OPTIMIZATION FRAMEWORK FOR NATURAL GAS STORAGE ASSETS SUBJECT TO PRICE UNCERTAINTY

by

Paul Jobinpicard

Submitted in partial fulfillment of the requirements
for the degree of Master of Applied Science

at

Dalhousie University
Halifax, Nova Scotia
July 2019

For my parents, who believed in me even when I didn't believe in myself.

Table of Contents

List of Tables

List of Figures

Abstract

Natural gas offers a clean and efficient fuel-burning method of power generation and heating. North America, specifically, features a sprawling network of natural gas pipelines and storage facilities that bring natural gas (NG) from source to customer. Like many commodities (oil, electricity, etc.), the price of NG is subject to uncertainty. Natural gas storage assets can allow marketing companies to profit by injecting or withdrawing gas at opportune moments. Even simple contracts; however, involve large numbers of decisions and constraints that must be considered. Furthermore, because of the financial nature of storage decisions, it is desirable to measure or include risk in the decision-making process. This thesis presents optimization framework (created for an industry partner) to serve as a decision support tool for natural gas storage assets. Mixed-integer-programming models are presented for two types of storage decisions: futures and cash. Model framework includes novel constraints that consider the impact of asset inventory on injection and withdrawal rates. Including the ratcheting constraints is found to be essential in determining feasible injection and withdrawal decisions when assets are subject to inventory ratcheting. For both cash and forward decisions, Stochastic, Robust and Distributionally Robust Optimization alternatives are presented to account for risk. Parameterization of the risk-management alternatives allows for a user to obtain tactical plans for storage assets while considering varying levels of risk tolerance and averseness.

List of Abbreviations Used

AMA	Asset Management Agreement
Btu	British Thermal Unit
CVaR	Conditional Value at Risk
DRO	Distributionally Robust Optimization
EL	Equally Likely
LP	Linear Program
MILP	Mixed Integer Linear Program
MMcf	Million Cubic Feet
NG	Natural Gas
RO	Robust Optimization
SAA	Sample Average Approximiation
USFTC	United States Futures Trading Commission
VaR	Value at Risk

Acknowledgements

Thank you to all of the professors, teaching assistants, administrators and staff who made my nearly seven years at Dalhousie a memorable and wonderful experience. Thank you, specifically, to the Industrial Engineering department and my advisors; Dr. Diallo and Dr. Venkatadri, who had a tremendous impact on my research. I would like to thank Dr. Ahmed Saif, who taught me many of the optimization approaches contained within. Additionally, I would like to thank our research partner for providing the inspiration for this research and to Alex, Gary and Nii, who were enjoyable colleagues throughout the project.

Chapter 1

Introduction

Natural gas (NG) plays an important role in global power generation and offers a cleaner alternative to coal and other fossil fuels. NG, considered a fossil fuel, is created by decaying plant and animal matter that is exposed to intense pressure over millions of years. NG is extracted from naturally occurring reservoirs, which are found all over the world. In 2017, global demand and consumption of NG saw extraordinary growth. NG global production nearly doubled the average production growth rate of the previous ten years, reaching 4% in 2017. The global NG consumption growth rate increased to 3%, the highest level since 2010 [1].

Many companies are finding opportunity in this growing industry, especially in North America. North America's open NG market creates opportunities for marketing companies and other third-party participants to profit from the stochastic nature of NG supply and demand. One particular area of interest for NG marketing companies is NG storage facilities. NG storage facilities present opportunities for companies to take advantage of fluctuating NG demand and prices by storing excess gas (when the price is low) or withdrawing stored gas to meet increased demands (high price). NG storage, trading and business decisions are often made using approximate approaches that rely on the trader's intuition and experience. Additionally, operational constraints and characteristics of NG storage facilities can be overlooked when making trading decisions.

Market uncertainty and different operational and practical constraints present an opportunity to examine NG storage trading decisions from an operations research perspective. This thesis outlines an optimization framework created for a utility company, to be used as a decision support tool for natural gas storage decisions. The next three sections of this chapter will serve as a primer for the natural gas industry in North America. These sections cover relevant concepts and terms used throughout this thesis. The last section will present the thesis motivation and outline.

1.1 NG Industry Overview

We begin by examining the work of Sturm [2], who provides a detailed history of the NG industry in the United States (and inherently all of North America), specifically, how the market developed into its current state. The Natural Gas Act was passed by Congress in 1938 to protect public interest and regulate the interstate industry created by a vast pipeline network. The Federal Power Commission was the government agency created to enforce the Natural Gas Act. As a response to wildly fluctuating prices, the United States Supreme Court ruled to pass the Phillips decision in 1954. In passing the Phillips decision, the Supreme Court ruled that the sale of natural gas at the source (wellhead) was subject to government regulation. This decision allowed the Federal Power Commission to gain control over NG prices.

The hope was that by regulating NG procurement costs they could control the price being paid by customers (the public). Government regulation essentially limited the price that procurement companies could charge pipeline companies. Regulation caused many NG producers to shift their focus from interstate pipelines to intrastate pipelines. Intrastate pipelines were governed locally and not as heavily regulated as interstate pipelines. Because intrastate pipelines offered higher profit margins, most interstate pipelines were struck with supply shortages in the 1970s. To combat this, the Natural Gas Policy Act was passed in 1978, reversing the regulation put in place by the Phillips decision. Regulation was thought to have protected consumers from NG producers driving prices up, but it instead caused a supply crisis.

Further deregulation continued and was overseen by the Federal Energy Regulatory Commission. The once heavily-regulated and stagnant NG market saw historic increases in business activity due to deregulation. Deregulation created the open market that exists today where companies (not just the pipeline companies) may purchase capacity in pipelines, storage facilities and enter into many different types of contracts involving NG production and transportation on a non-discriminatory basis. Storage facilities, specifically, saw a large paradigm shift due to deregulation. Originally the facilities were primarily used to balance supply and demand. After deregulation NG storage facilities began to see use from third-party (marketing) companies as both opportunities for profit and as risk-management tools.

1.2 North American NG Supply Chain

North America plays a vital role in current-day NG production. The United States and Canada are, respectively, the first and fourth largest producers of NG globally. The United States accounts for 21% of global NG production [3]. NG productivity in North America has increased over the last 15 years. Canadian NG production has remained relatively stable, NG production in the United States increased from around 50 billion cubic feet per day in 2006 to over 72 billion cubic feet per day by the end of 2016 [3]. While North America produces the largest amount of NG of any region, most of the production remains inside of North America. For example, all of Canada's NG exports are to the United States. This is most likely due to the close trading relationship between the United States and Canada. Another possible explanation for this is transatlantic transport of NG is costly compared to the other supply alternatives for the European market.

Increased production of NG has put more pressure on the North American NG supply chain. The North American NG supply chain can be separated into the upstream, midstream and downstream industries [4]. North America has a dense network of pipelines that connects the different echelons of the NG industry. Figures 1.1 and 1.2 display the pipeline network in the United States and Canada

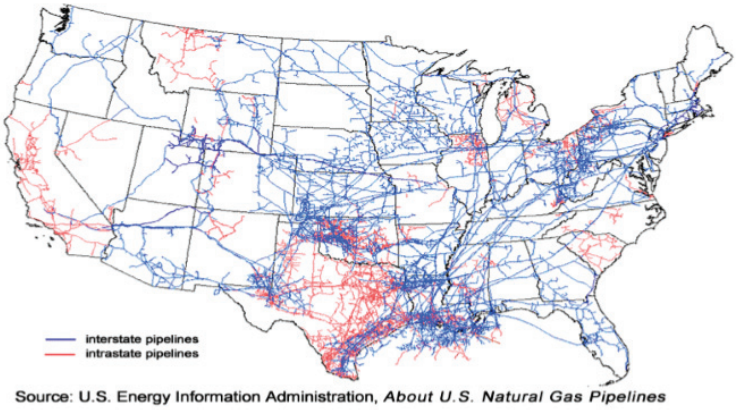

Figure 1.1: US NG Pipeline Network. Source: Energy Information Administration [4]

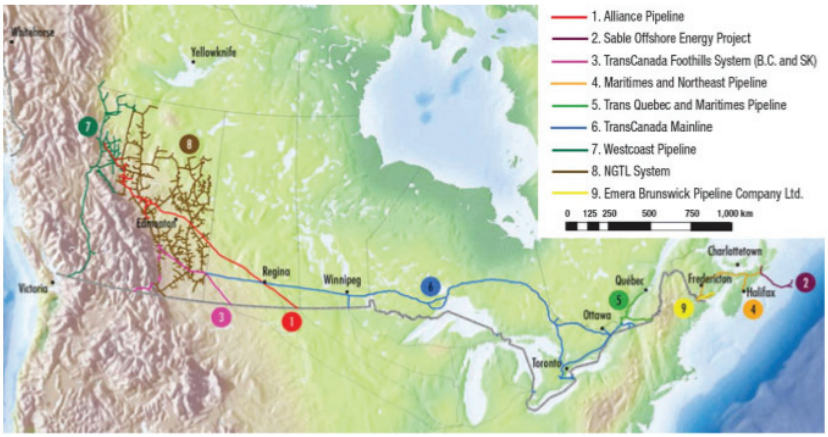

Figure 1.2: Canadian NG Pipeline Network. Source: Energy Information Administration [4]

Companies operating in the upstream industry are involved in exploration, drilling and production of NG. In the United States, most upstream companies are located in the Gulf of Mexico, where the largest reserves of NG are located. The downstream industry is made up of the local distribution companies that distribute gas to customers using local distribution pipelines [4]. Most of the demand in North America comes from the North Eastern United States. The cold winters, along with large population areas like New York City and Boston, create the high NG demand. The midstream industry contains NG processing plants and the subject of this study: NG storage facilities [4].

NG travels downstream from the exploration and procurement companies, through the vast pipeline network and finally gets distributed to end-users. NG is used in many capacities in North America such as: residential heating, commercial/industrial purposes (e.g. waste treatment) and power generation. Demand for NG is dynamic and unforeseen changes in demand put stress on the NG supply chain. NG storage facilities, located along the pipeline network, can be used to supplement unusually high demand and accommodate surplus during usually low demand. Storage facilities also create business opportunities for marketing companies, as they allow NG to be purchased during points of low demand (low price) and sold later during periods of high demand (high price).

NG storage facilities are (often) naturally occurring entities wherein NG can be

extracted or deposited according to an array of operational characteristics. NG storage facilities are used to account for fluctuating demand seen throughout the NG market in North America. For example, if a stretch of cold weather created a NG demand that could not be met by normal NG production, NG could be extracted from a storage facility to meet the demand. Similarly, if demand was lower than expected, the surplus NG could be injected for later use.

There are five different types of NG storage facilities utilized in North America: salt caverns, mines, aquifers, depleted reservoirs and hard-rock caverns [5]. Hard-rock caverns and mines are less commonly used than their counterparts, so their operational characteristics are not discussed in detail. Figure 1.3 provides an illustration of the different types of NG storage facilities.

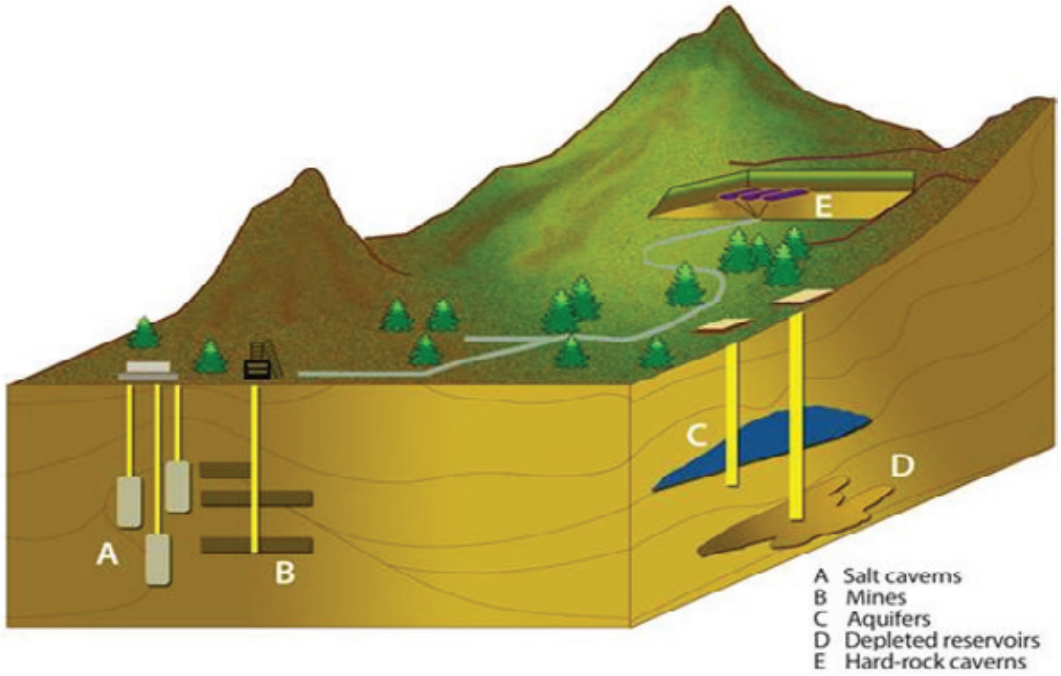

Figure 1.3: Different Types of NG Storage Facilities. Source: Energy Information Administration [4]

The different types of NG storage facilities and their distinguishing operating characteristics are discussed by Thompson et al. [5]. They classify four operating characteristics that differentiate NG storage facilities: base and working gas capacities, deliverability, injection capacity and cycling. Base capacity is the amount of gas required in a storage facility to keep sufficient pressure. This gas is almost never

removed from the facility. Working gas capacity is the amount of gas that is available to be bought and sold.

Deliverability is defined, by Thompson et al. [5], as the rate at which gas can be withdrawn from a NG storage facility, usually measured in million cubic feet per day (MMcf/d). Deliverability is often expressed in terms of its heat content (dekatherms/day or Btu). Injection capacity refers to the rate at which NG can be injected into a storage facility and is measured in the same way as deliverability. Finally, cycling is the number of times per year working gas volumes can be withdrawn from a storage facility.

Depleted reservoirs have the lowest deliverability and injection rates of the aforementioned NG storage facilities. Depleted reservoirs also contain very high amounts of base gas. Depleted reservoirs are common in the Northeastern United States, where 56% of the total US NG storage is located. Because of this, depleted reservoirs account for the majority of NG storage in the United States.

Salt caverns have much lower base gas requirements than depleted reservoirs. Salt caverns also provide the highest deliverability and injection rates of any NG storage facility. Additionally, salt caverns are capable of daily and monthly injections and withdrawals; making them very dynamic facilities. Most salt cavern storage facilities are located in the Gulf area of the United States, but there are some located in Michigan and Ontario. While salt cavern storage seems to have the most robust operating characteristics, because they are not naturally occurring close to the largest NG market (NE United States), they do not see as much use as depleted reservoirs. Aquifers possess operational characteristics that are generally between those of depleted reservoirs and salt caverns. Aquifers also tend to have very high base gas requirements (upwards of 80% of their total storage capacity).

Another important characteristic of NG storage facilities is the impact of inventory levels on deliverability. The deliverability rate of a facility decreases with inventory levels. It is easier to withdraw gas the higher the inventory levels of the facility are. The different operational characteristics of NG storage facilities are important to note, as they can impact future trading decisions. Natural gas storage contracts can be subject to "Ratcheting Agreements" wherein the company operating the storage field provides injection/withdrawal rates for varying inventory levels. This concept is

explored in the modeling section of this thesis.

With a basic understanding of the origins of the NG industry, the current state of the industry in North America and the concept of NG storage, the business model of NG in North America can be introduced.

1.3 NG Business Model

The current-day business model is a result of the aforementioned deregulation that occurred in the United States, and inherently Canada (as the large US market drove the entire North American NG industry). Opportunities for trading companies were created by the separation of business activities along the NG supply chain. Sturm [2] provides an overview of the NG market in North America. They divide the NG market into two segments: the physical (cash) market and the financial natural gas market.

The cash market involves daily transactions of NG between suppliers and buyers. The cash market involves here-and-now decisions, meaning gas can only be purchased *today*. Today's NG price is referred to as the spot price. While companies may plan to make cash purchases into the future, the only known price is the spot price; every other day's price is subject to change. The cash market, while inherently risky, can be used in tandem with NG storage facilities to make profits off large swings in the daily prices. For example: a marketing company (such as our client) may buy gas in the summer months when the price is normally lower and hold it until the winter. Then, if a cold-snap hits the Eastern United States, they can sell their stored gas on the cash market to hopefully make a large gain. The difference between purchase and selling price of NG is commonly referred to as the spread.

Because transactions can only be made on the spot price of NG in the cash market, there is a large degree of uncertainty regarding the future behavior of prices. Transacting in the forward NG market however, allows companies more flexibility and hedging opportunities than operating in the cash market. In the forward market, futures contracts are used as the financial instrument for purchases and sales of commodities. The United States Futures Trading Commission (USFTC) [6] provides an overview of futures contracts in the context of the commodities market in North America. They define a futures contract as an agreement to buy or sell a certain

amount of a commodity at an agreed upon price at specific points in the future. While most contracts enforce the delivery of the actual commodity, some futures agreements allow cash to be delivered in lieu of the commodity.

The agreed upon delivery date(s) are referred to as futures maturities. The USFTC indicates that trading companies often utilize futures as hedging tools to offset the risk imposed by the stochastic nature of commodities prices. Hedging refers to protecting oneself from loss by making other compensating transactions. It is indicated that "hedgers" make up a majority of futures traders [6].

The USFTC also defines another group of traders, referred to as "speculators". "Speculators" attempt to profit from the volatile nature of NG prices. For example: expecting a period of low demand, the futures price for a certain price may be low. A "speculator" may buy a forward contract to purchase gas for that period in the hopes that they can sell it at a later point for a large spread. NG storage facilities can again be utilized to store gas upon contract maturity, to be sold on either the cash or forward market, hopefully for a profit. While there are many other aspects to futures contracts and trading, they are outside the scope of this thesis.

Because the spot price of NG does not translate to a tradeable instrument [6], futures contracts are an essential aspect of NG trading and storage agreements. Storage agreements are often several months or years long. Because futures contracts can be bought for future delivery, they interface well with NG storage facilities, as the company can plan to inject/withdraw the specified amount of gas **at a fixed price**.

Operating on the forward market provides marketing companies with additional flexibility when purchasing and selling NG. While a futures contract outlines the purchase of NG at a fixed price, this price is subject to the current futures market. Like any commodity, the futures price of NG is constantly changing. Futures contracts provide companies with the opportunity to take advantage of the changes in the forward market.

Until delivery of the underlying commodity, a forward contract is simply a financial position a company takes. This position, as futures represent tradeable instruments, can be dropped at any point in time before maturity. For example: a company purchases gas in May for delivery in August, but three weeks later, as the futures market changes, they see a more profitable opportunity by selling the forward contract

for delivery in August. Therefore, the trader may sell their existing delivery position on the futures market.

Another important concept in futures markets is the forward curve. Platts [7] defines a forward curve as a "series of sequential prices either for future delivery of an asset or expected future settlements of an index.". It is important to understand that forward curves do not display projections for the evolution of NG spot prices, but rather the **current** price for future delivery of NG. Figure 1.4 , obtained from Platts [7], provides a graphical example of a forward price curve. Prices of the commodity of interest are plotted on the Y-axis against distinct points in time along the X-axis. Forward curve data can be obtained from exchanges, energy market data publishers

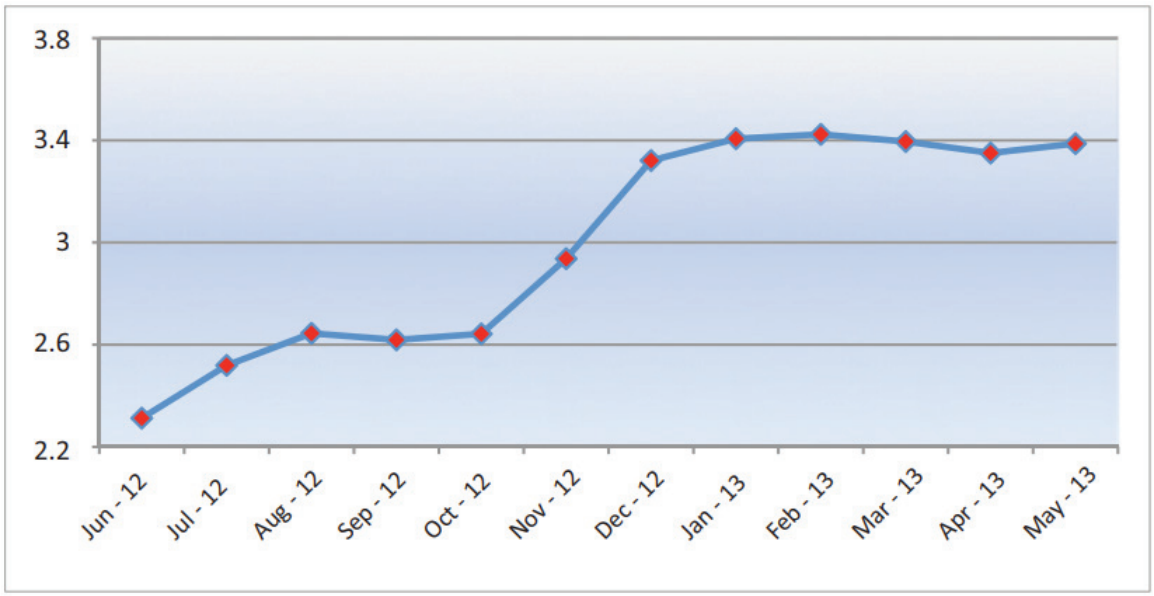

Figure 1.4: Sample Forward Price Curve. Source: Platts [7]

(who charge a fee for usage of their forward curve information), brokers, data distributors, system vendors, consensus curve publishers and internally developed curves. Forward curves utilize available data (historical data, weather data, market data) to estimate current futures prices. Forward price curves can be viewed as a "snapshot" of the current futures market, as the forward curve is subject to change throughout time.

Sturm [2] provides an example of a simple NG storage contract trading decision. Assume gas was purchased on the forward market for \$1.90/MMBtu. If the trader notices that NG futures prices next month are quoted at \$2.00/MMBtu, they must

decide if that spread is enough to cover the variable costs of the storage contract. Sturm provides a carrying and variable cost of $.085/MMBtu/d, which results in a profit of $.015/MMBtu. In most cases, contracts are made for quantities exceeding 10000 MMBtu, so this storage contract would be deemed very profitable. The following visual was adapted from Sturm to display the aforementioned storage transaction. While Sturm includes a holding cost in his example transaction, marketing compa-

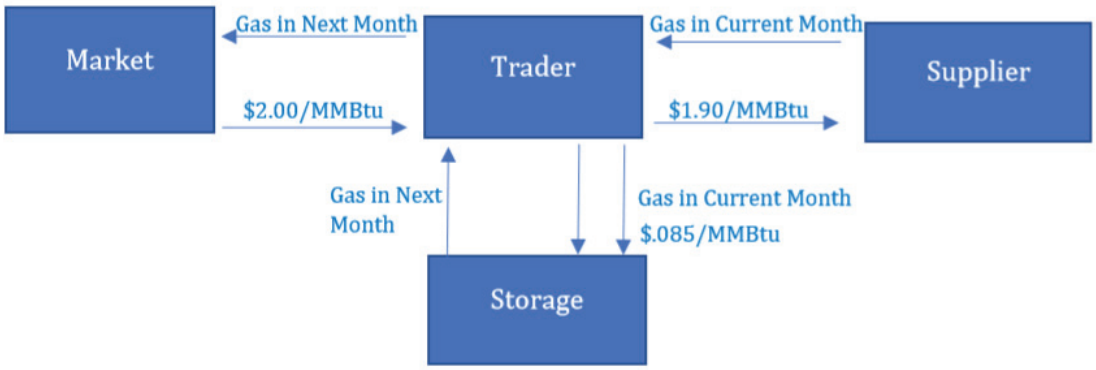

Figure 1.5: Simple NG Storage Agreement. Source: Sturm [2]

nies (including our client) often disregard holding costs, as they do not significantly impact the value of the storage decisions.

1.3.1 Behavior of NG Prices

The motivation behind NG storage contracts can be better understood by examining the historic behavior of NG. In general, the price of NG is lower in warmer months and higher in the winter when large cities' NG consumption increases due to heating. This behavior is confirmed by examining historical NG futures price data obtained from the New York Mercantile Exchange (NYMEX), which is one of the largest commodities futures exchanges in the world. Figure 1.6 displays the average futures price by month for the last 30 years. It should be noted that years 2000-2008 were excluded from this analysis as they were considered outliers due to the economic bubble and subsequent crash.

NG prices are generally historically higher during the winter months in North America and lower during warmer months. The peak in June is most-likely a result of increased power consumption caused by high temperatures in the summer months.

This behavior provides opportunity for gas to be purchased for lower prices in the summer and sold for a profit during the winter months.

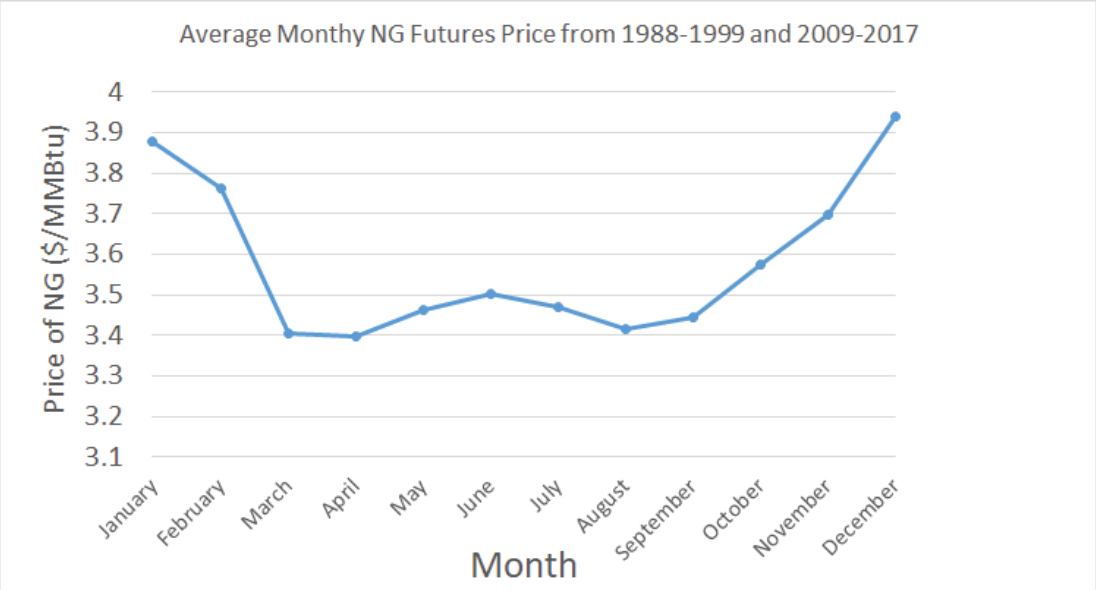

Figure 1.6: Historical NG futures prices by month from 1988-1999 and 2009-2017. Source: Index Mundi [8].

The impact of weather on NG prices can be further displayed by examining historical futures prices for 2005, as displayed by Figure 1.7. Here we see relatively stable monthly prices until August, where there is a distinct spike. In late August 2005, Hurricane Katrina devastated the Gulf of Mexico causing over $80 billion in damage. As previously discussed, the majority of United States' NG production is located in the Gulf of Mexico [2]. Much of the NG infrastructure was damaged or hindered by the 2005 hurricane season, which resulted in a supply shortage and record high NG prices. Again, companies can take advantage of extreme weather events by selling stored gas for extremely high prices.

While NG prices do exhibit seasonal tendencies, they are also extremely volatile. To emphasize this, a one-way Anova test was performed on average futures prices for each month. The hypothesis test is outlined by Figure 1.8. The results obtained from the Anova test are displayed in Figure 1.9. The resulting p-value is far greater than the significance level of the test, indicating there is no statistical difference between each month's mean historical price.

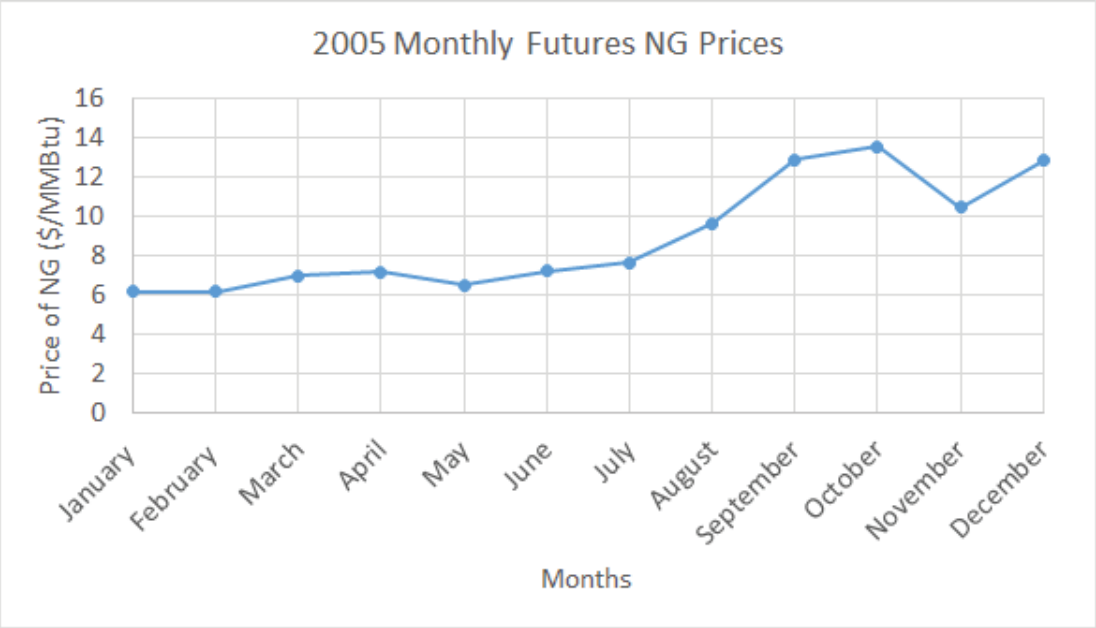

Figure 1.7: Historical NG futures prices by month for 2005. Source: Index Mundi [8].

Method

Null hypothesis	All means are equal
Alternative hypothesis	Not all means are equal
Significance level	α = 0.025

Equal variances were assumed for the analysis.

Figure 1.8: Anova hypothesis test, performed on monthly averaged futures prices. Source: Index Mundi [8].

Analysis of Variance

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Factor	11	8.396	0.7633	0.91	0.534
Error	235	197.694	0.8412		
Total	246	206.090			

Figure 1.9: Anova results for historical futures prices. Source: Index Mundi [8].

1.4 Thesis Motivation

This thesis was motivated by a collaboration between Dalhousie University and an industry partner, henceforth referred to as The Utility Company. The Utility Company utilizes NG storage assets to profit from changing NG prices. As prices in the forward and cash markets change, The Utility Company utilizes NG storage facilities to store gas purchased at lower prices until the gas can be sold at a higher price. Additionally, The Utility Company often enters into operational agreements where they are given possession of a NG storage asset and must fill the asset by the end of term for the least amount of money.

The problem is further complicated as The Utility Company may have access to a storage facility for multiple years, which results in a large number of possible injection and withdrawal decisions especially in the cash market. In addition to the sheer number of possible decisions, the storage contract is constrained by the operating characteristics of the facility (e.g. capacity and deliverability).

