Natural Gas Imbalance System Optimization

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY
CAMPUS MONTERREY

SCHOOL OF ENGINEERING
ENGINEERING GRADUATE PROGRAM

"Natural Gas Imbalance System Optimization"

PRESENTED AS PARTIAL FULFILLMENT FOR THE DEGREE OF:
DOCTOR IN ENGINEERING SCIENCES

PRESENTED BY:
GERARDO ALFREDO PÉREZ VALDÉS

MONTERREY, N.L. DECEMBER, 2009

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY
CAMPUS MONTERREY

SCHOOL OF ENGINEERING
ENGINEERING GRADUATE PROGRAM

Committee members recommend that the research project presented by Gerardo Alfredo Pérez Valdés be accepted as a partial requirement to obtain the academic degree:

Doctor in Engineering Sciences, Major in Industrial Engineering

Dissertation committee:

Dr. Vyacheslav V. Kalashnikov (Chair of the Committee)	Dr. Timothy I. Matis (Co-chair of the Committee)
Dr. Neale R. Smith	Dr. Milton L. Smith
Dr. Francisco R. Ángel-Bello	Dr. James L. Simonton

Dr. Alex Elías Zuñiga
Director
Doctorado en Ciencias de Ingeniería
December, 2009.

Natural Gas Imbalance System Optimization

Advisor: Vyachelav V. Kalashnikov

Advisor: Timothy I. Matis

Committee Members:

Milton L. Smith

Francisco R. Ángel-Bello

James L. Simonton

Neale R. Smith

Natural Gas Imbalance System Optimization

By
Gerardo Alfredo Pérez Valdés

Submitted to the School of Engineering on December, 2009,
in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Engineering Sciences

ACKNOWLEDGMENTS

First of all, my love and thanks to my parents, for all their support throughout all the years and trials in my life.

Thanks to my dear brothers, Arturo and David, for enduring me in everyday life; to my advisors, Slava Kalashnikov and Tim Matis, and all the other members of my committee, for their commitment to my development; to Marycarmen and Lalo, for life at lubbock would have been very difficult without the two of you; to Ceci, Krystel, Iván, Rocío, Jania, and Lupita, for their friendship and kindness; to all my fellow students from Monterrey and Lubbock, you rock guys!; and finally, to my professors, friends, and relatives, for everything you have given me, small and large things alike.

ABSTRACT

Following recent changes in the legislations regarding the natural gas supply chain, American and European firms found themselves facing numerous new problems. Particularly, the way in which a pipeline operator interacts with the natural gas shippers was affected. The new setting makes it more complicated for the pipeline operator to correct imbalances in the transportation system. In this work, we model the part of the supply chain that concerns the pipeline-shipper interactions, focusing on the cash-out penalization that is charged by the former when the latter creates imbalances. A series of mathematical optimization models, both deterministic and stochastic, are created to increasingly refine the abstraction of the system. Additionally, statistical support is provided to support the validity and usage of the data in the models. Several problem instances were computationally tested for each of the concepts introduced, and the evidence found is then used to define a decision making framework that contributes to the shipping company day to day decision process.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

In 1992 in the United States, and in 1995 in the European Union, a number of regulations were issued by the respective governmental institutions with the aim of effectively separating the control over several processes that formed the Natural Gas Supply Chain. The resulting market conditions are in favor of the independence of the transportation and the commercialization processes. As a result of this paradigm shift, and the subsequent restructuring of the market, the natural gas industry around the world now faces several issues in need of a systemic analysis.

Particularly interesting is a problem that results from *balancing* the fuel volumes over a distribution network. Such procedure is a direct concern for the Pipeline Operating Company (POC) in charge of the pipeline, since a certain level of control of the fuel volumes is required for the correct operation of the pipeline. Moreover, any Natural Gas Shipping Company (NGSC) is also concerned with the balance of the volumes because of the existence of legal impositions to avoid the creation of *imbalances* by them. Naturally, the business of a NGSC is to sell natural gas—moving it through the pipeline—to its clients: it has to fulfill contracted demands and then market any excess of fuel to achieve a maximum profit. In order to maximize its profits, the NGSC has to manage the volumes at every selling point, considering both balance, and selling costs and revenues.[1]

The situation the POC and NGSC face regarding pipeline balance can be modeled as a mixed integer nonlinear stochastic bilevel optimization problem, in which the NGSC point of view is modeled as one (upper) level of the problem, while the POC is modeled as the other (lower) level. The theoretical models and the experimental results derived from them are intended to be a framework and managing tool to help these actors of the natural gas supply chain. One particular goal is aiding the NGSCs in their day-to-day volume trade planning.

In the next sections of this chapter, we will elaborate in the history and nature of the natural gas market. Section 1.1 explains the development of natural gas as a common fuel, and section 1.2 describes the regulations that brought about the

[1] Taken from the author's [54].

paradigm shift and caused the problematic situation we analyze. Section 1.3 details the POC-NGSC system, its components and dynamics. Finally, section 1.4 defines the statement of the problematic situation as we see it, along with the assumptions needed in our analysis.

1.1 Overview of Natural Gas as a Fuel

Most of the natural gas we draw from the ground is hundreds of years old. As a fossil fuel, it has formed (there is no conclusive theory about this) by a combination of biological decomposition, ground pressure, and elevated temperatures. Natural gas is found along with oil in underground wells, with varying temperature levels determining the exact proportion of oil and gas formed. A type of natural gas, called *Biogenic Methane*, is produced by microorganisms decomposing living matter, though the economical relevance of this type of natural gas is barely emerging.[78].

The chemical composition of natural gas varies. It is not an actual chemical compound, but a mixture of organic gases formed mainly of methane, but also containing ethane, propane, butane, ethane, etc. (Table 1.1.)

Table 1.1. Typical Composition of Natural Gas. [78]

Methane	CH_4	70-90%
Ethane Propane Butane	C_2H_6 C_3H_8 C_4H_{10}	0-20%
Carbon Dioxide	CO_2	0-8%
Oxygen	O_2	0-0.2%
Nitrogen	N_2	0-5%
Hydrogen sulfide	H_2S	0-5%
Rare Gases	A, He, Ne, Xe	*traces*

The density of natural gas is lower than that of air, so it dissipates in the event of a leak. This is one advantage over other fuels, for it adds a security factor to end-users. Another benefit is the fact that natural gas is considered a *clean fuel*: it produces little to no harmful gases when burnt. At the time it is delivered to end users, natural gas is mostly methane, already purified in a process called *drying*, and which consists in the removal of most hydrocarbons other than CH_4. In this state,

as opposed to most fossils fuels, natural gas burns cleanly, that is methane combustion leaves water vapor and CO_2 as residuals:

$$CH_4 + 2O_2 \rightarrow 2H_2O + CO_2.$$

Natural Gas has a Higher Heating Value of 39.3MJ/m^3, and a Lower Heating Value of 34.6MJ/m^3. [9]

1.2 Natural Gas Market

Natural gas is considered the fossil fuel of the twenty-first century, as oil and carbon were considered the fossil fuels of the 20th and 19th centuries, respectively. In fact, natural gas is the fossil fuel with the fastest economical growing rate since the 1970s. [99, 28]

Nowadays, natural gas is the second fuel source after oil, having 20.7% of the world's energy production in the year 2005. As of the year 2006, 65.7% of the world's natural gas is produced at the countries belonging to the OECD (Australia, Austria, Belgium, Canada, the Czech Republic, Denmark, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Japan, South Korea, Luxembourg, Mexico, the Netherlands, New Zealand, Norway, Poland, Portugal, the Slovak Republic, Spain, Sweden, Switzerland, Turkey, the United Kingdom, and the United States,) and the former USSR (Armenia, Azerbaijan, Belarus, Estonia, Georgia, Kazakhstan, Kyrgyzstan, Latvia, Lithuania, Moldova, Russia, Tajikistan, Turkmenistan, Ukraine and Uzbekistan); see figure 1.1 for details. The biggest individual producer and exporter is Russia, while the United States is the world's largest importer, as shown in table 1.2.) [49]

Natural gas also contributes with a sizable part of electricity generation, with a share of 19.7% in 2005 (combined with gas-works gas.) Above half of the total natural gas world production is used as energy provider for the commercial, public services, agriculture and residential sectors, while the other half is employed in transportation, industry and non-energy related sectors.

According o the CIA World Factbook, an estimate using 2005 data for the world's production in 2007 was 2.854×10^12m^3, (producing between 98.74×10^{18}J and

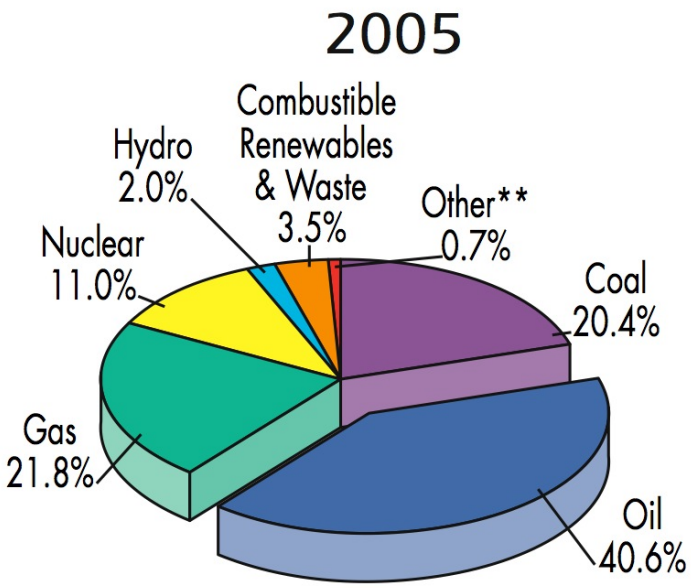


Figure 1.1. 2005 Fuel Shares of Total Energy Supply. ©OECD/EIA, 2007. *Key World Energy statistics*

112.16×10^{18}J. On the other hand, the estimated consumption would be of 3×10^{12}m^3 (between 103.8×10^{18}J and 117.9×10^{18}J.) [18]

1.2.1 Early History of Natural Gas Market

The earlier Greeks had already been in contact with natural gas wells, which they considered divine because of their apparently perpetual flames, actually a stream of

Table 1.2. Largest Producers, Importers and Producers of Natural Gas in 2006.

