastic process, and an arc represents the possibility of a state transition from one node on a given layer to a successor node on the next layer. Each arc is associated with a probability weight, and weights of all outgoing arcs of a node add up to one. Note that for a scenario tree, we would have to add the requirement that every node in stage $t$ has only one predecessor in stage $(t - 1)$.

Denote $N_t$ as the number of nodes in $t$, and $\bar{F}_{tn}$, $n \in [N_t]$, as the state of the stochastic process at node $n$ in $t$. Further, denote $\bar{F}_t = \{\bar{F}_{tn} : n \in [N_t]\}$ as the set of all possible states in the lattice layer corresponding to time $t$. Assuming that all state transition between nodes on consecutive stages have positive probabilities, the set of possible scenarios on the lattice is given by

$$\bar{F}_1 \times \bar{F}_2 \times \cdots \times \bar{F}_{T-1} \times \bar{F}_T.$$  \hspace{1cm} (21)

As the number of stages grows, the additional nodes needed to construct a lattice are those of the newly added stages, while the number of nodes in a scenario tree (with non-trivial
conditional distributions) grows exponentially in the number of stages. See Figure 1 for an illustrative comparison.

Our objective is to construct a scenario lattice such that the optimal policy for the lattice process yields a close to optimal policy for the true process. In this context, the discretization error is the difference between the optimal objective value of the stochastic optimization problem defined with the true process \(3\) and the value of the optimal policy for the lattice process implemented on the true process. From stability theory for stochastic programs, it is known that minimizing this error is closely linked to the Wasserstein distance between the discrete and the continuous process [Dupačová et al. 2003, Bally and Pagès 2003, Pflug and Pichler 2012].

While there exists an extensive literature on scenario tree generation (see Kaut and Wallace 2007 for a survey) there exists virtually no literature on scenario lattice generation for general MDPs.

### 3.2. Lattice Quantization

In this section, we describe a two-step method that discretizes a continuous process to a lattice using optimal quantization. In the first step, the method searches for the optimal nodes of the lattice by minimizing the Wasserstein distance between the nodes and the unconditional distributions at each stage. In the second step, the method estimates the transition probabilities between nodes on consecutive stages, for which we propose a new technique that performs this estimation in a recursive manner. Following ideas from moment matching, the new technique
ensures that the conditional expectations on the lattice coincide with the conditional expectations of the true process. As we will see in Section 4.3, this step is crucial to accurately approximate the indifference value of storage.

To minimize the distance between the price process and the lattice, we resort to optimal quantization. We can infer from Bally and Pagès (2003) that the discretization error depends on the stagewise Wasserstein distance between the true process $F$ and its discrete approximation $\bar{F}$. More specifically, at each stage $t$, we find quantizers $\bar{F}_{tn}$, $n \in [N_t]$, such that

$$\sum_{n \in N_t} \int_{\Gamma_{tn}} ||F_t - \bar{F}_{tn}||_2^2 \mathbb{P}(dF_t)$$

is minimal for $t = 2, \ldots, T$, where

$$\Gamma_{tn} = \left\{ x : n = \arg\min_m \left\{ ||x - \bar{F}_{tm}||_2^2, m \in [N_t] \right\} \right\}$$

is the Voronoi partition associated with the quantizers $\bar{F}_{tn}$, which serve as the nodes of the lattice in $t$.

Unfortunately, finding the optimal quantizers is an NP-hard problem (Aloise et al. 2009, Löhndorf 2016). Bally and Pagès (2003), faced with a similar problem, propose a method based on stochastic gradient descent which we will adopt here.

Denote $(\beta_k)_{k=1}^K$ as a sequence of stepsizes with $0 \leq \beta_k \leq 1$, $k \in [K]$. To solve the quantization problem using stochastic gradient descent, we draw random sequences $(\hat{F}_t^k)_{t=1}^T$ from the price process, for $k \in [K]$, and define

$$\bar{F}_{tn}^k = \begin{cases} 
\hat{F}_{tn}^{k-1} + \beta_k \left( \hat{F}_t^k - \hat{F}_{tn}^{k-1} \right) & \text{if } n = \arg\min_m \left\{ ||\hat{F}_t^k - \hat{F}_{tm}^{k-1}||_2, m \in [N_t] \right\}, \\
\hat{F}_{tn}^{k-1} & \text{otherwise},
\end{cases}$$

with $\bar{F}_{tn}^0 \equiv 0$, for $n \in [N_t]$, $t = 2, \ldots, T$, $k \in [K]$. Pagès and Printems (2003) show that if the sequence $(\beta_k)_{k=1}^K$ satisfies $\sum_{k=1}^{\infty} \beta_k = \infty$ and $\sum_{k=1}^{\infty} \beta_k^2 < \infty$, then the resulting nodes are local minimizers of (22).