Because of the large number of possible decisions and various operational constraints surrounding the facility, the process of analytically optimizing a storage asset requires much time and effort from the traders. The goal of the project was to create an optimization framework to aid in the decision-making process for natural gas storage assets. The hope was that, by using optimization techniques, the traders could obtain optimal tactical plans without hours of analysis. The models presented in this thesis were motivated by the project with The Utility Company, and several variants were implemented for future use.

1.5 Problem Statement

The thesis addresses two specific optimization models for NG storage assets subject to price uncertainty. Models are developed to provide traders at The Utility Company with optimal injection and withdrawal instructions such that the expected profit from the various transactions is maximized, subject to internally developed price simulations. Models were developed for two specific problems: Forward-to-Forward Transactions and The Summer Fill Problem.

In the Forward-to-Forward (F2F) problem, we consider optimizing a NG storage asset where the trader only makes transactions the futures market. The goal of this problem is to buy/sell forward contracts such that the cashflow of these transactions is maximized, subject to the internal price simulations. These transactions are constrained by the various operational constraints of the storage facility (e.g. deliverability and capacity). The trader must consider the operating characteristics as they determine the feasibility of their trades. This problem also includes situations where the NG storage facility is subject to inventory ratcheting. Inventory ratcheting can be imposed on NG storage facilities by their operating companies and reflects the physical characteristics of the facility along with the impacts of market liquidity. As a well becomes more full, the pressure increases and it becomes harder to inject gas. Additionally, when a NG well is full, a storage company will expect many withdrawals (as the well would be filled for periods of high demand). Because of this, the company operating the well will limit their deliverability based on the well inventory via a ratcheting agreement. The agreement complicates the feasible injections and withdrawals and must be considered when applicable.

The Summer Fill Problem considers the optimization of a NG storage asset where the trader only makes transactions on the cash market. Unlike for transactions on the forward market, NG cannot be purchased for future delivery on the cash market. In this problem the trader has access to a NG storage facility for the summer (April-October). In those 214 days, the trader must also fill the storage well to its capacity to prepare for increased winter demand. Because it does not take the entire summer to fill the well, the trader will also be making cash purchases and sales of NG to try to offset the cost of filling the well. The goal of this problem is to obtain the purchases/sales of NG on the cash market, such that the overall cashflow is maximized (subject to the internal price simulations), all while ensuring the well is full at the end of the month.

1.6 Thesis Contributions

This section of the thesis covers its scientific contributions. Contributions are assigned to the following categories: practical, academic, and data. Practical contributions refer to the contributions this thesis made to our industry partner. Academic

contributions refer to contributions towards academia. Finally, data contributions refer to contributions made through analysis of the provided data.

The inventory ratcheting constraints are a large practical contribution, as they allow a user to easily identify feasible injection and withdrawal amounts, when an asset is subject to inventory ratcheting. This analysis can be complex and requires much time for analysis. The thesis work also serves as an effective decision support tool for the industry partner, as they can use it to aid in NG storage decisions. Chance Constrained, Robust, Distributionally Robust and Utility modelling alternatives aid in risk management by taking the uncertainty of the price simulations into account. This framework can also assist in baseline valuation of a storage asset. Finally, the framework can be used to validate tactical plans (by ensuring feasibility).

Next, in terms of academic contributions, this thesis presents novel linear constraints to account for inventory ratcheting. This thesis shows the impact a ratcheting agreement can have on feasible injection/withdrawal amounts. This thesis also presents novel applications of Chance Constrained, Robust and Distributionally Robust Optimization. This thesis presents how each model framework can be used to account for uncertainty within the context of our problem. Finally, the concept of utility functions was applied to a NG storage well's inventory to serve as a risk management tool.

Data contributions can be divided into two sections: forward price simulations and cash price simulations. For forward price simulations, this thesis identified that volatility decreased with time (reversing-funnel). Additionally, the mean-centered nature of the price simulations was revealed. It was also determined that a mixture distribution can be fit to monthly prices. Data analysis also revealed data seasonality (higher prices in the winter months) and the natural spread of the price simulations. For the cash simulations, this thesis revealed increasing volatility with time (funnel effect). The daily prices were also found to be approximately normally distributed. Additionally, a dip in prices in October was identified and analysed. Analysis revealed both financial and customer causes for this dip.

1.7 Thesis Outline

The remainder of this thesis is organized as follows: first in Chapter 2, a literature review of existing and relevant operations research approaches pertaining to NG storage assets is performed. Next, the forward and cash market modeling frameworks are presented in Chapters 3 and 4, respectively. Both forward and cash chapters contain data analysis on provided price data in addition to model approaches presented. The results from each approach are discussed and analyzed. Finally, conclusions are drawn from the aforementioned analysis and recommendations for future work are provided.

Chapter 2

Literature Review

Before the modeling process began, a literature review was conducted to investigate existing approaches to the NG storage problem. The following chapter presents relevant approaches and discusses their applicability in the context of this project. The presented approaches were considered and used as references throughout the modeling process. While many approaches are covered, only approaches relevant to the case under study are presented in detail.

2.1 Linear Programming Approaches: Modeling NG Price Progression

Contesse et al. [9] present a mixed-integer approach for modeling NG purchases and transportation. Their model framework includes three parties: NG producers, transporters and local distribution companies. Their decision-support model allows the various complexities of purchasing and transporting NG to be considered and reflected in the optimal solution. They present an array of constraints that model elements related to the purchase and sale of gas, taking into account items such as: injection capacities, sales to different entities and transportation capacities. Their results showed that, while the resulting MIP formulation was quite large, it solved easily. They indicate that their model can be used to effectively support operational decision, as well as various contract negotiations.

Next, let us consider the linear programming model presented by Lai et al. [10] that pertains specifically to NG storage, which includes futures contracts as a trading operation. As previously discussed, futures contracts are frequently used throughout commodities trading as they help mitigate risk and are used as hedging tools. They consider a futures contract that contains N maturities, contained within a set $\mathcal{F} := \{0, ..., N-1\}$. Their standard notation $F(T_i, T_j)$ denotes the futures contract price at time T_i and maturity at time T_j where $j \geq i, \forall i, j \in \mathcal{F}$. Additionally, they define the spot price as $F(T_i, T_i)$. Note the spot price refers to the price of gas at the

current moment. By including this, they are able to consider both forward and cash transactions in their framework.

Next, they define $\mathbf{F_i} := (F_{ij}, j \in \mathcal{F},\ j \geq i),\ \forall i \in \mathcal{F}$ as the forward curve at time T_i. Their linear program utilizes the following parameters:

$\overline{x}$	Maximum NG inventory level, where $\overline{x} \in \mathbb{R}_+$
C^I	Constant injection capacity
C^W	Constant withdrawal capacity
c^W	Marginal withdrawal costs
c^I	Marginal injection costs
δ	One review period constant risk free discount factor
α^I	Commodity adjustment factor for injection
α^W	Commodity adjustment factor for withdrawal
s_i	Spot price of NG at T_i
$S_0^{i,j}(F_{0,i}, F_{0,j})$	Spread option on futures prices $F_{i,j}$ and $F_{i,i}$ at T_0.

The spread option, which is also presented by Carmona and Durrelman [11], is defined as:

$$S_0^{i,j}(F_{0,i}, F_{0,j}) := \delta^i\ \mathbb{E}\left[\left\{\delta^{j-i}\alpha^W \tilde{F}_{i,j} - (\alpha^I \tilde{F}_{i,i} + \delta^{j-i}c^W + c^I)\right\}\middle| F_{0,i}, F_{0,j}\right]. \tag{2.1}$$

Here $\tilde{F}_{i,j}$ is a random entity that is dependent on $F_{0,i}$, $F_{0,j}$. Colloquially, the spread option is equal to the value of injecting one unit of NG at time T_i and withdrawing it at time T_j. The commodity adjustment factors are used to model in-kind fuel losses during injection and withdrawal, where $\alpha^W \in (0, 1]$ and $\alpha^I \geq 1$. The spread option additionally accounts for the discount rate from period to period and the injection and withdrawal costs of the storage facility. Their model utilizes the following decision variables:

$q_{i,j}$	Amount of NG associated with spread option i, j
x_i	Inventory level at time i
y_0	Amount of NG sold at time T_0 (spot sale).

With that, their linear program is formulated as follows:

$$\max \quad s_0 y_0 + \sum_{i \in \mathcal{F}} \sum_{j \in \mathcal{F}, i<j} S_0^{i,j}(F_{0,i}, F_{0,j}) q_{i,j}, \tag{2.2}$$

s.t.

$$x_{i+1} = x_i \sum_{j\in\mathcal{F},\, j>i} q_{i,j} - y_0 \mathbf{1}\{i=0\} - \sum_{j\in\mathcal{F},\, j<i} q_{j,i}, \quad \forall i \in \mathcal{F} \tag{2.3}$$

$$x_i \leq \overline{x}, \quad \forall i \in \mathcal{F}\{0\} \cup \{N\}, \tag{2.4}$$

$$\sum_{j\in\mathcal{F},\, j>i} q_{i,j} \leq -C^I, \quad \forall i \in \mathcal{F} \setminus \{N-1\} \tag{2.5}$$

$$y_0 \leq C^W \tag{2.6}$$

$$\sum_{i\in\mathcal{F},\, i<j} q_{i,j} \leq C^W, \quad \forall j \in \mathcal{F} \setminus \{0\} \tag{2.7}$$

$$y_0 \geq 0 \tag{2.8}$$

$$q_{i,j} \geq 0, \quad i,j \in \mathcal{F}, i<j, \tag{2.9}$$

$$x_i \geq 0, \quad \forall i \in \mathcal{F} \setminus \{0\} \cup \{N\}. \tag{2.10}$$

The objective function seeks to maximize profit through both the spread option and the spot sale option. Their formulation allows for some (or all) of the inventory to be sold at T_0. Constraints (2.3) and (2.4) balance inventory levels and limit the inventory to the maximum capacity. Lai et al. use $\mathbf{1}\{.\}$ to represent that if event $\{.\}$ is true, the function equals 1, otherwise 0. Constraints (2.5)-(2.7) are capacity constraints and the remaining constraints are non-negativity constraints for the decision variables. While it is indicated that there are no closed-form formula for the spread option coefficient in the objective function, both Lai et al. [10] and Carmona and Durrleman [11] state that Kirks method (among others) can be used to numerically estimate said parameter. It should be noted that this formulation does not include any holding cost on the inventory. Lai et al. argue that holding costs could be added easily, if necessary.

Lai et al. [10] and Gray and Khandelwal [12] both indicate the results obtained by the LP policy can be improved through re-optimization. Re-optimization takes advantage of information that becomes available over time, re-optimizing at each maturity. One method for doing this is through Monte Carlo simulation. The LP policy can also be applied to solve the exact stochastic dynamic program presnted in Section 2.2, which is normally computationally intractable. Lai et al. also indicate that their linear programming approach can provide lower bounds on the value of storage,

which can be useful when benchmarking other heuristics and modeling methods. The linear programming approaches presented offer good estimates for the value of NG storage contracts, while considering the operational characteristics and constraints of the storage facilities. Their study utilizes constant injection and withdrawal parameters that do not account for the possibility of inventory ratcheting. As displayed in Section 3.2, ratcheting agreements can significantly impact the feasible amount of NG that can be injected/withdrawn in a certain period.

Nadarajah et al. [13] present relaxations of approximate linear programs to solve the real option management of commodity storage problem. They specifically reference management of a NG asset in their study. They emphasize the modeling difficulty caused by the exogenous nature of the forward price curve for NG, as described by Eydeland and Wolyniec [14]. This modeling difficulty causes exact models to be computationally intractable. They present an approximate linear programming approach, which uses lower-dimensional representations of the variables contained within the Markov decision processes that defines managing NG storage as a real option (discussed in detail in the following section). They achieve this by discretization of the forward curve defined by a Markov decision process. Solving the approximate linear programming equations allows for upper and lower bound estimates of the optimal trading policy. The authors also provide relaxations of their approximate linear programs, which were shown to outperform their associated approximate linear programs through numerical analysis. Their approaches offer a robust alternative for modeling the forward price progression of NG, remain computationally tractable and can provide acceptable upper and lower bounds on the value of NG storage. While their study does investigate the impact of different injection and withdrawal capacities (referred to as pairs), their study does not specifically mention the impacts of inventory ratcheting on the presented framework.

Byers [15] presents a linear optimization approach combined with Monte Carlo simulation to value commodity storage. Their study also applies the approach to a NG storage example. Their approach models both the intrinsic and extrinsic value of storage. They define the intrinsic value of storage as the buy low sell high opportunities created by the NG price spread. They refer to the extrinsic value as the uncertain or risk-laden component of a storage facilities value caused by the stochastic nature

of NG prices. The value of commodity storage is presented as a portfolio of forwards and options with different strikes and maturities where the strike price in terms of futures contracts refers to the agreed upon price of the underlying commodity. They present a two-stage optimization process. The first stage is a linear optimization that determines the purchase and sale of the commodity. The linear model insures flow balance constraints at the storage facility, ensures that commodities are bought and sold correctly (in the proper sequence), among other practical constraints. The second stage of their optimization examines the results of the first stage to determine if the position of the portfolio has changed from the previous day, thus changing the extrinsic value of the portfolio. Their approach allows for the risk involved with NG storage decisions to be modeled. Finally, they indicate that path dependencies arise in commodities storage because the inventory available today impacts the set of decisions on a given day. Like other approaches, this model treats the injection and withdrawal capacities as static quantities. Byers [15] applies their approach to two different examples of NG storage. The two-tiered model approach increased tractability and allowed for the extrinsic value of NG to be modeled. Their model provides a simple approach of valuing commodity storage. They also provide insight into the typical actions of a trader. They indicate that a rational trader will always sell the maximum allowable volume and purchase the maximum allowable volume.

In practice, the computational tractability of linear models is attractive. Linear models can also be used as heuristic approaches for solving dynamic modeling approaches. While the above approaches do provide insight into the NG storage problem, they differ from the work of this thesis due to their modeling of NG price progression. As discussed later, for this project the traders utilize a discrete set of price forecasts to aid in their decision-making.

2.2 Dynamic Programming Approaches: Modeling NG Price Progression

While linear programming approaches can provide insight into potential NG storage decisions, industry practitioners often prefer high-dimensional models that model the full dynamics of NG price progression [10]. The model presented by Black [16], which is a high-dimensional forward model for the evolution of futures prices, is used as an

example of the type of price progression models preferred in practice by Eydeland and Wolyniec [14] and Lai et al [10]. In the Black model, the evolution of futures prices are defined by driftless geometric Brownian motion (with maturity-specific constant volatility $\sigma_i > 0$ and standard Brownian motion increment $dZ_i(t)$ [10,16].

Geometric Brownian motion is continuous-time stochastic process that is used to model randomly varying quantities (e.g. the price of a NG futures contract in this case). While the details of Brownian motion are not discussed in this study, Eyedland and Wolyniec [14] and Sigman [17] provide overviews of geometric Brownian motion and discuss its use in option pricing.

The Black model, as presented by Lai et al. [10], is as follows:

$$\frac{dF(t,T_i)}{F(t,T_i)} = \sigma_i dZ_i(t), \qquad \forall i \in \mathcal{F} \tag{2.11}$$

$$dZ_i(t)dZ_j(t) = \rho_{i,j}dt, \qquad \forall i,j \in \mathcal{F} \tag{2.12}$$

Here $\rho_{i,j} \in (-1,1)$ and refers to a constant correlation coefficient that relates maturity times T_i and T_j. The above is an N-factor model and is an acclaimed example of the high-dimensional forward models preferred in practice [10].

Consider again the work of Lai et al. [10], which was adapted from a periodic review model from Secomendi [18]. In this approach, inventory review periods correspond to futures price maturities. Here the previously mentioned notation and conventions for F_i and T_i is utilized (as shown in (2.2)-(2.13)). This model utilizes mostly the same parameters to the one presented on page 16, with the exception of the spread option parameter, which is no longer present.

$\overline{x}$	Maximum NG inventory level, where $\overline{x} \in \mathbb{R}^+$
C^I	Constant injection capacity
C^W	Constant withdrawal capacity
c^W	Marginal withdrawal costs
c^I	Marginal injection costs
δ	One review period constant risk free discount factor
α^I	Commodity adjustment factor for injection
α^W	Commodity adjustment factor for withdrawal
s_i	Spot price of NG at T_i

Using the above parameters, the following action sets are defined for injection,

withdrawal and any review time as

$$\mathcal{A}^I(x) := [\max\{C^I, (x - \overline{x})\}, 0], \tag{2.13}$$

$$\mathcal{A}^W(x) := [0, \min\{x, C^W\}], \tag{2.14}$$

$$\mathcal{A}(x) := \mathcal{A}^I(x) \cup \mathcal{A}^W(x). \tag{2.15}$$

The action sets ensure that only feasible amounts of NG can be injected or withdrawn from a storage facility. For example, if the amount of available inventory is less than the withdrawal capacity, then only that available amount can be extracted. Next, the reward associated with action a at time T_i is refined as $r(a, s_i)$. With that, the reward function is defined as

$$r(a,s) := \begin{cases} (\alpha^I s + c^I)a & \text{if } a \in \mathbb{R}_-, \\ 0 & \text{if } a = 0, \qquad \forall s \in \mathbb{R}_+, \\ (\alpha^W s + c^W)a & \text{if } a \in \mathbb{R}_+. \end{cases} \tag{2.16}$$

In this formulation the convention is established such that injections are negative actions and withdrawals are positive. This corresponds to the purchase of NG that occurs before injection, meaning the company takes on negative cash flows and vice versa for withdrawals. Examining the reward function in more detail it can be seen that the spot price of NG wholly determines the reward from the storage decision. Finally, they define the following exact stochastic dynamic program (for reference $\mathcal{L}$ refers to the feasible set of inventory levels defined by [0,$\overline{x}$]):

$$V_N(x_N, \mathbb{F}_N) := 0, \qquad \forall x_N \in \mathcal{L}, \tag{2.17}$$

$$V_i(x_i, \mathbb{F}_N) := \max_{a \in \mathcal{A}(x_i)} r(a, s_i) + \delta \mathbb{E}\left[V_{i+1}(x_i - a_i, \tilde{\mathbf{F}}_{i+1}) | \mathbf{F}'_i\right] \tag{2.18}$$

It is revealed that due to the high-dimensional state space of the model when applied in practice, the above formulation is computationally intractable. Lai et al. [10] state that stochastic dynamic programming is the natural approach to solve the NG storage valuation problem, but the high-dimensional price evolution models preferred by traders makes application of the exact model impossible.

Additionally, Lai et al. [10] state that the linear program defined by (2.2)-(2.13) can be used to create a feasible policy for the exact dynamic programming model. They detail how this might be done, but it is outside the scope of this thesis. Two different approaches are presented to confront the intractability of the exact dynamic program. The first approach is based entirely on the information available at T_0, which makes the approach based on the intrinsic value of storage. This model assumes that a transaction is made in the forward market at T_0, which differs from the previous approach. The intrinsic model is described as follows:

$$U_N^I(x_N; \mathbf{F}_0) := 0, \qquad \forall x_N \in \mathcal{L}, \tag{2.19}$$

$$U_i^I(x_i, F_0) := \max_{a \in \mathcal{A}(x_i)} r(a, \mathbf{F}_{0,i}) + \delta U_{i+1}^I(x_i - a, \mathbf{F}_0), \qquad \forall i \in \mathcal{F}, x_i \in \mathcal{L} \tag{2.20}$$

The second approach for improving the tractability of the dynamic programming model provided by Lai et al. [10] is by utilizing approximate dynamic programming (ADP) on a reformulated version of the exact model described by (2.13)-(2.18). The value of the approximate dynamic program is computed using Monte Carlo simulation. It is revealed that the approximate approach produced tight upper and lower bounds on the value of storage, when applied in their study. Similar to the linear programming approach (2.1)-(2.10), Lai et al. state that the approximate dynamic programming approach can take advantage of re-optimization.

Parsons [19] presents a two-factor tree model for valuation of NG storage assets. Unlike many approaches, their approach includes inventory ratcheting in their model framework. Their results indicate that including inventory ratcheting produces "pockets of high-optionality". They state that the behavior is caused by different advantages arises when on either side of a ratchet. Their model framework includes a mean reverting process to represent the price of NG. Their work confirms the importance of including ratcheting agreements in model framework for valuation of NG storage assets.

Chen and Forsyth [20] present a semi-Lagrangian approach that models the NG storage as a stochastic control problem. They identify the NG storage problem as a stochastic control problem that results in a Hamilton-Jacobi-Bellman equation. They present a semi-Lagrangian approach for solving the aforementioned equation. Their

approach allows for the model to consider different aspects of spot-price models for NG including: mean reversion, seasonality and price jumps. Esteve et al. [21] also utilize a stochastic control approach for pricing swing options which can be useful for modeling the option of NG storage.

Bjerksund et al. [22] provide an approach that seeks to maximize the intrinsic value of NG storage. The state that a common approach in literature for valuation of NG storage is to utilize simple models for the price process in tandem with complex optimization techniques. In their study they take an alternative approach by utilizing a complex price model with a simplified decision rule (maximization of the intrinsic value). The optimal storage decisions are found by solving a dynamic program representing the intrinsic value of the asset. Their results propose that their combination of a complex pricing model with a simplified decision rule is the appropriate approach for valuing NG storage.

Yi [23] presents three methodologies for valuation of NG storage assets, including an approach that utilizes Monte Carlo simulation in tandem with stochastic dual programming. This approach only considers the spot price of NG in storage valuation. Their approach also incorporates bid and ask prices into the optimization framework. The "bid" is the offer the NG buyer makes on the gas, whereas the "ask" is the price set by the seller. Their results indicate that, when considering only the spot price of NG, a higher-than-market value is placed on storage. They also state the importance of accurate pricing models on storage valuation.

Lai et al. [24] present methods for the valuation of NG storage at a liquefied natural gas (LNG) terminal. They state that the exact valuation of the option of storing LNG is computationally intractable. In their study, they develop a tractable heuristic model for valuation of NG storage as LNG. Model framework includes the shipping of LNG, evolution of NG prices and inventory control. Their model specifically utilizes Markov Decision processes in tandem with Monte Carlo simulation among other approaches outside the scope of this thesis. Their results indicate that their approach provides an accurate valuation of NG storage and is tractable enough for industry use. While not referring specifically to the NG storage problem, this study involve similar operational characteristics underground storage facilities such as fuel loss.

Clearly, modeling the price progression of NG complicates accurately valuing storage assets. Carmona and Durrleman [10] specifically discuss the modeling difficulty of price progression in energy markets. They state that, while a simple relationship exists between spot price and forward prices for futures and forward contracts in regular markets, this relationship does not exist in commodities markets. For this reason, modeling approaches for the progression of price is more difficult for commodities, inherently making the NG storage problem more difficult. Modeling difficulty is specifically attributed to the seasonality of commodities markets and their mean-reverting nature.

2.3 Robust and Stochastic Optimization: Frameworks for Optimization Under Uncertainty

Robust and Stochastic optimization are utilized to account for uncertainty in optimization problems. As previously stated, NG prices are subject to a high degree of uncertainty throughout time. To investigate their applicability in the context of the NG storage problem, a brief review of related approaches was conducted.

Barbry et al. [25] develop a robust optimization framework to model the risk-averse nature of electricity storage decisions. Methods for characterizing price uncertainty in the day-ahead New York electricity market are presented. Factors such as total demand, market prices and wind power contributions were considered when characterizing price uncertainty. Next, they present a decision model that uses a "budget" to model the traders' loss-aversion when making a storage decision. Results from their study indicate that, by implementing a robust bidding strategy, the risk of financial loss on energy storage decisions can be reduced when compared to a nominal strategy. Additionally, this was obtained with acceptable decreases in expected profit when compared to the nominal strategy. This approach displays the impact of risk-averse optimization frameworks on commodities purchase and storage decisions.

Another relevant RO approach from literature is the work of Lorca et al. [26] In their study, they present a multi-stage RO model to account for uncertainty in solar and wind power in the context of the unit commitment problem. In their study, they construct dynamic uncertainty sets to account for the stochasticity that renewable

resources can posses. While not referring specifically to NG, their study does include energy storage in the model framework. Results from their study revealed that their methods allow multi-stage robust unit commitment problems to be efficiently solved, while considering uncertainty created by solar and wind power generation. They indicate that their model approach offers advantages in cost and reliability over traditional uncertainty set and deterministic approaches.

Mokrian and Stephen [27] present a stochastic framework for optimizing energy storage, specifically electricity. Electricity, like NG, is capable of being stored and is subject to varying supply and demand. Their paper summarizes different storage technologies and presents model framework that includes linear programming, multi-stage stochastic and dynamic programming approaches. They indicate that, while the dynamic programming approach outperformed the others, it suffers from tractability issues due to the state space of the problem. They state that the stochastic approach is flexible in choice of differing pricing models and is capable of dealing with many uncertainties. This problem is similar to NG storage as their electricity storage is subject to operational constraints that govern the potential storage actions.

Next, consider the work of Bajram and Can [28], who present a stochastic programming approach for multi-period portfolio optimization. They construct a piecewise linear utility function to model risk and possible recourse of financial decisions, which allows stochastic programming to be applied. Their work involves generating scenario trees to model the underlying uncertainty around most portfolio transactions. Each node in the decision tree corresponds to a set of asset returns during a specific time-period. They indicate that their model, while a simplification in several ways, is an effective tool for multi-stage asset liability management decisions. This allows the investor the chance to evaluate different trading options, and possible recourse. While not referring specifically to NG storage or trading, their work still provides an outline of how stochastic optimization can be used in financial and risk applications.

Aouam et al. [29] present stochastic and quadratic programming approaches for natural gas procurement. Their work considers local distribution companies who can choose from different contract types to purchase NG to satisfy customer demand. They present two strategies to model NG procurement, which they refer to as naive and dynamic strategies. Stochastic programming is utilized in the proposed dynamic

strategies. They present several stochastic models in their framework including a risk-neutral and variance-based model. Naive strategies utilize simple procurement rules. Using simulation, it is shown that dynamic strategies, wherein the probability distribution for future NG prices is estimated, possess more risk when considering modeling error. Additionally, they show that in the presence of accurate price modeling, the dynamic strategies are more effective. Convex combinations for dynamic and naive strategies are then used to create robust bidding strategies formulated as a quadratic program. Their results indicate that the robust framework benefits from the cost effectiveness of dynamic strategies and from the low risk associated with naive strategies. While not referring specifically to optimizing a NG storage asset, this work does provide valuable insights into modeling NG transactions.

Kannan et al. [30] present a stochastic optimization model for planning natural gas purchases, storage, transportation and deliverability. Unlike other approaches from literature, this approach models the deliverability as a dynamic quantity, as it can be impacted by changes in supply and demand. Their framework considers that decisions can be made on yearly, monthly, daily and inter-day bases. They utilize discrete scenarios to represent changes in supply and demand patterns during different seasons, which is similar to the methods used by Knowles [31]. The goal of their model is to minimize all associated costs across all supply and demand scenarios. Results indicate that effective gas procurement strategies are obtained through use of their model. They also indicate that, because deliverability is sensitive to change, they suggest the model be used to negotiate deliverability in contracts or to justify use of a storage facility. This work provides insight into modeling the various operational characteristics and transactions involved with NG sales and purchases.

2.4 Conclusion

While the presented approaches do provide insights into modeling the NG storage problem, there are some differences between the approaches from literature and our thesis problem. Some presented approaches deal with optimizing energy storage or financial instruments unrelated to NG. While seemingly unrelated, these approaches provide insights into optimization techniques for commodities subject to price uncertainty. Many of the NG storage-specific approaches presented model the injection

and withdrawal capacities as constant and do not consider dynamic inventory ratcheting. As will be revealed in Section 4, the injection and withdrawal capacities can be subject to change based on different inventory levels in the facility.

Additionally, most of the approaches presented in Sections 2.1 and 2.2 include modeling the evolution of NG prices in their framework. This is the complicating aspect of many of the presented approaches, as the high-dimensionality of modeling NG price progression often limits the tractability of the framework. This thesis does not seek to model the price progression of NG in the optimization framework, but rather utilizes all information available to the trader to make the most informed decision. The models presented in Section 2.3 did not refer specifically to the NG storage problem, but do provide valuable insights into modeling uncertainty and risk in the context of the NG storage problem.

Chapter 3

Model Framework for Futures-to-Futures Transactions

This chapter of the thesis presents the various modeling approaches created to optimize NG storage asset (profit maximization), while considering only futures transactions. This means that only the option to buy and sell futures contracts is considered. In this section, it is assumed that a trader has access to a NG storage well and must decide on the appropriate quantities to inject/withdraw such that profit is maximized. The storage facility is also subject to pre-defined operating characteristics, which limit the feasible injection and withdrawal quantities. We also assume that the trader has access to a discrete set of price simulations which contain simulated futures price paths for a finite time-horizon. Sample price simulations for a 24-month period were provided by The Utility Company and used to test model approaches. This is the same information used by traders at The Utility Company in their decision-making process. The models presented in this section were developed to be used as a decision support tool for use at The Utility Company. It is reasonable to assume that The Utility Company is a relatively small player in the NG market, therefore their actions do not impact the underlying price of NG.

The remainder of this chapter is organized as follows: first sample price simulation data provided by The Utility Company is presented and analyzed. Next, a base model for futures-to-futures transactions with constant deliverability is presented. To account for dynamic deliverability, a model is presented that contains novel constraints to account for the change in deliverability as the well inventory levels change. As the price data possess uncertainty and variation, several model variants are presented to help mitigate risk. Chance-constrained, Robust and Distributionally Robust model variants are presented and their results are discussed. Because futures contracts are financial instruments, recourse can be taken on past decisions if more profitable opportunities arise. Because of this, a model that considers past decisions is developed

and presented. The impact of the various model variants on this model are also presented and discussed. Solver and computer specifications can be found in Appendix A. Python codes for model approaches are found in Appendices B-E.

3.1 Price Simulations

Before the modeling process began, it was important to understand the price simulations utilized by traders at The Utility Company. The price simulations are developed internally using proprietary methods. The price simulations are provided to the traders and are used to assess the potential value a storage asset presents. In the case of futures transactions, the price simulations model different instances of the futures curve at the current moment in time as presented in Figure 1.4. As previously stated, the futures curve presents the current market price of a commodity on a futures exchange.

The sample price simulations provided by The Utility Company spanned a 24-month period and presented predictions for monthly futures prices from January 2019 to December 2020. 962 distinct price predictions were provided for each month in the 24-month horizon. To begin the analysis, the mean price and standard deviation of predicted prices were calculated for each month. The average monthly price follows the general trend outlined in Section 1.3 with the colder months exhibiting higher prices. The period January 2019 through March 2019, specifically sees very high prices when compared to the other months in the time-horizon. The price simulations reflect the seasonal tendencies of NG prices discussed in the literature.

Figure 3.2 also reveals that the standard deviation of the price predictions generally decreases over time. January 2019 through March 2019, specifically, possess much higher standard deviations than any other month in the data set. This increased volatility is caused by the increased expected liquidity seen in those months. This trend is reflected (although to a smaller magnitude) in the standard deviation of the other winter months in the time horizon. In general, the data shows more volatility in months with higher expected prices. To further explore the price data the distribution of price on a monthly basis can be examined. Specifically, the price distribution for January 2019, July 2019 and September 2020 is examined such that the analysis is not biased to one month's particular price distribution.