Rank	Role		
	Producer	Exporter	Importer
1st	Russia	Russia	United States
2nd	United States	Canada	Germany
3rd	Canada	Norway	Japan
4th	Iran	Algeria	Italy
5th	Norway	Netherlands	Ukraine

gas escaping from the underground deposits, burning due to either a natural or purposeful igniting agent. While the early cultures of Europe and Asia Minor found these occurrences of natural gas odd and even feared them, the Chinese, around 500 BC, were more practical and developed ways to transport the escaping gas to their urban centers, effectively building the very first pipelines.

The first commercial use of natural gas happened in England around 1785. This form of gas was produced from coal, not extracted from wells. During the 18th and the 19th centuries, the Americans dug wells specifically intended to obtain natural gas, which was chiefly used for illumination purposes at the time. With the invention of the Bunsen burner (in 1885,) and the development of the first large-scale pipelines (around 1891,) the American Natural Gas Industry started the process of turning natural gas into one of the main sources of energy for the country. Reliable long-distance pipelines, though, did not appear in the US until the 1920s.

By 1938, the United States Government first regulated the natural gas industry, believing that a controlled environment was necessary for the product. This action, known as the *Natural Gas Act of 1938*, was used as the main guideline for establishing and operating the Federal Energy Regulatory Commission (FERC.) [90] However, by the early second half of the 20th century, several administrative problems prompted the authorities to reconsider the way they had envisioned the market conditions for this commodity.[79, 99]

1.2.2 Reforming Regulations of the Market

Let us imagine a little town whose main source of heating is natural gas. Because of the relatively small size of the local market there, there is no reason for more than one company to deploy pipelines to service the town: whatever the gas needs of this town are, they can be satisfied with a single supplier. This pipeline is operated by what we call a Pipeline Operating Company, whose main business activity is to transport natural gas towards and within the town. Depending on the number of Natural Gas Shipping Companies around, we might have one of two cases: (1) There are NGSCs wishing to gain access the town's market and are required to use the existing pipeline—controlled by the POC; or (2) no NGSCs are

interested in entering the local market, thus the availability of natural gas is completely dependent on the decision of the POC.

Case (1), from a free-market point of view, seems adequate, as it generates competence and consequently benefits for the clients. What would happen, however, if the same POC that operates the only pipeline in town were also one of the competing NGSC? The main business activity of the POC may still be that of managing the pipeline, but it can also obtain revenue from marketing gas, just like the other NGSC. Here, it is difficult to think of fair competition, as the POC has the ability to dictate conditions of operation of the pipeline that favor its role as NGSC. Case (1) could then "degenerate" into case (2), with all the economic impacts that such situation implies. By the end of the twentieth century, governments over the world realized this situation and started to issue regulatory measures to deal with it.

Because of this tendency, competitive service and free-market approaches have driven several changes in international regulations of natural gas in the last twenty years. [80] From a business point of view, the natural gas industry in the world is going from being heavily regulated into a situation where it is more "market based." [29, 99]

Nevertheless, the Natural Gas Act of 1938 and its subsequent modifications failed to prevent the existence of situations like the one exposed in at the beginning of this section. The problem was not overlooked [80], and by 1992, the FERC issued Order 636, also known as "The Restructuring Rule" [26]. This obligated the POCs to separate their marketing and transporting operations and either sell their shipping-related assets to existing NGSCs, or forming affiliate—yet independent—companies under specific rules [27]. Similar situations were faced by the European operators under *Regulation (EC) 1775* [48]. Before FERC Order 636, POCs purchased the natural gas that they transported [29], hence, they had a large amount of influence on prices and the usage of their equipment and facilities, i.e., it was a market in which they had the last word regarding virtually any operation. Today, and for the most part, Natural Gas Shippers and Suppliers have more freedom to produce and commercialize the commodity. [89, 106, 21]

The companies involved in the shipping sector had to undergo deep structural changes in order to face the new policies. Particularly, we have witnessed a

mandatory division of the gathering, marketing and transmission operations, forcing the companies involved to accommodate to the new conditions. In terms of economic landscapes, there has been a change in the paradigm where companies are finding new business opportunities. Middle- and end-users have been granted the benefit to choose their service providers so that they better suit their needs.

1.2.3 Impact of the Reform

Both the Restructuring Rule and Regulation (EC) 1775 have had deep impacts in the way natural gas markets are now constituted and, according to the American Gas Association, had also produced efficiencies in it [90]. Current supply dynamics and issues, as well as future tendencies appear thoroughly described in [96].

In the next few sections, we give a brief overview of the conditions that constitute this paradigm.

Pipeline Operating Companies

As it often occurs when a company is forced to change its practices in benefit of fair competition, the POCs have turned out the most affected of all the participants in this sector.

- Liker it was discussed before, POCs were effectively forced to cease any marketing operation, which led to diminished earnings.
- Resulting from this, they have had to redesign their transportation prices to cover their operating costs in ways other than marketing gas [26].
- POCs now have problems regarding the maintenance of the technically required volumes of natural gas in their facilities [92]. This was easier to perform when they were allowed gathering operations and unregulated control of their pipeline.
- The construction of "hub centers" to interconnect different pipelines has been indirectly encouraged, in order to allow for buying/selling operations from third parties to go smoothly [29].

- In some cases, though, reimbursement strategies had to be implemented to alleviate the transition costs for the POCs [26].

POCs were given certain privileges to cope with these imposed restrictions, such as proposing rates to penalizing the creation of imbalances. [90].

Natural Gas Shipping Companies

Since the regulations described above were in part devised to open the market to shipping companies, a number of advantages were granted to them:

- NGSCs now compete in a market more independent of the influence of other companies, and can target specific areas of such market to improve productivity.

- New companies can now afford the logistical and economical ventures of entering the natural gas market as NGSCs.

- Such new companies have been allowed to increase their natural gas production, relieving some of the pressure to meet demand [29].

- Smaller companies that had already made investments in the energy market have now a greater chance to succeed in business [48].

- Nevertheless, the new restrictions that POCs experience have hindered certain NGSCs operations in some instances, like the availability of no-notice service. [100]

- In the EU, the advent of Regulation (EC) 1775/2007 has made some NGSCs nervous about a possible emergency of POCs with distributing capabilities, which is seen as a drawback in the policy. [37]

Regulating Agencies

Governments in the world have also experienced a number of changes associated with the new paradigms.

- Regulating agencies had to be formed, and continuous pondering and amendments are required to keep the regulations they are to uphold up to date and providing useful, reasonable restrictions.

- In the US, the FERC has now direct control only over the POCs and Local Distribution Companies. This means, among other things, that the prices NGCSs charge do not depend directly on regulatory agencies. [79]

- In order to verify the activities of the NGSCs, new institutions have been created to guide [31] and monitor [10] the general development of the industry.

- Courts have been called to solve legal cases because many litigations now belong to different sectors and involve different companies [100].

- Tax reforms need to be made to allow the POC to conduct their business without losing money. [92]

Consumers and Local Distribution Companies

- End-users and local distribution companies have now a wider range of options to choose from when contracting a service, and are no longer subject to the desires of a POC.

- New companies are able to enter the market and satisfy the increasing demand for fuel and energy.

- The proposals are aimed at providing choice, fair prices, cleaner energy and secure supply to this sector [48].

- The amount of natural gas transported by interruptible methods (such as shipping trucks) have decreased because of the more accessible pipelines. [86]

- The efficiency of the infrastructure of the sector has reportedly increased. [26]

1.3 The Shipper-Pipeline System

As we have stated above, the main objective of this work is to obtain a theoretical model and decision tools for the NGSCs and other actors of the natural gas supply

chain, so that they can more easily operate in the context of an imbalanced market. Other aspects of the supply chain, such as production, marketing, contracting, refining, etc., are described in section 1.3.1. They are, however, not directly relevant to the process herein analyzed and are presented only for providing a background to the study. The models developed in the next chapters will cover the balance process of the network (by) transporting fuel from one zone to another, as well as the economical information needed to accurately assess the optimal solution rending the safest, most profitable way of action of the NGSC.

We will deal here only with a single NGSC-POC system i.e. there is no more than one NGSC involved, so that the POC does not have the option to auction imbalances (though certain real-world system hold these as possible.)

1.3.1 Natural Gas Supply Chain

The natural gas supply chain consists of the following stages: Exploration, Extraction, Processing, Transportation, Storage and Distribution. In practice, the first three are typically carried on by the same company, as well as the last two. This is not mandatory, nor is it always the case: there are firms dedicated exclusively to processing, or to storage. Given the amount of possible economies of scale, it is desirable to take such multiple roles. Transportation firms, however, have been banned access to other links of the chain, either because of the importance of their operations or because of the acceptance of "naturally monopolistic" nature of their activities. [81]

Additionally, after the 1990s regulations, there has been an emergence of financial markets of many kinds related to natural gas, adding a fifth link—marketing—to the natural gas supply chain. A simple yet comprehensive description of the natural gas supply chain and its associated processes can be found in [25].

A schematic view of the natural gas supply chain as it is described below can be seen in figure 1.2

Exploration, Extraction, and Processing

Exploration is the processes of surveying both continental and offshore potential sources, using seismological tools like magnetometers and gravimeters to determine

the existence, size and viability of a Natural Gas extraction point. Computer simulations in 2D, 3D and 4D; and data analysis tools, aid today's geologist in this stage. In the US, most natural gas basins are located in a corridor having Wyoming in one end and Texas, Louisiana and Alabama in the other. [97]

Once the feasibility of digging a new well has been validated, a group of geologists and geophysicists determine the ideal conditions for the placements of the drilling crews and equipment, prior to the actual extraction that is made using different kinds of equipment for both onshore and offshore extraction.