To estimate the transition probabilities between lattice nodes at subsequent stages, let us fix the nodes of the lattice as $\bar{F}_{tn} \equiv \bar{F}_{tn}^K$ and denote $p_t$, $t \in [T - 1]$, as the $|N_t| \times |N_{t+1}|$ transition matrix between layers $t$ and $t + 1$ with elements $p_{tnm}$, where each $p_{tnm}$ defines the (conditional) probability of a state transition from $\bar{F}_{tn}$ to $\bar{F}_{t+1,m}$.
Bally and Pagès (2003) propose to estimate the transition probabilities by

\[
    p_{tnm} = \frac{\sum_{k=1}^{K} I_{\Gamma_{tn}}(\hat{F}^k_t) I_{\Gamma_{tm}}(\hat{F}^k_{t+1})}{\sum_{k=1}^{K} I_{\Gamma_{tn}}(\hat{F}^k_t)}, \quad n \in [N_t], \ m \in [N_{t+1}], \ t \in [T-1],
\]

(25)
with \( I_A \) as the indicator function of the set \( A \). We refer to this method as *forwards estimation* since probabilities are estimated from forward simulations of the price process.

A problem with forwards estimation is that, for any given node in \( t < T \), the (conditional) expected successor state on the lattice does not exactly match the conditional expected successor state for the continuous process but usually slightly deviates from this value. This, in turn, introduces a bias into the optimal policy on the lattice, which leads to a value function that overestimates the value of storage, as we will demonstrate in Section 4.3.

We propose to estimate the transition probabilities in a different manner. Instead of estimating the transition probabilities during forward simulations, we adjust the predecessor nodes layer by layer while going backwards in time. We refer to this method as *backwards estimation*. In contrast to forwards estimation, adjusting the nodes of the lattice going backwards ensures that the conditional means of the lattice equal those of the true process, thereby eliminating the aforementioned bias.

Assume that nodes \( \bar{F}^k_{t+1,m}, \ m \in [N_{t+1}] \), are fixed and have already been corrected. Since we have to correct the nodes in \( t + 1 \), before we can compute the conditional means of the nodes in \( t \), we begin in \( T - 1 \). Before correcting node \( n \) in \( t \), we estimate the vector of transition probabilities, \( p_{tn} \), based on a sequence of sample state transitions, \((\hat{F}^k_{t+1})_{k=1}^{K}\) from (3) conditional on the values stored in \( \bar{F}^k_{tn} \), by setting

\[
    p_{tnm} = K^{-1} \sum_{k=1}^{K} I_{\Gamma_{tm}}(\hat{F}^k_t), \quad m \in [N_{t+1}], \ n \in [N_t], \ t = T - 1, \ldots, 1.
\]

(26)
Using these probabilities, the conditional expectation on the lattice for a node, \( \bar{F}^K_{tn} \), is given by

\[
    \mathbb{E}[\bar{F}^k_{t+1} | \bar{F}^K_{tn}] = \sum_{m=1}^{N_{t+1}} p_{tnm} \bar{F}^k_{t+1,m}, \ n \in [N_t], \ t \in [T-1],
\]

(27)
whereas its conditional expectation for the multivariate GBM is

\[
    \mathbb{E}[\bar{F}^k_{t+1} | \bar{F}^K_{tn}] = \bar{F}^K_{tn} \circ (e^{\mu_t}, \ldots, e^{\mu_T}),
\]

(28)
with $\circ$ as point-wise product. To ensure that these two values match for a given node, we correct the relevant elements of the state vector, $\bar{F}_t = (\bar{F}_{t1}, \ldots, \bar{F}_{tT})$,

$$\bar{F}_{tnj} = \sum_{m=1}^{N_{t+1}} p_{tnm} \bar{F}_{t+1,m} e^{-\mu j}, \quad j = t + 1, \ldots, T, \quad n \in [N_t], \quad t = T - 1, \ldots, 1. \quad (29)$$

Note that, as long as we can explicitly calculate a correction step as in (29), this algorithm can be adapted to any other Markov process.

The following proposition shows that matching conditional expectations also ensures that the conditional expectations of states at later stages matches the conditional expectations of the true process.