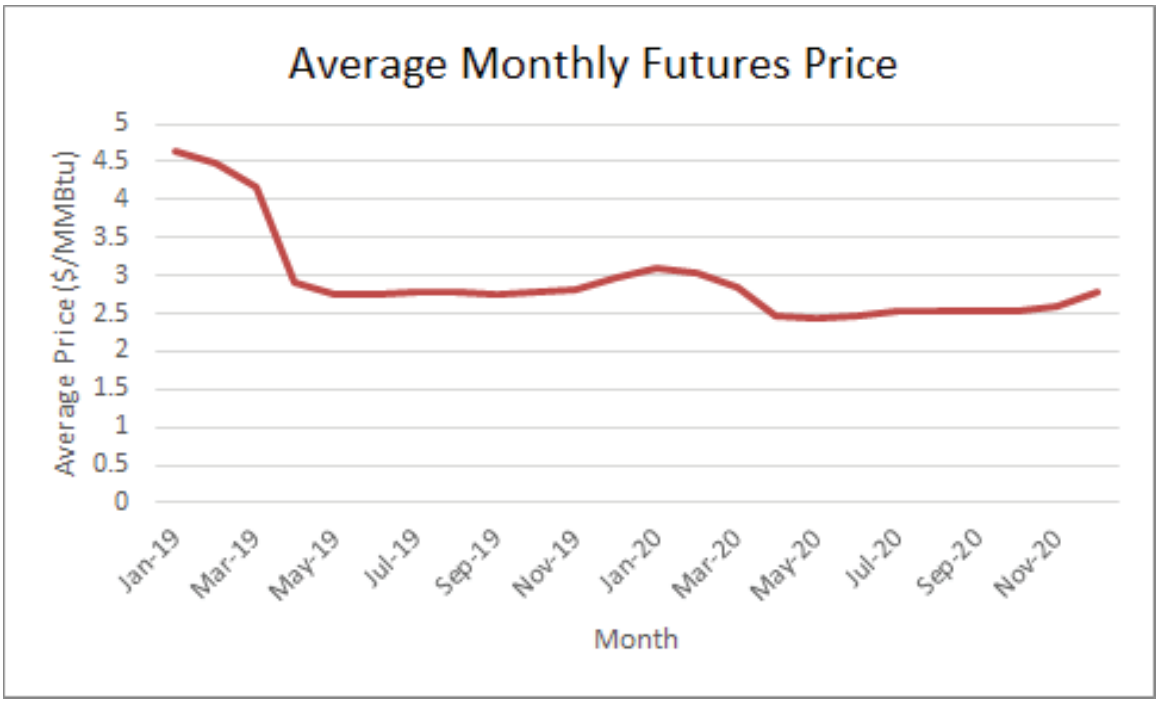

Figure 3.1: Average monthly futures price of NG, according to price simulations.

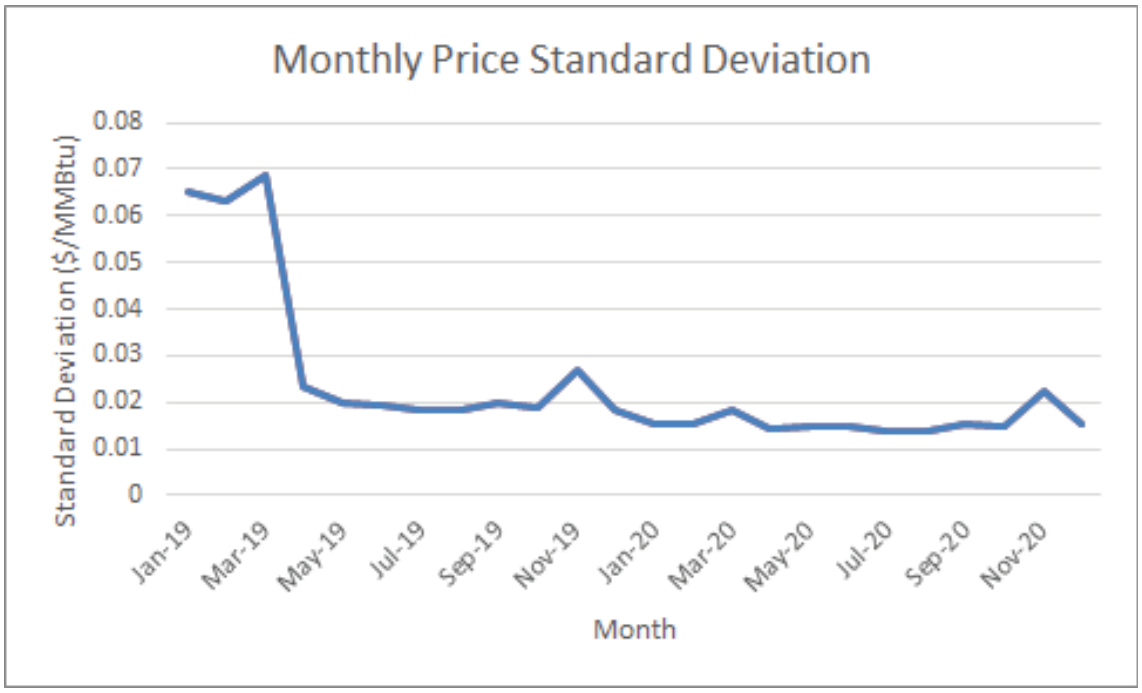

Figure 3.2: Standard deviation of monthly futures prices, from the price simulations.

Figures 3.3 through 3.5 show that the monthly price is highly mean centered according to the price simulations. January, specifically possess a very high central

tendency with a large spread. July's prices see variation of a lower magnitude with a lower central tendency and September's price distribution seems to fall between that of January and July.

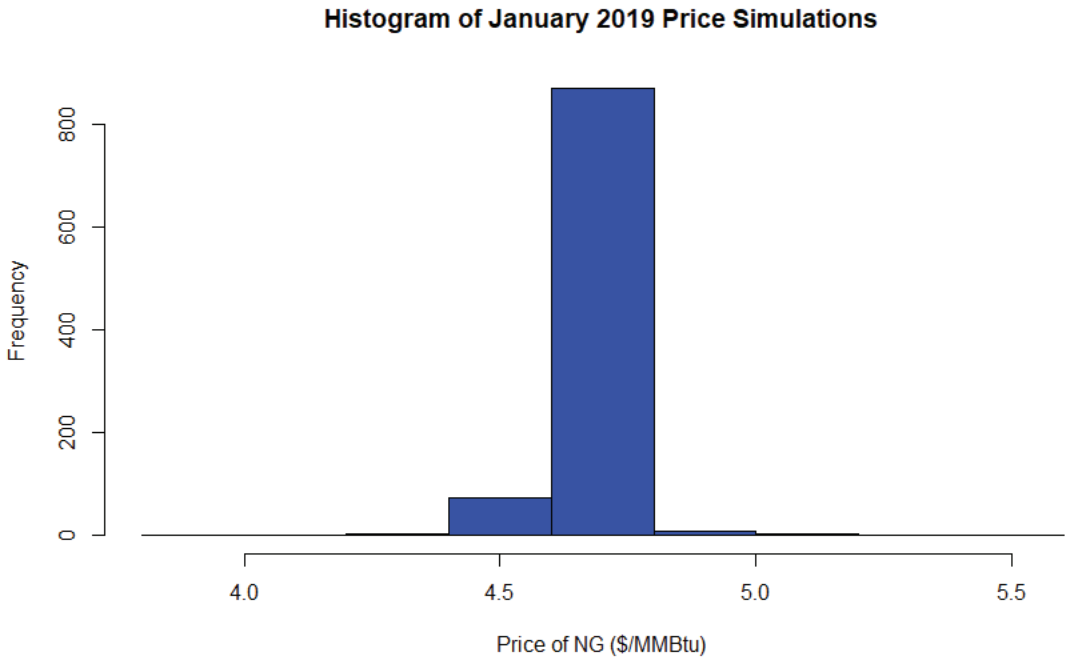

Figure 3.3: Distribution of prices for January 2019, according to price simulations

To investigate if any distributional assumptions could be made, the price distributions of each month in the time horizon were tested for significant fits amongst known probability distributions. Because the prices in each month are so highly mean-centered, no known probability distribution produces a significant fit. The histograms do indicate that most price simulations produce prices close to the overall mean in each month. Additionally, it is a reasonable assumption that monthly prices follow general symmetric distributions.

Overall, the price simulations appear to be highly mean-centered, which could be a reflection of the prediction methods utilized at The Utility Company. The data reflects the typical seasonal trends of natural gas prices. The price simulations, while mean centered, do posses variation. This variation must be accounted for in the decision making process, as without its consideration, the trader could be vulnerable to unfavorable price realizations that result in financial losses.

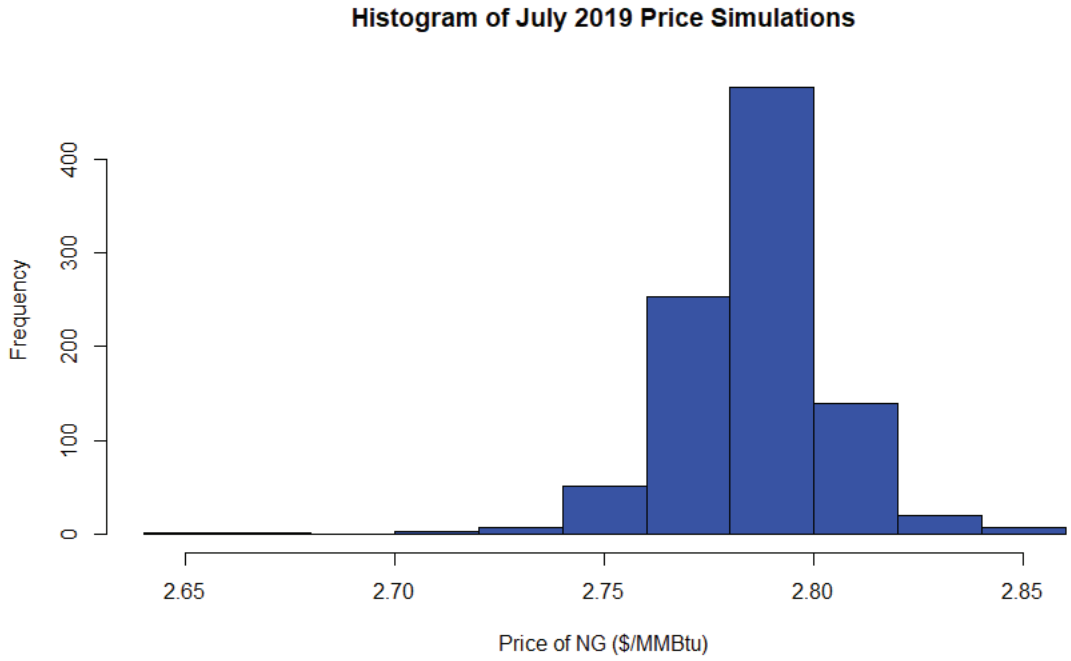

Figure 3.4: Distribution of prices for July 2019, according to price simulations

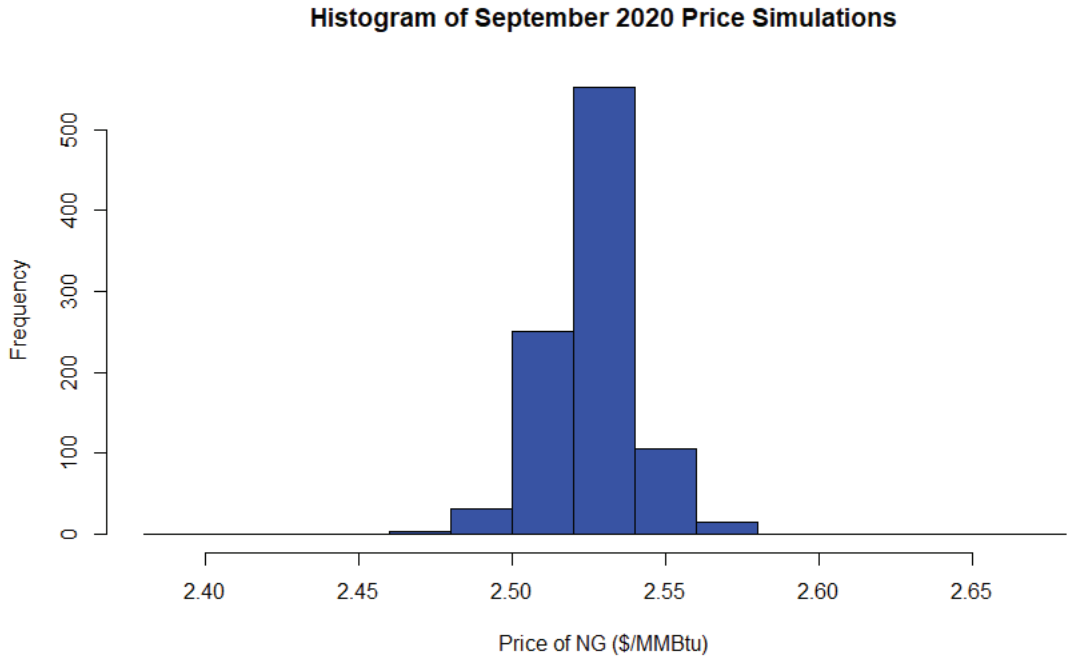

Figure 3.5: Distribution of prices for September 2020, according to price simulations

3.2 Equally-Likely Mixed Integer Approach: Constant Deliverability

As no probability distribution can be fit to the data and the data is symmetric and highly mean-centered, each of the price simulations can be safely considered equally-likely. This is due to the highly mean-centered nature of the data. This means that, for each month, there is a set $s \in S$ of 962 price predictions. Consider also a set of time periods, $t \in T$, that represents the planning horizon. Here the time periods represent the discrete decision-making intervals, in our case months. We also assume that all storage actions (injection, withdrawal or do-nothing) occur between time periods (ex. if an injection decision is made at time $t-1$, the specified amount of gas is injected and available for sale/withdrawal by time t).

To account for the various costs and operational constraints of NG storage facilities, the following parameters must be included:

p_{st}	futures price of NG at time t according to prediction s
Cap	maximum capacity available for storage
f	maximum injection quantity during a single time period t
g	maximum withdrawal quantity during a single time period t

Because the transactions refer to financial instruments, which have the option of being sold/bought before maturity (physical delivery of the gas into the storage asset), fuel loss is not included in the present framework. This assumption was validated with the trading team at The Utility Company, as they indicated that fuel loss is not a significant factor in valuation of a storage asset, when considering futures-to-futures transactions. Additionally, they indicated that variable costs for injection and withdrawal can be similarly ignored.

The following decision variables are used to formulate the mixed-integer program:

v_t	quantity of NG purchased to be injected during time period t
w_t	quantity of NG withdrawn to be sold during time period t
I_t	inventory of NG at the beginning of time period t

$$z_t = \begin{cases} 0, & \text{if no withdrawal occurs during time period } t, \\ 1, & \text{if no injection occurs during time period } t. \end{cases}$$

There is no need to specify a case for z_t for the do-nothing option, as this option

is achieved by setting either the injection of withdrawal variable to zero. With that, the mixed-integer program's objective function is defined as follows:

$$\max \quad \frac{1}{|S|} \sum_{s \in S} \sum_{t \in T} p_{st} \cdot [-v_t + w_t] \tag{3.1}$$

Clearly, the above objective function is equivalent to simply using the expected value of the price in month t, according tto the price predictions in S. Thus the deterministic objective function is written as:

$$\max \quad \sum_{t \in T} \overline{p}_t \cdot [-v_t + w_t] \tag{3.2}$$

where $\overline{p}_t$ is simply the average price in time t, according to the price projections. The objective function seeks to maximize the expected spread from injection and withdrawal (purchases and sales) of NG, according to the price simulations. It should be noted that there is no need to include the time value of money in this approach. As these transactions are occurring on the futures market, the transactions all happen at the current moment and therefore represent the present value of the transaction. Next, the following constraints are introduced to enforce the physical constraints on the storage asset:

$$v_t \leq f \cdot (1 - z_t), \qquad \forall t \in T \tag{3.3}$$

$$w_t \leq g \cdot z_t, \qquad \forall t \in T \tag{3.4}$$

$$I_t = I_{t-1} + v_{t-1} - w_{t-1}, \qquad \forall t \in T \tag{3.5}$$

$$I_t \leq Cap, \qquad \forall t \in T \tag{3.6}$$

$$I_0 = I_{|T|}, \tag{3.7}$$

$$v_t, w_t, I_t \geq 0, \quad z_t, \in \{0, 1\}, \qquad \forall t \in T \tag{3.8}$$

Constraints (3.3) and (3.4) ensure that the physical deliverability constraints of the storage facility are obeyed, i.r. not withdrawing more gas in a time period than possible. Additionally, these constraints reinforce the assumption that an injection and withdrawal do not occur during the same time interval. Constraint (3.5) ensures flow balance and constraint (3.6) ensures the available capacity of the storage field

is obeyed. Lastly, aside from non-negativity constraints, (3.7) ensures the well is returned to its initial inventory level at the end of the contract. This model is similar to the linear programming approach presented by Lai et. al [10], but does not include the evolution of NG prices in the optimization process. Rather our approach uses the provided price simulations to evaluate the value of the storage asset. With that, the complete formulation is as follows:

$$\max \quad \sum_{t \in T} \overline{p}_t \cdot [\, -v_t + w_t \,] \tag{3.2}$$

s.t.

$$v_t \leq f \cdot (1 - z_t), \qquad \forall t \in T \tag{3.3}$$

$$w_t \leq g \cdot z_t, \qquad \forall t \in T \tag{3.4}$$

$$I_t = I_{t-1} + v_{t-1} - w_{t-1}, \qquad \forall t \in T \tag{3.5}$$

$$I_t \leq Cap, \qquad \forall t \in T \tag{3.6}$$

$$I_0 = I_{|T|}, \tag{3.7}$$

$$v_t, w_t, I_t \geq 0, \quad z_t, \in \{0, 1\}, \qquad \forall t \in T \tag{3.8}$$

Note the similarities between this model and the one presented by [10]. In addition to the price simulations, sample operating characteristics in Table 3.1 are used to test run the model.

Table 3.1: Operating characteristics provided by The Utility Company.

f	10,000 MMBtu/day
g	15,000 MMBtu/day
I_0	0
Cap	1,000,000 MMBtu

In the initial approach, maximum injection and withdrawal rates are assumed constant. As will be discussed in later sections, this assumption can lead to over-valuation of a storage asset when the storage agreement is subject to inventory ratcheting. Additionally, it was assumed that the client has an initial operational inventory of 0. As the clients fiscal year begins on January 1, it was safe to assume an empty inventory. It should be noted the initial inventory level can significantly change tactical plans.

Using the provided data, the following results and inventory profile were obtained (note the system and solver characteristics are described within Appendix A):

Table 3.2: EL model results-constant deliverability.

Expected Profit	$ 518,518.06
Average Inventory Level	468854.17 MMBtu
Number of Storage Actions	17
Computation Time	2.363 Seconds
Model Size	74 Continuous Variables, 24 Integer Variables & 145 Constraints

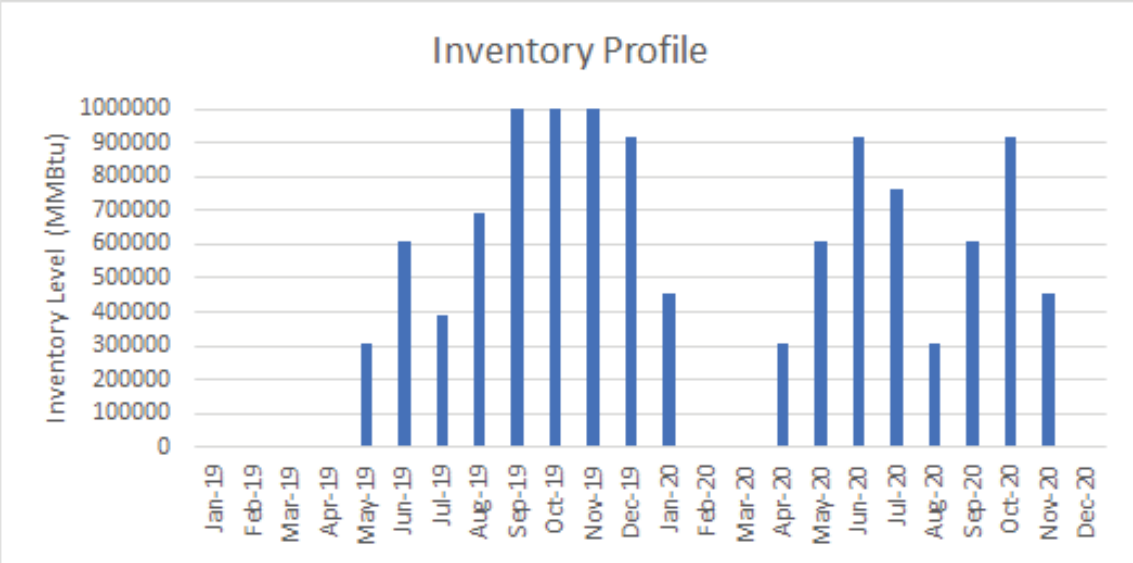

Figure 3.6: Inventory profile for futures-to-futures model, excluding ratcheting.

The results in Table 3.2 show no tractability issues and the model made various injection/storage decisions to optimize the expected profit. The model performs no actions in several months as depicted in Figure 3.6, which is an encouraging result, as the "do nothing" option can sometimes be the one that results in the highest by the behavior exhibited in Figure 3.1. Figure 3.1 shows that the model filled the storage asset in months with lower-than-expected prices such that gas could be withdrawn and sold in months with high expected prices.

While this simple approach seems effective, it does not account for instances where the deliverability is impacted by changes in inventory level. Because of this, this approach places a high upper bound on the expected value of the storage asset.

3.3 Equally-Likely Mixed Integer Approach With Inventory Ratcheting

In some cases, the storage asset may be subject to a ratcheting agreement, which is a set of parameters, outlined by the company operating the NG well, indicating the impact of different inventory levels on deliverability. The ratcheting constraints provided by The Utility Company are shown in Table 3.3.

Table 3.3: Sample ratcheting agreement, provided by The Utility Company.

Inv. Lower Bound	**Inv. Upper Bound**	**Max Daily Inj. Qty.**	**Max Daily With. Qty**
0	150,000	10,000	4,000
150,000	300,000	8,000	8,000
300,000	1,000,000	6,000	15,000

The ratcheting agreement indicates that, when the inventory level falls between certain upper and lower bound, it results in corresponding injection and withdrawal capacities. This sample ratcheting agreement shows that as the well inventory increases it gets harder to inject, but easier to withdraw gas. Conversely, as the inventory levels drop it gets easier to inject, but harder to withdraw. The ratchets represent not only pressure physics but are also contractual obligations for firm operations. The storage well is typically shared between several players through asset management agreements (AMAs) where each firm has rights to a certain volume. The ratchets used in this thesis are representative and ensure that one party cannot significantly impact another firm's ability to inject or withdraw. Subject to this ratcheting agreement, assuming constant injection and withdrawal capacities is incorrect. To illustrate this, consider the injection the previous model made in May. Assuming constant injection and withdrawal capacities, one can inject 305,000 MMBTu of gas in a given month (assuming 30.5 days in a month), and this amount is injected by the model. Using the above table the May injection limit is calculated to be 274,000 MMBtu. Because of this inventory ratcheting must be included in the model framework.

To implement ratcheting, consider a set $\ell \in L$ of pieces (or brackets) of the ratcheting function. For the table above: $\ell \in \{0, 1, 2\}$, for example. Additionally, ℓ_0 refers to the first ratcheting bracket and $\ell_{|L|}$ refers to the last ratcheting bracket. Next, the following parameters are defined with respect to the ratcheting characteristics of the facility:

UB_ℓ Upper bound on the inventory for bracket, ℓ, of the ratcheting function
LB_ℓ Lower bound on the inventory for bracket, ℓ, of the ratcheting function
f_ℓ Monthly injection capacity, when in ratcheting bracket ℓ
g_ℓ Monthly withdrawal capacity, when in ratcheting bracket ℓ

Here the parameters used to represent constant delverability in the previous approach are indexed by ℓ to reflect inventory ratcheting. For the purposes of this study, the monthly injection and withdrawal capacities for each ratcheting level were computed assuming 30.5 days in each month. Monthly specific injection/withdrawal capacities can easily be included by indexing f and g by t, but when included in the framework it was shown to have no impact on the optimal solution.

Next, the following variables must be introduced:

$$r_{t\ell} = \begin{cases} 1, & \text{if } LB_\ell \leq I_t \leq UB_\ell, \\ 0, & \text{otherwise.} \end{cases}$$

$\lambda_{t\ell k}$ Maximum proportion of bracket k used for injection in time t when starting in bracket ℓ

$\mu_{t\ell k}$ Maximum proportion of bracket k used for withdrawal in time t when starting in bracket ℓ

$$x_{t\ell k} = \begin{cases} 1, & \text{if bracket } k \text{ is selected for withdrawal when starting in} \\ & \text{bracket } \ell \text{ at time period } t, \\ 0, & \text{otherwise.} \end{cases}$$

$$c_{t\ell k} = \begin{cases} 1, & \text{if bracket } k \text{ is selected for injection when starting in} \\ & \text{bracket } \ell \text{ at time period } t, \\ 0, & \text{otherwise.} \end{cases}$$

Next, a series of constraints are introduced to enforce ratcheting. These constraints replace (3.3) and (3.4) in the initial formulation.‘ First, the ratchet for the start of the month must be identified:

$$I_t \leq Cap + (UB_l - Cap) \cdot r_{tl}, \qquad \forall t \in T, \ell \in L \tag{3.9}$$

$$I_t \geq LB_\ell r_{t\ell}, \qquad \forall t \in T, \ell \in L \tag{3.10}$$

These constraints will set a binary variable, $r_{t\ell}$ to 1 *iif* the inventory levels are within a certain bracket's inventory bounds. The next constraint simply ensures we

can only begin the month in one ratcheting bracket.

$$\sum_{\ell \in L} r_{t\ell} = 1, \quad \forall t \in T \tag{3.11}$$

The following constraints ensure the amount injected or withdrawn cannot exceed the limit, as defined by the ratcheting function. This is done by making the injection and withdrawal amounts the sum-product of the proportion of time spent in each bracket and the corresponding injection or withdrawal rate.

$$v_t \leq \sum_{\ell \in L} \sum_{k \geq \ell} \lambda_{t\ell k} f_k, \quad \forall t \in T \tag{3.12}$$

$$w_t \leq \sum_{\ell \in L} \sum_{k \leq \ell} \mu_{t\ell k} g_k, \quad \forall t \in T \tag{3.13}$$

Because, when injecting, the inventory level will never decrease, brackets below the initial level do not need to be considered. Additionally, when withdrawing, it is impossible to see an increase in inventory, therefore it is not necessary to consider ratcheting levels above the one the month started in. This logic is reflected in the above constraints.

The last constraints ensure that, based on where the inventory started in a month, the ratcheting agreement is obeyed. The binary variables c_{tlk} and x_{tlk} are used to force the model to pick the right bracket first, in cases where the bracket choice is inconsequential. These constraints must be built for each possible case, that is, constraints must be built for every possible combination of feasible ratchets in a given month.

$$\lambda_{t\ell k} \leq \frac{UB_\ell - I_t}{f_\ell} + M(1 - r_{t\ell}), \quad \forall t \in T, \ell \in L, k = \ell, \ell \neq \ell_{|L|} \tag{3.14}$$

$$\lambda_{t\ell k} \leq \frac{UB_k - UB_{k-1}}{f_k} + M(1 - r_{t\ell}), \quad \forall t \in T, \ell, k \in L, k > \ell \tag{3.15}$$

$$\sum_{k \geq \ell} \lambda_{t\ell k} = r_{t\ell}, \quad \forall t \in T, \ell \in L \tag{3.16}$$

$$\mu_{t\ell k} \leq \frac{I_t - LB_\ell}{g_\ell} + M(1 - r_{t\ell}), \quad \forall t \in T, \ell \in L, k = \ell, \ell \neq \ell_0 \tag{3.17}$$

$$\mu_{t\ell k} \leq \frac{LB_{k+1} - LB_k}{g_k} + M(1 - r_{t\ell}), \quad \forall t \in T, \ell, k \in L, k < \ell \tag{3.18}$$

$$\sum_{k \leq \ell} \mu_{t\ell k} = r_{t\ell}, \qquad \forall t \in T, \ell \in L \tag{3.19}$$

$$\lambda_{t\ell k} \leq c_{t\ell k}, \qquad \forall t \in T, \ell, k \in L, k \geq \ell \tag{3.20}$$

$$c_{t\ell k} \geq c_{t\ell k'}, \qquad \forall t \in T, \ell, k, k' \in L, k \geq \ell, k' > k \tag{3.21}$$

$$\mu_{t\ell k} \leq x_{t\ell k}, \qquad \forall t \in T, \ell, k \in L, k \leq \ell \tag{3.22}$$

$$x_{t\ell k} \geq x_{t\ell k'}, \qquad \forall t \in T, \ell, k, k' \in L, k \leq \ell, k' < k \tag{3.23}$$

$$z_t, r_t, x_{tlk}, c_{tlk} \in \{0, 1\}, \qquad \forall t \in T \tag{3.24}$$

Again, when injecting, the deliverability can only decrease during a month, and vice versa when withdrawing, which is the logic used for the k subscript. (3.14) calculates the maximum proportion of the month that gas can be injected at using the rate of the initial ratcheting bracket (the ratchet the well started the month in). Colloquially, how much time can we inject at the initial ratcheting rate for a given month. (3.15) calculates the maximum proportion of the month gas can be injected at, across all remaining ratcheting levels. Again, here we do not need to consider ratcheting levels below initial, as when injecting, the inventory level will always increase. (3.16) ensures that the proportions of the month spent injecting at each ratcheting level are equal to 1, if the well began in bracket ℓ. (3.17)-(3.19) are similar to (3.14)-(3.16), but refer to the withdrawal of gas. Finally, aside from non-negativity constraints, (3.20)-(3.23) ensure the ratchets are utilized in the proper order (e.g. sequentially).