Immediately after extraction, natural gas is considered *wet*. This variant of natural gas is loaded with Natural Gas Liquids (NGLs,) which are those hydrocarbons other than methane (usually, but not uniquely, ethane, propane, butane and pentane,) that render the gas inappropriate for usage as fuel. Wet natural gas is transported from the well to processing facilities in what is called the *gathering system*, a pipeline network of low pressure, low diameter ducts that are typically owned by one of the companies involved in the preliminary steps of the supply chain. Once in the processing plant, the NGLs are removed from the natural gas in a process called *drying*.

Dry gas is required not just by the end-users (for reasons of quality,) but also by POCs. Pipeline operators usually impose restrictions regarding the "quality" of the natural gas that is injected into their ducts, demanding specific parameters of composition, temperature and BTU content. Once introduced into the pipeline, natural gas is mixed with the inputs of other companies, so a certain quality of gas is necessary to safeguard the interests of all the NGSC using the same pipeline. [97]

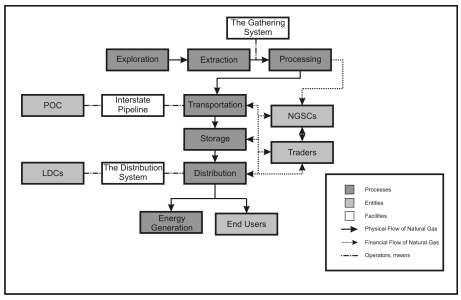


Figure 1.2. The Natural Gas Supply Chain (compare to that in [25])

Transportation

This stage contains the different methods through which natural gas is moved among the several facilities that compose the supply chain. This process is usually carried on in pipelines with the aid of compressing and storing equipment. Pipelines are conduits used to mode natural gas from the wells to the processing plants (the gathering system,) from the processing plants to the distribution centers (or *pool zones*) across the nation (the interstate/intrastate pipeline system,) and from the distribution centers to its consuming points (the distribution system.) Pipelines are owned and operated by Pipeline Operating Companies, whom by the end of the 1990s are effectively prohibited from participating in most other natural gas processes in both Europe. There are about 80 major interstate pipelines in the US, with 20 percent of them servicing the 80% of the market supply.

Both the gathering and the distribution systems are usually owned by the companies that primarily employ them (extraction or processing plants in the first case; Local Distribution Companies in the second,) so these systems are not technically considered part of the transportation chain link. In this work, we consider a POC as the operator of an interstate pipeline. The segmentation of the natural gas market (i.e., the creation of submarkets within localized regions,) and the consequent formation of long-distance, intrastate pipelines, creates management situations where even inside a country, export and import phenomena occur [45]. The pooled regression analysis shown later in the stochastic models chapter of this work is in part a response to this market segmentation. It aims to providing tools for identifying local markets behaving differently with respect to their consumption/price econometrics.

Pipeline Operating Companies usually have high fixed costs, derived from the inherent pipeline construction. Furthermore, these costs are considered sunk since the pipeline has little to none alternative usage. Once installed, however, the maintenance and operation costs of a pipeline are relatively low. Another economical advantage for the POC is the fact that, except for exceptionally large markets, a single pipeline is more than enough to satisfy the necessities of a given location. Because of all this, (interstate) pipelines are accepted as being in a naturally monopolistic market. [51]

Storage

A few years ago, natural gas was used almost exclusively as heating fuel, and as such, it was a strongly seasonal commodity stored for consumer peak seasons. With the employment of natural gas in electricity generation, and with the regulations like FERG Order 636 and 637, storage nowadays serves a purpose of commercial reasons: natural gas is now a market driven commodity in need of storage that allows for a lucrative market managing.

The general denomination for a facility where natural gas is stored for its subsequent delivery to selling points is that of *Pool Zone*. Pool zone may also refer to the temporary storage points where natural gas is kept by the processors before being sold or shipped.

Pool zones are very important points of activity to Natural Gas Shipping Companies, who act as intermediaries (financial-wise only, not physically) between Processing and Local Distribution Companies. A NGSC arranges for the injection and extraction of natural gas into/from the pipeline. It buys gas from the processing companies and then sells it to the LDCs, accommodating volumes across different pool zones.

An extensive study of natural gas storage policies and uses and their impact in the natural gas market can be found in [19]. There is explained how storage is arguably the most important factor in the flexibility of the natural gas supply chain.

Distribution

This is the final stage of the supply chain. Once the gas has been delivered into a pool zone for its storage, it can be forwarded to the various costumers by a Local Distributing Company. The LDCs are in charge of the delivery of the fuel to the end-users, and can be public (governmental,) or owned by private investors. More often than not, LDCs take ownership of the natural gas after it leaves the pool zones, ending the involvement of any prior actor, like the NGSCs. [81].

In large communities, several LDCs can share a market, though usually LDCs enjoy exclusivity in natural gas supply in their region, creating a monopolistic market similar to that of the interstate pipelines. For example, Natural Gas Mexico is the sole distributor of natural gas in the city of Monterrey, Mexico. Its

competitors offer other heating alternatives considered substitutes to natural gas, such as LP gas. An increasing number of countries are pushing for open-access regulations to LDCs, similar to what they already have for the POC and the interstate pipelines.[51]

Marketing

Since the beginning of the modern reformed market, many companies find it economically feasible to develop natural gas marketing, selling and buying operations in a financial environment. Although not a supply chain link *per se*, market operators have a large amount of influence in the actions of the other actors.

There exist *physical gas* markets and *financial gas* markets. Physical gas markets function with physical gas contracts to deliver actual fuel from one firm to another during a stipulated time, in either wholesale or retail formats. The volumes of gas traded in such spot markets grew rapidly since the advent of the open-access transportation and the liberalizing reforms. [25]

Financial gas markets are used for managing price risk and not necessarily for physical delivery. They are highly volatile markets with financial institutions as intermediates for the risk-transferring schemes. The most common forms of financial markets for natural gas are *forwards* (single time payment future sale contracts,) *swaps* (risk-based payment management,) *futures* (commodity supply management,) and *options* (future price arrangement.)

Additionally, some POCs have developed Transportation Markets, which are based in the purchase and sale of pipeline prerogatives among the different NGSCs they serve. [75] The *Primary Transportation Market* uses Transportation Contracts, which may be firm (they give the NGSC rights over capacity and transportation regardless of the season) or interruptible (the POC determines the schedule of the deliveries depending of pipeline capacity.) Hybrid versions of these two kinds of contracts also exist.

In certain cases *Secondary Transportation Markets* may arise. When NGSCs (or any other holder of transportation contract rights) are legally able to do so, they can sell of their pipeline access to other firms: capacity is then used as a financial object to bid for in this market. Once such transaction is completed, the contract

holders notify the POC about this change of capacity rights. Certain POCs, however, do not concede property rights to their client NGSCs, negating the (direct) existence of this secondary market.[51]

1.3.2 The NGSC-POC Subsystem

Since the issuing of the Natural Gas Market regulations, a new and complex set of relationships between the transportation managers (POCs,) and the firms in charge of the trading, purchase and sale of the natural gas (NGSCs) were created. This subsystem of the larger natural gas supply chain is of special interest because of the many ways POCs and NGSCs have devised to cope with the liberalization of the natural gas market. This section refers solely to the components of this stage of the supply chain, to which we will refer as *the NGSC-POC (sub)system*: once the natural gas has been bought from the Processing Company by the NGSC, it is considered to have been *input into the NGSC-POC system*.

After leaving the processing plant, natural gas volumes will be bought by a NGSC, who will arrange for their delivery into several pool zones. The locations of these pool zones may span throughout a whole country, like the US, or a smaller region. This is done by contracting the services of a POC to transport the gas through its pipeline. The NGSC-POC contracts can vary in their duration, yet because of the relative inflexibility of the infrastructure attending a certain point, long-term contracts are usually made, where duration is considered an important factor. [84, 85]

Gas can typically flow in one physical direction (*upstream-downstream*, or left to right in figure 1.3.) The NGSC controls the financial flow of gas between the Processing Companies and the LDCs at the Pool Zones. The POC controls the physical flow according to its contracts with the NGSCs. Compare this scheme with that in page 12 in Juris, 1998b [52].

The gas bought from the Processing Plants is injected into the pipeline and mixed with the natural gas owned by other NGSCs that possibly use the same pipeline. Each NGSC then arranges for the extraction of the fuel and its storage in one of several pool zones downstream. The NGSC must specify, before the extraction occurs, the volume that is going to be withdrawn from the pipeline at every pool

zone, and the total extraction figures have to match (unless otherwise arranged with the POC) the total volume injected. At the pool zone, the NGSC sells the gas to the LDC(s) operating in the zone. These will deliver the fuel to the end-users trough their own systems.

The gas leaves the NGSC-POC subsystem when it is sold to the LDCs, that is, when it leaves its destination pool zone.

Operating Issues

In case that the NGSC extracts a volume of natural gas at a pool zone other than the volume stated in its contract with the POC, an *imbalance* is created. Positive imbalances occur when the NGSC has not extracted as much fuel as it has announced (thus there is more natural gas in a certain pool zone); negative imbalances, on the contrary, happen when the NGSC extracts more gas than that originally stated (leaving the pool zone with a lack of fuel.) POCs allow for certain imbalance tolerance levels, within which no balancing action is taken and the system is considered controlled (balanced.) If one or more imbalances exist outside the tolerance levels, we say that a *System Imbalance* occurs, or equivalently, that the system is imbalanced. System imbalances have repercussions in the financial aspect of the market: whoever caused the imbalance has to make up for it, either compensating the gas, paying a fine, and so forth.