**Proposition 2.** Let $(F_t)_{t \in [T]}$ be the MGBM process defined in (3), and let $(\bar{F}_t)_{t \in [T]}$ be the corresponding lattice process. Assume that nodes and probabilities of the lattice are such that

$$E(\bar{F}_{t+r}|\bar{F}_{t+r-1} = \bar{F}_{t+r-1,n}) = E(F_{t+r}|F_{t+r-1} = \bar{F}_{t+r-1,n}), \quad \forall \ n \in [N_{t+r-1}], \quad \forall \ r \in [s],$$

then

$$E(\bar{F}_{t+s}|\bar{F}_t = \bar{F}_{tk}) = E(F_{t+s}|F_t = \bar{F}_{tn}), \quad n \in [N_t].$$

**Proof.** W.l.o.g. $(F_t)_{t \in [T]}$ takes values in $\mathbb{R}$. Note that because the process is GBM, we have

$$E(F_{t+s}|F_t = x) = e^{sx},$$

where $\mu$ is the drift of the process. We show the result for $s = 2$, the general case follows by induction.

$$E(\bar{F}_{t+2}|\bar{F}_t = \bar{F}_{tk}) = \sum_{j \in N_{t+2}} p_{tkj} \sum_{m \in N_{t+2}} p_{tjm} \bar{F}_{t+2,m} = \sum_{j \in N_{t+1}} p_{tkj} E(\bar{F}_{t+2}|\bar{F}_{t+1} = \bar{F}_{t+1,j})$$

$$= \sum_{j \in N_{t+1}} p_{tkj} e^{\mu \bar{F}_{t+1,j}} = e^{\mu} \sum_{j \in N_{t+1}} p_{tkj} \bar{F}_{t+1,j} = e^{\mu} E(\bar{F}_{t+1}|\bar{F}_t = \bar{F}_{tk})$$

$$= e^{\mu} \bar{F}_{tk} = E(F_{t+2}|F_t = \bar{F}_{tk}) \quad \square$$

### 3.3. Stochastic Dual Dynamic Programming on Lattices

In this section, we show how SDDP can be used to approximate the optimal policy for the discrete price process represented by the lattice and how this policy can be implemented under the true continuous price process.
SDDP iteratively alternates between simulating the approximate optimal policy and updating the approximate post-decision value functions, \( \hat{V}_t \), which, in turn, define the approximate optimal policy of the next iteration.

At each iteration, \( l \in [L] \), SDDP performs a *forward pass*, during which a sequence of state transitions, \( (\hat{F}_t^l)_{t=2}^T \), is drawn from the lattice process to generate a sequence of sample resource states, \( (\hat{R}_t^l)_{t=2}^T \). Denote \( \hat{V}_t^l \) as the approximate post-decision value function after the \( l \)-th iteration. Then, sample resource states can be obtained by following the incumbent approximate optimal policy, i.e.,

\[
\hat{R}_{t+1}^l = \arg\max_{\pi_t \in \Pi_t} \left\{ C_t(\hat{F}_{tn}^l, \pi_t) + \hat{V}_{t-1}^l(\hat{F}_{tn}^l, R_t^l(\pi_t)) \right\}, \quad t \in [T-1], \quad l \in [L],
\]

with \( \hat{R}_1^l = R_1 \) and \( \hat{V}_1^l \equiv 0 \) (i.e., for \( l = 1 \), \( v_t = 0 \) in (18)).

After each forward pass, SDDP performs a *backward pass* adding new hyperplanes to the sets of supporting hyperplanes that define the approximate post-decision value functions.

To obtain the post-decision value function at any node of the lattice, we have to evaluate the nested CVaR of the value functions at all possible successor nodes. To accomplish this, we adapt the method proposed by Philpott et al. (2013) for lattices. In contrast to other methods for handling the nested CVaR, this approach requires no reformulation of the model and does not augment the state space of the dynamic program.

The method of Philpott et al. (2013) is based on the idea of reweighing transition probabilities. In general, for a discrete random variable \( X \) with possible realizations \( X_1, \ldots, X_M \) and corresponding probabilities \( p_1, \ldots, p_M \), the risk measure \( \rho_{\alpha, \lambda} \) has the dual representation

\[
\rho_{\alpha, \lambda}(X) = \min_{\xi \in \mathcal{U}} \sum_{i \in [M]} p_i \xi_i X_i,
\]

where

\[
\mathcal{U} = \left\{ \xi \in \mathbb{R}^M : \sum_{i \in [M]} \xi_i p_i = 1, \xi_i = (1 - \lambda) + \lambda \eta_i, 0 \leq \eta_i \leq \frac{1}{\alpha}, \forall i \in [M] \right\}.
\]

Using (16), this enables us to write

\[
\hat{V}_t(\hat{F}_{tn}, R_{t+1}) = \min_{\xi \in \mathcal{U}} \sum_{m \in [N_{t+1}]} p_{tnm} \xi_m \hat{V}_{t+1}(\hat{F}_{t+1,m}, R_{t+1}).
\]
Hence, $\rho_{ \alpha, \lambda}$ can be viewed as an expectation with a changed probability measure. The supergradients can be constructed from this representation by noting that

$$\partial_{R_{t+1}} \bar{V}_t(F_{tn}, R_{t+1}) = \sum_{m \in [N_{t+1}]} p_{tnm} \xi_m^* \partial_{R_{t+1}} V_{t+1}(\bar{F}_{t+1,m}, R_{t+1}),$$

(34)

where the weights, $\xi_m^*$, $m \in [N_{t+1}]$, are optimal for $R_{t+1}$ in (33).