Certain levels of the ratcheting function require different constraints than others. (3.14) is an example of such a case. If the inventory level at the start of the month is in the last bracket, injections will be made at this rate for the whole month (if choosing to inject). Similarly, (3.17) enforces that, if a withdrawal is made (when starting in the lowest bracket) then you will never withdraw at another rate. This model is general and can be applied to any number of ratchets. With that, the complete formulation is as follows:

$$\max \quad \sum_{t \in T} \overline{p}_t \cdot [\, -v_t + w_t \,] \tag{3.2}$$

s.t.

$$I_t \leq Cap + (UB_l + Cap) \cdot r_{tl}, \quad \forall t \in T, \ell \in L \tag{3.9}$$

$$I_t \geq LB_\ell r_{t\ell}, \quad \forall t \in T, \ell \in L \tag{3.10}$$

$$\sum_{\ell \in L} r_{t\ell} = 1, \quad \forall t \in T \tag{3.11}$$

$$v_t \leq \sum_{\ell \in L} \sum_{k \geq \ell} \lambda_{t\ell k} f_k, \quad \forall t \in T \tag{3.12}$$

$$w_t \leq \sum_{\ell \in L} \sum_{k \leq \ell} \mu_{t\ell k} g_k, \quad \forall t \in T \tag{3.13}$$

$$\lambda_{t\ell k} \leq \frac{UB_\ell - I_t}{f_\ell} + M(1 - r_{t\ell}), \quad \forall t \in T, \ell \in L, k = \ell, \ell \neq \ell_{|L|} \tag{3.14}$$

$$\lambda_{t\ell k} \leq \frac{UB_k - UB_{k-1}}{f_k} + M(1 - r_{t\ell}), \quad \forall t \in T, \ell, k \in L, k > \ell \tag{3.15}$$

$$\sum_{k \geq \ell} \lambda_{t\ell k} = r_{t\ell}, \quad \forall t \in T, \ell \in L \tag{3.16}$$

$$\mu_{t\ell k} \leq \frac{I_t - LB_\ell}{g_\ell} + M(1 - r_{t\ell}), \quad \forall t \in T, \ell \in L, k = \ell, \ell \neq \ell_0 \tag{3.17}$$

$$\mu_{t\ell k} \leq \frac{LB_{k+1} - LB_k}{g_k} + M(1 - r_{t\ell}), \quad \forall t \in T, \ell, k \in L, k < \ell \tag{3.18}$$

$$\sum_{k \leq \ell} \mu_{t\ell k} = r_{t\ell}, \quad \forall t \in T, \ell \in L \tag{3.19}$$

$$\lambda_{t\ell k} \leq c_{t\ell k}, \quad \forall t \in T, \ell, k \in L, k \geq \ell \tag{3.20}$$

$$c_{t\ell k} \geq c_{t\ell k'}, \quad \forall t \in T, \ell, k, k' \in L, k \geq \ell, k' > k \tag{3.21}$$

$$\mu_{t\ell k} \leq x_{t\ell k}, \quad \forall t \in T, \ell, k \in L, k \leq \ell \tag{3.22}$$

$$x_{t\ell k} \geq x_{t\ell k'}, \quad \forall t \in T, \ell, k, k' \in L, k \leq \ell, k' < k \tag{3.23}$$

$$I_t = I_{t-1} + v_{t-1} - w_{t-1}, \quad \forall t \in T \tag{3.5}$$

$$I_t \leq Cap, \quad \forall t \in T \tag{3.6}$$

$$I_0 = I_{|T|} \tag{3.7}$$

$$z_t, r_t, x_{tlk}, c_{tlk} \in \{0, 1\}, \quad \forall t \in T \tag{3.24}$$

$$v_t, w_t, I_t, \lambda_{tlk}, \mu_{tlk} \geq 0, \quad \forall t \in T \tag{3.25}$$

The model was run using the provided data and the obtained results and inventory profile are displayed in Table 3.4 and Figure 3.7 respectively.

Table 3.4: EL model including ratchets results.

Expected Profit	$ 376,299.70
Reduction In Expected Profit From No-Ratchets Model	38 %
Average Inventory Level	376,103.625 MMBtu
Number of Storage Actions	18
Computation Time	5.46 Seconds
Model Size	270 Continuous Variables, 112 Integer Variables, 1489 Constraints

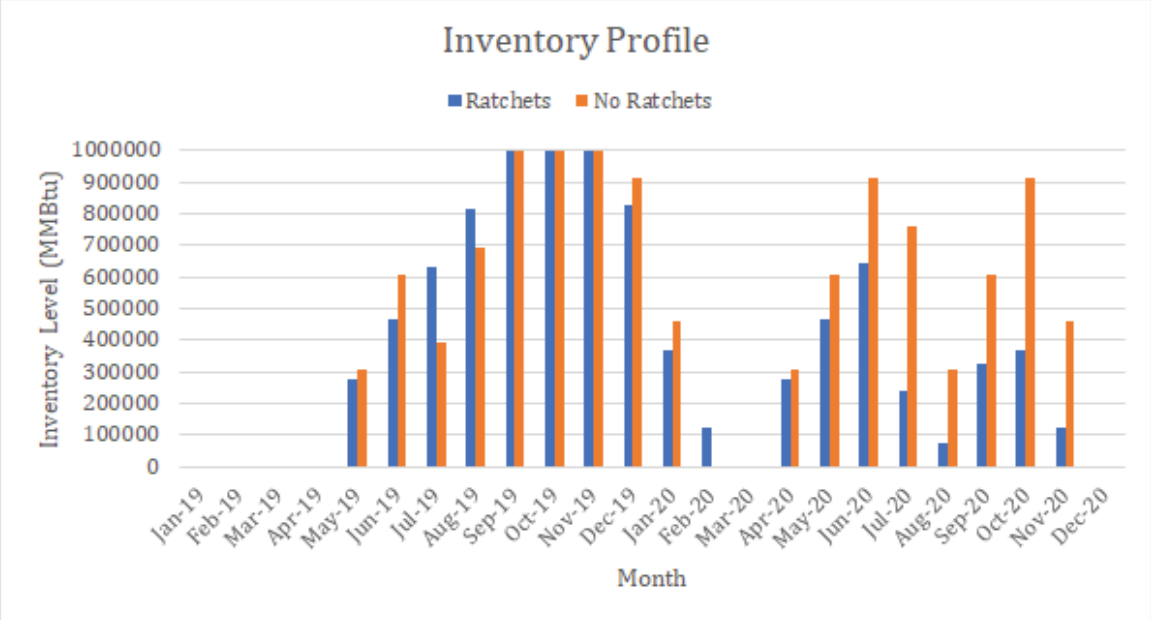

Figure 3.7: Comparison between inventory profiles when including/excluding ratcheting.

These results indicate that including the ratcheting constraints (if ratcheting is imposed on storage agreement) is essential in properly evaluating the potential value of a storage asset. The above results mean that ratcheting parameters severely impact the quantities that can be injected or withdrawn, and inherently lowers the expected profit from a storage contract.

To ensure the ratcheting parameters worked as intended, the injection/withdrawal limits for a given month were calculated analytically and compared to the optimal solution (for the calculations 30.5 days per month was utilized). The results displayed in Table 3.5 confirm that the ratcheting constraints function as intended.

Table 3.5: Analytical injection limit calculations and comparison to model results.

June 2019			
Max Proportion of Month in Ratchet 1	0.893	**Quantity**	163,500 MMBtu
Max Proportion of Month in Ratchet 2	0.107	**Quantity**	26,000 MMBtu
Total Injection Limit (per ratcheting agreement)			189,500 MMBtu
Amount Injected By Model:			189,500 MMBtu

In this section novel constraints for representing a NG storage asset ratcheting agreement have been presented. The presented model is general, and can be applied to ratcheting agreement with any number of levels. Results indicate that ignoring the impacts of inventory ratcheting results in overestimating feasible injection and withdrawal amounts. Additionally, a more accurate value can be placed on the NG storage asset by including inventory ratcheting in the formulation.

3.3.1 Accounting for Past Decisions

Here we expand on the Equally-Likely MILP model with ratcheting by considering previous positions held on a storage asset. As the simulated price curves are subject to change, it is indeed possible that the injection/storage positions taken previously, may not be the best choice, given the new information. Because this model assumes transactions are made on the futures market, the positions taken can be sold/bought on today's futures market with a brokerage fee incurred. For robustness, our model should consider the positions held and the new information and, if necessary, augment the injection/withdrawal decisions if the changes increase the expected profit. Because held positions represent sunk costs, only the quantity of gas purchased or sold needs to

be accounted for. To account for previous financial positions, the following parameters must be introduced:

K_t	Existing injection position for period t
H_t	Existing withdrawal position for period t
ϵ	Brokerage fee for augmenting position

The introduced parameters will allow the model to consider past decisions in the optimization process. Next, the following variables must be introduced:

Q_t	Amount of injection position dropped at time t
R_t	Amount of withdrawal position dropped at time t
$\Omega_t =$	$\begin{cases} 1, & \text{if injection position exists at time period } t, \\ 0, & \text{otherwise.} \end{cases}$
$\beta_t =$	$\begin{cases} 1, & \text{if withdrawal position exists at time period } t, \\ 0, & \text{otherwise.} \end{cases}$

To account for positions the objective function must be updated so the cost/benefit of augmenting the previous decisions can be accounted for. The revised objective function becomes:

$$\max \sum_{t \in T} p_t \left[-v_t + w_t + (1-\epsilon)Q_t - (1+\epsilon)R_t \right]. \tag{3.26}$$

Constraints must be introduced that ensure that the amount of injection/withdrawal position that is dropped does not exceed amounts held by previous positions (e.g. a company cannot sell a contract they do not own).

$$Q_t \leq H_t, \quad \forall t \in T \tag{3.27}$$

$$R_t \leq K_t, \quad \forall t \in T \tag{3.28}$$

A set of constraints must be introduced to ensure that, if a position for injection exists in a month, no withdrawals can be made during that month (and vice-versa).

$$M\beta_t \geq K_t - R_t, \quad \forall t \in T \tag{3.29}$$

$$M\Omega_t \geq H_t - Q_t, \quad \forall t \in T \tag{3.30}$$

$$v_t \leq (1-\beta_t)M, \qquad \forall t \in T \tag{3.31}$$

$$w_t \leq (1-\Omega_t)M \qquad \forall t \in T \tag{3.32}$$

As previously held injection/withdrawal positions impact projected inventory of the well, some of the constraints from the base formulation must be modified to ensure proper inventory balance and feasible injection/withdrawal quantities. Modified versions of constraints (3.5), (3.12) and (3.13) are as follows:

$$v_t + H_t - Q_t \leq \sum_{\ell \in L} \sum_{k \geq \ell} \lambda_{t\ell k} f_k, \qquad \forall t \in T, \tag{3.33}$$

$$w_t + K_t - R_t, \leq \sum_{\ell \in L} \sum_{k \leq \ell} \mu_{t\ell k} g_k, \qquad \forall t \in T, \tag{3.34}$$

$$I_t = I_{t-1} + v_{t-1} - w_{t-1} + H_{t-1} - Q_{t-1} - K_{t-1} + R_{t-1}, \quad \forall t \in T. \tag{3.35}$$

These constraints will allow modifications made to previous positions to be accounted for in the working inventory. This ensures that if an injection position is dropped, the gas is no longer available for sale and withdrawal, and vice-versa for withdrawal positions. The complete formulation is then written as (note the set notation used was shortened to comfortably align all equations):

$$\max \sum_{t \in T} p_t \left[-v_t + w_t + (1-\epsilon)Q_t - (1+\epsilon)R_t \right] \tag{3.26}$$

s.t.

$$I_t \leq Cap + (UB_l + Cap) \cdot r_{tl}, \qquad \forall t \in T, \ell \in L \tag{3.9}$$

$$I_t \geq LB_\ell r_{t\ell}, \qquad \forall t, \ell \tag{3.10}$$

$$\sum_{\ell \in L} r_{t\ell} = 1, \qquad \forall t \tag{3.11}$$

$$v_t + H_t - Q_t \leq \sum_{\ell \in L} \sum_{k \geq \ell} \lambda_{t\ell k} f_k, \qquad \forall t \tag{3.35}$$

$$w_t + K_t - R_t, \leq \sum_{\ell \in L} \sum_{k \leq \ell} \mu_{t\ell k} g_k, \qquad \forall t \tag{3.36}$$

$$\lambda_{t\ell k} \leq \frac{UB_\ell - I_t}{f_\ell} + M(1 - r_{t\ell}), \qquad \forall t, \ell, k = \ell, \ell \neq \ell_{|L|} \tag{3.14}$$

$$\lambda_{t\ell k} \leq \frac{UB_k - UB_{k-1}}{f_k} + M(1 - r_{t\ell}), \qquad \forall k \in L, k > \ell, t, \ell \quad (3.15)$$

$$\sum_{k \geq \ell} \lambda_{t\ell k} = r_{t\ell}, \qquad \forall t, \ell \quad (3.16)$$

$$\mu_{t\ell k} \leq \frac{I_t - LB_\ell}{g_\ell} + M(1 - r_{t\ell}), \qquad \forall t, \ell, k = \ell, \ell \neq \ell_0 \quad (3.17)$$

$$\mu_{t\ell k} \leq \frac{LB_{k+1} - LB_k}{g_k} + M(1 - r_{t\ell}), \qquad \forall k < \ell, t, \ell \quad (3.18)$$

$$\sum_{k \leq \ell} \mu_{t\ell k} = r_{t\ell}, \qquad \forall t, \ell \quad (3.19)$$

$$\lambda_{t\ell k} \leq c_{t\ell k}, \qquad \forall k \geq \ell, t, \ell \quad (3.20)$$

$$c_{t\ell k} \geq c_{t\ell k'}, \qquad \forall k \geq \ell, k' > k, t, \ell \quad (3.21)$$

$$\mu_{t\ell k} \leq x_{t\ell k}, \qquad \forall k \leq \ell, t, \ell \quad (3.22)$$

$$x_{t\ell k} \geq x_{t\ell k'}, \qquad \forall k \leq \ell, k' < k, t, \ell \quad (3.23)$$

$$Q_t \leq H_t, \qquad \forall t \quad (3.27)$$

$$R_t \leq K_t, \qquad \forall t \quad (3.28)$$

$$\beta_t \leq K_t - R_t, \qquad \forall t \quad (3.29)$$

$$\Omega_t \leq H_t - Q_t, \qquad \forall t \quad (3.30)$$

$$M\beta_t \geq K_t - R_t, \qquad \forall t \quad (3.31)$$

$$M\Omega_t \geq H_t - Q_t, \qquad \forall t \quad (3.32)$$

$$v_t \leq (1 - \beta_t)M, \qquad \forall t \quad (3.33)$$

$$w_t \leq (1 - \Omega_t)M, \qquad \forall t \quad (3.34)$$

$$I_t = I_{t-1} + v_{t-1} - w_{t-1} + H_{t-1} - Q_{t-1} - K_{t-1} + R_{t-1}, \qquad \forall t \quad (3.37)$$

$$v_t, w_t, I_t, Q_t, R_t \lambda_{tlk}, \mu_{tlk} \geq 0, \qquad \forall t \quad (3.36)$$

$$z_t, r_t, x_{tlk}, c_{tlk,t}, \Omega_t \in \{0, 1\}, \qquad \forall t \quad (3.37)$$

To ensure the model functioned as intended, two different samples of price data were selected to simulate the change in information over time. The first data sample was randomly selected from the price data, and produced the results and inventory profile described by Table 3.6 and Figure 3.8.

To test the model, the price in May 2019 was set to an extremely high value. If working properly, the model should drop the injection position for May 2019 and

Table 3.6: Results for futures model with positions, first run.

Expected Profit	$337,894.18
Number of Storage Actions	17

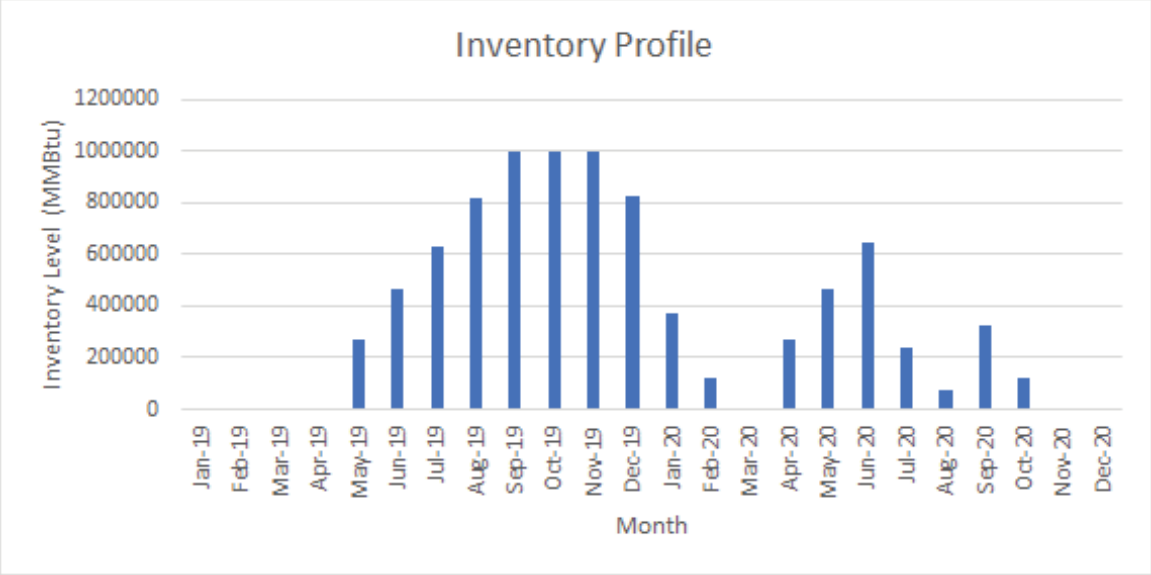

Figure 3.8: Inventory profile for futures model with positions, first run.

attempt to fill the well such that a withdrawal in May can be made.

Table 3.7: Results after May price set extremely high.

Expected Profit	$13,178,655.93
Number of Storage Actions	22

The results show that the model augmented the previous decisions to increase the overall profit. Instead of injecting in May 2019, the model dropped that position (at a cost) and changed its decisions such that a withdrawal in May 2019 could be accommodated. In the remaining sections of this chapter, it is assumed that no positions for injection or withdrawal exist on the storage asset.

3.4 Chance Constrained Model

The previous approaches take the price simulations into account by simply using their expected value. While the base model is a useful decision support tool for a trader, it does not provide the user with much information. Because of the financial nature

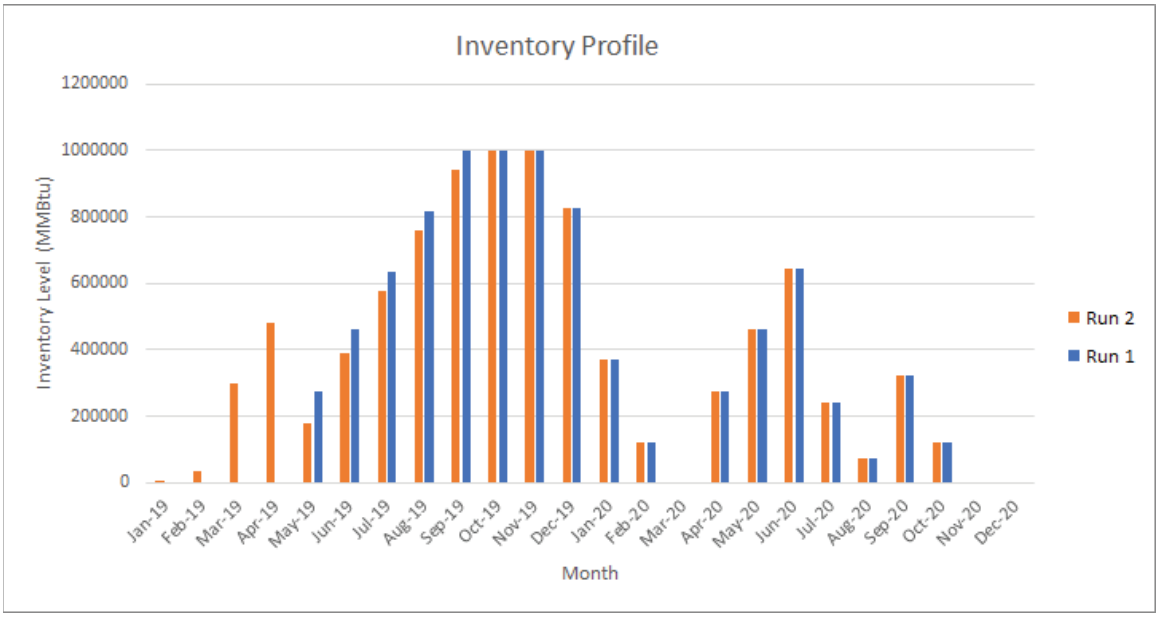

Figure 3.9: Change in inventory profile when May price set to high value.

of our problem, including risk in the model framework becomes especially important. Consider the following chance constraint, with risk level α and desired profit level b:

$$P\left\{\sum_{t\in T} p_t \cdot [\,-v_t + w_t\,] \geq b\right\} \geq 1-\alpha. \tag{3.38}$$

This constraint requires that the probability of meeting some desired profit level, b is at least $1-\alpha$. As stated by Uryasev and Rockafellar [32], the chance constraint represents the value at risk. In this case, the chance constraint is used to require the optimal solution to have some probabilistic guarantee of meeting some desired profit level, given the provided price data. This section presents two approaches for including a chance constraint in our model framework: the first method approximates the chance constraint using conditional value at risk (CVaR), and the second uses sample average approximation (SAA).

3.4.1 Chance Constraint: CVaR Approximation

To apply the CVaR approximation of the chance constraint outlined by (3.41), the procedure described by Uryasev and Rockafellar [32] is utilized. First the chance

constraint is written in the following (equivalent) form:

$$P\left\{b - \sum_{t\in T} p_t \cdot [\,-v_t + w_t\,] \leq 0\right\} \geq 1 - \alpha. \tag{3.39}$$

Next, the constraint can be expressed linearly by introducing auxiliary variables θ and u_s and the parameter q_s which represents the probability of price simulation s. Using the equally-likely assumption, each scenario probability for the provided data is equal to $\frac{1}{962}$. The price at time t according to simulation s is denoted by p_{st}. To include the chance constraint, the following constraints are added to the model presented in Section 3.2:

$$\theta + \alpha^{-1} \sum_{s\in S} q_s u_s \leq 0, \tag{3.40}$$

$$u_s \geq b - \sum_{t\in T} p_{st} \cdot [\,-v_t + w_t\,] - \theta, \qquad \forall s \in S, \tag{3.41}$$

$$u_s \geq 0, \qquad \forall s \in S. \tag{3.42}$$

Adding these constraints to the model presented in Section 3.3 leads to the following complete formulation (obtained by modifying model in Section 3.2):

$$\max \quad \sum_{t\in T} \overline{p}_t \cdot [\,-v_t + w_t\,] \tag{3.2}$$

s.t.

$$I_t \leq Cap + (UB_l + Cap) \cdot r_{tl}, \qquad \forall t \in T, \ell \in L \tag{3.9}$$

$$I_t \geq LB_\ell r_{t\ell}, \qquad \forall t \in T, \ell \in L \tag{3.10}$$

$$\sum_{\ell\in L} r_{t\ell} = 1, \qquad \forall t \in T \tag{3.11}$$

$$v_t \leq \sum_{\ell\in L} \sum_{k\geq\ell} \lambda_{t\ell k} f_k, \qquad \forall t \in T \tag{3.12}$$

$$w_t \leq \sum_{\ell\in L} \sum_{k\leq\ell} \mu_{t\ell k} g_k, \qquad \forall t \in T \tag{3.13}$$

$$\lambda_{t\ell k} \leq \frac{UB_\ell - I_t}{f_\ell} + M(1 - r_{t\ell}), \qquad \forall t \in T, \ell \in L, k = \ell, \ell \neq \ell_{|L|} \tag{3.14}$$

$$\lambda_{t\ell k} \leq \frac{UB_k - UB_{k-1}}{f_k} + M(1 - r_{t\ell}), \qquad \forall t \in T, \ell, k \in L, k > \ell \tag{3.15}$$

$$\sum_{k \geq \ell} \lambda_{t\ell k} = r_{t\ell}, \qquad \forall t \in T, \ell \in L \tag{3.16}$$

$$\mu_{t\ell k} \leq \frac{I_t - LB_\ell}{g_\ell} + M(1 - r_{t\ell}), \qquad \forall t \in T, \ell \in L, k = \ell, \ell \neq \ell_0 \tag{3.17}$$

$$\mu_{t\ell k} \leq \frac{LB_{k+1} - LB_k}{g_k} + M(1 - r_{t\ell}), \qquad \forall t \in T, \ell, k \in L, k < \ell \tag{3.18}$$

$$\sum_{k \leq \ell} \mu_{t\ell k} = r_{t\ell}, \qquad \forall t \in T, \ell \in L \tag{3.19}$$

$$\lambda_{t\ell k} \leq c_{t\ell k}, \qquad \forall t \in T, \ell, k \in L, k \geq \ell \tag{3.20}$$

$$c_{t\ell k} \geq c_{t\ell k'}, \qquad \forall t \in T, \ell, k, k' \in L, k \geq \ell, k' > k \tag{3.21}$$

$$\mu_{t\ell k} \leq x_{t\ell k}, \qquad \forall t \in T, \ell, k \in L, k \leq \ell \tag{3.22}$$

$$x_{t\ell k} \geq x_{t\ell k'}, \qquad \forall t \in T, \ell, k, k' \in L, k \leq \ell, k' < k \tag{3.23}$$

$$\theta + \alpha^{-1} \sum_{s \in S} q_s u_s \leq 0 \tag{3.41}$$

$$u_s \geq b - \sum_{t \in T} p_{st} \cdot [\, -v_t + w_t \,] - \theta, \qquad \forall s \in S \tag{3.42}$$

$$I_t = I_{t-1} + v_{t-1} - w_{t-1}, \qquad \forall t \in T \tag{3.5}$$

$$I_t \leq Cap, \qquad \forall t \in T \tag{3.6}$$

$$I_0 = I_{|T|} \tag{3.7}$$

$$u_s \geq 0, \qquad \forall s \in S \tag{3.43}$$

$$\theta \text{ u.r.} \tag{3.43}$$

$$z_t, r_t, x_{tlk}, c_{tlk} \in \{0, 1\}, \qquad \forall t \in T \tag{3.24}$$

$$v_t, w_t, I_t, \lambda_{tlk}, \mu_{tlk} \geq 0, \qquad \forall t \in T \tag{3.25}$$

To ensure the constraints worked as intended, the model was run with a desired profit level of \$0 with a risk level of $\alpha = 0.5$. Such low profit levels and confidence levels should produce the same result as the deterministic model.

Table 3.8: Chance Constrained-CVaR Model Results

Expected Profit	\$ 376,299.70
Number of Storage Actions	18
Computation Time	6.84 Seconds
Model Size	1233 Continuous Variables, 112 Integer Variables, 3414 Constraints

The same result as the deterministic model was obtained as depicted in Table 3.8, the chance-constrained model functioned as intended. The number of constraints in the model significantly increased from the previous two approaches. This is because we treat each simulated price path as a scenario. This approach could become restrictive if the number of price simulations was increased to an extremely large number, but this is unlikely. To further investigate the impact of adding (3.41) & (3.42) to the formulation, the desired profit level (b) was varied and the corresponding maximum achievable confidence level (1-α) was determined.

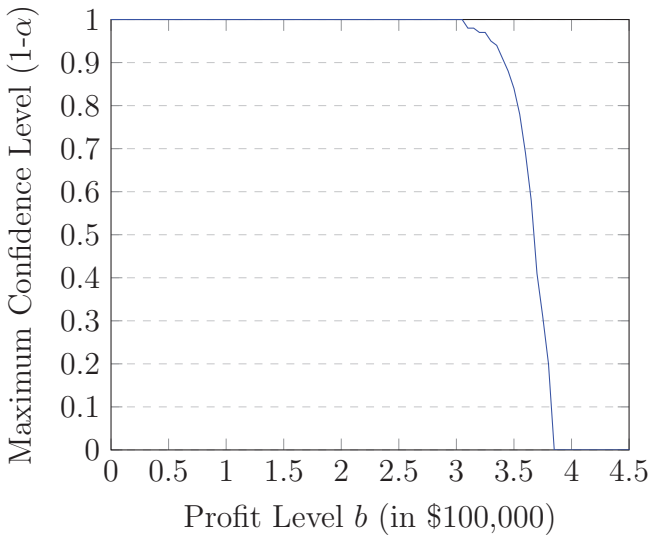

Figure 3.10: Maximum confidence level for varying values of b

As the desired profit level is increased, the maximum achievable confidence level decreases. Figure 3.10 shows some interesting behavior, as there is no decrease in the confidence level until approximately b = \$300,000, where the graph sees a very steep decline. It should be noted that for all profit levels $\gtrapprox$ \$ 390,000 there can be no probabilistic guarantee made on that profit level.

The behavior of Figure 3.10 is due to the highly mean-centered nature of the price data. Because the data is so centered around the mean, there are not many scenarios that exist wherein the expected profit greatly differs from the mean expected profit. This means that, when the desired profit level is low, the maximum confidence level (1-α) is not greatly impacted by increases in b. However, once the profit level increases past the optimal solution, the mean expected profit, and because the data is mean

centered, the probability of surpassing the mean expected profit sharply decreases to 0. Figure 3.10 also indicates that, given the price simulations, it is very easy to make less than $300,000, as the steep decline does no begin until after this profit level. From the traders' perspective, this indicates a easily-achievable baseline profit for the NG storage facility.

3.4.2 Chance Constraint-Sample Average Approximation

Another method for linearly approximating the chance constraint outlined by (3.40) is provided by Ahmed and Shapiro [33]. They utilize Sample Average Approximation to tractably reformulate a chance constraint. First the chance constraint must be written as:

$$P\{b - \sum_{t \in T} p_t \cdot [\, -v_t + w_t \,] \geq 0\} \leq \alpha \tag{3.44}$$

The above constraint is then approximated using a Sample Average Approximation constraint taking the form of:

$$\frac{1}{S} \sum_{s \in S} \mathbb{1}(b - \sum_{t \in T} p_{st} \cdot [\, -v_t + w_t \,]) \leq \alpha, \tag{3.45}$$

which can be included in the formulation by again considering the set $s \in S$ of sample price simulations and by introducing a binary variable o_s. Implementing SAA adds the following constraints to the formulation presented in Section 3.2:

$$b - \sum_{t \in T} p_{st} \cdot [\, -v_t + w_t \,] \leq M o_s, \qquad \forall s \in S, \tag{3.46}$$

$$\sum_{s \in S} o_s \leq \alpha |S|, \tag{3.47}$$

$$o_s \in \{0, 1\}, \qquad \forall s \in S. \tag{3.48}$$

In this formulation α represents the level of risk involved with storage decisions (probability the profit threshold is not met, according to the simulated data). Colloquially, this model provides a storage plan while ensuring, according to the simulated prices, the risk α and profit threshold b are respected. The complete SAA formulation

is as follows:

$$\max \quad \sum_{t\in T} \overline{p}_t \cdot [\, -v_t + w_t \,] \tag{3.2}$$

s.t.

$$I_t \le Cap + (UB_l + Cap) \cdot r_{tl}, \qquad \forall t \in T, \ell \in L \tag{3.9}$$

$$I_t \ge LB_\ell r_{t\ell}, \qquad \forall t \in T, \ell \in L \tag{3.10}$$

$$\sum_{\ell\in L} r_{t\ell} = 1, \qquad \forall t \in T \tag{3.11}$$

$$v_t \le \sum_{\ell\in L} \sum_{k\ge\ell} \lambda_{t\ell k} f_k, \qquad \forall t \in T \tag{3.12}$$

$$w_t \le \sum_{\ell\in L} \sum_{k\le\ell} \mu_{t\ell k} g_k, \qquad \forall t \in T \tag{3.13}$$

$$\lambda_{t\ell k} \le \frac{UB_\ell - I_t}{f_\ell} + M(1 - r_{t\ell}), \qquad \forall t \in T, \ell \in L, k = \ell, \ell \ne \ell_{|L|} \tag{3.14}$$

$$\lambda_{t\ell k} \le \frac{UB_k - UB_{k-1}}{f_k} + M(1 - r_{t\ell}), \qquad \forall t \in T, \ell, k \in L, k > \ell \tag{3.15}$$

$$\sum_{k\ge\ell} \lambda_{t\ell k} = r_{t\ell}, \qquad \forall t \in T, \ell \in L \tag{3.16}$$

$$\mu_{t\ell k} \le \frac{I_t - LB_\ell}{g_\ell} + M(1 - r_{t\ell}), \qquad \forall t \in T, \ell \in L, k = \ell, \ell \ne \ell_0 \tag{3.17}$$

$$\mu_{t\ell k} \le \frac{LB_{k+1} - LB_k}{g_k} + M(1 - r_{t\ell}), \qquad \forall t \in T, \ell, k \in L, k < \ell \tag{3.18}$$

$$\sum_{k\le\ell} \mu_{t\ell k} = r_{t\ell}, \qquad \forall t \in T, \ell \in L \tag{3.19}$$

$$\lambda_{t\ell k} \le c_{t\ell k}, \qquad \forall t \in T, \ell, k \in L, k \ge \ell \tag{3.20}$$

$$c_{t\ell k} \ge c_{t\ell k'}, \qquad \forall t \in T, \ell, k, k' \in L, k \ge \ell, k' > k \tag{3.21}$$

$$\mu_{t\ell k} \le x_{t\ell k}, \qquad \forall t \in T, \ell, k \in L, k \le \ell \tag{3.22}$$

$$x_{t\ell k} \ge x_{t\ell k'}, \qquad \forall t \in T, \ell, k, k' \in L, k \le \ell, k' < k \tag{3.23}$$

$$b - \sum_{t\in T} p_{st} \cdot [\, -v_t + w_t \,] \le M o_s, \qquad \forall s \in S \tag{3.47}$$

$$\sum_{s\in S} o_s \le \alpha |S| \tag{3.48}$$

$$I_t = I_{t-1} + v_{t-1} - w_{t-1}, \qquad \forall t \in T \tag{3.5}$$

$$I_t \le Cap, \qquad \forall t \in T \tag{3.6}$$

$$I_0 = I_{|T|} \tag{3.7}$$

$$z_t, r_t, x_{tlk}, c_{tlk} \in \{0,1\}, \qquad \forall t \in T \tag{3.24}$$

$$v_t, w_t, I_t, \lambda_{tlk}, \mu_{tlk} \geq 0, \qquad \forall t \in T \tag{3.25}$$

$$o_s \in \{0,1\}, \qquad \forall s \in S \tag{3.49}$$

Like the CVaR approach, SAA was validated with the deterministic model by selecting a profit level of \$0 and α =0.5, as this should produce the same results. The results in Table 3.9 confirm the formulation functions as intended.