System imbalances also endanger the functional state of the pipeline, which requires certain amounts of pressure to operate adequately. Achieving required

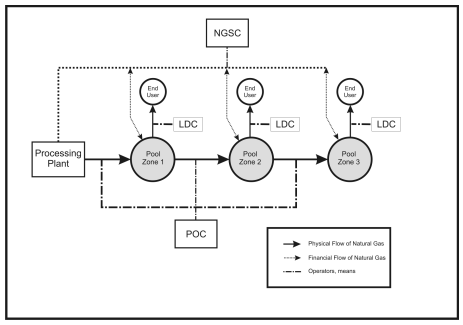


Figure 1.3. Flow of Natural Gas.

pressure levels is more difficult when the POC has no direct control over the volumes in the pipeline.

To remedy this, the POC uses practices known as *system balancing procedures*, which have the purpose of reducing imbalances in one or all pool zones. Balancing the system has a crucial role in the development of a competitive market, [32] since it provides flexibility to the NGSCs options to conduct a dynamic business .

Balancing the NGSC-POC system can be done in several ways, depending on the nature of the market and the ownership rights over the gas granted to the NGSCs. The POC can compensate positive and negative imbalances caused by the same NGSC in nearby zones, charging a fine to the involved parties for the operative expenses. Another example is injecting/extracting volumes from a positive/negative imbalanced zone into an alternate storage facility and charge purchase/sale/storage fees to the NGSCs. Usually, POCs are allowed to determine the nature and frequency of their balancing practice, as long as it adheres to the existent regulations [44], which normally rule over responsibilities, fairness, frequency, trading, costs, etc. [32]

In this work, since we are allowing a single-NGSC system and no additional storage of resources, we will consider that the POC deals with system imbalances by means of a *Cash-Out Penalizations* imposed to the NGSCs whenever they incur n imbalances.

The Balancing Process

During a (previously arranged) number of days, the NGSC can extract natural gas from the pipeline in each of the pool zones it attends. This *operating period* can be, for example, one month, two weeks, and so on. If it is economically advisable for it, the NGSC can create imbalances in the pool zones, as long as the volumes remain within the limits allowed by the POC in every pool zone and in the system in general. After the operating period is over, the POC will determine the best way to balance the current state of the system, moving gas downstream from the points where there are positive imbalances to the points where there are negative imbalances. Positive imbalances cannot be physically moved upstream to

compensate negative imbalances, so instead POC credits the NGSC with moved gas and somehow stores the additional volume elsewhere.

At the time this final day balancing process ends, it is required that all imbalances in every pool zone bear the same sign (i.e., all non-negative, or all non-positive.) If the final configuration consists of negative imbalances, the POC will charge a previously announced amount for the volume of gas that was extracted from the pipeline, at a price per volume that may be different at each pool zone. In this case, the POC is effectively selling gas to the NGSC. Since sales are not the actual business of the POC, its sale price is ordinarily higher than that of the Processing Companies: the NGSC is expected to have made good profits from the extra gas extracted so that this situation becomes economically feasible.

On the contrary, if the final configuration includes positive imbalances, then the POC acts as though buying the natural gas from the NGSC. Indeed, the POC should pay less than an end-user for whatever gas it is acquiring, for it was not its interest to buy fuel in the first place. Furthermore, since the pipeline is not allowed to sell natural gas to end users or LDCs, the only uses it might have for the purchased fuel are those concerned with its operating necessities (such as pumps fueling) and balancing methods.

Exact pricing of the imbalances is a major point of interest for the players of the supply chain: it defines the actions each participant will take when deciding upon injection, extraction, and balancing schemes. Sophisticated mathematical models involving specific industries are constantly being developed, for no balancing policy as of now is considered optimal [61].

1.3.3 Concerns for the Involved Parties

In the NGSC-POC system, both involved parties have a clear—and not always compatible —set of objectives. This difference causes the NGSC-POC system to be modeled with tools that allow for multi-objective or multi-level treatment.

Objectives of the NGSC

The NGSC is a firm whose main goal is to be able to sell the natural gas it owns to its clients, satisfying whatever contracts it has previously signed and then

marketing the exceeding volumes (if such exists) with secondary customers if the price they can afford is advantageous to the NGSC.

During the operating period, the NGSC may encounter that it can sell additional gas—at a given day and pool zone—than that originally planned at higher prices than those previously estimated. Creating a negative imbalance may then be advisable as long as the high prices compensate the risk for an imbalance penalization. The NGSC can possible compensate for this negative imbalances with positive imbalances in other pool zones in order to keep within allowed limits. This practice may continue during the entire operating period, eventually resulting in a *final day configuration*. The NGSC knows that the POC will rearrange this configuration (in the fashion described below) to obtain a more balanced *final configuration*; the rules are known to both parties at the beginning of the operating period. Depending on the nature of this configuration, the NGSC may be charged or paid a certain amount of money: if the amount charged is less than the revenue generated from the sale of the additional gas that created the imbalances in the first place, then the actions of the NGSC are economically feasible and it would have earned additional profits. (See table 1.3)

Creating imbalances, however, is an uncertain business, as the penalization occurs only after the operating period ends, and by that time the prices in each pool zone will likely have experienced changes. As the imbalances for any given day depend in some way on the imbalances the day before, it is necessary to have the most accurate forecast of future demands and prices in each pool zone so that good business decisions are made. Day-to-day prediction of prices and demands can be made considering several factors, such as the installed base of appliances, industry operation, temperatures, etc; different extents of the forecasting time window require equally different techniques and forecasting models[68]. Specifically, prices tend to have unpredictable behaviors when analyzed in the short run, with some authors using systemic analysis to determine the nature, causes and possible future occurrences of spikes in the natural gas price time series. [35]

Hence, the NGSC has two main objectives: (1) manage the daily imbalances in each pool zone so that it obtains the largest revenue, considering the stochastic nature of the prices and consumption of the market; and (2) finishing the operating

period with a final day configuration, derived from the daily imbalances, that will force the POC to charge it a low penalization compared to the revenue that caused the necessity of having imbalances.

Objectives for the POC

In contrast with the NGSC, the POC aims for a balanced system that will guarantee functionality for all the NGSCs using the facilities, while reducing the amount of money that is exchanged in its operations. Its main functions [52] are:

- Intake of the agreed volumes and qualities of gas at injection points
- Transportation through the (interstate) pipeline
- Extraction of gas of agreed volumes and quality at agreed delivery points
- Maintenance of a balance between injection/extraction
- Maintenance of a specific pressure
- Maintenance of a specific calorific value of gas

From our point of view, the most important interaction between the NGSC and the POC happens when the latter arranges the former final day configuration and transforms it into the final configuration. This is done with three objectives in mind: (1) to reduce the imbalance volumes in each pool zone (by compensating positive and negative imbalances); (2) to obtain an all-non-negative or all-non-positive imbalance configuration; and (3) to do so in such a way that the cash-out (payment/penalization) from/to the POC to/from the NGSC is the lowest possible.

Table 1.3. Costs and Earnings for the NGSC [57]

	Penalization from the POC	(positive or negative)
+	Revenue generated from contracted gas sales	(positive)
+	Revenue generated by additional gas sales	(positive)
+	Costs for reserving pipeline capacity	(negative)
+	Costs for unmet gas contracts	(negative)
=	Profit for the NGSC	(cost if negative)

Table 1.4. Costs and Earnings for the POC

	Revenue for moving gas downstream	(positive)
+	Cost for crediting gas upstream	(negative)
+	Revenue for having negative final configuration imbalances	(positive)
+	Revenue or cost for having positive final configuration imbalances	(positive or negative)
=	Profit for the POC	(cost if negative)

The balance period will then see the POC moving positive imbalances downstream (charging the NGSC for this) and crediting positive imbalances upstream (giving money to the NGSC.) If the resulting final configuration contains negative imbalances, the POC will make further charges to the NGSC; if the final configuration contains positive imbalances, the POC may pay or charge the NGSC. Both the POC and the NGSC are aware at all times of the parameters that determine the payments (figure 1.4.)

The survey of the variables involved in the pipeline operations is ultimately done by a human manager. Expert systems [47], like the one in [102], may provide reliable information to a pipeline manager with the objective of reducing the inherent human inconsistency and proneness to make mistakes, which have reportedly had large financial consequences [101].

Any positive (negative) profit for the POC is regarded as a cost (earning) for the NGSC in tables 1.3 and1.4. The only economical activity for the POC considered here is the balance penalization, whereas the NGSC engages in the sale of the natural gas, thus its objectives are different and include such activities.

We must stress the fact that whatever movements of gas the POC can make subject to the decisions the NGSC has already made and the final day configuration resulting from it. This makes the problem of maximizing the NGSC revenue not only a multi-objective optimization, but also a bilevel optimization problem, in which the NGSC will make a decision *knowing* which course of action the POC will take in response. The POC, however, has to contempt with the state of the system the NGSC has left, yet, the former has certainty in its information.

1.4 Problem Statement

It is the intention of this work to present a theoretical and numerical framework that will allow a single NGSC to maximize its profit (minimize its costs) by solving an optimization problem of the form

$$\begin{aligned} \text{minimize} \quad & f(x, y(x); P, D) && (1.1) \\ \text{subject to} \quad & x \in \mathbb{X}, && (1.2) \end{aligned}$$

where x is a vector containing the decision variables controlled by the NGSC, such as the amount of gas notified to the POC to be extracted, the amount of gas actually extracted, the imbalance created, etc; $\mathbb{X}$ is a set representing the physical and financial constraints x is subject to; $y = y(x)$ is a vector mapping of x which represents the response of the POC to the NGSC actions (see below); P and D are parameters regarding the price and consumption of natural gas, which may or may not be considered stochastic in nature. The real objective function f represents the cost the NGSC incurs, taking into consideration the penalization derived from creating imbalances [57], the amount of gas sold, the cost of reserving network capacity with the POC [75], and so on.

Vector y will be formed by variables that represent the possible actions of the POC once the NGSC has made its decision, i.e., the balancing process the POC has to undergo to reduce the total amount of imbalances in its system. This "response" concept leads to a **bilevel** optimization problem in which the upper level is described in (1.1)-(1.2), and the lower level corresponds to the minimization problem the POC solves:

$$\begin{aligned} \text{minimize} \quad & g(x, y(x)) && (1.3) \\ \text{subject to} \quad & y \in \mathbb{Y}, && (1.4) \end{aligned}$$

where the set $\mathbb{Y}$ represents the constraints the POC must adhere to when balancing the network. Function $g(x, y(x))$ is the objective of the POC, namely, it represents the cash-out between the parties to be minimized.