As the optimal weights are easily identified by sorting the values $V_{t+1}(\bar{F}_{t+1,m}, R_{t+1}), t \in [T-1]$, the above method can be used to construct value function approximations during the backward pass similar to the expectation case.

We adapt this idea to the piecewise-linear approximate post-decision value functions. In this setting, the values corresponding to (33) at sample resource states, $\hat{R}_l$, are given by

$$a_l(F_{tn}, \hat{R}_{l+1}) = \sum_{m=1}^{N_{t+1}} p_{tnm} \xi_m^* \max_{\pi_{t+1} \in \Pi_{t+1}(\hat{R}_{l+1})} \left\{ C_{t+1}(F_{t+1,m}, \pi_{t+1}) + \hat{V}_{t+1}(F_{t+1,m}, \hat{R}_{t+2}(\pi_{t+1})) \right\},$$

(35)

$n \in [N_t], t = T - 1, \ldots, 1, l \in [L].$

Accordingly, as in (34), the slope vectors of the supporting hyperplanes at $\hat{R}_{l+1}$, are given by

$$b_l(F_{tn}, \hat{R}_{l+1}) = \partial_{\hat{R}_{l+1}} a_l(F_{tn}, \hat{R}_{l+1}), n \in [N_t], t = T - 1, \ldots, 1, l \in [L],$$

(36)

which can be obtained from the dual solution of (35).

The convergence properties of the algorithm are summarized in the following proposition.

**Proposition 3.** 1. The approximate value functions are upper bounds of the problem’s true value functions.

2. The approximate optimal policy converges to the optimal policy in finitely many iterations.

**Proof.** Note that the function $R_T \mapsto \hat{V}_{T-1}(F_{T-1,n}, R_T)$ is concave, since the initial resource state only appears in the right-hand side of the optimization problem in $T$. The hyperplanes are supergradients and hence upper bounds of the function. The minimum of these upper bounds remains an upper bound. An inductive argument over the stages establishes 1.

To prove 2, note that as all maximization problems are linear, the functions $R_T \mapsto \hat{V}_{T-1}(F_{T-1,n}, R_T)$ are actually piecewise-linear, and therefore equal to the minimum of a
finite number of supergradients (see Philpott and Guan 2008, Lemma 1, and Remark 1 above). Assume that the algorithm stops adding hyperplanes to the approximate post-decision value functions after iteration \( n \in \mathbb{N} \), and there is a sampling path \((\bar{F}_{1,n_1}, \ldots, \bar{F}_{T-1,n_{T-1}})\) such that \( \bar{F}_{T-1,n_T} = \bar{F}_{T-1,n} \) leading to a resource state \( R_T \) at the end of period \( T - 1 \) with \( \hat{V}_{T-1}(F_{T-1,n}, R_T) < \hat{V}_{T-1}(F_{T-1,n}, R_T) \). Then, by the Borel-Cantelli Lemma, the sequence \((F_{1,n_1}, \ldots, F_{T-1,n_{T-1}})\) will be sampled in some iteration \( n' > n \). Accordingly, a new supporting hyperplane will be added at \( R_T \), which contradicts the choice of \( n \). \( \square \)

To speed up the algorithm, we use the \( \varepsilon \)-approximation of Löhndorf et al. (2013), which rejects hyperplanes during the backward pass that do not improve the approximation by more than \( \varepsilon > 0 \). If \( \varepsilon \) is set to zero each post-decision value function node will consist of exactly \( l \) hyperplanes after iteration \( l \), as in conventional SDDP. Discarding hyperplanes leads to a looser approximation but also decreases problem size and hence computational effort as we will see in Section 4.3.

Since the approximate value function obtained by SDDP are upper bounds of the true value functions on the lattice, the optimal objective value of the optimization problem in \( t = 1 \) is an upper bound for the true value on the lattice. Let us refer to this value as \textit{lattice upper bound},

\[
\text{LUB} = \max_{\pi_1 \in \Pi_1(R_1)} \left\{ C(F_1, \pi_1) + \hat{V}_1^L(F_1, R_2) \right\}.
\]