Table 3.9: Chance Constrained-SAA Model Results

Expected Profit	\$ 376,299.70
Number of Storage Actions	18
Computation Time	7.21 Seconds
Model Size	270 Continuous Variables, 1076 Integer Variables, 2452 Constraints

The desired profit level b was varied and the corresponding maximum confidence level was determined (1-α). Figure 3.11 displays the results from both SAA and the CVaR approximations. As expected, the two methods differ slightly, but share similar behavior.

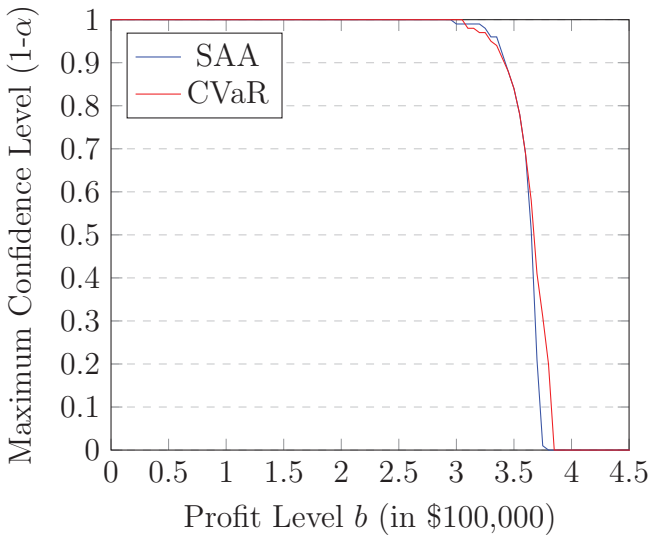

Figure 3.11: Maximum confidence level for varying profit levels

Again, the steep decline in (1-α) can be attributed to the mean-centered nature of the data. While the SAA approach contains less constraints than the CVaR model, the number of integer variables increased significantly. For this reason the SAA model suffered from poor model run times as the chance constraint became more restrictive (e.g., as the desired profit level was increased past the optimal solution, making any probabilistic guarantee difficult to obtain). For this reason, the CVaR approximation would be preferred in practice when the number of scenarios is large. The presented approaches allow for a chance constraint to be included in the MIP framework. Including the chance constraint allows a trader to have a probabilistic guarantee on their expected performance, subject to the price simulations. While a chance constraint allows for the uncertainty of the price data to be considered, some situations may call for a more risk-averse decision-making approach.

3.5 Robust Optimization: Budgeted Uncertainty

While stochastic optimization allowed us to model a chance constraint on profitability, it does not account for decisions where the decision-maker is risk-averse. In general, the variation seen in the simulated prices increases when the market is expected to be more liquid. For example, the average standard deviation of price predictions for January-March is \$0.0655, whereas the remaining months possess an average standard deviation of \$0.0177. The January-March trimester normally sees high liquidity, as cold weather increases the demand for NG. If a trader had access to a well during these months, they might be more risk averse and desire a more conservative solution such that they are protected if the price varies. To create a risk-averse model framework, Robust Optimization can be applied [34]. Robust optimization models specifically aim to minimize the worst-case scenario loss by generating more conservative optimal solutions [34].

Similar to [25], we will implement a RO model with budgeted uncertainty. Using budgeted uncertainty allows the trader to adjust their level of conservativeness when making a storage decision. This is an attractive quality, as can allow the tacit knowledge of the trader to be included into the model by adjusting the uncertainty budget. Because, on the futures market, there is no value in postponing the decision until the uncertainty is revealed, the problem can be formulated as a single-stage RO model

[34].

To formulate the RO model with budgeted uncertainty, the procedure outlined by Bertsimas and Sim [35] is followed. Consider again the deterministic objective function utilized in the base model from Section 3.2 (where p_t is the average price, according to price simulations):

$$\max_{v_t, w_t \geq 0} \quad \sum_{t \in T} \overline{p}_t \cdot [\, -v_t + w_t \,]. \tag{3.49}$$

Because the uncertain parameter, p_t, only impacts the objective function, the constraints from the nominal problem remain unchanged in the robust formulation. To reformulate the nominal problem as a RO problem with budgeted uncertainty, we first define the following uncertainty set:

$$\Xi : [\ \ p \in \mathbb{R}^{|T|} : p_t = p_t^{min} + (p_t^{max} - p_t^{min}) z_t \ , 0 \leq z_t \leq 1, \sum_{t \in T} z_t \leq \Gamma \ \ ]. \tag{3.50}$$

Next, letting $\hat{p} = p_t^{max} - p_t^{min}$, the RO problem becomes (leaving out the other constraints from the nominal problem, as they are not impacted by uncertainty):

$$\max_{v_t, w_t} \quad \min_{z_t} \ \ [p_t^{min} + \hat{p}_t z_t](-v_t + w_t) \tag{3.51}$$

$$0 \leq z_t \leq 1, \quad \forall t \in T \quad (n_t) \tag{3.52}$$

$$\sum_{t \in T} z_t \leq \Gamma \quad (y) \tag{3.53}$$

Taking the dual of the inner problem results in the following complete formulation with a modified objective function and the addition of constraint (3.57).

$$\max \sum_{t \in T} p_t^{min}(-v_t + w_t) + \sum_{t \in T} n_t + \Gamma y \tag{3.54}$$

s.t.

$$n_t + y \leq [-v_t + w_t]\hat{p}, \qquad \forall t \in T \tag{3.55}$$

$$I_t \leq Cap + (UB_l + Cap) \cdot r_{tl}, \qquad \forall t \in T, \ell \in L \tag{3.9}$$

$$I_t \geq LB_\ell r_{t\ell}, \qquad \forall t \in T, \ell \in L \tag{3.10}$$

$$\sum_{\ell \in L} r_{t\ell} = 1, \qquad \forall t \in T \tag{3.11}$$

$$v_t \leq \sum_{\ell \in L} \sum_{k \geq \ell} \lambda_{t\ell k} f_k, \qquad \forall t \in T \tag{3.12}$$

$$w_t \leq \sum_{\ell \in L} \sum_{k \leq \ell} \mu_{t\ell k} g_k, \qquad \forall t \in T \tag{3.13}$$

$$\lambda_{t\ell k} \leq \frac{UB_\ell - I_t}{f_\ell} + M(1 - r_{t\ell}), \qquad \forall t \in T, \ell \in L, k = \ell, \ell \neq \ell_{|L|} \tag{3.14}$$

$$\lambda_{t\ell k} \leq \frac{UB_k - UB_{k-1}}{f_k} + M(1 - r_{t\ell}), \qquad \forall t \in T, \ell, k \in L, k > \ell \tag{3.15}$$

$$\sum_{k \geq \ell} \lambda_{t\ell k} = r_{t\ell}, \qquad \forall t \in T, \ell \in L \tag{3.16}$$

$$\mu_{t\ell k} \leq \frac{I_t - LB_\ell}{g_\ell} + M(1 - r_{t\ell}), \qquad \forall t \in T, \ell \in L, k = \ell, \ell \neq \ell_0 \tag{3.17}$$

$$\mu_{t\ell k} \leq \frac{LB_{k+1} - LB_k}{g_k} + M(1 - r_{t\ell}), \qquad \forall t \in T, \ell, k \in L, k < \ell \tag{3.18}$$

$$\sum_{k \leq \ell} \mu_{t\ell k} = r_{t\ell}, \qquad \forall t \in T, \ell \in L \tag{3.19}$$

$$\lambda_{t\ell k} \leq c_{t\ell k}, \qquad \forall t \in T, \ell, k \in L, k \geq \ell \tag{3.20}$$

$$c_{t\ell k} \geq c_{t\ell k'}, \qquad \forall t \in T, \ell, k, k' \in L, k \geq \ell, k' > k \tag{3.21}$$

$$\mu_{t\ell k} \leq x_{t\ell k}, \qquad \forall t \in T, \ell, k \in L, k \leq \ell \tag{3.22}$$

$$x_{t\ell k} \geq x_{t\ell k'}, \qquad \forall t \in T, \ell, k, k' \in L, k \leq \ell, k' < k \tag{3.23}$$

$$I_t = I_{t-1} + v_{t-1} - w_{t-1}, \qquad \forall t \in T \tag{3.5}$$

$$I_t \leq Cap, \qquad \forall t \in T \tag{3.6}$$

$$I_0 = I_{|T|} \tag{3.7}$$

$$z_t, r_t, n_t, x_{tlk}, c_{tlk} \in \{0, 1\}, \qquad \forall t \in T \tag{3.56}$$

$$v_t, w_t, I_t, \lambda_{tlk}, \mu_{tlk} \geq 0, \qquad \forall t \in T \tag{3.25}$$

$$y \leq 0 \tag{3.57}$$

The model was evaluated with $\Gamma = 0$, which produced the following results:

Because of how the uncertainty set Ξ was defined, using a budget of $\Gamma = 0$, the problem simply reduces to the nominal problem where $p_t = p_t^{\min}$. This explains the larger expected profit than the deterministic results, as using the minimum increases

Table 3.10: Budgeted Uncertainty Model Results

Expected Profit	$430,452.53
Number of Storage Actions	18
Computation Time	7.45 Seconds
Model Size	295 Continuous Variables, 112 Integer Variables, 1514 Constraints

the profitability of certain injection/storage decisions. Like all the other approaches, the model run time is low and reasonable for the problem size. Γ was varied to investigate its impact on the optimal solution (expected profit), with the results are displayed in Figure 3.12.

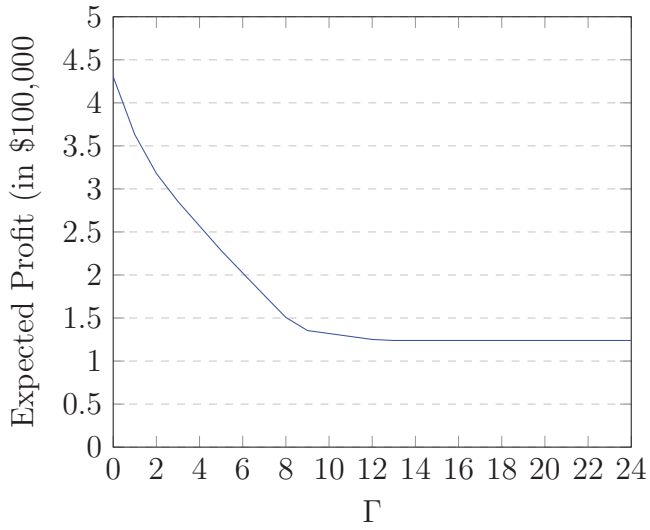

Figure 3.12: Impact of Γ on Optimal Objective Function

As the uncertainty budget is increased, the expected profit decreases. In the context of the storage problem this means, in general, injecting/withdrawing in smaller quantities and less frequently. Comparing $\Gamma = 0$ and $\Gamma = 24$, one can observe a large difference in the expected profit. With no uncertainty budget (and the nominal problem for that matter) there are 19 storage actions taken over the 24 month time-span. This means there are only five months where no action is performed. Conversely, with the maximum uncertainty budget only ten actions are taken. This means the storage facility is idle for over half the contract length. This is a reflection of the model considering the risk-adversity (as selected) and changing its solution accordingly.

While Robust Optimization is useful for protecting oneself from poor realization of prices, like other robust approaches, the solutions tend to perform poorly in average [34]. To investigate this, the uncertainty budget was set to $\Gamma = 6$ & $\Gamma = 20$ and the model was run. The optimal solutions for the different uncertainty budgets were then evaluated on each of the 962 different price scenarios (Figure 3.13).

Figure 3.13 shows that as the uncertainty budget is increased, and more conservative solutions are generated, the optimal solution performs worse when evaluated across all price simulations. Some interesting behavior is the difference between expected profit of the most conservative model ($\Gamma = 20$) and the less conservative model ($\Gamma = 6$) is not very significant. The less conservative model does outperform the highly conservative model when applied to the price scenarios, however.

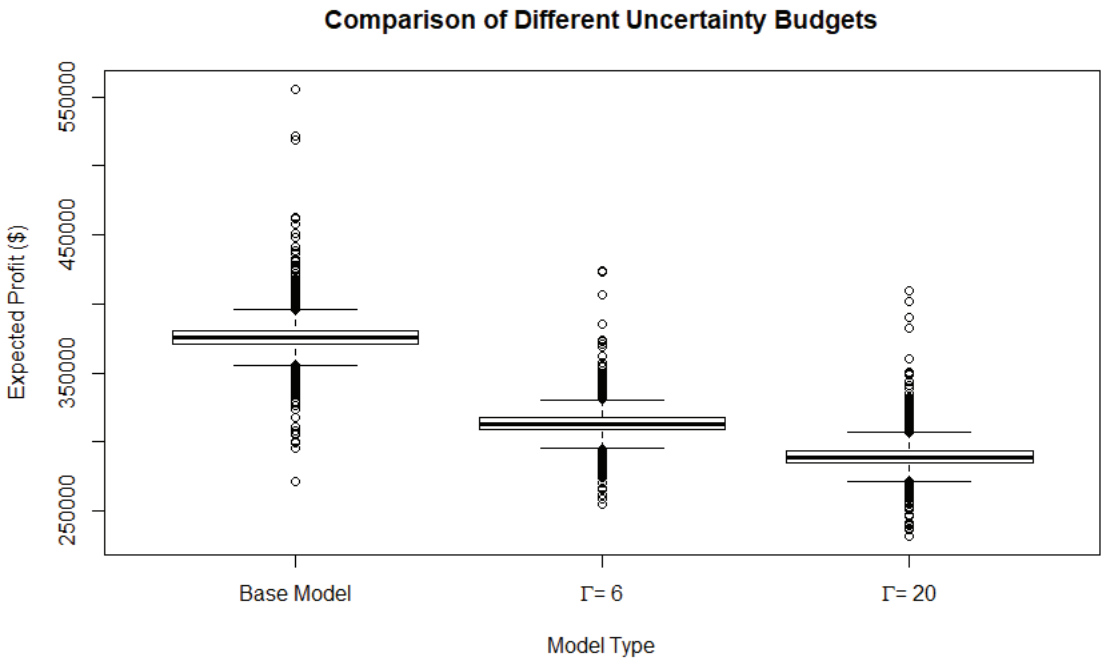

Figure 3.13: Boxplot displaying the impact of increasing the uncertainty budget on expected performance

Figure 3.13 can also be used to support the use of the robust framework. Consider the worst performing price curve for the base model. This results in an expected profit of around $275,000. This is less than a 7% difference in worst-case scenario performance, according to the price simulations and when compared to $\Gamma = 6$. The robust solution does not perform well against the price simulations because the simulations

do not represent the situation the solution has guarded itself against. The robust framework is utilized to provide injection and storage decisions such that the user is protected from changes in the expected prices (according to the simulations). While the robust solutions perform poorly in average, in times of high market volatility, they can be employed to develop risk-averse tactical plans that are not as vulnerable to poor realizations of futures prices, like the base model is. Additionally, by utilizing a budgeted uncertainty budget, this framework allows the trader to vary the uncertainty budget (which is simply a scalar) to investigate its impact on tactical storage decisions.

3.6 Distributionally Robust Optimization

As previously stated, robust formulations aim to minimize the expected loss according to the worst scenario. In the previous section, a budgeted uncertainty set was utilized to vary the conservativeness of solutions. As displayed, these solutions tend to perform poorly in average. Distributionally Robust Optimization (DRO) can be used to offer a risk-averse framework that is not as over-conservative as robust solutions [36]. This is done by assuming the probability distribution of the uncertain parameter is partially known. In this case, it is assumed that each of the price scenarios is equally likely. Using DRO, the equally-likely assumption can be treated as an estimate, and solutions can be generated such that the trader is protected if these probabilities change. To implement a DRO framework, a model utilizing a phi-divergence ambiguity set is utilized, as outlined by [36].

To begin, we assume the probability distribution of price, P, belongs to a set of distributions D, referred to as the distributional ambiguity set. Let $\hat{q}_s$ represent the estimate of the probability of scenario s (in our case $\hat{q}_s = \frac{1}{962}, \forall s \in S$), and let q_s represent the scenario probability (as determined by the model), we can define the following phi-divergence ambiguity set where $\mathcal{M}$ represents the set of all probability measures on the measurable space:

$$\mathcal{D}(\hat{Q}, \rho) := \left\{ P \in \mathcal{M} \middle| I_\phi(q, \hat{q}) = \sum_{s \in S} \hat{q}_s \phi(\frac{q_s}{\hat{q}_s}) \leq \rho \right\} \tag{3.58}$$

For simplicity, variation distance phi-divergence is used, such that:

$$I(q, \hat{q}) = \sum_{s \in S} |q_s - \hat{q}_s|. \tag{3.59}$$

With that, the DRO problem can be formulated as follows (without the nominal problem constraints, as they are unaffected by uncertainty):

$$\max_{v_t, w_t \geq 0} \min_{q_s, d_s \geq 0} \sum_{s \in S} \sum_{t \in T} q_s [p_{ts}(-v_t + w_t)] \tag{3.60}$$

s.t.

$$\sum_{s \in S} q_s = 1 \quad (\theta) \tag{3.61}$$

$$-\sum_{s \in S} d_s \geq -\rho \quad (\pi) \tag{3.62}$$

$$d_s \geq \hat{q}_s - q_s, \quad \forall s \in S \quad (\psi_s^+) \tag{3.63}$$

$$d_s \geq q_s - \hat{q}_s, \quad \forall s \in S \quad (\psi_s^-). \tag{3.64}$$

Note the variable d_s was introduced to account for the absolute value in the variation distance calculation, and the inner problem dual variables are contained in parentheses. Taking the dual modifies the objective and adds constraints (3.67) and (3.68) to formulation from Section 3.2, as follows:

$$\max \quad \theta - \rho\pi + \sum_{s \in S} \psi_s^+ \hat{q}_s - \sum_{s \in S} \psi_s^- \hat{q}_s \tag{3.65}$$

s.t.

$$\theta \leq \sum_{t \in T} p_{ts}(-v_t + w_t) - \psi_s^+ + \psi_s^-, \quad \forall s \in S \tag{3.66}$$

$$\psi_s^+ + \psi_s^- \leq \pi, \quad \forall s \in S \tag{3.67}$$

$$I_t \leq Cap + (UB_l + Cap) \cdot r_{tl}, \quad \forall t \in T, \ell \in L \tag{3.9}$$

$$I_t \geq LB_\ell r_{t\ell}, \quad \forall t \in T, \ell \in L \tag{3.10}$$

$$\sum_{\ell \in L} r_{t\ell} = 1, \quad \forall t \in T \tag{3.11}$$

$$v_t \leq \sum_{\ell \in L} \sum_{k \geq \ell} \lambda_{t\ell k} f_k, \qquad \forall t \in T \tag{3.12}$$

$$w_t \leq \sum_{\ell \in L} \sum_{k \leq \ell} \mu_{t\ell k} g_k, \qquad \forall t \in T \tag{3.13}$$

$$\lambda_{t\ell k} \leq \frac{UB_\ell - I_t}{f_\ell} + M(1 - r_{t\ell}), \qquad \forall t \in T, \ell \in L, k = \ell, \ell \neq \ell_{|L|} \tag{3.14}$$

$$\lambda_{t\ell k} \leq \frac{UB_k - UB_{k-1}}{f_k} + M(1 - r_{t\ell}), \qquad \forall t \in T, \ell, k \in L, k > \ell \tag{3.15}$$

$$\sum_{k \geq \ell} \lambda_{t\ell k} = r_{t\ell}, \qquad \forall t \in T, \ell \in L \tag{3.16}$$

$$\mu_{t\ell k} \leq \frac{I_t - LB_\ell}{g_\ell} + M(1 - r_{t\ell}), \qquad \forall t \in T, \ell \in L, k = \ell, \ell \neq \ell_0 \tag{3.17}$$

$$\mu_{t\ell k} \leq \frac{LB_{k+1} - LB_k}{g_k} + M(1 - r_{t\ell}), \qquad \forall t \in T, \ell, k \in L, k < \ell \tag{3.18}$$

$$\sum_{k \leq \ell} \mu_{t\ell k} = r_{t\ell}, \qquad \forall t \in T, \ell \in L \tag{3.19}$$

$$\lambda_{t\ell k} \leq c_{t\ell k}, \qquad \forall t \in T, \ell, k \in L, k \geq \ell \tag{3.20}$$

$$c_{t\ell k} \geq c_{t\ell k'}, \qquad \forall t \in T, \ell, k, k' \in L, k \geq \ell, k' > k \tag{3.21}$$

$$\mu_{t\ell k} \leq x_{t\ell k}, \qquad \forall t \in T, \ell, k \in L, k \leq \ell \tag{3.22}$$

$$x_{t\ell k} \geq x_{t\ell k'}, \qquad \forall t \in T, \ell, k, k' \in L, k \leq \ell, k' < k \tag{3.23}$$

$$I_t = I_{t-1} + v_{t-1} - w_{t-1}, \qquad \forall t \in T \tag{3.5}$$

$$I_t \leq Cap, \qquad \forall t \in T \tag{3.6}$$

$$I_0 = I_{|T|} \tag{3.7}$$

$$z_t, r_t, x_{tlk}, c_{tlk} \in \{0, 1\}, \qquad \forall t \in T \tag{3.24}$$

$$v_t, w_t, I_t, \lambda_{tlk}, \mu_{tlk} \geq 0, \qquad \forall t \in T \tag{3.25}$$

$$\psi_s^-, \psi_s^+ \geq 0, \qquad \forall s \in S \tag{3.68}$$

$$\theta \text{ } u.r., \pi \geq 0 \tag{3.69}$$

To test the model's size and tractability $\rho = 1$ is considered. The results from implementation are displayed in Table 3.11. Here a more conservative solution than the nominal problem is obtained, which is as expected. The results also show that there are no tractability issues with the model, although the model size has increased with the additional continuous variables and constraints.

Next, ρ was varied to investigate its impact on the optimal solution, as displayed

Table 3.11: DRO Model Using Phi-Divergence Results

Expected Profit	$ 301,160.70
Number of Storage Actions	16
Computation Time	11.23 Seconds
Model Size	2196 Continuous Variables, 112 Integer Variables, 3413 Constraints

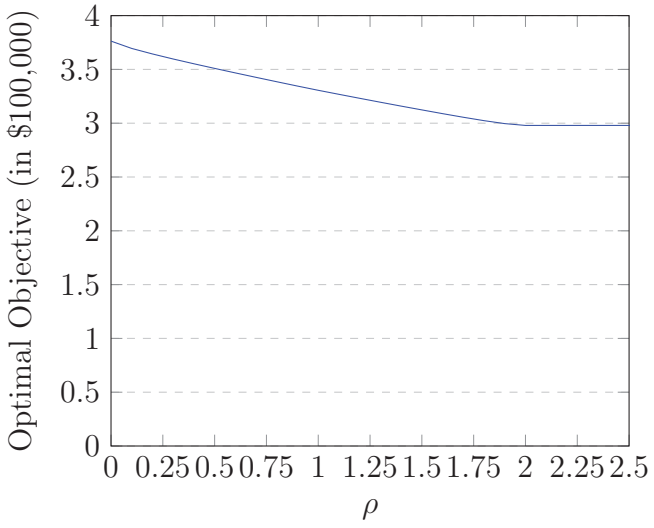

Figure 3.14: Impact of ρ on Optimal Objective

by Figure 3.14. As ρ (level of conservativeness) is increased, the expected profit decreases. We can also see that, even at its most conservative, DRO produces a less conservative solution than highly-conservative robust solutions. This approach lessens the reliance on the equally-likely price scenario assumption and can help protect the user if those probabilities change. Like before, two arbitrary values of ρ were selected and the corresponding optimal solutions were evaluated on each of the provided price simulations, producing Figure 3.15.

The results displayed in Figure 3.15 demonstrate the possible advantages of using a DRO framework. For both levels of conservativeness selected, the DRO model improves the worst-case scenario performance, while impacting the average expected profit by less than 5%. Additionally, the DRO solutions significantly outperform the robust approach when applied to the price simulations. Another interesting result is

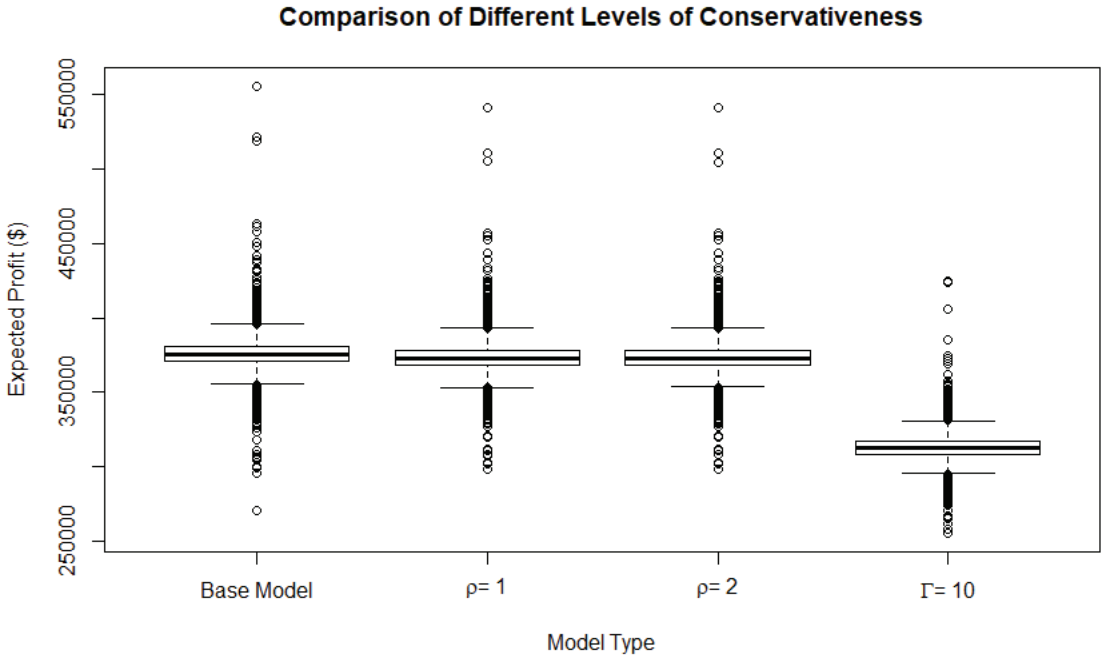

Figure 3.15: Boxplot comparing RO, DRO and Base Model Results

that the DRO model only slightly decreases the expected profit of the highest performing scenarios. The DRO results had much better expected performance than the other robust solutions. Additionally, the DRO solutions only slightly decreased profit of the best performing scenarios, especially when compared to the robust solutions. These results are encouraging, as they support the application of the DRO models in practice. The DRO framework presented similarly features an adjustable parameter to measure risk-averseness. Like the robust approach, this quality is attractive in practice, as it allows the trader to adjust ρ to different levels and investigate the impact on the provided tactical plan. The DRO framework allows for a risk-averse attitude towards injection and withdrawals to be made, without the sacrifices in expected performance made by the budgeted robust framework.

3.7 Summary

In this chapter several model variants have been presented for optimizing a NG storage asset subject to price uncertainty. The approaches utilize novel constraints that reflect an inventory ratcheting agreement. These constraints prove essential for accurately

assessing the amount of gas that can be injected or withdrawn in a month. As the provided price simulations possess uncertainty, several model variants were presented to allow varying levels of risk-averseness considered in decision-making. The chance constrained model allows a trader to ensure a probabilistic guarantee on some desired performance level. This is an attractive feature and is useful in decision-making. The robust approach creates a worst-case oriented framework that can be applied in times of high perceived risk. While the robust solutions were shown to perform poorly against the price simulations, the robust model can is still useful in some situations. Finally, to allow for a less-conservative solution than robust, DRO approach was utilized. The DRO results allowed for a risk-averse decision-making attitude, without sacrificing potential profits.

It should be noted that the presented approach does not account for the forward evolution of prices, as done by many in the literature. As this framework was developed as a decision-support tool, the approaches were created to utilize the provided price simulations to model NG prices. An extension of the presented work is to build the evolution of futures prices into the framework. As mentioned by [10], due to the high-dimensionality of modeling futures price progression, the resulting optimization model containing futures price modeling is often intractable. With that, the presented approach is a useful optimization tool that allows a trader to investigate different tactical plans for a storage field. The ratcheting constraints, specifically, allow a trader to validate proposed strategies to ensure the desired injection and withdrawal amounts are possible given any inventory ratcheting. The presented approaches can be modified for the daily cash market, which is the subject of the following chapter.

Chapter 4

Model Framework for Cash-to-Cash Transactions

This chapter of the thesis deals with the optimization of a natural gas storage asset considering cash transactions. As previously mentioned, the NG market is divided into the financial sector (futures contracts) and cash. Cash transactions (referred to as spot sale) can only be made for the **current** time-period. Unlike the futures model, there is no recourse that can be taken on held positions. Once gas is purchased or sold, the commodity is delivered the next day. Our model framework is specifically developed for and applied to the summer fill of a NG storage asset. In this case, it is assumed that the trader has access to the storage asset over the course of the summer and is required to have the well full by the end of the term (214 days). Like before, the storage asset is subject to operational characteristics that impact the feasible injection and withdrawal decisions. It is also assumed that each day the trader has access to price simulations that are used to aid in decision-making, as is the case at The Utility Company. Additionally, for the summer fill, the trader must ensure the storage asset is full by the end of the term, while attempting to profit from good price spreads in the interim. Again, a general assumption was made that The Utility Company is a relatively small player in the NG market, therefore their transactions do not impact the underlying price of NG.