Conceptual Assumptions

The following assumptions are considered when trying to produce a suitable model of the form (1.1) - (1.2), (1.3)-(1.4)

Assumption 1. *There is only one NGSC company acting as the upper level decision maker, which will be referred as the leader. Equivalently, there is only one POC as the lower level decision maker, which will be called the follower.*

Assumption 2. *Both the POC and the NGSC act rationally, and their courses of action/decision making guidelines are known to each other.*

Assumption 3. *The operating period is finite and predefined, known to both leader and follower since the beginning of the process.*

Assumption 4. *The NGSC controls the extraction of natural gas, and consequently the creation of imbalances, in all the pool zones and in every day in the process.*

Assumption 5. *The mathematical function the POC uses to penalize the NGSC for producing imbalances in the system, $g(x, y)$, is known to both parties and remains unchanged through the operating period.*

Assumption 6. *The costs incurred by the NGSC when booking capacity in any pool zone are significantly lower than the sale price the NGSC can obtain in that zone.*

Assumption 7. *The NGSC will have a contracted demand in each pool zone it must satisfy. Not satisfying the contracted demand implies that the NGSC has to pay for every unit of unmet demand at a higher price than any sale price that could be obtained from selling gas an equivalent amount of gas in that zone.*

Assumption 8. *Any positive amount of gas remaining in a given pool zone and day after the contracted demand has been satisfied can be sold in that zone at a price typically higher than the regular sale price.*

Assumption 9. *Prices and consumptions, as seen by the NGSC, can be forecasted with relatively good certainty, i.e., there are not foreseeable changes in trend from the observed historical data for these figures.*

Assumption 10. *Every possible loss of gas (due to leaks, usage as pumps fuel, etc;) during the transportation process conducted by the POC is considered to be a fixed, deterministic percentage of the volume moved.*

Assumption 11. *Physical movement of gas throughout the pool zones is exclusively downstream: there is a starting pool zone (the one topologically closest to the well) and a final pool zone (the one at the end of the network.) In order to move gas from the starting zone to the last one, one would need to pump gas through every middle pool zone, and this direction cannot be physically reversed.*

Assumption 12. *The POC acts in such a way that any action it performs seeks to benefit both himself and the NGSC: any decision variables controlled by the POC are controlled so that they minimize the absolute loss for either party.*

Most of these assumptions underlie the models described in chapter 3; however, specific models may need to ignore one or the other, in which case it will be explicitly stated. Stochastic models in chapter 4 may also require additional assumptions.

Chapter Summary

In this chapter, we have talked about the qualities of natural gas as a fuel, its history, and its current market situation. The last twenty years have brought about a number of changes in the natural gas industry, and such changes have had deep repercussions in the natural gas supply chains, in both the EU and US. While the future of the natural gas market as it looks now can merely be speculated, several authors have proposed scenarios in which the occurrence/non-occurrence of certain important events may deeply affect the outcome of the natural gas market dynamics. [2]

The nature of the reforms and their impact in a particular subsystem of the supply chain are detailed, explaining in particular the relationship between Pipeline Operating Companies and the Natural Gas Shipping Companies. In this NGSC-POC (sub)system, the concept of imbalance appears as both a tool for the creation of dynamic markets, and as a problematic trait that the involved parties have to address properly. We state our problem as that of obtaining a decision

making tool whose purpose is to aid the NGSC to face the balancing method. This is done using bilevel mathematical programming; the fundamental nature and assumptions for the whole work are lastly detailed.

CHAPTER 2
LITERATURE REVIEW

This chapter offers references to models formulated to solve different kinds of problems found in the natural gas industry. Most of these models have resemblances to those we develop in this work as they all, in one way or another, use networks and time series to describe the subsystems of the natural gas supply chain.

There are many interests involved when studying natural gas. Looking at the entire, industry-wide system, one may want to model the optimal deployment of wells, refining facilities, local networks, interstate pipelines, etc. Once these are set, there is a need to optimize their operation parameters: production size and cost, pipeline throughput, pumping costs, equipment costs, etc. There is the economical aspect of the problem: the amount of fuel that is to be sold in the spot market, in the futures' market, in the capacity markets, etc. Since the problem we attempt to examine lies around the middle of the natural gas supply chain as we described it, we will focus, in this section, on the models that specifically deal about pipeline's network transportation models, price and demand management, and specially, multilevel formulation of the NGSC-POC subsystem.

Section 2.1 discusses the basic notions of natural gas network models used in the rest of the sections; it also presents the most basic network models found in literature. Section 2.2 deals with models that operate and optimize the network based in physical and operating costs, and section 2.3 presents models using risk management and forecasting tools when determining the parameters—and thus the decisions—to be made. Finally, the most important part of this chapter, section 2.4, displays several references to bilevel approaches to natural gas imbalance management, which are largely used to lay the foundations of chapters 3 and 4.

2.1 Basic Gas Network Models

Perhaps the most basic model for a pipeline network in which one can still find interesting properties is the one in Cremer, (2002) [20]. Reproduced in figure 2.1, the simple network consists of two injection/extraction nodes, and a single-conduit pipeline linking them.

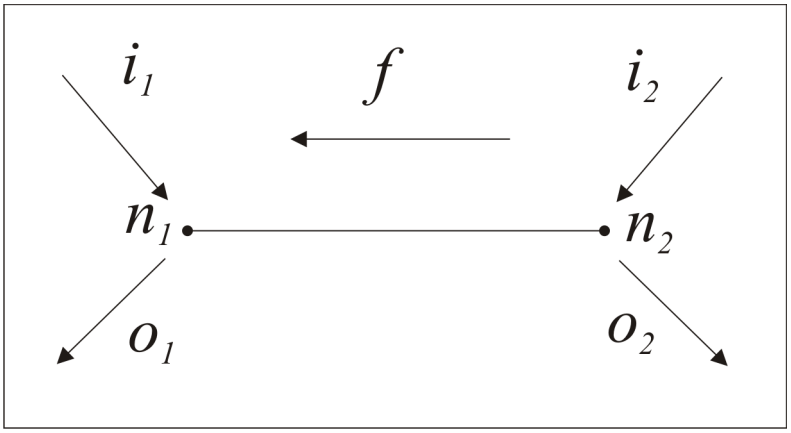


Figure 2.1. A Simple, Illustrative Natural Gas Pipeline Network. [20]

Injection of gas is represented with variables i_1 and i_2, whereas extraction is represented with variables o_1 and o_2. Variable f is physical movement, or flow of gas, from node n_2 towards node n_1. The authors consider the potential existence of a flow in the opposite direction to f, but make it clear that, in such case, the reverse flow will be insignificant when compared to flow f, which is thus named the *dominant flow*.

Another interesting feature of this simple network model is that it shows how transportation costs, given certain conditions, need not be proportional to distance. Cremer discusses the particular case when the objective function is modeled including Ramsey monopoly terms [65].

Another simple way to present a basic formulation for natural gas networks are transportation problems. Indeed, one could represent a natural gas supply line as a set of nodes and arcs in a graph. Imagine that we have a source node—a natural gas well— and a sink node—a consumer or LDC delivery point—in every node of the equivalent to a segmented line (figure 2.2)

This is a simple, multi-node cyclic network [87]. Each node may be both a source and a sink node, though one would likely demand: (1) that this cannot occur not at the same time, and (2) that there are no leaks in the process. Under these conditions, we have a typical *transportation problem* in the sense of Monge [76]. Furthermore, if we drop assumption (2) above i.e. we allow leaks to be considered,

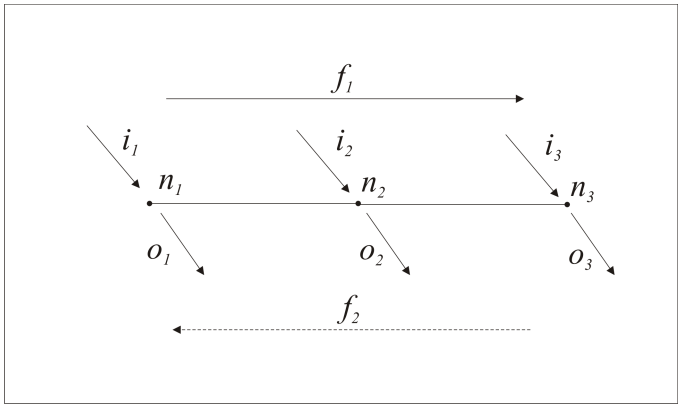


Figure 2.2. A Linear Gas Network

then we have an *imbalanced* transportation problem. Modeling transportation problems with such assumptions is done, for example, in [107].

Note that the dominant flow, f_1 in figure 2.2, runs from node n_1 to node n_3. The inverse flow, f_2, is merely financial in nature, meaning that it has no physical gas in motion, and consequently it causes no operating costs or leaks.

2.2 Models Focused in Optimizing the Physical Operation

This section illustrates models whose main concern is the conservation of proper operating conditions (pressure level, for the most part,) and the minimization of the costs of such maintenance (by optimizing the number of pumps, compressors, etc.)

A complete modeling including production, transportation, markets and other features of the natural gas supply chain is presented in Tomasgard (2007) [98]. This model makes use of relevant data from the Norwegian firms Gassco (a state-owned POC [36]) and Statoil (the country's largest fuel—and overall—company. [91]) The objective function modeled form the point of view of the POC and aims to optimize the *routing* of the gas while maintaining volume, quality and pressure levels in the pipeline within operating limits. It also uses a modified transportation model as the one discussed in the previous section, while adds storage, spot price sales, and future sales as features of the model.