While we cannot derive an upper bound of the problem under the true process, we can easily obtain a lower bound for the optimal expectation (\( \lambda = 0 \)), which we refer to as the \textit{process lower bound} (PLB). Denote \( \hat{\pi}^* \) as the optimal policy obtained under the lattice process. To compute the PLB, we draw \( K \) random sequences \((\hat{F}_k^k)_{t=1}^T\) from the true price process, and then use the following policy to make decisions,

\[
\hat{\pi}'_t \equiv \arg \max_{\pi_t \in \Pi_t(R_t)} \left\{ C(\hat{F}_t^k, \pi_t) + \hat{V}_t(\hat{F}_t^k, R_{t+1}(\pi_t)) \right\}
\]

whereby \( n = \arg \min_m \left\{ \| \hat{F}_t^k - \hat{F}_{tn} \|_2^2, m \in [N_t] \right\} \). The PLB is the sample average of the immediate rewards, which are accumulated over time following the policy \( \hat{\pi}' \),

\[
\text{PLB} = C_1(F_1, \hat{\pi}_1^*) + K^{-1} \sum_{k=1}^K \sum_{t=2}^T C_t(\hat{F}_t^k, \hat{\pi}'_t)
\]
Although the LUB is not an upper bound for the true problem, it is reasonable to conclude that if, for the risk-neutral case ($\lambda = 0$), the gap between LUB and PLB is small, the value function approximations are sufficiently accurate and the lattice is a good approximation of the true process.

4. Numerical Results

4.1. Instances

For our numerical analysis, we use the exact same parameters as reported in Lai et al. (2010). Their setup combines real price data with storage contract characteristics from the literature. Lai et al. (2010) calibrate four driftless versions of the MGBM price process, using the information available at the closing of NYMEX Henry Hub on 2006-01-03 (Spring), 2006-06-01 (Summer), 2006-08-31 (Fall), and 2006-12-01 (Winter). The interest rates as reported by the Dept of Treasury on the four selected days define the monthly discount factor of the model. The initial state of the MGBM, $F_1$, is given by the spot price and the futures prices of the first 23 maturities on each of the four days. Implied volatilities of the 23 futures prices are obtained from prices of NYMEX call options on natural gas futures. Storage instances vary in their injection (withdrawal) limits ranging from 0.15 mmBtu/month (0.3 mmBtu/month) for instance A to 0.45 mmBtu/month (0.9 mmBtu/month) in instance C. If not stated otherwise, model instance A with Spring prices will serve as our base case. The resulting problems have 24 decision stages. All constant model parameters are summarized in Table 5 in the Appendix.

Since we assume the market to be incomplete, we estimated the drift parameters using the same data as Lai et al. (2010) to facilitate a comparison of the results (see Table 4 in the Appendix).

In this setup, there are 23 prices, 23 contracts, and 1 storage, so that we end up with a 24-dimensional resource state and a 23-dimensional environmental state at the time of valuation. Classic stochastic-dynamic programming with a look-up table representation and ten possible states in each dimension would end up with a total of $10^{47}$ states. Similarly, if we were to replace a 100-node-per-stage lattice with the equivalent scenario tree, we would end up with $100^{23} = 10^{46}$ terminal nodes. Both alternatives are computationally intractable.
4.2. Implementation

Unless stated otherwise, for our numerical experiments, we ran SDDP for 1000 iterations and constructed a lattice with 1000 nodes per stage and backwards estimation as base case. The algorithm rejects supporting hyperplanes that improve the current approximation by less than \( \varepsilon = 0.001 \).

All lattices were built on the basis of \( K = 10^6 \) price scenarios for the quantization and forwards estimation step as well as \( K = 10^3 \) state transitions sampled from each node for backwards estimation. As stepsize rule for the stochastic gradient algorithm we used

\[
\beta_k = \frac{100}{k + 1000}, \quad k \in [K].
\]  

(39)

See Figure 2 for a visual comparison of the evolution of the spot price over time for different lattice sizes. Inspection of the graphs does not reveal any glaring inconsistencies between the true process and lattices with varying number of nodes per stage. Quite the contrary, important process characteristics seem to remain intact after quantization, namely mean, variance, and skewness of the unconditional distribution as well as serial dependence.

The rolling intrinsic policy can be obtained by performing forward passes without a value function, i.e., \( \bar{V}_t \equiv 0, \ t \in [T] \). In contrast to Lai et al. (2010), we do not consider future injection and withdrawal cost (see Section 2.2). Nevertheless, we found that the resulting discounted rewards deviate by less than 2\% from the values reported in Lai et al. (2010), and we therefore decided to ignore this detail in favor of model parsimony.

The implementation of the algorithms is available as a Java library called QUASAR. Computations were conducted remotely on a shared virtual machine (Dual-Xeon E5-2650) with 16 available CPU threads and 64 GB memory.

4.3. Approximation Quality

Lai et al. (2010) show that the rolling intrinsic policy is near-optimal under risk-free pricing. We can therefore use the rolling intrinsic policy as benchmark for this case. We are particularly

\(^1\) QUASAR is available from \texttt{http://www.quantego.com}
interested in the difference between the rolling intrinsic value and LUB/PLB to study the influence of lattice quantization on solution quality.