The remainder of this chapter is formulated as follows: first data analysis is performed on the provided price simulations. Next, a baseline MIP model for daily storage decisions is presented and applied to the provided data. Next, various models with inventory-level constraints are presented as a possible approach for risk-management. Finally, DRO is applied to the cash model and the results are discussed. Again, Solver and computer specifications can be found in Appendix A. Python codes for cash model approaches are found in Appendices F & G.

4.1 Price Simulations

As in the futures-to-futures case, the trader has access to a discrete set of price simulations for natural gas prices. In this case the simulations model the evolution of the spot-price of NG over time. Unlike the futures curve, only the current time-period is transactable in the cash market. This means that decisions can only be made one day at a time, and positions for future injections and withdrawals can only be planned, but not acted upon.

The provided price simulations span from April 1 to October 31 2019. For each day in the time-horizon, 24 different simulated price paths are generated. Note this number is significantly lower than the amount generated for futures decisions, which is a refection of the difficulty in modeling cash prices. Figures 4.1 and 4.2 display the average price and standard deviation of prices, according to the provided simulations. Here slightly different behavior than the futures-to-futures price data can be

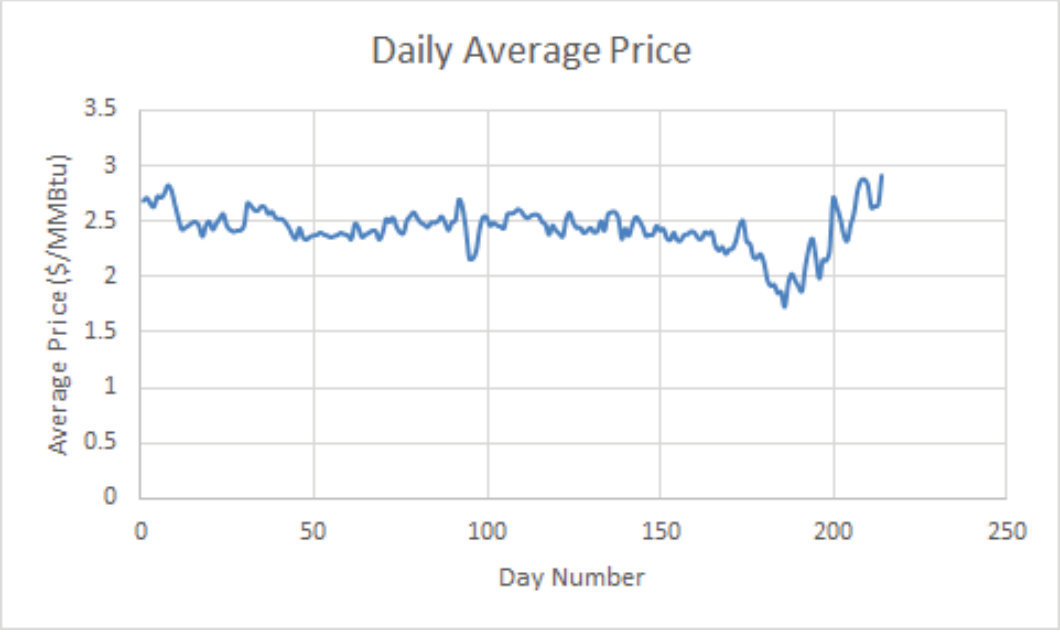

Figure 4.1: Average daily prices, according to price simulations.

observed. The standard deviation increases over time, which is most-likely a result of the increased volatility seen in the market as winter approaches. The daily prices see much fluctuation throughout the summer months, and begin to rise as the winter approaches. Histograms were generated to investigate the distribution of daily prices,

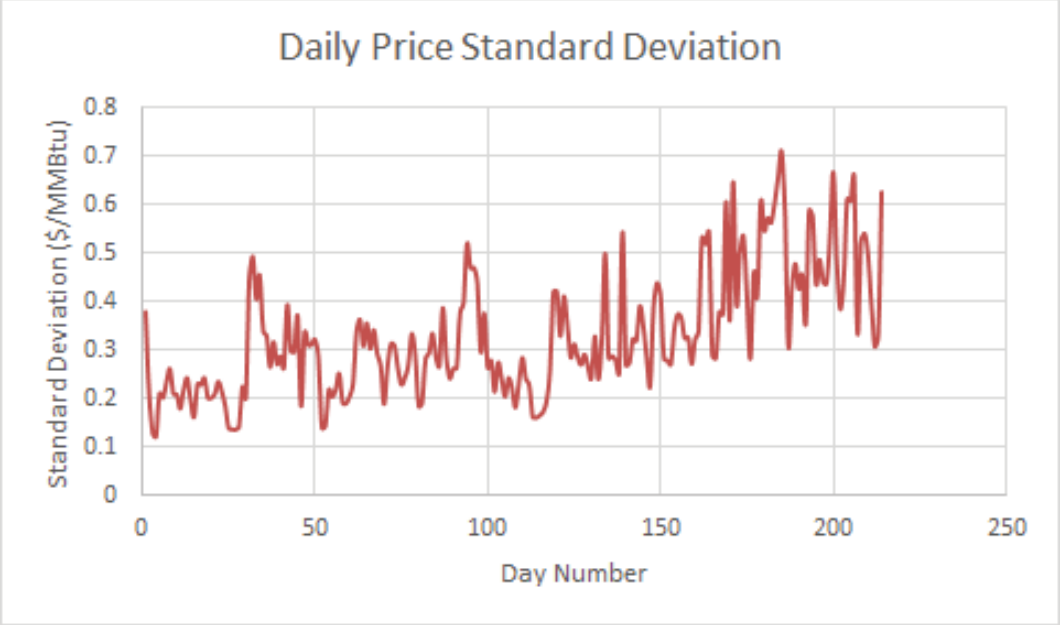

Figure 4.2: Standard deviation of daily prices, according to price simulations

an example of which is displayed in Figure 4.3. The daily prices appear to follow a general symmetric, mean-centered distribution. The distribution of cash prices is much less mean-centered than the futures data and approximately 75% of the daily prices are normally distributed.

4.2 Daily Model-Equally-Likely MIP

To model cash decisions, the futures-to-futures framework presented in the previous chapter can be modified. Each of the simulated price paths is assumed to be equally-likely. This means that now, for each day, there is a set of price predictions, $s \in S$ of size 24. The summer fill occurs over a planning horizon of 214 days therefore $|T| = 214$. Unlike the futures market, the transactions can only be taken for the current day. In this case, it can be assumed that the trader knows today's price for certain. Current decisions are denoted by t_0. Because of this we now define the set T as follows: $T =: \{0, 1, ..., |T|\}$.

For this specific storage asset, it was assumed that there is no inventory ratcheting agreement in place, and the facility is subject to constant injection and withdrawal capacities. Additionally, because the cash market involves the physical purchase and sale of NG, it is important to model fuel loss, as in cash transactions it is considered

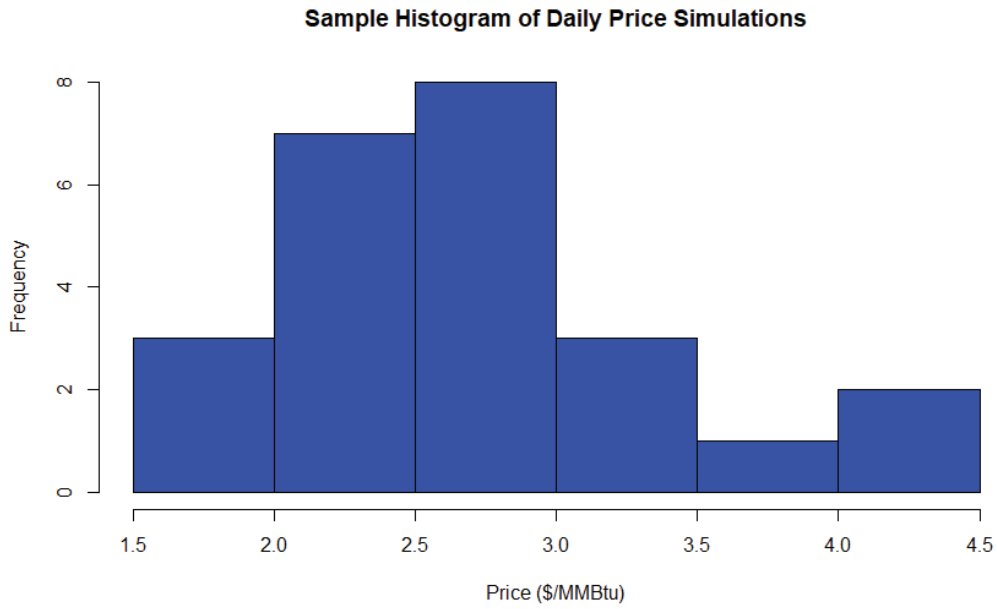

Figure 4.3: Sample Distribution of Price Simulations

to have a large impact on injection and withdrawal amounts. In this case each cash transaction incurs a transaction fee. The following parameters are utilized in the cash model:

p_0	Current spot price of NG, in \$ per MMBtu
p_{st}	cash price of NG at time t according to prediction s
Cap	maximum capacity available for storage
f	maximum injection quantity during a single time period t
g	maximum withdrawal quantity during a single time period t
ϵ	Transaction fee for purchases/sales of NG, in %
δ	Fuel loss rate for injections/withdrawals, in %
I_0	Initial inventory level

To formulate the model, the following variables are used:

v_t	quantity of NG purchased to be injected during time t
w_t	quantity of NG withdrawn to be sold during time t

I_t inventory of NG at the beginning of time period t

$$z_t = \begin{cases} 0, & \text{if no injection occurs during time period } t, \\ 1, & \text{if no withdrawal occurs during time period } t. \end{cases}$$

With that, the daily (cash) model is formulated as follows:

$$\max \quad p_0[-(1+\epsilon)v_0 + (1-\epsilon)w_0] + \sum_{t\in T} \overline{p}_t[-(1+\epsilon)v_t + (1-\delta)(1-\epsilon)w_t] \tag{4.1}$$

s.t.

$$v_t \leq f \cdot (1 - z_t), \qquad \forall t \in T \tag{4.2}$$

$$w_t \leq g \cdot z_t, \qquad \forall t \in T \tag{4.3}$$

$$I_t = I_{t-1} + (1-\delta)v_{t-1} - w_{t-1}, \qquad \forall t \in T \setminus \{0\} \tag{4.4}$$

$$I_t \leq Cap, \qquad \forall t \in T \setminus \{0\} \tag{4.5}$$

$$I_{|T|} = Cap \tag{4.6}$$

$$v_t, w_t, I_t \geq 0, \qquad \forall t \in T \tag{4.7}$$

$$z_t \in \{0, 1\}, \qquad \forall t \in T \tag{4.8}$$

This is model is an adapted version of the model presented in Section 3.2 with spot sales and a fill requirement for the well (4.6). This framework must also include the spot price of NG as an additional parameter, as this would be known and not a function of the simulations. The fuel loss rate and brokerage fees are also included in this model. Fuel loss in imposed on the actual quantity being injected, but not on the withdrawn quantity in constraint (4.4). This is to reflect the gas being completely removed from the available inventory. The fuel loss is imposed on withdrawals in the objective function, as fuel loss impacts the quantity available to sell. The brokerage fees were accounted for by increasing the injection price by ϵ % and reducing the withdrawal price by the same amount. As seen in (4.1) and (4.4), the withdrawal at period $t-1$ results in a revenue of $(1-\delta)(1-\epsilon)w_{t-1}$ after accounting for fuel loss and brokerage fees.

It should be noted that while the model will output decisions for each day in the project horizon, only today's action can be taken. Decisions into the future are used to create a **projected** inventory profile that (given all current information) that will result in the optimal cash flow and ensure today's decision is feasible (given the fill requirement). Each day, as the model is re-optimized using updated price simulations (and spot prices) the optimal decisions may change. Because future actions cannot be

taken, there is no recourse (as seen in the futures model). For the cash model, once a decision has been made and gas has been injected or withdrawn, upon re-optimization, only the starting inventory is required to be updated

Unlike in the futures model case, the future transactions in the cash market are impacted by the time value of money. In general, this would mean that any of the projected future transactions should be discounted by $(1+\eta)^t$, where η represents a discount rate per period. For this specific application a value of $\eta = 0$ is utilized because The Utility Company discourages its traders from conservative decision-making. This particular company is extremely profitable and they have a strong asset position to offset any possible losses. So this space is created separately from their investments, in bonds for example.

Along with the price simulations, the following parameters were provided to test the model:

Table 4.1: Operational characteristics for daily model, provided by The Utility Company.

f	20,000 MMBtu/day
g	40,000 MMBtu/day
Cap	1,000,000 MMBtu/day
I_0	0 MMBtu
ϵ	1%
δ	1%

The model was implemented using the provided data, with the results and inventory profile displayed by Figure 4.4 and Table 4.2. Using the expected price from the simulations, the model attempts to inject and withdraw gas throughout the summer before finally filling the asset by the end of term. The model does not begin the actual fill until later in the term, and this behavior is confirmed when re-examining Figure 4.1. There is a large dip in average price between days 175 and 200, which would make this the most advantageous time to inject. This delay in the summer fill presents a risk-management issue for trader. While, according to the price simulations the price will be low at these times, the predicted prices are uncertain and subject to change. If the fill is delayed until the end of the summer and then a weather event causes massive price increases, the trader will be extremely vulnerable. To manage this risk, several approaches can be taken.

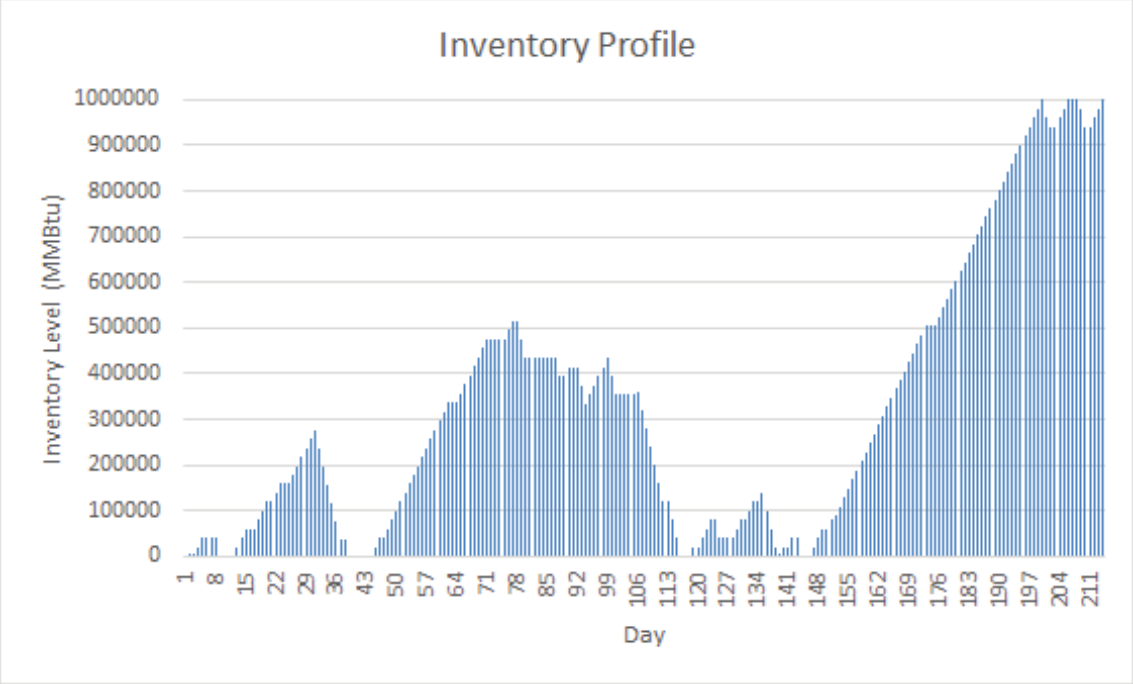

Figure 4.4: Inventory profile obtained by equally-likely base model.

Table 4.2: EL Daily Model Results.

Expected Cash Flow	-$ 2,110,632.62
Number of Actions	152
Number of Idle Days	62
Computation Time	4.26 Seconds

4.3 Daily Model-Baseline Fill Constraint

One approach to manage the aforementioned risk is for the trader to utilize tacit knowledge to decide on inventory waypoints they want the summer fill to meet. In this case, the trader could select inventory levels that must be maintained from a certain time-period and onwards, such that they are not delaying the fill too long and leaving themselves open to changes in price. Letting F represent the desired fill level and t' represent the day at which the fill constraint must be enforced after, the daily formulation can be updated as (where (4.9) is added to the formulation):

$$\max \quad p_0[-(1+\epsilon)v_0 + (1-\epsilon)w_0] + \sum_{t \in T} \overline{p}_t[-(1+\epsilon)v_t + (1-\delta)(1-\epsilon)w_t] \qquad (4.1)$$

s.t.

$$I_t \geq F, \qquad \forall t \geq t' \tag{4.9}$$

$$v_t \leq f \cdot (1 - z_t), \qquad \forall t \in T \tag{4.2}$$

$$w_t \leq g \cdot z_t, \qquad \forall t \in T \tag{4.3}$$

$$I_t = I_{t-1} + (1 - \delta)v_{t-1} - w_{t-1}, \qquad \forall t \in T \setminus \{0\} \tag{4.4}$$

$$I_t \leq Cap, \qquad \forall t \in T \setminus \{0\} \tag{4.5}$$

$$I_{|T|} = Cap \tag{4.6}$$

$$v_t, w_t, I_t \geq 0, \qquad \forall t \in T \tag{4.7}$$

$$z_t \in \{0, 1\}, \qquad \forall t \in T \tag{4.8}$$

In practice, a trader could include as many of these constraints as they desire. Additionally, by indexing F by t a trader could specify inventory requirements for each time-period in the summer fill. This constraint allows the trader additional flexibility and the opportunity to inject their tacit knowledge into the modeling process. In discussions with the traders, it was determined that for the trader to feel comfortable with the proposed fill strategy, the well should remain at least half full from day 148 and onward. The cash model was implemented using the provided data, with the results and inventory profile displayed by Table 4.3 and Figure 4.5, respectively. The quantities in parentheses indicate the difference between the present and baseline model.

Table 4.3: Daily model results with baseline fill constraint.

Expected Cash Flow	-$ 2,144,729.24 (-1.59%)
Number of Actions	136 (-16)
Number of Idle Days	78 (+16)
Computation Time	5.23 Seconds

The inventory profile has been "lifted" compared to the base model results. Interestingly, the large change in the inventory profile did not significantly impact the expected performance when compared to the base model. From a trader's perspective, this tactical plan is more realistic, as it does not push the fill of the asset until very late in the term. Additionally, this approach increased the number of projected

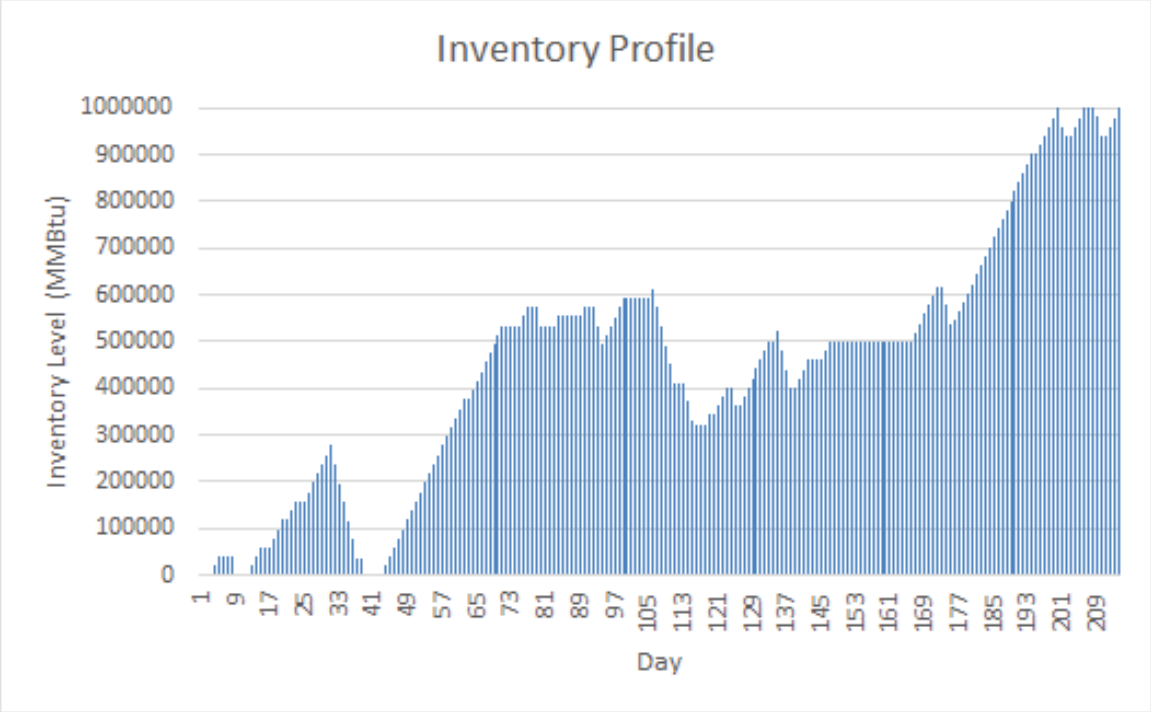

Figure 4.5: Inventory profile obtained using specified fill constraint.

idle days, which provides the trader with additional flexibility to respond to changes in prices.

4.3.1 Daily Model-Complex Fill Constraints Using Utility Functions

The previous section presented a very simple approach to managing risk, with respect to the well inventory level. The provided approach can provide time-specific inventory constraints, but this would require manual calculations from the trader. This section expands on the baseline fill constraints by generating different constraints for the inventory profile based on the risk-adversity of the trader. To create risk-management constraints on the physical inventory level, the concept of exponential utility can be utilized. Utilizing exponential utility functions can allow for different inventory profiles to be generated, based on the trader's risk adversity. As stated by Norstad [37], utility functions are utilized frequently in financial applications, as they allow traders to vary their risk tolerance to investigate the impact on expected returns. In this case, the utility function is generated on the inventory level, as it is the measure requiring different risk tolerances. Consider the following utility functions, where t is

used to represent the current moment in time and K is used as a scaling parameter for each of the three functions. Additionally the shape of the functions are determined by a parameter a, where $0 \leq a < 1$.

$$I(t) = K_1 t \tag{4.10}$$

$$I(t) = K_2(1 - e^{at}) \tag{4.11}$$

$$I(t) = K_3 e^{at} \tag{4.12}$$

These functions can be used to generate three tiers of desired inventory profiles. The first function, which is linear, represents a risk-neutral attitude towards the inventory level, where (4.11) represents a risk averse attitude and (4.12) represents a risk-seeking attitude. These functions can be used by the trader to generate different desired inventory profiles for the summer fill, based on their risk-tolerance. These functions can be explicitly included in the base formulation as follows:

$$\max \quad p_0[-(1+\epsilon)v_0 + (1-\epsilon)w_0] + \sum_{t \in T} \overline{p}_t[-(1+\epsilon)v_t + (1-\delta)(1-\epsilon)w_t] \tag{4.1}$$

s.t.

$$I_t \geq I(t), \qquad \forall t \geq t' \tag{4.13}$$

$$v_t \leq f \cdot (1 - z_t), \qquad \forall t \in T \tag{4.2}$$

$$w_t \leq g \cdot z_t, \qquad \forall t \in T \tag{4.3}$$

$$I_t = I_{t-1} + (1-\delta)v_{t-1} - w_{t-1}, \qquad \forall t \in T \setminus \{0\} \tag{4.4}$$

$$I_t \leq Cap, \qquad \forall t \in T \setminus \{0\} \tag{4.5}$$

$$I_{|T|} = Cap \tag{4.6}$$

$$v_t, w_t, I_t \geq 0, \qquad \forall t \in T \tag{4.7}$$

$$z_t \in \{0, 1\}, \qquad \forall t \in T \tag{4.8}$$

To investigate the impact of adding these constraints, the following cases were considered:

The approaches listed in Table 4.4 were created with the trading team to reflect

Table 4.4: Different inventory profile, risk-management approaches

Profile #	**Risk Level**
1	Risk-neutral with no half-full constraint
2	Risk-neutral, where well is required to be half-full by day 148
3	Risk-seeking with no half-full constraint
4	Risk-seeking, where well is required to be half-full by day 148
5	Risk-averse with no half-full constraint
6	Risk-averse, where well is required to be half-full by day 148

real-world decision-making. The approaches each require different minimum inventory thresholds from day 148 and onwards. Figure 4.6 displays the profiles that the fill constraints will enforce. The risk-seeking inventory profiles would not require the well to be full until much later in the time-horizon. Conversely, the risk-averse functions require the well be full much earlier in the time horizon. The parameter K was numerically calculated for each of the approaches, and their values are reported in Appendix H.

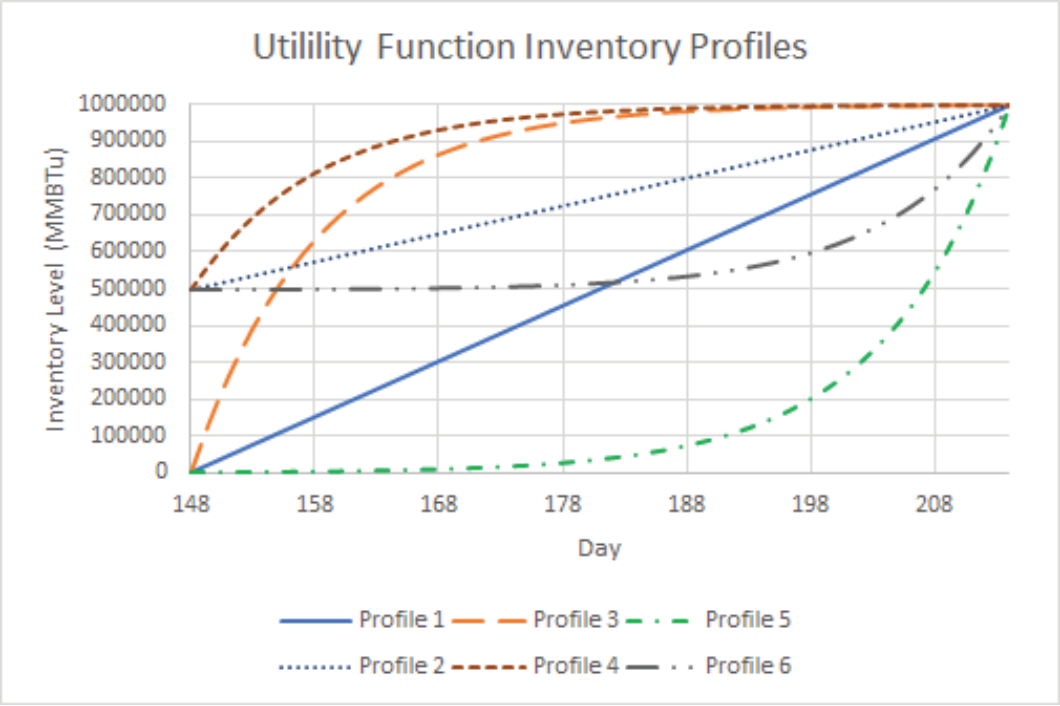

Figure 4.6: Graph displaying different inventory profile constraints based on utility functions

The different utility functions were applied to the model. Figure 4.7 displays a boxplot of expected performance subject to the price simulations of each of the utility

approaches, in addition to the base model and simple fill constraint as implemented in the previous section. The results show that profiles 1 and 3 result in the same optimal solution as the base model. This indicates that the enforced inventory profile did not impact the optimal solution. For the remaining approaches, the expected performance decreased when compared to the base model. As expected, profiles 2 and 5 performed the worse, as these were conservative. In all cases the variance in the optimal solution across the scenarios remained relatively constant. These results reveal the limitations of managing risk from an inventory perspective. In this case, the price is the only parameter subject to uncertainty. Attempting to mitigate risk from an inventory perspective and not a price perspective seems to result in lower expected performance.

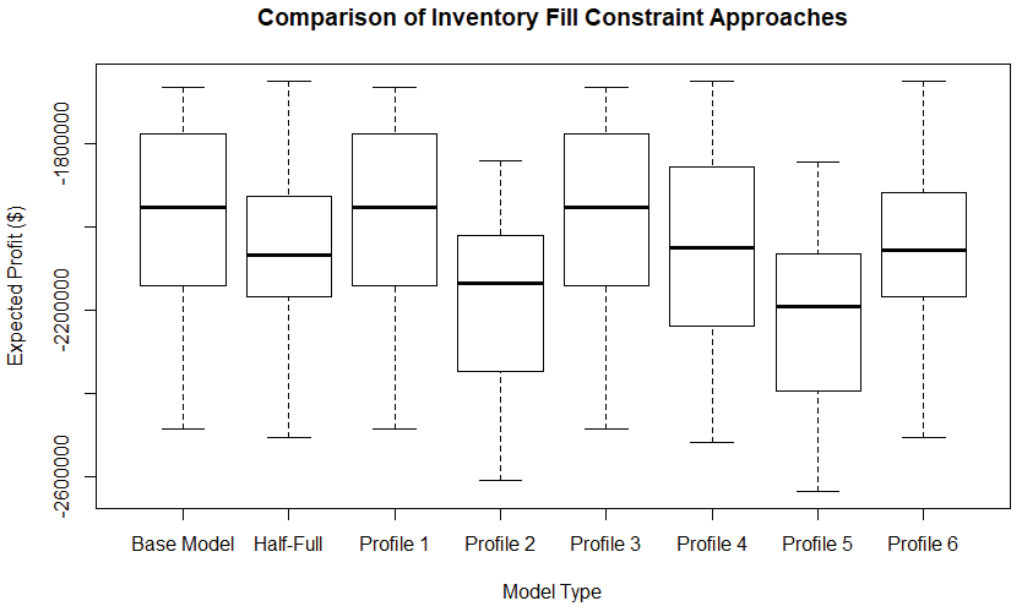

Figure 4.7: Graph displaying different inventory profile constraints based on utility functions

The inventory fill constraints are useful from a practical perspective. While enforcing constraints on the inventory level using utility decreased the expected performance, different risk attitudes can provide the trader with additional flexibility when filling the well. By requiring the well meet certain inventory levels at certain times, the trader obtains more flexibility due to possible idle days. In general, the risk-seeking profiles seem to have little value, as they normally result in the same

solution as the risk-neutral utility functions. In the base model case, the delayed fill did not allow much flexibility, as the trader must constantly inject for a long period to fill the well by the end of term. Inventory constraints can alleviate this concern by allowing the trader to require certain inventory level requirements subject to their risk tolerance.