The later model is complemented with that in Midthun *et al* (2008.) This expansion shows the role of a benign (non profit-seeking) POC solving routing issues

by marketing booking of capacity in the pipeline and selling it in both a spot market and a futures market. Such situation is modeled as a Generalized Nash Equilibrium [24] (in a setting more akin to our own study) in which there are several large and small NGSCs competing for capacity in the network. Perhaps the most important result we find in this work regards the normally employed Weymouth equations [73]—which model the pressure in the pipelines and are commonly used by engineers—and how can they be effectively substituted by common bounds for the variables, allowing for faster computation times, albeit in a reduced feasible space.

Another model that includes physical aspects of the pipeline optimization is that in Wong [108], where dynamic programming is used to solve a modified transportation problem including special nodes that indicate the existence of compressors. Optimal operating conditions are then decided applying dynamic programming.

Moreover, in Wu *et al* [109], we find a transportation model focused on reducing the network complexity (e.g., reducing large network trees to line graphs lake the one in figure 2.2.) The objective function is formulated so as to minimize the operating costs resulting from running the compressors that control flow and pressure in the pipeline. Relaxation techniques for hard constraints are also given. The first half of the article discusses mathematical models regarding the technical aspects of the pipeline (compressor's running parameters, fuel costs, etc.); whereas the second half presents the aforementioned network models. These are later expanded in [77], where a method for finding a feasible initial solution is developed so that it can be feed into the model in [109].

Rios-Mercado has studied different aspects of natural gas networks. In [12], network routing problems are solved by finding feasible flows and then using non-sequential dynamic programming to deliver a set of feasible pressures. An alternate procedure to the dynamic approach is shown in [34], A Generalized Reduced Gradient [46] is used to produce both feasible flows and pressures. Both techniques are compared in [11], where near-optimal solutions produced are improved using heuristic taboo search [43]: since this routing problem is done in a non-convex solution space, guaranteeing a global optimum is not easily done. More information about this type of models can be found in [87, 104, 13, 33].

2.3 Models Including Stochastic Elements

This section discusses models that center predominantly in the analysis of stochastic systems. Stochastic variables are sometimes added as external variables to expand more traditional models instead of turning the variables of those models into stochastic elements.

Uncertainty in temperatures is a concern for Maggioni *et al* [70], where stochastic differential equations [3] are used to estimate and forecast environmental temperatures included in the objective functions of a two-stage stochastic model. The objective function here is the expected value of the NGSC profit and it depends on several different variables of diverse natures.

Midthun *et al* [75], additionally to the capacity booking model mentioned before, shows a stochastic modeling of an investment portfolio. Both of these models include stochastic elements in their objective functions:

- In the capacity booking models, there are three alternative objective functions: maximizing the flow of the pipeline, maximizing the economic value of that flow, or maximizing the social surplus [38]. The stochastic variables are the spot prices, modeled as a number of price scenarios.

- The portfolio optimization problem contains a scenario setting for its stochastic modeling. Instead of modeling the variables for each scenario as independent, it used a *node formulation* that allows for a reduced number of decision variables and the omission of an explicit set of *non-anticipativity constraints*, but somewhat complicates the expressions for the constraints. The stochastic variables are again the spot prices, but the objective function represents the production and the transportation and storage related compressors operating costs, which is to be minimized, in a typical *multistage stochastic optimization model with recourse.* [60]

Although all these models are dedicated to links of the supply chain other than the NGSC-POC subsystem, they are relevant to this work since they show (1) a branching scenario tree approach to the stochastic optimization problem, (2) bilevel characteristics, and (3) the usage of multi-stage stochastic optimization with

recourse, all of them important tools considered in the models developed in chapter 4.

Looking at the forecasting and estimation tools, Balestra and Nerlove [5] presents an early work in estimation of natural gas demand/consumption. This works on a rather small-scale, as demand is considered driven by the acquisition of appliances. The paper then goes on using time series analysis through least-square estimations to round up the forecast framework.

Gomez-Villalba and Ramos stated [39] that "it is necessary to provide (firms) with optimization tools to manage risk derived from price uncertainty." To this purpose, they present a transportation and contracting model for natural gas and electricity pricing, treating commodities prices as market-based. This model deals with the minimization of the consumer energy supply cost. Though not completely analogous to our problem statement (the decision maker and constraints of the optimization models are completely different in both nature and behavior,) this article shows an important similitude to our work: scenario trees are used, along with an economical justification of their use, and an extensive glossary on scenario tree-related stochastic terms is presented. The algorithm [40] used in this work to obtain natural gas prices forecast is based in time series analysis of Brent spot prices and an ARIMA [15] model. An important assumption regarding forecast residuals is done here and we promptly adopt it: we need to consider that the distribution of the price residuals in the prediction time is the same as their distribution in the historical sample.

Other common methods to forecast Natural Gas consumption are Artificial Neural Networks [16], and Fuzzy Neural Networks [16, 103]. Artificial Neural Networks are attractive for NGSCs because they allow them to map nonlinear relationships between regressors (wind speed, temperature, etc.) and the response variable (consumption.) This is often done when the scope of the forecast is such that no long-term relationships are considered reliable. On the other hand, Fuzzy Neural Networks are often used when the reliability of the data inputted is poor or too general (e.g., when we have only point-estimates of highly variable statistics.)

Recent works using Gompertz-type algorithm in the natural gas consumption of Spain was done in [42], developing complex techniques that reportedly improved

upon more conventional forecasting methods in the same setting, though further work is admittedly needed to add more explicative factors hence providing deeper analysis of the market.

2.4 Models Including Bilevel Problems

Finally, we present the models that were the inspiration of this dissertation work. The works in this section have a closer relationship to the developments in chapters 3 and 4.

Bilevel Programming is a branch of modern Multilevel Optimization Theory [57, 74]. The mathematical tools that constitute the bilevel programming field are extensions of regular mathematical programming and game theory models [7]. Optimization problems with more than one decision makers are usually modeled as equilibrium or game models, and when one of those decision makers has strategic advantages or timing differences over its competitors, bilevel programs often arise.

The bilevel formulation of a NGSC-POC subsystem is initially done in [88]. The NGSC becomes the upper level decision maker, whereas the POC is the lower level decision maker. The authors address the inherent difficulty of bilevel programs [57] by using simulated annealing [64] to solve a heuristic algorithm: a high-quality approximation that guarantees good solutions in short time instead of optimal solution in prohibitive running times.

This problem is later taken by Kalashnikov and Rios-Mercado [57, 74] and reformulated, modeling the lower level as a variational inequality problem [50] and using a parameterized version of this inequality as a penalty function for the upper level, thus turning the original bilevel problem into a single-level problem, albeit a complex one. The model presented in these articles is a bilevel, mixed integer problem with a quadratic objective function and logical (e.g., if a then b ...) constraints.

The authors in [58] claim that the problem above described can be split in two "almost completely independent stages": finding the lower level solution for a given upper level decision vector, and verifying the feasibility of the obtained solution for the upper level. An algorithm to iteratively solve this formulation is presented, which also has the advantage to removing the lower level integer variable and

passing it onto the upper level, which enormously eases the calculation of the solution.

The method proposed in [58] is implemented computationally in two different ways in [59]: an iterative inexact penalization method [71], and a direct method based in the Nealder-Mead Simplex method [83]. The authors' intention is to prove the superiority of the iterative inexact penalization method over the direct one in terms of convergence, stability and running performance.

Dempe *et al* [23] transform the lower level problem in [57] into a generalized transportation problem: the upper level is split in a branch and bound-like manner, turning the lower level into a problem with sinks and source nodes instead of pool zones. Convergence and conditions for this transformation, as well as an algorithm to efficiently solve this problem are provided.

While all the latter bilevel formulations deal only with penalization cash-out costs for the NGSC imposed from the POC, [22] integrates the NGSC expected revenue changes into the problem, modifying the upper level function to include linear terms. This work is continued in [53], where the analysis of the changes this modification creates in the optimal points is illustrated, and gross revenue, instead of revenue changes, is added to the objective function.

Chapter Summary

Modeling work regarding natural gas covers a wide amount of subjects, ranging from chemical and physical qualities that wells and refining facilities have to make, to technical parameters the pipelines have to control, to financial and economic tools over gas networks. This chapter resumes works regarding the models pertinent to this dissertation work—optimization and forecasting models in the natural gas industry—as well as alternate models that start from different assumptions and have different objectives, but nevertheless provide insight regarding Natural Gas Optimization frameworks.

The central matter of this chapter is section 2.4, which describes bilevel optimization models that are used as the foundation for the contents appearing later in this work. Those algorithms deal almost exclusively with the cash-out penalization that the POC ultimately charges or pays to the NGSC because of the

creation of imbalances; we intent to expand the models in that section so that they are compatible with the assumptions and intentions stated in chapter 1.

CHAPTER 3
DETERMINISTIC MODELS

The purpose of this chapter is to exhibit the mathematical models devised to pursue the objectives of this research work, as well as the algorithms that will be used to computationally solve the optimization models. Specifically, we deal now with the deterministic approach to formulate and solve a bilevel optimization problem with the characteristics described in section 1.4. The basic notions for this are given in section 3.1.

Section 3.1.1 shows the process by which mathematical models has been developed, inspired from the original source in [88]. Later, in section 3.2, theoretical work is described to reduce the models thus obtained into linear and almost-linear equivalent versions that are intended to ease the computational solutions. The algorithms employed to solve this theoretical models are lastly given in section 3.3.

3.1 Bilevel Programming Model

In many decision processes, we find that the variables involved are controlled by different entities, or decision makers, each one pursuing independent—and often opposed—goals. The basic mathematical models developed to abstract these situations are informally called (mathematical) "games". A particular type of game is one in which the decision makers (the "gamers") have different levels of influence, or take their decisions at different times, i.e., there is an underlying hierarchy in the game structure. An approach often followed with these types of games is to focus in one level of decision, and model every other level as assumptions.