We varied parameters of the price process, the quantization method, number of nodes per stage, and the rejection threshold, ceteris paribus vis-à-vis the base case. Table 1 reports the total number of hyperplanes (Hyps), the number of linear programming problems solved until termination (LPs solved), the total computation time (Time), the lattice upper bound (LUB), the process lower bound (PLB), and the rolling intrinsic value (RI). All policies are executed for $10^5$ sample paths generated with the true process. Each run was repeated five times to study the variability of LUB and PLB under different random seeds.

Under default algorithm settings, the difference between the rolling intrinsic value and the PLBs is below 1.4 percent, irrespective of transition probability estimation method and season.
### Table 1  Influence of different algorithmic settings on solution quality

Although the difference is small, backwards estimation yields consistently higher rewards than forwards estimation and even marginally higher rewards than the rolling intrinsic policy for the Winter prices. Nevertheless, the difference in rewards between both policies is small when executed under the true process. We conclude that the approximate optimal policies are near-optimal.

The most notable result of the analysis concerns the difference between forwards estimation and backwards estimation with respect to the objective value of the MDP as given by the LUB. With forwards estimation, the LUB exhibits a substantial upwards bias, which implies that the optimal value of storage cannot be accurately calculated. With backwards estimation, by contrast, the gap between LUB and PLB vanishes and the bias is effectively eliminated. We conclude that, by achieving consistency between approximate value (LUB) and true value (PLB), the approximated optimal objective value of the MDP reflects the correct value of storage.

Changing the cut rejection threshold has a smaller influence on the PLB than on the LUB, where the latter increases as the threshold increases and decreases as the threshold decreases. Decreasing the threshold from $\varepsilon = 10^{-3}$ to $\varepsilon = 10^{-5}$ increases the number of hyperplanes by a factor of 3 (4), thereby increasing computation time by 37 (51) percent when backwards
estimation (forwards estimation) was used.

Changing number of nodes has an even smaller effect than the threshold. Computation times and the number of hyperplanes scale fairly linearly in the number of nodes. With 10000 nodes per stage and backwards estimation, the difference between process lower bound and rolling intrinsic value is not significant, which indicates that the optimal discrete policy eventually converges towards a near-optimal policy if the number of nodes is sufficiently large. Nevertheless, even with only 100 nodes per stage, the LUB (PLB) deviates less than 3 (2) percent from the rolling intrinsic value, which indicates that quality solutions can be obtained at low computational cost.

Although solving the optimization problem entails solution of several million LPs and produces a large number of supporting hyperplanes, an efficient parallel implementation keeps run times well below 30 minutes (with default settings), which highlights the practical applicability of our approach.

4.4. Indifference Pricing

For our next analysis, we turn towards studying the performance of our approach using indifference pricing. We use the MGBM with the estimated drifts as physical measure. Following Lai et al. (2010), we created 12 instances of the model that combine one out of the four variants of the price process with one out of the three possible storage capacity limits. We evaluate four policies: the first policy (EV) maximizes the expected value; the second policy (Low) maximizes a weighted combination of expected value and CVaR with $\lambda = 0.1$ and $\alpha = 0.05$; the third policy (High) maximizes only the CVaR with $\alpha = 0.05$ and $\lambda = 1$; and the fourth policy is the rolling intrinsic policy (RI).

Table 2 reports the computational results, including the indifference value of storage under the corresponding risk measure (LUB) as well as the expected discounted rewards from following the optimized policy (PLB) and its standard deviation for $10^5$ sample paths generated with the true process. Only in the risk-neutral case, LUB is equal to PLB. In all other cases, LUB is smaller than PLB, as it includes the tail loss which is lower than the expected value. This
Indifference pricing of natural gas storage contracts