4.4 Daily Model-Distributionally Robust Optimization

Similarly as for the futures models, DRO can be applied to allow a risk-averse attitude to be taken towards decision-making. The same DRO approach, utilizing a phi-divergence ambiguity set as per [36], was applied to the base cash model, which results in the following formulation:

$$\max_{v,w,I,\pi,\psi,\theta} \quad p_0[-(1+\epsilon)v_0 + (1-\epsilon)w_0] + \theta - \rho\pi + \sum_{s\in S}\psi_s^+\hat{q}_s - \sum_{s\in S}\psi_s^-\hat{q}_s \tag{4.14}$$

s.t.

$$\begin{aligned}
\theta \leq\ & p_0[-(1+\epsilon)v_0 + (1-\epsilon)w_0] \\
&+ \sum_{t\in T} p_{ts}[-(1+\epsilon)v_t + (1-\delta)(1-\epsilon)w_t] - \psi_s^+ + \psi_s^-, \quad \forall s \in S && (4.15)\\
\psi_s^+ + \psi_s^- &\leq \pi, \quad \forall s \in S && (4.16)\\
v_t &\leq f\cdot(1-z_t), \quad \forall t \in T && (4.2)\\
w_t &\leq g\cdot z_t, \quad \forall t \in T && (4.3)\\
I_t &= I_{t-1} + (1-\delta)v_{t-1} - w_{t-1}, \quad \forall t \in T\setminus\{0\} && (4.4)\\
I_t &\leq Cap, \quad \forall t \in T\setminus\{0\} && (4.5)\\
I_{|T|} &= Cap && (4.6)\\
v_t, w_t, I_t &\geq 0, \quad \forall t \in T && (4.7)\\
z_t &\in \{0,1\}, \quad \forall t \in T && (4.8)\\
\psi_s^-, \psi_s^+ &\geq 0, \quad \forall s \in S && (4.17)\\
\theta\ \ u.r., \pi &\geq 0 && (4.18)
\end{aligned}$$

Note that, as the spot price is not subject to uncertainty from the price simulations, its term is unaffected by the reformulation. A parameter ρ is utilized to allow the user to vary their level of risk-adversity. Using the provided data, ρ was varied to investigate its impact on the expected cash flow, with the results displayed by Figure 4.8.

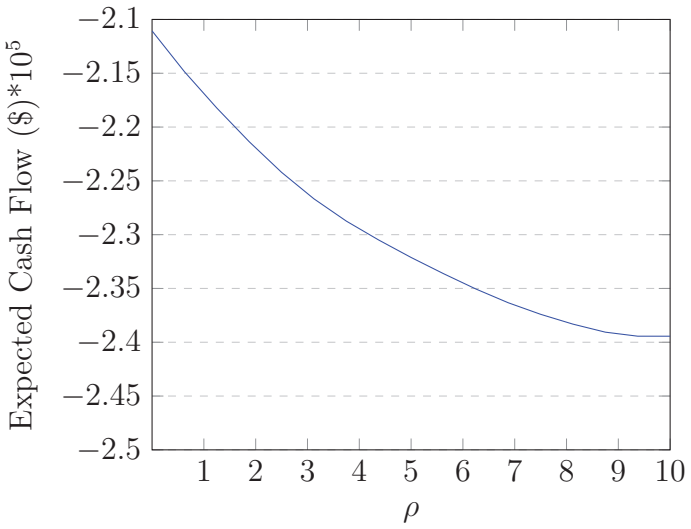

Figure 4.8: Impact of ρ on expected cash flow

Like the other models presented, this formulation possesses no tractability issues. The results are consistent with those of Chapter 3, as ρ is increased, the expected profit decreases. Clearly there is a trade-off between risk-adversity and expected performance. To further explore the impact of the DRO cash formulation, the model was run at two different ρ values, and a box-plot was generated from the optimal solution based on the price simulations.

The results displayed in Figure 4.9 support the use of the DRO framework. While both levels of ρ resulted in a tactical plan with decreased expected performance, the variance of the DRO solutions is far less than the other approaches. This indicates that the DRO approaches are mitigating the risk generated by price uncertainty more effectively than constraints placed on the inventory profile. The DRO framework significantly improves the worst-case performance when compared to the fill constraint approaches. As cash prices can be subject to change, the DRO framework allows for decisions to be made, such that the trader is protected from changes in the price simulations. This framework allows for risk to be more effectively managed and

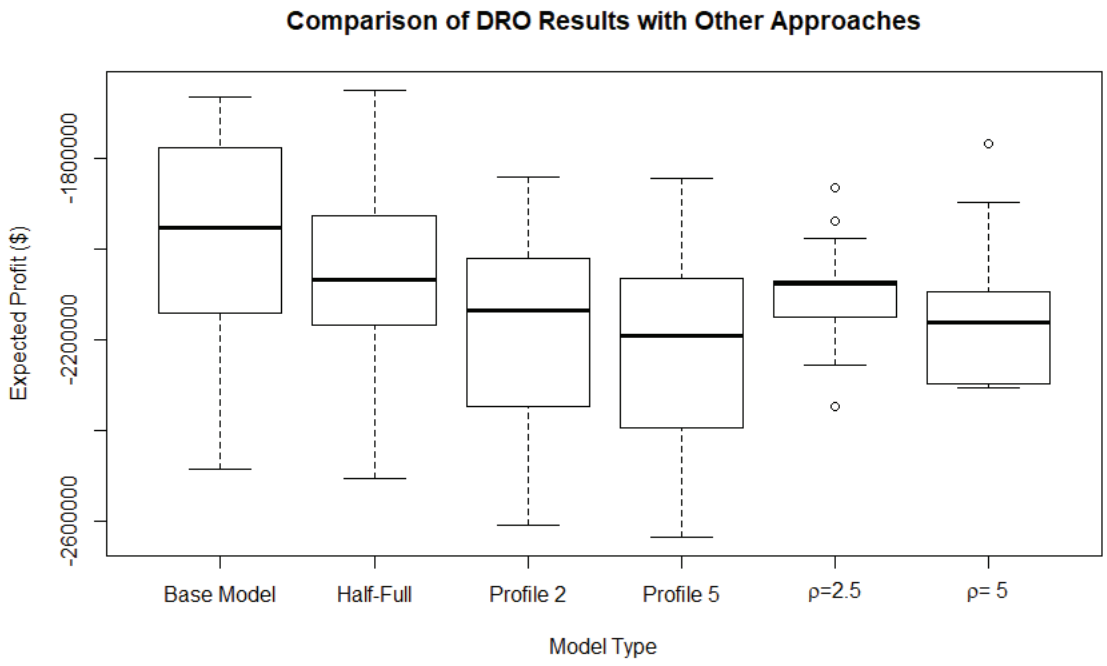

Figure 4.9: Comparison of DRO results with base model, simple fill constraint and utility fill constraints

still provides the flexibility generated by idle days, as done by the fill constraint approaches. Because this approach utilizes ρ as an adjustable parameter, it provides the trader with the opportunity to investigate different tactical plans based on their risk-adversity.

4.5 Daily Model-Robust Optimization

While the budgeted robust approach was not applied to the cash model, there are certain scenarios where the trader will be *extremely* risk averse. An example of this is near the end of the summer fill period. As the end of the term approaches, the trader will possess a hyper risk-averse attitude towards storage decisions. If there are only a few days left until the well is required to be full and a withdrawal is made, the trader becomes very exposed to changes in prices for the remaining days. If the price increases significantly, the trader will still have to purchase gas to replenish the amount withdrawn for sale. For this case, a model alternative that offers highly conservative

solutions should be explored. While budgeted uncertainty provides conservative solutions, to generate highly conservative solutions, a box-uncertainty set can be utilized in a robust framework (as per [34]). Here the trader is less interested in the flexibility afforded by the Budgeted Uncertainty set, and more interested in obtaining a highly conservative solution. To formulate the problem we must first define our uncertainty set, Ξ, as per [34].

$$\Xi := [\ \ p \in \mathbb{R}^{|T|} : p_t \geq p_t^{min}, p_t \leq p_t^{max}\ \ , \forall t \in T] \tag{4.19}$$

With that, the robust formulation is as follows (leaving out the nominal problem constraints as they are not impacted by uncertainty) with the dual variables contained in parantheses.

$$\max_{v,w,I,z}\ \ p_0[-(1+\epsilon)v_0+(1-\epsilon)w_0]+\ \min_{p\in\Xi}\ \ \sum_{t\in T} p_t[-(1+\epsilon)v_t+(1-\delta)(1-\epsilon)w_t] \tag{4.20}$$

s.t.

$$p_t \geq p_t^{min}, \quad \forall t \in T \quad (\pi_t^-) \tag{4.21}$$

$$p_t \leq p_t^{max}, \quad \forall t \in T \quad (\pi_t^+) \tag{4.22}$$

By taking the dual of the inner problem, the final tractable formulation (adapted from the base model) is as follows:

$$\max_{v,w,I,z,\pi^-,\pi^+}\ \ p_0[-(1+\epsilon)v_0 + (1-\epsilon)w_0] + \sum_{t\in T}[p^{min}\pi_t^- + p^{max}\pi_t^+] \tag{4.23}$$

s.t.

$$\pi_t^- + \pi_t^+ = -(1+\epsilon)v_t + (1-\delta)(1-\epsilon)w_t, \quad \forall t \in T \tag{4.24}$$

$$v_t \leq f \cdot (1 - z_t), \quad \forall t \in T \tag{4.2}$$

$$w_t \leq g \cdot z_t, \quad \forall t \in T \tag{4.3}$$

$$I_t = I_{t-1} + (1-\delta)v_{t-1} - w_{t-1}, \quad \forall t \in T \setminus \{0\} \tag{4.4}$$

$$I_t \leq Cap, \quad \forall t \in T \setminus \{0\} \tag{4.5}$$

$$I_{|T|} = Cap \tag{4.6}$$

$$v_t, w_t, I_t \geq 0, \qquad \forall t \in T \tag{4.7}$$

$$z_t \in \{0, 1\}, \qquad \forall t \in T \tag{4.8}$$

$$\pi_t^- \geq 0, \pi_t^+ \leq 0, \qquad \forall t \in T \tag{4.25}$$

Where (4.23) replaces (4.1) and (4.24) and (4.25) are added to the nominal formulation. This approach allows the daily price to vary between the minimum and maximum simulated value such that a solution is generated that is protected from poor price realizations. The model was implemented using the provided data, and the following results and inventory profile were obtained were the quantities in parentheses denote the difference between this approach and the base model:

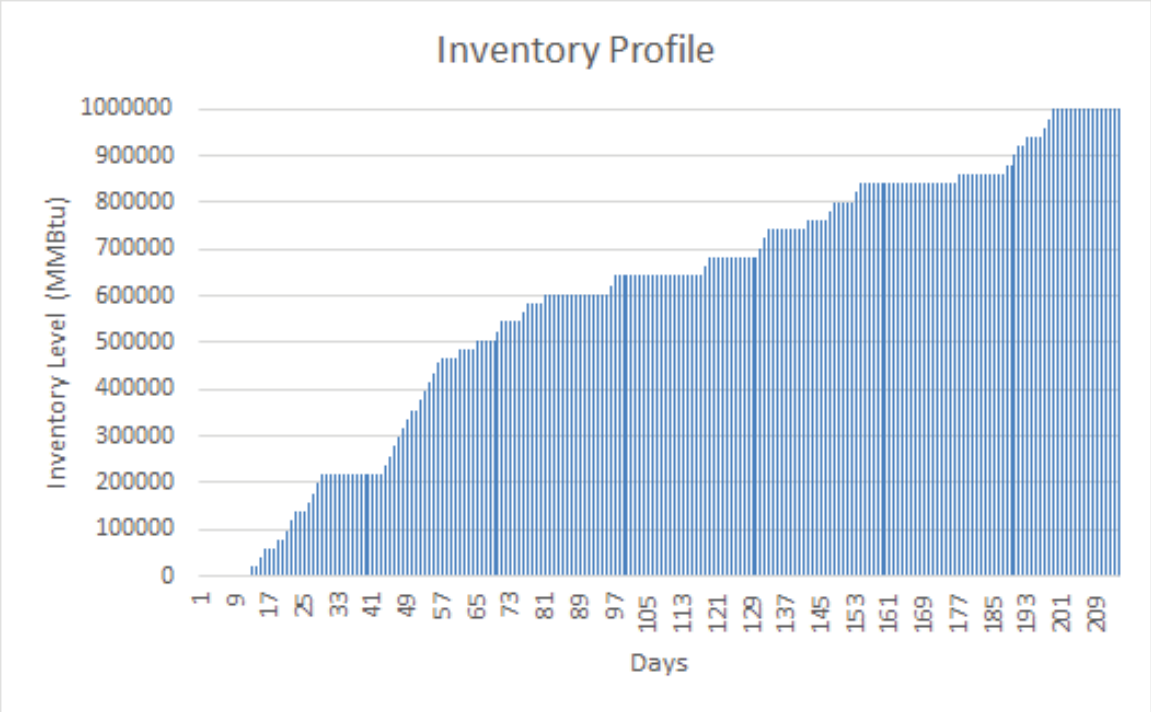

Figure 4.10: Inventory profile obtained from the box-uncertainty set model.

Table 4.5: Box-uncertainty cash model results.

Expected Cash Flow	-$ 2,771,542,79 (-31.73%)
Number of Actions	51 (-101)
Number of Idle Days	78 (+101)
Computation Time	6.45 Seconds

As expected, the box approach produces a highly conservative solution, when

compared to the nominal problem. This results in a much lower expected profit than the nominal solution. The inventory profile also reveals interesting results. Due to the highly risk-averse attitude the RO approach takes, the model chooses only to inject gas throughout the summer fill. Colloquially, the model chose the days to inject in the absolute worst case scenario. Generally, this means the model considered the highest price when injecting and lowest price when withdrawing. Because of this, no withdrawals were made. This solution provides the best days to inject, in the worst case scenario subject to the simulated prices. This approach has applications in the later stages of the summer fill, where more conservative plans for the remaining days should be made, as the approaching winter will increase the price volatility. The results also show the model injecting in smaller quantities. Instead of injecting as much as possible on fewer days, the box solution spread the injections out to protect itself from price uncertainty.

To further explore the robust results, a box-plot was generated from the robust solution by evaluating it for each of the price simulations, as displayed by Figure 4.11. While the box solution suffers from the worst expected performance, the results do have some interesting implications. While the robust solution did not involve selling any gas throughout the summer, its performance overlaps with every model approach. This indicates that one could obtain a better cashflow by simply injecting gas all summer when compared to approaches where 60+ withdrawals are made. These results display the risk involved with the cash transactions. The box solution offers an advantage over the fill constraints as the variance is significantly lower.

While the robust solution is not practical when implemented across the entire summer, the highly-risk averse solutions generated by the box model can be utilized in times of extremely high volatility (e.g. winter months or during an extreme weather event). This framework would allow the user to be very protected from unfavorable price realizations.

4.6 Summary

In this chapter, several formulations for the optimization of cash-to-cash storage transactions has been presented. While the case considered did not include inventory ratcheting, the ratcheting constraints applied to the futures model could easily be

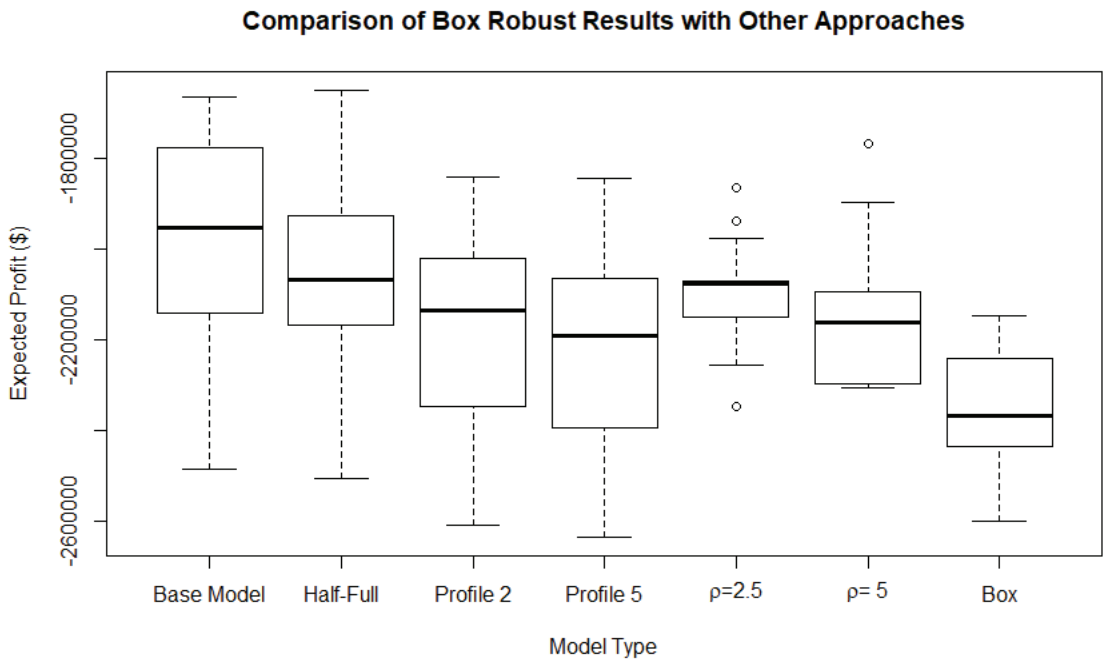

Figure 4.11: Comparison of Box results with base model, simple fill constraint, utility fill constraints and DRO

applied to the cash case. Several variants of the cash model were presented. The first set of variants involved placing constraints on the inventory levels of the well. These constraints were shown to have little value and do not seem effective in managing the uncertainty seen in the price simulations. The fill constraints do provide the trader with additional flexability, as certain approaches result in more idle days where the trader is free to make any decision.

Like in the futures model case, DRO was applied to the cash framework such that risk-averse solutions could be generated. The DRO models slightly lowered the expected performance, while significantly improving the worst-case performance of the model, when compared to the other approaches. The DRO models are practical, as they feature an adjustable parameter than can allow for a trader to test the impact of different risk-tolerances on the optimal solution. The DRO solutions additionally significantly reduce the variance of the optimal solution's expected performance.

Finally, a robust framework was presented utilizing a box uncertainty set. This approach generates highly conservative solutions, which can be used to help guard the trader from the uncertainty present in the price simulations. While the expected

performance of the RO solutions is poor, the variance of the expected performance is reduced when compared to the base model and fill constraint approaches. Additionally, the expected performance of the box approach overlapped with every other approach. This is an interesting result as the box solution did not involve selling any gas. These results indicate that, even if many sales of gas are made, there is a chance the net cash flow will be worse than if gas had only been injected.

Chapter 5

Conclusions and Recommendations for Future Work

This work presents an optimization framework developed as a decision-support tool for natural gas storage decisions. The futures framework presented allows for price simulations and operational characteristics of a NG storage facility to be considered in the modeling process. The ratcheting scheme presented is novel, and was developed specifically for this thesis. Not only can the presented models be used to provide optimal injection and withdrawal decisions, but it can be used by the trader to assess the feasibility of their injection and storage decisions. Including the ratcheting constraints, when applicable, was shown to have a significant impact on the amount of gas available for injection/withdrawal.

To allow additional flexibility, the ability to alter past decisions was added to the model framework. This allows the model to consider existing injection/storage decisions and the option to alter those decisions, if deemed more profitable. It should be noted that this approach was developed to best utilize the discrete price simulations utilized in practice. To obtain a more accurate value of storage, the evolution of the futures price curve must be considered. As mentioned by [10]-[21], accurately and tractability modeling the progression of the futures curve is difficult and requires high-factor models. While this approach does not contain the forward dynamics of NG prices, it serves as an effective decision support tool and takes advantage of the futures instruments to offer the trader recourse.

To account for the uncertainty in the price simulation data, several model variants were presented. The first variant utilized a chance-constraint to allow the user some probabilistic guarantee on performance. This model approach allows for the uncertainty in the data to be better visualized, indicating that due to the data's mean centered nature, the probabilistic guarantee on performance sharply decreases once the desired profit level is increased past the deterministic solution.

To provide a risk-averse framework, a robust alternative was presented that utilized a budgeted uncertainty set. As expected, as the uncertainty budget was increased, the solution becomes more conservative. The robust solutions were shown to perform poorly when evaluated on their expected performance. To improve the expected performance, DRO was applied to offer less conservative, risk-averse solutions. The DRO solutions improve the worst-case performance while only slightly impacting expected performance. To further explore the impacts of DRO on the futures model, other divergence measures can be considered. As the framework was developed for practical application, a simple divergence measure was utilized, other measures may provide interesting results.

The second part of this thesis presented the framework for the cash market. The cash framework functioned similarly to the futures framework, with some small differences. As cash decisions can only be made one day at a time, there is no reason to include recourse, as done in the futures model. Like the futures model, the cash models considered the provided price simulations and the physical constraints on the storage facility to provide optimal injection and storage decisions. For cash decisions, model framework was developed to manage risk in two ways: from an inventory perspective and from a price perspective.

Model variants that placed constraints on the physical inventory level were presented. As the trader was required to fill the well by the end of the summer, different alternatives were explored to offer more flexible inventory profiles for the trader. The first approach simply required a minimum inventory level be maintained from some time-period on. This constraint was shown to have little impact on the optimal solution while creating more idle days (which provides the trader with more flexibility.

Next, the concept of utility functions was utilized to investigate the impact of different risk-tolerances on fill constraints. These constraints were shown to change the injection and withdrawal decisions based on the risk level. The risk-neutral and seeking profiles allowed the well to be filled much later in the summer, which resulted in a higher profit. The risk-averse profiles did not allow a late fill, and resulted in lower profits. Most importantly, all of the approaches that placed constraints on the inventory level, while often impacting the optimal solution, had very little impact on the variance of the expected profit.

Similar to in the futures case, DRO was applied to the cash model. DRO cash solutions were shown to perform slightly worse than the deterministic solution (when compared to the price simulations), while reducing solution variance and improving worst-case performance. The DRO solutions result in more idle days, which is an attractive quality to a trader. Finally, to present a highly risk-averse framework, a robust formulation utilizing a box uncertainty set was utilized. This model's solutions were shown to perform very poorly in average (across an entire summer), but this approach has value when applied in times of very high uncertainty and volatility.

Additional expansions on the presented framework include amalgamating the futures and cash models into one unifyed framework. This would allow both cash and futures decisions to be considered simultaneously. Other types of agreements could be considered in this framework as well (e.g. bal-month agreements). To create a unifying framework, the futures progression of futures prices must be included in the modeling process. As stated in the literature review, dynamic programming seems to be the natural approach for this problem. Another possible extension is to consider a portfolio of storage facilities instead of the singular case, as examined in this thesis. Additionally, viewing the NG storage well as a part of the NG supply chain with various financial and physical commitments is a useful extension of this work. The Summer Fill Problem can also be formulated as a multi-stage stochastic, RO and DRO problem due to the nature of the cash transactions (can delay decisions). Finally, ratcheting can be explored as a function of pipeline capacity, not just contractual obligations.

Bibliography

[1] British Petroleum. Natural gas production. https://www.bp.com/en/global/corporate/energy-economics/statistical-review-of-worldenergy/natural-gas/natural-gas-production.html, 2018. [Online; accessed 23-July-2018].

[2] Fletcher J. Sturm. *Trading natural gas: cash futures options and swaps*. PennWell Books, 1997.

[3] Natural Resources Canada. Natural gas facts. https://www.nrcan.gc.ca/energy/facts/natural-gas/20067, 2018. [Online; accessed 23-July-2018].

[4] Energy Information Administration. Energy explained, Oct 2017. URL https://www.eia.gov/energyexplained/.

[5] Matt Thompson, Matt Davison, and Henning Rasmussen. Natural gas storage valuation and optimization: A real options application. *Naval Research Logistics*, 56(3):226238, 2009. doi: 10.1002/nav.20327.

[6] UCFTC. Futures market basics. https://www.cftc.gov/ConsumerProtection/EducationCenter/FuturesMarketBasics/index.htm, 2018. [Online; accessed 23-July-2018].

[7] Platts. A look forwardunderstanding forward curves in energy markets. https://platts.com/IM.Platts.Content/Products-\-Services/Products/RKSR004_0512_SpecialReportUnderstandingForwardCurvesEn-\-ergyMarkets.pdf, 2018. [Online; July 5, 2018].

[8] NYMEX & CME Group Index Mundi. Historical ng futures price data. Unpublished raw data, 2018. [Online; July 17, 2018].

[9] Luis Contesse, Juan Carlos Ferrer, and Sergio Maturana. A mixed-integer programming model for gas purchase and transportation. *Annals of Operations Research*, 139(1):39–63, 2005.

[10] Guoming Lai, François Margot, and Nicola Secomandi. An approximate dynamic programming approach to benchmark practice-based heuristics for natural gas storage valuation. *Operations research*, 58(3):564–582, 2010.

[11] René Carmona and Valdo Durrleman. Pricing and hedging spread options. *Siam Review*, 45(4):627–685, 2003.

[12] J Gray and P Khandelwal. Realistic gas storage models ii: Trading strategies. *Commodities Now*, (September):1–5, 2004.

[13] Selvaprabu Nadarajah, François Margot, and Nicola Secomandi. Relaxations of approximate linear programs for the real option management of commodity storage. *Management Science*, 61(12):3054–3076, 2015.

[14] Alexander Eydeland and Krzysztof Wolyniec. *Energy and power risk management: New developments in modeling, pricing, and hedging*, volume 206. John Wiley & Sons, 2003.

[15] Joe Wayne Byers. Commodity storage valuation: A linear optimization based on traded instruments. *Energy Economics*, 28(3):275–287, 2006.

[16] F Black. The pricing of commodity contracts. *Journal of Financial Economics*, 3(1-2):167–179, 1976.

[17] K. Sigman. Geometric brownian motion. http://www.columbia.edu/ ks20/FE-Notes/4700-07-Notes-GBM.pdf, 2006. [Online; July 27, 2018].

[18] Nicola Secomandi. Optimal commodity trading with a capacitated storage asset. *Management Science*, 56(3):449–467, 2010.

[19] Cliff Parsons. Quantifying natural gas storage optionality: a two-factor tree model. *Journal of Energy Markets*, 6(1):95–124, 2013.

[20] Zhuliang Chen and Peter A Forsyth. A semi-lagrangian approach for natural gas storage valuation and optimal operation. *SIAM Journal on Scientific Computing*, 30(1):339–368, 2007.

[21] Christophe Barrera-Esteve, Florent Bergeret, Charles Dossal, Emmanuel Gobet, Asma Meziou, Rémi Munos, and Damien Reboul-Salze. Numerical methods for the pricing of swing options: a stochastic control approach. *Methodology and computing in applied probability*, 8(4):517–540, 2006.

[22] Petter Bjerksund, Gunnar Stensland, and Frank Vagstad. Gas storage valuation: Price modelling v. optimization methods. *The Energy Journal*, pages 203–227, 2011.

[23] L. Y. Natural gas storage valuation (master's thesis, georgia institute of technology). 2007.

[24] Guoming Lai, Mulan X Wang, Sunder Kekre, Alan Scheller-Wolf, and Nicola Secomandi. Valuation of storage at a liquefied natural gas terminal. *Operations Research*, 59(3):602–616, 2011.

[25] Adrien Barbry, Miguel F Anjos, Erick Delage, and Kristen R Schell. Robust self-scheduling of a price-maker energy storage facility in the New York electricity market. *Energy Economics*, 78:629–646, 2019.

[26] Alvaro Lorca and Xu Andy Sun. Multistage robust unit commitment with dynamic uncertainty sets and energy storage. *IEEE Transactions on Power Systems*, 32(3):1678–1688, 2016.

[27] Pedram Mokrian, Moff Stephen, et al. A stochastic programming framework for the valuation of electricity storage. In *26th USAEE/IAEE North American Conference*, pages 24–27. Citeseer, 2006.

[28] M Can N Bajram. A stochastic programming approach for multi-period portfolio optimization. *Southeast Europe Journal of Soft Computing*, 1(2):1678–1688, 2012.

[29] Tarik Aouam, Ronald Rardin, and Jawad Abrache. Robust strategies for natural gas procurement. *European Journal of Operational Research*, 205(1):151–158, 2010.

[30] AE Bopp, VR Kannan, SW Palocsay, and SP Stevens. An optimization model for planning natural gas purchases, transportation, storage and deliverability. *Omega*, 24(5):511–522, 1996.

[31] Thomas Knowles. A stochastic programming model for gas supply planning. In *Proceedings of the National Decision Sciences Institute Conference*, pages 1254–1256, 1994.

[32] Stanislav Uryasev and R Tyrrell Rockafellar. Conditional value-at-risk: optimization approach. In *Stochastic optimization: algorithms and applications*, pages 411–435. Springer, 2001.

[33] Shabbir Ahmed and Alexander Shapiro. Solving chance-constrained stochastic programs via sampling and integer programming. In *State-of-the-Art Decision-Making Tools in the Information-Intensive Age*, pages 261–269. Informs, 2008.

[34] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust optimization*, volume 28. Princeton University Press, 2009.

[35] Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations research*, 52(1):35–53, 2004.

[36] Aharon Ben-Tal, Dick Den Hertog, Anja De Waegenaere, Bertrand Melenberg, and Gijs Rennen. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2):341–357, 2013.

[37] John Norstad. An introduction to utility theory. *Unpublished manuscript at http://homepage. mac. com/j. norstad*, 1999.