Multilevel Programming, on the contrary, deals with mathematical programming problems that consider the whole hierarchical structure. If there are only two participants, we have a bilevel programming problem: one decision maker (dubbed the follower, or lower level decision maker) has its choices constrained by the decisions made by the other decision maker (dubbed the leader, or upper level decision maker.)

Typically, we model the lower level as a regular optimization problem whose feasible set $\Gamma = \Gamma(\mathbf{x})$ is a mapping of the values that correspond to the leader's

decision variables, $\mathbf{x}$. The follower's constraints and objective function will consequently change whenever they include variables controlled by the leader. *By knowing the logic used by the follower, the leader can successfully influence the problem's outcome to its convenience, even though it is not in complete control of the decision.*

Multilevel Programming are a well known and important field of Optimization Theory, with applications in environmental sciences, politics, economy, databases, network design, transportation, and logistics [57]. In this work, we attempt to apply multilevel programming (or rather, bilevel programming) to the NGSC-POC subsystem described in section 1.3.2, according to the guidelines given in section 1.4.

3.1.1 Deterministic Optimization Models

This section presents the deterministic models that abstract the NGSC-POC subsystem. They will be used later as the basis for the stochastic models in chapter 4.4.

Note that, even though the solution of a stochastic formulation is the ultimate goal of the dissertation work, the deterministic models developed are not lacking in their own research merit, both theoretically and numerically speaking.

Each one of the deterministic models in this chapter deals with some of the following subjects:

- the source problem used as a base
- the expansion of its objective function to include variables of interest
- the linearization of the later problem's upper level
- the linearization of the later problem's lower level

3.1.2 Mixed Integer Bilevel Cash-Out Problem

Let us first recapitulate our problem statement, paraphrasing the explanation given in Kalashnikov and Ríos-Mercado [57]: we are concerned with one NGSC, the leader in our bilevel problem; and one POC, the follower; in a particular problem setting. The NGSC controls gas extraction from the pipeline each day within certain

predetermined, finite operative period divided in equal-length intervals (which may be days, weeks, etc.) Without loss of generality, we will from now on refer to these intervals as days. The extraction takes place across many points, called *Pool Zones*. The extractions made by the NGSC may indirectly create imbalances, which are differences between the amount of gas the NGSC has planned/booked—at the beginning of this process—, and the actual extracted volumes.

If the amount of natural gas extracted from the pipeline is *greater* than the amount originally stated, we have a *negative* imbalance. Conversely, if the amount of gas extracted from the pipeline is *lower* than the amount stated, we have a *positive* imbalance. The NGSC controls the extraction amounts—and consequently, the creation of imbalances —mostly independently in each pool zone and day, subject to the constraints explained later.

Once the NGSC has finished its extracting operations, the POC will have the chance to rearrange the final day imbalances, moving gas form positive imbalance pool zones to negative imbalance pool zones. The goal of this process is to leave only nonnegative or nonpositive imbalances in all pool zones, obtaining the minimal cost/revenue from this rearrangement, as the resulting final imbalances are used to determine the NGSC to POC (or POC to NGSC) cash-out derived from imbalance penalization.

The notation given in tables (3.1)-(3.6) is used throughout the rest of this chapter to represent the variables involved in the process above, with some symbols also used in chapter 4.

Table 3.1. Sets for Problem $PR1(x, s, y, u, v, z, q)$.

N	Maximum number of days in the process; $N \in \mathbf{Z}^{++}$.
P	Number of pool zones; $P \in \mathbf{Z}^{++}$.
$\mathbf{T}$	Set for Days; $T = \{1, 2, ..., N\}$.
$\mathbf{J}$	Set for Pool Zones; $J = \{1, 2, ..., P\}$.

Table 3.2. Upper Level Parameters for Problem $PR1(x, s, y, u, v, z, q)$.

x_{ti}^L, x_{ti}^U	Lower and upper bounds for the daily imbalance x_{ti} at day t in zone i; $t \in \mathbf{T}$, $i \in \mathbf{J}$.
x_t^L, x_t^U	Lower and upper bounds for the sum of the daily imbalances at day t; $t \in \mathbf{T}$.
s_{ti}^L, s_{ti}^U	Bounds on balance swing s_{ti} from day $t-1$ to day t, in pool zone i; $t \in \mathbf{T}$, $i \in \mathbf{J}$.
x_{0i}	Imbalance at the beginning of day 1 in pool zone $i \in \mathbf{J}$.

Table 3.3. Lower Level Parameters for Problem $PR1(x, s, y, u, v, z, q)$.

e_{ij}	Percentage of gas used/lost when moving one unit of gas from zone i to zone j; $i, j \in \mathbf{J}$.
f_{ij}	Forward haul cost for moving one unit of gas from zone i to zone j; $i, j \in \mathbf{J}$.
b_{ij}	Backward credit given to the NGSC after moving one unit of gas from zone i to zone j;$i, j \in \mathbf{J}$.
r_i	Linear penalization coefficients in zone i; $i \in \mathbf{J}$.
δ_i	Quadratic penalization coefficients in zone i; $i \in \mathbf{J}$.

Table 3.4. Upper Level Decision Variables for Problem $PR1(x, s, y, u, v, z, q)$.

x_{ti}	Imbalance at the end of day t in zone i; $t \in \mathbf{T}$, $i \in \mathbf{J}$.
s_{ti}	Imbalance swing from day $t-1$ to day i in zone i; $t \in \mathbf{T}$, $i \in \mathbf{J}$.

Table 3.5. Lower Level Decision Variables for Problem $PR1(x, s, y, u, v, z, q)$.

y_i	Final imbalance in zone i; $i \in \mathbf{J}$.
u_{ij}	Volume of gas moved from zone i to zone j; $i, j \in \mathbf{J}$.
v_{ij}	Gas credited from zone j to zone i; $i, j \in \mathbf{J}$.
z	Total cash-out for the Natural Gas Shipping Company.

Table 3.6. Auxiliary Variable for Problem $PR1(x, s, y, u, v, z, q)$.

q	Binary variable equal to 1(0) if final imbalances y_i are all non-negative (non-positive.) In case $y_i = 0, i \in \mathbf{J}$, we accept $q = 1$.

The model proposed in [57] is given by (3.1a)-(3.1e), (3.2a)-(3.2k). It will be labeled as $PR1(x, s, y, u, v, z, q)$. Variable z is controlled by the lower level decision maker but appears in the upper level's objective function; conversely, variable x is controlled by the upper level but appears in several constraints of the lower level.

Upper Level Model:

$$\text{Minimize:} \quad h_1(x, s, z) = z \tag{3.1a}$$

Subject to:

$$x_{ti}^L \leq x_{ti} \leq x_{ti}^U, \quad t \in \mathbf{T}, i \in \mathbf{J}; \tag{3.1b}$$

$$s_{ti}^L \leq s_{ti} \leq s_{ti}^U, \quad t \in \mathbf{T}, i \in \mathbf{J}; \tag{3.1c}$$

$$x_t^L \leq \sum_{i \in \mathbf{J}} x_{ti} \leq x_t^U, \quad t \in \mathbf{T}; \tag{3.1d}$$

$$x_{ti} = x_{t-1,i} + s_{ti}, \quad t \in \mathbf{T}, i \in \mathbf{J}. \tag{3.1e}$$

Constraints (3.1b) limit the size of the imbalances in every pool zone and day. Constraints (3.1c) limit the change that imbalances in a given zone can have form one day to another: sizable imbalances, or sizable changes thereof would disrupt the regular operating pressure in the pipeline, thus the POC imposes all NGSC upper and lower bounds to both variables. In the same manner, the total imbalance throughout the whole system is bounded above and below by constraints (3.1d). Constraints (3.1e) express the natural relationship between the imbalance variables x_{ti} and the imbalance swing variables s_{ti}.

Lower Level Model:

$$\text{Minimize:} \quad g_1(x, y, u, v, z, q) = z^2 \tag{3.2a}$$

Subject to:

$$y_j = x_{Ni} + \sum_{i:i<j} (1 - e_{ij}) u_{ij} - v_{ij} + \sum_{k:k>j} v_{jk} - u_{jk};\ j \in \mathbf{J}; \tag{3.2b}$$

$$\sum_{j:i<j} u_{ij} + \sum_{k:k<i} v_{ki} \leq \max\{0, x_{N,j}\};\ j \in \mathbf{J}; \tag{3.2c}$$

$$u_{ij} \leq \begin{cases} x_{N,i} & \text{if } x_{N,i} > 0 \text{ and } x_{N,j} < 0, \\ 0 & \text{otherwise;} \end{cases} \tag{3.2d}$$

$$v_{ij} \leq \begin{cases} x_{N,j} & \text{if } x_{N,j} > 0 \text{ and } x_{N,i} < 0, \\ 0 & \text{otherwise;} \end{cases} \tag{3.2e}$$

$$\min\{0, x_{N,i}\} \leq y_i \leq \max\{0, x_{N,i}\};\ i \in \mathbf{J}; \tag{3.2f}$$

$$-M(1 - q) \leq y_i \leq Mq;\ i \in \mathbf{J}; \tag{3.2g}$$

$$z = \sum_{i \in \mathbf{J}} [\delta_i (y_i)_+^2 - r_i y_i] - \sum_{(i,j):i<j} v_{ij} b_{ij} + \sum_{(i,j):i<j} f_{ij} (1 - e_{ij}) u_{ij}; \tag{3.2h}$$

$$y_i, z \in \mathbb{R};\ i \in \mathbf{J}; \tag{3.2i}$$

$$u_{ij}, v_{ij} \geq 0;\ i, j \in \mathbf{J}; \tag{3.2j}$$

$$q \in \{0, 1\}. \tag{3.2k}$$

Expression (3.2b) defines the relation between the *final day imbalances* x_{Ni}, which are ultimately determined by the NGSC manipulation at the end of day N, and the *final imbalances* y_i, which are produced after the POC's rearrangement of the resulting x_{Ni}'s. The final imbalance at zone i, denoted by y_i, is equal to whatever imbalance was already there before the rearrangement (the first term) plus whatever gas is moved towards there (second and fourth terms,) minus whatever gas was moved from there (third and fifth terms.)