<table>
<thead>
<tr>
<th>Instance</th>
<th>Storage Value (LUB)</th>
<th>Exp Dsc Rewards (PLB)</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Low</td>
<td>High</td>
<td>I</td>
</tr>
<tr>
<td>Spring A</td>
<td>4.25</td>
<td>3.88</td>
<td>3.76</td>
</tr>
<tr>
<td>Spring B</td>
<td>5.22</td>
<td>4.65</td>
<td>4.31</td>
</tr>
<tr>
<td>Spring C</td>
<td>5.60</td>
<td>4.86</td>
<td>4.40</td>
</tr>
<tr>
<td>Summer A</td>
<td>4.81</td>
<td>4.40</td>
<td>4.26</td>
</tr>
<tr>
<td>Summer B</td>
<td>6.20</td>
<td>5.62</td>
<td>5.29</td>
</tr>
<tr>
<td>Summer C</td>
<td>6.62</td>
<td>5.91</td>
<td>5.48</td>
</tr>
<tr>
<td>Fall A</td>
<td>4.18</td>
<td>3.74</td>
<td>3.53</td>
</tr>
<tr>
<td>Fall B</td>
<td>6.28</td>
<td>5.56</td>
<td>5.10</td>
</tr>
<tr>
<td>Fall C</td>
<td>7.45</td>
<td>6.57</td>
<td>5.94</td>
</tr>
<tr>
<td>Winter A</td>
<td>1.80</td>
<td>1.26</td>
<td>0.93</td>
</tr>
<tr>
<td>Winter B</td>
<td>2.46</td>
<td>1.73</td>
<td>1.17</td>
</tr>
<tr>
<td>Winter C</td>
<td>2.73</td>
<td>1.94</td>
<td>1.27</td>
</tr>
</tbody>
</table>

EV = maximize expected discounted rewards, Low = maximize nested CVaR with $\alpha = 0.05$, $\lambda = 0.1$, High = maximize nested CVaR with $\alpha = 0.05$, $I$ = intrinsic value, RI = rolling intrinsic value

Table 2: Indifference value of storage and discounted rewards (in $/mmBtu) with different risk measures

The results show that the rolling intrinsic approach is not optimal for decision-makers who are not perfectly risk averse. This result contributes to the literature which currently views the theoretical finding that the indifference value cannot be obtained from the expected discounted rewards, as it is done in rolling intrinsic valuation.

Moreover, the table shows that there exists a clear trade-off between high expectation and low standard deviation of the discounted rewards. A risk-neutral decision-maker who maximizes the expectation achieves the highest rewards but is also exposed to a significant amount of risk. Perfect risk aversion can be achieved by following the rolling intrinsic policy, which minimizes risk exposure but also leads to lower rewards. These findings are consistent across all problem instances and confirm our theoretical investigation from Proposition 1.

Let us now take a look at the distribution of the discounted rewards under different policies as they evolve over time. Figure 3 shows four fan charts of the cumulative discounted reward distribution for different values of $\alpha$ and $\lambda$.

The plots indicate that maximizing the expected value invites speculation with a high downside risk but also a high upside potential. While the corresponding policy invests in the beginning and postpones realization of rewards to the end of the time horizon, the risk-averse policy with $\lambda = 1.0$ realizes the largest fraction of the reward upfront with only marginal gains over time.
Figure 3  Fan charts of accumulated rewards for different risk preferences which visualize the unconditional distributions of rewards with dark colors around the median and lighter colors for more extreme quantiles of the distribution. Additionally, the unconditional mean is indicated by a black solid line with the average terminal profit on the right-hand side of each graph.

rolling intrinsic value as near-optimal. While our results confirm this view for risk-free pricing, the results for indifference pricing show that the rolling intrinsic policy is clearly suboptimal for a risk-neutral decision-maker.

4.5. Backtest with Historical Data

The previous section presented a comparison of different policies for simulated price paths. In this section, we report how these policies perform in a real trading situation when evaluated against actual prices instead of simulations. In particular, we are interested in a mutual comparison of indifference pricing and rolling intrinsic valuation based on historical futures prices.

We collected data of historical 12-month price futures curves of NYMEX Henry Hub natural gas futures from 1991 to 2015. We used instance A and computed rewards that accumulate
over a 12-month time horizon with zero interest rate from March to February. To estimate the
parameters of the MGBM price process, we used data from the respective previous 5 years for
each of the problems.

We investigated the decisions of a hypothetical gas trader who purchases a storage contract
each year and then manages the contract over the course of the year. This requires to solve
the optimization problem over a receding horizon and update model parameters accordingly.
Starting in 1996, each year began with a clean slate and a 12-month planning horizon. After
one problem was solved, the immediate reward from implementing the first-stage decision was
recorded and the final resource state was passed on as initial state to the subsequent 11-month
problem. Before solving the subsequent problem, MGBM parameters were reestimated and a
new lattice was built. This process was repeated over a receding horizon until only 1 month was
left. Then, a new contract was purchased. The resulting sample of records contains 20 annual
and 240 monthly reward observations.

The results of the backtest for different policy parameters are summarized in Table 3. The
rolling intrinsic policy which serves as benchmark lead to an average annual reward of 0.775
($/mmBtu) with a standard deviation of of 0.815. The table reports the difference in average
rewards between the rolling intrinsic policy and the approximate optimal policies for different
parameters, along with the p-values of a two-sided t-test for equal means.