Appendix A

Computer and Solver Specifications

Computer and Solver Specifications	
Solver	Gurobi
Processor	Intel Core i7-8550U CPU @ 1.80 GHZ
GPU	NVIDIA TITAN RTX
Memory/RAM	16 GB

Appendix B

Futures Model Code: No Ratchets

```
import time
start_time = time.time()
from pulp import *

from openpyxl import load_workbook

from pulp import solvers

wb=load_workbook(filename= 'Price Curves 12-5-18.xlsx')

sheet_ranges=wb['n=30']

ws=wb['n=30']

Horizon=12
N=30
M=10000000
pieces=3
T=list(range(Horizon))
S=list(range(N))
L=list(range(pieces))
p=[[0 for j in range(Horizon)] for i in range(N)]
r=[[0 for j in range(Horizon)] for i in range(N)]
b=[0 for l in range(pieces)]
f=[0 for l in range(pieces)]
g=[0 for l in range(pieces)]
```

```
b[0]=150000
b[1]=300000
b[2]=1000000

f[0]=10000*30.5
f[1]=8000*30.5
f[2]=6000*30.5

g[0]=4000*30.5
g[1]=8000*30.5
g[2]=15000*30.5

Cap=1000000

d=.97

for i in range(N):

    for j in range(Horizon):

       p[i][j]=ws.cell(row=i+2, column=j+2).value

prob= LpProblem("EL-NR", LpMaximize)

v = LpVariable.matrix("v", (T),0, None, LpContinuous)
w = LpVariable.matrix("w", (T),0, None, LpContinuous)
I = LpVariable.matrix("I", (T),0, None, LpContinuous)
y = LpVariable.matrix("y", (T) ,0, 1, LpBinary)
```

```
z = LpVariable.matrix("z", (T), 0, 1, LpBinary)
a = LpVariable.matrix("a", (T,L), 0, 1, LpBinary)

prob+= ((d/N)*(lpSum([lpSum([-v[t]*p[s][t]+w[t]*p[s][t] for t in T])
for s in S])))

for t in T:

        prob+= w[t]<=I[t]

        prob+= w[t]<=M*z[t]

        prob+= v[t]<=M*(1-z[t])

        prob+= I[t]<=Cap

        prob+= lpSum([a[t][l] for l in L])==1

        prob+= v[t]<=lpSum([a[t][l]*f[l] for l in L])

        prob+= w[t]<=lpSum([a[t][l]*g[l] for l in L])

        prob+= lpSum([a[t][l]*b[l] for l in L])>=I[t]

#for t in T:
        #for l in L:
                #prob+= a[t][l]*b[l]>=I[t]

for t in T:
    if t>0:
        b=t-1
        prob+= I[t]==I[b]+v[b]-w[b]
```

```
prob+= I[Horizon-1]>=I[0]

prob+= I[0]==0

prob.solve()

print("Max profit:", prob.objective.value())

for v in prob.variables():
   if v.varValue>0:
        print(v.name, "=", v.varValue)

print("--- %s seconds ---" % (time.time() - start_time))
```

Appendix C

Futures Model Code: With Ratchets

```
import time
start_time = time.time()
from pulp import *

from openpyxl import load_workbook

from pulp import solvers

import xlsxwriter

wb=load_workbook(filename= 'Price Curves 12-5-18.xlsx')

sheet_ranges=wb['Sheet1']

ws=wb['Sheet1']

Horizon=24

pieces=3

T=list(range(Horizon))

TN=list(range(25))

L=list(range(pieces))
```

```
p=[0 for j in range(Horizon)]

UB=[0 for l in range(pieces)]
LB=[0 for l in range(pieces)]
f=[0 for l in range(pieces)]
g=[0 for l in range(pieces)]

for j in range(Horizon):
    p[j]=ws.cell(row=965, column=j+2).value
    print (p[j])

UB[0]=150001
UB[1]=300001
UB[2]=1000000

LB[0]=0
LB[1]=150000
LB[2]=300000

f[0]=10000*30.5
f[1]=8000*30.5
f[2]=6000*30.5

g[0]=4000*30.5
g[1]=8000*30.5
g[2]=15000*30.5

Cap=1000000

M=Cap
```

```
d=.97



prob= LpProblem("EL-NR", LpMaximize)


v = LpVariable.matrix("v", (T),0, None, LpContinuous)
w = LpVariable.matrix("w", (T),0, None, LpContinuous)
I = LpVariable.matrix("I", (T),0, None, LpInteger)
z = LpVariable.matrix("z", (T), 0, 1, LpBinary)
lb= LpVariable.matrix("lambda", (T,L,L), 0, 1, LpContinuous)
mu= LpVariable.matrix("mu", (T,L,L), 0, 1, LpContinuous)
r=  LpVariable.matrix("r", (T,L), 0 ,1, LpBinary)
zz= LpVariable.matrix("zz", (T,L,L),0, 1,LpBinary)
xx= LpVariable.matrix("xx", (T,L,L),0,1,LpBinary)

prob+= (lpSum([p[t]*(-v[t]+w[t]) for t in T]))


for t in T:

        prob+= w[t]<=I[t]

        prob+= w[t]<=M*z[t]

        prob+= v[t]<=M*(1-z[t])

        prob+= I[t]<=Cap
```

```
        prob+= lpSum([r[t][l] for l in L])==1


for t in T:
        prob+= I[t]<=lpSum([r[t][l]*UB[l] for l in L])
        prob+= I[t]>=lpSum([r[t][l]*LB[l] for l in L])
###restricting INJ and WITH quantities
for t in T:
##    #IF starting in R1
    prob+=v[t]<=lb[t][0][0]*f[0] +
    lb[t][0][1]*f[1]+lb[t][0][2]*f[2]+lb[t][1][1]*f[1]
    +lb[t][1][2]*f[2] + lb[t][2][2]*f[2]
    prob+=w[t]<=mu[t][0][0]*g[0] + mu[t][1][1]*g[1] +
    mu[t][1][0]*g[0]+mu[t][2][2]*g[2]
    + mu[t][2][1]*g[1] + mu[t][2][0]*g[0]
#limiting the proportions on lambda
for t in T:
    #IF Starting in R1
    prob+=lb[t][0][0]<= ((UB[0]-I[t])/f[0])+M*(1-r[t][0])
    prob+=lb[t][0][1]<= ((UB[1]-UB[0])/f[1])+M*(1-r[t][0])
    prob+=lb[t][0][2]<= ((UB[2]-UB[1])/f[2])+M*(1-r[t][0])
    prob+=lb[t][0][0]+lb[t][0][1]+lb[t][0][2]==r[t][0]
    #rules for lambda selection
    prob+=lb[t][0][0]<=zz[t][0][0]
    prob+=lb[t][0][1]<=zz[t][0][1]
    prob+=lb[t][0][2]<=zz[t][0][2]
    prob+=zz[t][0][0]>=zz[t][0][1]
    prob+=zz[t][0][0]>=zz[t][0][2]
    prob+=zz[t][0][1]>=zz[t][0][2]
    #IF starting in R2
    prob+= lb[t][1][1]<=((UB[1]-I[t])/f[1])+M*(1-r[t][1])
    prob+= lb[t][1][2]<=((UB[2]-UB[1])/f[2])+M*(1-r[t][1])
```

```
    prob+= lb[t][1][1]+lb[t][1][2]==r[t][1]
    #rules for lambda selection
    prob+=lb[t][1][1]<=zz[t][1][1]
    prob+=lb[t][1][2]<=zz[t][1][2]
    prob+=zz[t][1][1]>=zz[t][1][2]
    #IF starting in R3
    #prob+= lb[t][2][2]<=((UB[2]-I[t])/f[2])+M*(1-r[t][2])
    prob+= lb[t][2][2]==r[t][2]
for t in T:
##    #IF Starting in R1
      prob+=mu[t][0][0]==r[t][0]
##    #IF Starting in R2
      prob+=mu[t][1][1]<=((I[t]-LB[1])/g[1])+M*(1-r[t][1])
      prob+=mu[t][1][0]<=((LB[1]-LB[0])/g[0])+M*(1-r[t][1])
      prob+=mu[t][1][1]+mu[t][1][0]==r[t][1]
####    #rules for lambda selection
      prob+=mu[t][1][1]<=xx[t][1][1]
      prob+=mu[t][1][0]<=xx[t][1][0]
      prob+=xx[t][1][1]>=xx[t][1][0]
    #IF Starting R3
      prob+=mu[t][2][2]<=((I[t]-LB[2])/g[2])+M*(1-r[t][2])
      prob+=mu[t][2][1]<=((LB[2]-LB[1])/g[1])+M*(1-r[t][2])
      prob+=mu[t][2][0]<=((LB[1]-LB[0])/g[0])+M*(1-r[t][2])
      prob+=mu[t][2][2]+mu[t][2][1]+mu[t][2][0]==r[t][2]
    #rules for lambda selection
      prob+=mu[t][2][2]<=xx[t][2][2]
      prob+=mu[t][2][1]<=xx[t][2][1]
      prob+=mu[t][2][0]<=xx[t][2][0]
      prob+=xx[t][2][2]>=xx[t][2][1]
      #prob+=xx[t][2][2]>=xx[t][2][0]
      prob+=xx[t][2][1]>=xx[t][2][0]
```

```
for t in T:
    for l in L:
         for k in L:
            prob+=lb[t][l][k]>=0
            prob+=mu[t][l][k]>=0


     #for t in T:
        #for l in L:
                #prob+= a[t][l]*b[l]>=I[t]

for t in T:
    if t>0:
        b=t-1
        prob+= I[t]==I[b]+v[b]-w[b]

prob+= I[Horizon-1]>=I[0]

prob+= I[0]==0

prob.solve(GUROBI())

print("Max profit:", prob.objective.value())

#for v in prob.variables():
   #if v.varValue>0:
        #print(v.name, "=", v.varValue)
#ff=0
#for name, c in list(prob.constraints.items()):
        #ff=ff+1

#print (ff)
```

```
resultsbook=xlsxwriter.Workbook('Results.xlsx')

worksheet=resultsbook.add_worksheet()

for t in T:
    worksheet.write(0,t,value(v[t]))
    worksheet.write(1,t,value(w[t]))
    worksheet.write(2,t,value(I[t]))

resultsbook.close()
sys.exit()

print("--- %s seconds ---" % (time.time() - start_time))
```

Appendix D

Futures Model Code: Budgeted Robust

```
import time
start_time = time.time()
from pulp import *

from openpyxl import load_workbook

from pulp import solvers
import xlsxwriter

wb=load_workbook(filename= 'Price Curves 12-5-18.xlsx')

sheet_ranges=wb['Sheet1']

ws=wb['Sheet1']

Horizon=24

pieces=3

T=list(range(Horizon))

TN=list(range(25))

L=list(range(pieces))

pmin=[0 for j in range(Horizon)]
```

```
phat=[0 for j in range(Horizon)]

UB=[0 for l in range(pieces)]
LB=[0 for l in range(pieces)]
f=[0 for l in range(pieces)]
g=[0 for l in range(pieces)]

for j in range(Horizon):
    pmin[j]=ws.cell(row=969, column=j+2).value
    phat[j]=ws.cell(row=970, column=j+2).value

print (pmin)
print ("break")
print (phat)


UB[0]=150001
UB[1]=300001
UB[2]=1000000

LB[0]=0
LB[1]=150000
LB[2]=300000

f[0]=10000*30.5
f[1]=8000*30.5
f[2]=6000*30.5

g[0]=4000*30.5
g[1]=8000*30.5
```

```
g[2]=15000*30.5

Cap=1000000

M=Cap

d=.97

budget=24


prob= LpProblem("EL-NR", LpMaximize)

v = LpVariable.matrix("v", (T),0, None, LpContinuous)
w = LpVariable.matrix("w", (T),0, None, LpContinuous)
I = LpVariable.matrix("I", (T),0, None, LpInteger)
z = LpVariable.matrix("z", (T), 0, 1, LpBinary)
lb= LpVariable.matrix("lambda", (T,L,L), 0, 1, LpContinuous)
mu= LpVariable.matrix("mu", (T,L,L), 0, 1, LpContinuous)
r = LpVariable.matrix("r", (T,L), 0 ,1, LpBinary)
q = LpVariable.matrix("q", (T), None, 0, LpContinuous)
n = LpVariable("n", None, 0, LpContinuous)
zz= LpVariable.matrix("zz", (T,L,L),0, 1,LpBinary)
xx= LpVariable.matrix("xx", (T,L,L),0,1,LpBinary)

prob+= lpSum([pmin[t]*(-v[t]+w[t]) for t in T])
+lpSum([q[t] for t in T])+budget*n

for t in T:

        prob+= w[t]<=I[t]
```

```
        prob+= w[t]<=M*z[t]

        prob+= v[t]<=M*(1-z[t])

        prob+= I[t]<=Cap

        prob+= lpSum([r[t][l] for l in L])==1

        prob+= q[t] + n <= phat[t]*(-v[t]+w[t])


for t in T:
        prob+= I[t]<=lpSum([r[t][l]*UB[l] for l in L])
        prob+= I[t]>=lpSum([r[t][l]*LB[l] for l in L])
###restricting INJ and WITH quantities
for t in T:
##    #IF starting in R1
    prob+=v[t]<=lb[t][0][0]*f[0] + lb[t][0][1]*f[1]
    + lb[t][0][2]*f[2]
    +lb[t][1][1]*f[1] +lb[t][1][2]*f[2] + lb[t][2][2]*f[2]
    prob+=w[t]<=mu[t][0][0]*g[0] + mu[t][1][1]*g[1]
    + mu[t][1][0]*g[0]
    +mu[t][2][2]*g[2] + mu[t][2][1]*g[1] + mu[t][2][0]*g[0]
#limiting the proportions on lambda
for t in T:
    #IF Starting in R1
    prob+=lb[t][0][0]<= ((UB[0]-I[t])/f[0])+M*(1-r[t][0])
    prob+=lb[t][0][1]<= ((UB[1]-UB[0])/f[1])+M*(1-r[t][0])
    prob+=lb[t][0][2]<= ((UB[2]-UB[1])/f[2])+M*(1-r[t][0])
    prob+=lb[t][0][0]+lb[t][0][1]+lb[t][0][2]==r[t][0]
    #rules for lambda selection
    prob+=lb[t][0][0]<=zz[t][0][0]
```

```
    prob+=lb[t][0][1]<=zz[t][0][1]
    prob+=lb[t][0][2]<=zz[t][0][2]
    prob+=zz[t][0][0]>=zz[t][0][1]
    prob+=zz[t][0][0]>=zz[t][0][2]
    prob+=zz[t][0][1]>=zz[t][0][2]
    #IF starting in R2
    prob+= lb[t][1][1]<=((UB[1]-I[t])/f[1])+M*(1-r[t][1])
    prob+= lb[t][1][2]<=((UB[2]-UB[1])/f[2])+M*(1-r[t][1])
    prob+= lb[t][1][1]+lb[t][1][2]==r[t][1]
    #rules for lambda selection
    prob+=lb[t][1][1]<=zz[t][1][1]
    prob+=lb[t][1][2]<=zz[t][1][2]
    prob+=zz[t][1][1]>=zz[t][1][2]
    #IF starting in R3
    #prob+= lb[t][2][2]<=((UB[2]-I[t])/f[2])+M*(1-r[t][2])
    prob+= lb[t][2][2]==r[t][2]
for t in T:
##    #IF Starting in R1
      prob+=mu[t][0][0]==r[t][0]
##    #IF Starting in R2
      prob+=mu[t][1][1]<=((I[t]-LB[1])/g[1])+M*(1-r[t][1])
      prob+=mu[t][1][0]<=((LB[1]-LB[0])/g[0])+M*(1-r[t][1])
      prob+=mu[t][1][1]+mu[t][1][0]==r[t][1]
####    #rules for lambda selection
      prob+=mu[t][1][1]<=xx[t][1][1]
      prob+=mu[t][1][0]<=xx[t][1][0]
      prob+=xx[t][1][1]>=xx[t][1][0]
    #IF Starting R3
      prob+=mu[t][2][2]<=((I[t]-LB[2])/g[2])+M*(1-r[t][2])
      prob+=mu[t][2][1]<=((LB[2]-LB[1])/g[1])+M*(1-r[t][2])
      prob+=mu[t][2][0]<=((LB[1]-LB[0])/g[0])+M*(1-r[t][2])
      prob+=mu[t][2][2]+mu[t][2][1]+mu[t][2][0]==r[t][2]
```

```
    #rules for lambda selection
      prob+=mu[t][2][2]<=xx[t][2][2]
      prob+=mu[t][2][1]<=xx[t][2][1]
      prob+=mu[t][2][0]<=xx[t][2][0]
      prob+=xx[t][2][2]>=xx[t][2][1]
      #prob+=xx[t][2][2]>=xx[t][2][0]
      prob+=xx[t][2][1]>=xx[t][2][0]

for t in T:
    for l in L:
         for k in L:
            prob+=lb[t][l][k]>=0
            prob+=mu[t][l][k]>=0



     #for t in T:
        #for l in L:
                #prob+= a[t][l]*b[l]>=I[t]

for t in T:
    if t>0:
        b=t-1
        prob+= I[t]==I[b]+v[b]-w[b]

prob+= I[Horizon-1]>=I[0]

prob+= I[0]==0

prob.solve(GUROBI())

print("Max profit:", prob.objective.value())
```

```
resultsbook=xlsxwriter.Workbook('Results.xlsx')

worksheet=resultsbook.add_worksheet()
for t in T:
    worksheet.write(0,t,value(v[t]))
    worksheet.write(1,t,value(w[t]))
    worksheet.write(2,t,value(I[t]))

resultsbook.close()
sys.exit()

print("--- %s seconds ---" % (time.time() - start_time))
```

Appendix E

Futures Model Code: DRO

```
}
import time
start_time = time.time()
from pulp import *

from openpyxl import load_workbook

from pulp import solvers

import xlsxwriter

wb=load_workbook(filename= 'Price Curves 12-5-18.xlsx')

sheet_ranges=wb['Sheet1']

ws=wb['Sheet1']

Horizon=24

pieces=3

T=list(range(Horizon))

TN=list(range(25))

L=list(range(pieces))
```

```
S=list(range(962))


qhat=[(1/962) for i in S]
p=[[0 for s in S] for i in T]
UB=[0 for l in range(pieces)]
LB=[0 for l in range(pieces)]
f=[0 for l in range(pieces)]
g=[0 for l in range(pieces)]

for j in range(Horizon):
    for i in S:
        p[j][i]=ws.cell(row=i+2, column=j+2).value


UB[0]=150001
UB[1]=300001
UB[2]=1000000

LB[0]=0
LB[1]=150000
LB[2]=300000

f[0]=10000*30.5
f[1]=8000*30.5
f[2]=6000*30.5

g[0]=4000*30.5
```

```
g[1]=8000*30.5
g[2]=15000*30.5

Cap=1000000

M=Cap

d=.97

budget=1

rho=2



prob= LpProblem("EL-NR", LpMaximize)


v = LpVariable.matrix("v", (T),0, None, LpContinuous)
w = LpVariable.matrix("w", (T),0, None, LpContinuous)
I = LpVariable.matrix("I", (T),0, None, LpInteger)
z = LpVariable.matrix("z", (T), 0, 1, LpBinary)
lb= LpVariable.matrix("lambda", (T,L,L), 0, 1, LpContinuous)
mu= LpVariable.matrix("mu", (T,L,L), 0, 1, LpContinuous)
r = LpVariable.matrix("r", (T,L), 0 ,1, LpBinary)
q = LpVariable.matrix("q", (T), None, None, LpContinuous)
pi = LpVariable("pi", 0, None, LpContinuous)
wp= LpVariable.matrix("wp", (S), 0, None, LpContinuous)
wm= LpVariable.matrix("wm", (S), 0, None, LpContinuous)
theta=LpVariable("theta", None, None, LpContinuous)
zz= LpVariable.matrix("zz", (T,L,L),0, 1,LpBinary)
```

```
xx= LpVariable.matrix("xx", (T,L,L),0,1,LpBinary)

prob+= theta - rho*pi+lpSum([wp[s]*qhat[s] for s in S])
-lpSum([wm[s]*qhat[s] for s in S])

for s in S:
    prob+= theta <= lpSum([p[t][s]*(-v[t]+w[t]) for t in T])
    - wp[s] + wm[s]
    prob+= wp[s]+wm[s] <= pi

for t in T:

        prob+= w[t]<=I[t]

        prob+= w[t]<=M*z[t]

        prob+= v[t]<=M*(1-z[t])

        prob+= I[t]<=Cap

        prob+= lpSum([r[t][l] for l in L])==1

for t in T:
        prob+= I[t]<=lpSum([r[t][l]*UB[l] for l in L])
        prob+= I[t]>=lpSum([r[t][l]*LB[l] for l in L])
###restricting INJ and WITH quantities
for t in T:
##    #IF starting in R1
    prob+=v[t]<=lb[t][0][0]*f[0] + lb[t][0][1]*f[1]
    + lb[t][0][2]*f[2]
    +lb[t][1][1]*f[1] +lb[t][1][2]*f[2] + lb[t][2][2]*f[2]
    prob+=w[t]<=mu[t][0][0]*g[0] + mu[t][1][1]*g[1]
```

```
    + mu[t][1][0]*g[0]
    +mu[t][2][2]*g[2] + mu[t][2][1]*g[1] + mu[t][2][0]*g[0]
#limiting the proportions on lambda
for t in T:
    #IF Starting in R1
    prob+=lb[t][0][0]<= ((UB[0]-I[t])/f[0])+M*(1-r[t][0])
    prob+=lb[t][0][1]<= ((UB[1]-UB[0])/f[1])+M*(1-r[t][0])
    prob+=lb[t][0][2]<= ((UB[2]-UB[1])/f[2])+M*(1-r[t][0])
    prob+=lb[t][0][0]+lb[t][0][1]+lb[t][0][2]==r[t][0]
    #rules for lambda selection
    prob+=lb[t][0][0]<=zz[t][0][0]
    prob+=lb[t][0][1]<=zz[t][0][1]
    prob+=lb[t][0][2]<=zz[t][0][2]
    prob+=zz[t][0][0]>=zz[t][0][1]
    prob+=zz[t][0][0]>=zz[t][0][2]
    prob+=zz[t][0][1]>=zz[t][0][2]
    #IF starting in R2
    prob+= lb[t][1][1]<=((UB[1]-I[t])/f[1])+M*(1-r[t][1])
    prob+= lb[t][1][2]<=((UB[2]-UB[1])/f[2])+M*(1-r[t][1])
    prob+= lb[t][1][1]+lb[t][1][2]==r[t][1]
    #rules for lambda selection
    prob+=lb[t][1][1]<=zz[t][1][1]
    prob+=lb[t][1][2]<=zz[t][1][2]
    prob+=zz[t][1][1]>=zz[t][1][2]
    #IF starting in R3
    #prob+= lb[t][2][2]<=((UB[2]-I[t])/f[2])+M*(1-r[t][2])
    prob+= lb[t][2][2]==r[t][2]
for t in T:
##    #IF Starting in R1
      prob+=mu[t][0][0]==r[t][0]
##    #IF Starting in R2
      prob+=mu[t][1][1]<=((I[t]-LB[1])/g[1])+M*(1-r[t][1])
```

```
      prob+=mu[t][1][0]<=((LB[1]-LB[0])/g[0])+M*(1-r[t][1])
      prob+=mu[t][1][1]+mu[t][1][0]==r[t][1]
####    #rules for lambda selection
      prob+=mu[t][1][1]<=xx[t][1][1]
      prob+=mu[t][1][0]<=xx[t][1][0]
      prob+=xx[t][1][1]>=xx[t][1][0]
    #IF Starting R3
      prob+=mu[t][2][2]<=((I[t]-LB[2])/g[2])+M*(1-r[t][2])
      prob+=mu[t][2][1]<=((LB[2]-LB[1])/g[1])+M*(1-r[t][2])
      prob+=mu[t][2][0]<=((LB[1]-LB[0])/g[0])+M*(1-r[t][2])
      prob+=mu[t][2][2]+mu[t][2][1]+mu[t][2][0]==r[t][2]
    #rules for lambda selection
      prob+=mu[t][2][2]<=xx[t][2][2]
      prob+=mu[t][2][1]<=xx[t][2][1]
      prob+=mu[t][2][0]<=xx[t][2][0]
      prob+=xx[t][2][2]>=xx[t][2][1]
      #prob+=xx[t][2][2]>=xx[t][2][0]
      prob+=xx[t][2][1]>=xx[t][2][0]

for t in T:
    for l in L:
         for k in L:
            prob+=lb[t][l][k]>=0
            prob+=mu[t][l][k]>=0


     #for t in T:
        #for l in L:
                #prob+= a[t][l]*b[l]>=I[t]

for t in T:
    if t>0:
```

```
        b=t-1
        prob+= I[t]==I[b]+v[b]-w[b]

prob+= I[Horizon-1]>=I[0]

prob+= I[0]==0

prob.solve(GUROBI())

print(rho, prob.objective.value())

resultsbook=xlsxwriter.Workbook('Results.xlsx')

worksheet=resultsbook.add_worksheet()

for t in T:
    worksheet.write(0,t,value(v[t]))
    worksheet.write(1,t,value(w[t]))
    worksheet.write(2,t,value(I[t]))

resultsbook.close()
sys.exit()
#for v in prob.variables():
   #if v.varValue>0:
        #print(v.name, "=", v.varValue)
#ff=0
#for name, c in list(prob.constraints.items()):
        #ff=ff+1

#print (ff)

print("--- %s seconds ---" % (time.time() - start_time))
```

Appendix F

Base Cash Model Code

```
import time
start_time = time.time()

from pulp import *

from openpyxl import load_workbook

import xlsxwriter

wb=load_workbook(filename= 'Book1.xlsx')

sheet_ranges=wb['Sheet1']

ws=wb['Sheet1']

Horizon=214

scenario=24

T= list(range(Horizon))

S= list(range(scenario))

#If using scenarios#
#p=[[0 for s in S] for t in T]
#for t in T:
```

```
    #for s in S:

        #p[t][s]=ws.cell(row=s+2, column=t+2).value
#IF using average#
p=[0 for t in T]

for t in T:
    p[t]=ws.cell(row=26, column=t+2).value

print (p)

Cap=1000000

M=Cap

cin=20000

cout=40000

a=.01

delta=.01


prob= LpProblem("EL-NR", LpMaximize)


v = LpVariable.matrix("v", (T),0, None, LpContinuous)
w = LpVariable.matrix("w", (T),0, None, LpContinuous)
wp = LpVariable.matrix("wp", (T),0, None, LpContinuous)
```

```
I = LpVariable.matrix("I", (T),0, None, LpContinuous)
y = LpVariable.matrix("y", (T) ,0, 1, LpBinary)
z = LpVariable.matrix("z", (T), 0, 1, LpBinary)

prob+= lpSum([p[t]*(-((1+a)*v[t])+((1-a)*wp[t])) for t in T])

for t in T:

    prob+= v[t]<=cin

    prob+= w[t]<=cout

    prob+=  wp[t]==(1-delta)*w[t]

    prob+= w[t]<=I[t]

    prob+= w[t]<=M*z[t]

    prob+= v[t]<=M*(1-z[t])

    prob+= I[t]<=Cap

for t in T:
    if t>0:
        b=t-1
        prob+= I[t]==I[b]+((1-delta)*v[b])-w[b]

prob+= I[213]==1000000

prob+= w[213]==0

prob+= v[213]==0
```

```
for t in T:
    if t>145:
        prob+= I[t]>=500000

prob+= I[0]==0

prob.solve()

print("Max profit:", prob.objective.value())

#for vv in prob.variables():
    #if vv.varValue>0:
        #print(vv.name, "=", vv.varValue)


resultsbook=xlsxwriter.Workbook('Results.xlsx')

worksheet=resultsbook.add_worksheet()

for t in T:
    worksheet.write(0,t,value(v[t]))
    worksheet.write(1,t,value(w[t]))
    worksheet.write(2,t,value(I[t]))

resultsbook.close()
sys.exit()

print("--- %s seconds ---" % (time.time() - start_time))
```

Appendix G

DRO Cash Model Code

```
import time
start_time = time.time()

from pulp import *

from openpyxl import load_workbook

import xlsxwriter

wb=load_workbook(filename='Book1.xlsx')

sheet_ranges=wb['Sheet1']

ws=wb['Sheet1']

Horizon=214

scenario=24

T= list(range(Horizon))

S= list(range(scenario))

#If using scenarios#
p=[[0 for s in S] for t in T]
var=[0 for t in T]
```

```
for t in T:

    var[t]=ws.cell(row=30,column=t+2).value
    for s in S:
        p[t][s]=ws.cell(row=s+2, column=t+2).value
#IF using average#
#p=[0 for t in T]

#for t in T:
    #p[t]=ws.cell(row=26, column=t+2).value

print (p)

q=[(1/24) for s in S]

Cap=1000000

M=Cap

cin=20000

cout=40000

a=.01

delta=.01

rho=1

prob= LpProblem("EL-NR", LpMaximize)
```

```
v = LpVariable.matrix("v", (T),0, None, LpContinuous)
w = LpVariable.matrix("w", (T),0, None, LpContinuous)
wp = LpVariable.matrix("wp", (T),0, None, LpContinuous)
pi=LpVariable("pi", 0, None, LpContinuous)
theta=LpVariable("theta", None, None, LpContinuous)
psip=LpVariable.matrix("psip", (S), 0, None, LpContinuous)
psim=LpVariable.matrix("psim", (S), 0, None, LpContinuous)
I = LpVariable.matrix("I", (T),0, None, LpContinuous)
y = LpVariable.matrix("y", (T) ,0, 1, LpBinary)
z = LpVariable.matrix("z", (T), 0, 1, LpBinary)

prob+= theta - rho*pi + lpSum([psip[s]*q[s] for s in S])
- lpSum([psim[s]*q[s] for s in S])

for s in S:

    prob+= theta <= lpSum([p[t][s]*(-((1+a)*v[t])+
    ((1-a)*wp[t])) for t in T])-psip[s]+psim[s]

    prob+= psip[s]+psim[s] <= pi

for t in T:

    prob+= v[t]<=cin

    prob+= w[t]<=cout

    prob+=  wp[t]==(1-delta)*w[t]

    prob+= w[t]<=I[t]
```

```
    prob+= w[t]<=M*z[t]

    prob+= v[t]<=M*(1-z[t])

    prob+= I[t]<=Cap

for t in T:
    if t>0:
        b=t-1
        prob+= I[t]==I[b]+((1-delta)*v[b])-w[b]

prob+= I[213]==1000000

prob+= w[213]==0

prob+= v[213]==0

#for t in T:
    #if t>145:
        #prob+= I[t]>=500000

prob+= I[0]==0

prob.solve()

totalvar=0

for t in T:
    totalvar=totalvar+(((value(v[t])+value(w[t]))**2)*var[t])

print("Max profit:", prob.objective.value(),totalvar)
```

```
#for vv in prob.variables():
    #if vv.varValue>0:
        #print(vv.name, "=", vv.varValue)


resultsbook=xlsxwriter.Workbook('Results.xlsx')

worksheet=resultsbook.add_worksheet()
for t in T:
    worksheet.write(0,t,value(v[t]))
    worksheet.write(1,t,value(w[t]))
    worksheet.write(2,t,value(I[t]))

resultsbook.close()
sys.exit()

print("--- %s seconds ---" % (time.time() - start_time))
```

Appendix H

Robust Cash Model Code

```
import time
start_time = time.time()

from pulp import *

from openpyxl import load_workbook

import xlsxwriter

wb=load_workbook(filename= 'DailyData.xlsx')

sheet_ranges=wb['Sheet2']

ws=wb['Sheet2']

Horizon=214

scenario=24

T= list(range(Horizon))

S= list(range(scenario))

#If using scenarios#
#p=[[0 for s in S] for t in T]
#for t in T:
```

```
    #for s in S:

        #p[t][s]=ws.cell(row=s+2, column=t+2).value
#IF using average#
p=[0 for t in T]
pmin=[0 for t in T]
pmax=[0 for t in T]

for t in T:
    p[t]=ws.cell(row=26, column=t+2).value
    pmin[t]=ws.cell(row=27, column=t+2).value
    pmax[t]=ws.cell(row=28, column=t+2).value

print (p)



Cap=1000000

M=Cap

cin=20000

cout=40000

a=.01

delta=.01
```

```
prob= LpProblem("EL-NR", LpMaximize)


v = LpVariable.matrix("v", (T),0, None, LpContinuous)
w = LpVariable.matrix("w", (T),0, None, LpContinuous)
wp = LpVariable.matrix("wp", (T),0, None, LpContinuous)
pi= LpVariable.matrix("pi", (T), 0, None, LpContinuous)
lb= LpVariable.matrix("lb", (T), None, 0, LpContinuous)
I = LpVariable.matrix("I", (T),0, None, LpContinuous)
y = LpVariable.matrix("y", (T) ,0, 1, LpBinary)
z = LpVariable.matrix("z", (T), 0, 1, LpBinary)

prob+= lpSum([pmin[t]*pi[t]+ pmax[t]*lb[t] for t in T])

for t in T:

    prob+= v[t]<=cin

    prob+= w[t]<=cout

    prob+= pi[t]+lb[t]==(-((1+a)*v[t])+((1-a)*wp[t]))

    prob+=  wp[t]==(1-delta)*w[t]

    prob+= w[t]<=I[t]

    prob+= w[t]<=M*z[t]

    prob+= v[t]<=M*(1-z[t])

    prob+= I[t]<=Cap
```

```
for t in T:
    if t>0:
        b=t-1
        prob+= I[t]==I[b]+((1-delta)*v[b])-w[b]

prob+= I[213]==1000000

prob+= w[213]==0

prob+= v[213]==0

#for t in T:
    #if t>145:
        #prob+= I[t]>=500000

prob+= I[0]==0

prob.solve()

print("Max profit:", prob.objective.value())

#for vv in prob.variables():
    #if vv.varValue>0:
        #print(vv.name, "=", vv.varValue)


resultsbook=xlsxwriter.Workbook('Results.xlsx')

worksheet=resultsbook.add_worksheet()

for t in T:
    worksheet.write(0,t,value(v[t]))
```

```
    worksheet.write(1,t,value(w[t]))
    worksheet.write(2,t,value(I[t]))

resultsbook.close()
sys.exit()

print("--- %s seconds ---" % (time.time() - start_time))
```

Appendix I

Parameters Used in Utility Functions

Utility Fill Parameter Values	
Approach 1	K=15152.52
Approach 2	K=7575.758
Approach 3	K=1362.221
Approach 4	K=681.1106
Approach 5	K=1000000
Approach 6	K=50681.1