The results show that all indifference pricing policies lead to higher rewards than the rolling
intrinsic policy. In line with the findings from the simulation study, we observe a clear trade-off
between reward and risk. The difference to the rolling intrinsic value is in the range of 8 to 40
percent and is more sensitive towards the risk preferences than to algorithmic parameters.
Although the difference is greatest for the risk-neutral case, the result is not significant at the 90% confidence level, due to the high variability of the reward distribution. Setting $\lambda \geq 0.5$ decreases risk as expected, and $p$-values decrease accordingly, so that for these instances differences become statistically significant.

Varying the number of nodes per stage had almost no effect. This confirms the result of the simulation study which showed that coarse discretizations already provide quality solutions.

The results demonstrate that the rolling intrinsic policy is not optimal for a risk-neutral decision maker and more profit can be made by solving the underlying stochastic-dynamic optimization problem. We conclude that gas storage valuation benefits from using indifference pricing instead of risk-free pricing, which provides additional evidence for the incompleteness of gas markets.

5. Conclusion

We present a novel approach for valuation of gas storage contracts using indifference pricing that fully embraces the incompleteness of gas markets. The underlying decision problem is modeled as a high-dimensional stochastic-dynamic optimization problem that jointly considers futures trading and storage operation and explicitly models a trader’s risk preferences.

We introduce ADDP as a generic framework for solving discrete-time, continuous-state, risk-averse MDPs and demonstrate that the method is capable of handling the high-dimensional optimization problem which has been considered intractable so far, (e.g. Lai et al. 2010, Nadarajah et al. 2015). The algorithm relies on a discretization of the continuous-state futures price process to a scenario lattice. We find that the approximate optimal policy is highly sensitive to how well the lattice matches the conditional first moment of the price process. To avoid a severe upward bias in the valuation, we propose a novel backwards estimation method which ensures that the conditional means of the lattice coincide with the means of the true process.

Lai et al. (2010) and Secomandi (2015) provide theoretical and empirical support for the near-optimality of the rolling intrinsic value under risk-free pricing. We extend this result by showing that, in an incomplete market, the rolling intrinsic solution is only optimal for decision-makers who are perfectly risk-averse. In all other cases, we demonstrate in a simulation study that
the rolling intrinsic solution leads to sub-optimal results. We further strengthen this finding by conducting a comprehensive backtest using historical data and show that more profit can be made by following an indifference pricing policy.

Future work should look at the effect of different futures price models, for example, the model proposed in Clewlow and Strickland (2000), as well as the interplay of spot and futures trading.

References


Appendix A: Problem Parameters

<table>
<thead>
<tr>
<th>$\mu_t$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0092</td>
<td>0.0095</td>
<td>0.0101</td>
<td>0.0101</td>
<td>0.01</td>
<td>0.0097</td>
<td>0.0098</td>
<td>0.01</td>
<td>0.01</td>
<td>0.0099</td>
<td>0.0101</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\mu_t$</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0102</td>
<td>0.0108</td>
<td>0.0113</td>
<td>0.0114</td>
<td>0.0103</td>
<td>0.0107</td>
<td>0.0111</td>
<td>0.0113</td>
<td>0.0116</td>
<td>0.0111</td>
<td>0.0112</td>
<td>0.0109</td>
<td></td>
</tr>
</tbody>
</table>

Table 4 Drift parameters of the physical measure.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Notation</th>
<th>Instance</th>
<th>Value(s)</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage limit</td>
<td>$\pi$</td>
<td></td>
<td>1.00 mmBtu</td>
<td></td>
</tr>
<tr>
<td>Initial resource state</td>
<td>$R_i$</td>
<td></td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Injection (withdrawal) limit</td>
<td>$\overline{7}, \overline{\overline{7}}$</td>
<td>A 0.15 (0.30) mmBtu</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>B 0.30 (0.60) mmBtu</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>C 0.45 (0.90) mmBtu</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Injection cost</td>
<td>$c^i$</td>
<td></td>
<td>0.02 US$/mmBtu</td>
<td></td>
</tr>
<tr>
<td>Withdrawal cost</td>
<td>$c^w$</td>
<td></td>
<td>0.01 US$/mmBtu</td>
<td></td>
</tr>
<tr>
<td>Injection loss</td>
<td>$d^i$</td>
<td></td>
<td>0.99 mmBtu/mmBtu</td>
<td></td>
</tr>
<tr>
<td>Withdrawal loss</td>
<td>$d^w$</td>
<td></td>
<td>1.01 mmBtu/mmBtu</td>
<td></td>
</tr>
<tr>
<td>Interest rates</td>
<td>$\delta$</td>
<td>Spring 4.74 % p.a.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Summer 5.05 % p.a.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fall 5.01 % p.a.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Winter 4.80 % p.a.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time horizon</td>
<td>$T$</td>
<td></td>
<td>24 month</td>
<td></td>
</tr>
</tbody>
</table>

Table 5 Parameters for the numerical study.