The expression in line (3.2c) prevents cyclical movements of gas: for any given zone, the amount of gas moved forward or backward from there cannot be higher than any positive imbalance initially there. Constraints (3.2d)-(3.2e) prevent any

other type of movement (forwards or backwards) other than that carried on from a positive imbalanced zone to a negative imbalanced zone.

Line (3.2f) prevents positive imbalances to grow bigger, and negative imbalances to decrease further: the sign of final imbalances y_i must be the same as the corresponding final day imbalance x_{Ni}. Also, final day imbalances must all share the same sign—this is a business requirement—; constraint (3.2g) guarantees this by employing the auxiliary variable q.

Lastly, constraint (3.2h) defines the relationship of the variable representing the penalization amount, z, with the final day imbalances, final imbalances, and forward/backward moved volumes. The second term represents the amount credited in favor of the NGSC by the POC for "moving" gas upstream, an action that is performed only "in paper" as the network does not allow for physical upstream movements of gas (hence no gas loss/usage.) For every unit of gas moved backwards from zone j to zone i, the POC credits the NGSC with b_{ij}.

The third term represents the cost for the shipper to physically move gas downstream: the POC has to use pumps to perform this operation, which causes operating costs that are charged to the NGSC. The percentage of gas lost in this movement, e_{ij}, is covered by the NGSC, so for every unit of gas moved forwards from zone i to zone j, the NGSC is charged by the POC with an effective price of $f_{ij}(1 - e_{ij})$.

The first term, $\sum_{i \in \mathbf{J}} [\delta_i (y_i)_+^2 - r_i y_i]$, where $(y_i)_+ = \min\{y_i, 0\}$, has a slightly more complex justification: it represents a piecewise, derivable quadratic cost function that effectively serves as a 'purchase/sale' price from the POC to the NGSC: if there is a negative imbalance, the NGSC is taking extra gas in the pipeline, buying it at the constant unit price of r_i. Conversely, if there is a positive imbalance, the NGSC is leaving gas in the pipeline, selling it to the POC at a price of $(\delta_i - r_i)$ per square unit bought. Terms δ_i are artificial parameters added to limit the amount of gas the NGSC would likely sell to the POC: if $\delta_i > 0$, there is then a stationary point (a global minimum,) for function $[\delta_i (y_i)_+^2 - r_i y_i]$ for pool zone i (see figure 3.1.) Beyond this point, the earnings of the NGSC for "selling" gas to the POC decrease until they become negative: the NGSC is leaving gas in the pipeline *and* paying the POC to receive it! Consequently, there is a strong motivation for the NGSC to

maintain any positive final imbalances day x_{Ni}'s around the stationary point—or at least, not allowing them to grow too large—, so that the POC cannot create y_i's larger than the intersection point of the purple line in any given pool zone.

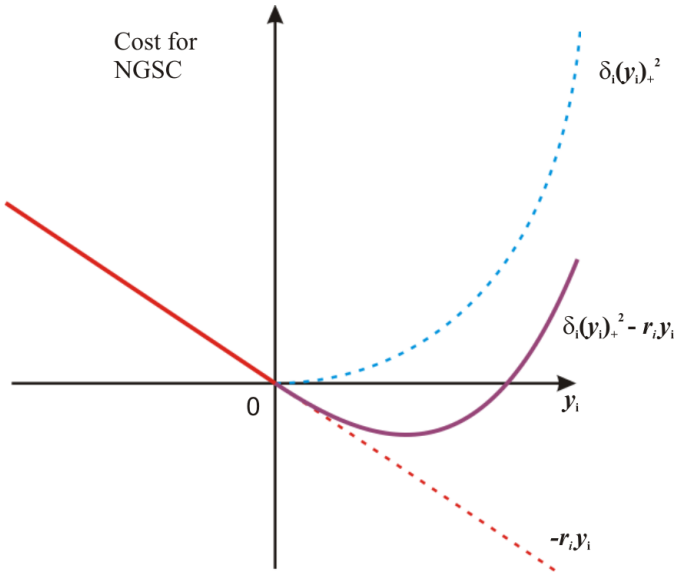


Figure 3.1. Objective Function (3.2h)'s First Term for an Arbitrary Pool Zone.

Finally, note that the objective function of the POC in the lower level is not $-z$ (the negative cost for the NGSC i.e. the total cost for the POC,) as one would naturally assume, but instead z^2. This represents the willingness of the POC to minimize both its and the NGSC's possible losses when conducting business with each other (assumption12.) Business-wise, it would not be advisable for a NGSC to engage in a contract with a POC whose stated purpose is to undermine the former!

3.1.3 Price Deviation Model Variation

Problem $PR1(x, s, y, u, v, z, q)$ describes a NGSC-POC subsystem concerned only with pool zone imbalances and the derived penalization cash-out between the participants. In order to make this problem closer to reality, we add sale and cost terms for the NGSC operations, as well as variables representing the actual volumes of gas extracted and booked in every pool zone. These changes affect solely the upper level; the lower level parameters, decision variables, and constraints remain for the most part unchanged.

An early attempt to introduce sale and price considerations in the optimization model was done by this work authors in [22, 53], where the objective function in (3.1a) was modified to be:

$$h_2(x, s, z) = z + \sum_t \sum_i d_{ti} x_{ti}, \tag{3.3}$$

where $\Delta = (d_{ti}) \in \mathbf{R}^{N \times P}$ is a price deviation matrix, i.e., the sale price for pool zone i at day t differs from the expected price by d_{ti}. This modification would ultimately affect revenues from gas sales, so decisions regarding the size of the imbalance in that particular pool zone and day would be affected too. The problem obtained by substituting (3.1a) with (3.3) is labeled as $PR2(x, s, y, u, v, z, q; \Delta)$.

$PR2(x, s, y, u, v, z, q; \Delta)$ is but a simple modification to problem $PR1$: it adds nothing but a linear term on the objective function. Results regarding running time, convergence, and comparisons between optimal values of functions h_1 and h_2 performed by the authors are presented in the references already given.

There are many shortcomings in the $PR2(x, s, y, u, v, z, q; \Delta)$ approach. First, it is a deterministic model, so its decision-making ability is limited to present different instances with different values for matrix Δ (e.g., prices going up, down, positive-only or negative-only deviations, etc.) Furthermore, the value of function h_2 holds little meaning to the decision maker; it does not represent any actual amount of interest for the NGSC. Hence, we expanded from this formulation and added actual gas volumes (instead of only imbalances,) as well as sale prices (instead of price deviations) and consumptions.

3.1.4 Price, Demand and Extraction Volume Model

This section presents the modification to the model explained at the end of previous section; specifically, we include now aspects of the NGSC-POC like extraction volume, consumer price and demand.

Tables 3.8 to 3.7 show the variables and parameters introduced for this formulation.

Variable EA_{ti} represents the amount of gas actually extracted from pool zone i at time t, whereas variable EP_{ti} is the amount of gas *announced to be extracted*, or

Table 3.7. Price and Demand Related Parameters Added to Problem $PR1$.

DC_{ti}	Expected demand at day t in pool zone i; $t \in \mathbf{T}$, $i \in \mathbf{J}$.
DM_{ti}	Maximum amount of gas that is possible to extract in pool zone i at time t; $t \in \mathbf{T}$, $i \in \mathbf{J}$.
Π^c_{ti}	Unit price for the first DC_{ti} units of gas extracted/sold (contracted gas) at time t in zone i; $t \in \mathbf{T}$, $i \in \mathbf{J}$.
Π^e_{ti}	Unit price for whatever units of gas extracted/sold beyond DC_{ti} at time t in pool zone i; $t \in \mathbf{T}$, $i \in \mathbf{J}$.
CI_{ti}	Cost for undelivered contracted gas unit at time t in pool zone i; $i \in \mathbf{J}$.
CR_{ti}	Cost for reserved gas capacity at time t in pool zone i; $t \in \mathbf{T}$, $i \in \mathbf{J}$.

Table 3.8. Extraction and Booking Decision Variables Added to Problem $PR1$.

Upper Level Decision Variables	
EA_{ti}	Amount of gas extracted/sold from zone i at day t; $t \in \mathbf{T}$, $i \in \mathbf{J}$.
EP_{ti}	Amount of gas notified to be extracted at day t in zone i; $t \in \mathbf{T}$, $i \in \mathbf{J}$.

booked, by the NGSC before the whole process had begun. Using the new variables, we can deduce a new, more illustrative definition for the imbalance variables:

$$x_{ti} = EP_{ti} - EA_{ti},\ i \in \mathbf{J}, t \in \mathbf{T}. \tag{3.4}$$

If $EP_{ti} > EA_{ti}$, we are extracting less than our stated/injected amount, creating a positive imbalance x_{ti} in zone i at time t. On the contrary, if $EP_{ti} < EA_{ti}$, we are then extracting more gas than the amount initially injected, thus creating a negative imbalance x_{ti}.

For every day t and pool zone i, the NGSC has agreed to provide an actor external to the NGSC-POC system with a certain volume of natural gas so that the LDC distributes it to its own costumers. This figure is represented by parameter DC_{ti} (assumption 7.) For every unit of gas the NGSC does not satisfies, it receives a penalization of CI_{ti}. Also, for every unit of gas injected, EP_{ti}, the NGSC has to pay to the POC CR_{ti} for the use of the pipeline. This covers all operative costs the POC

might incur while dealing with the normal, daily process of pumping gas throughout the process, but not expenses derived from final day balancing of the system.

For every unit of gas extracted and delivered by the NGSC to the (external) LDC within the limit DC_{ti}, the NGSC earns Π^c_{ti}. Any amount of gas extracted beyond this ($EA_{ti} - DC_{ti}$, in case that $EA_{ti} > DC_{ti}$) will be sold by the NGSC, according to assumption 8, to secondary customers who pay Π^e_{ti} per unit of gas. The NGSC cannot inject nor extract more gas than DM_{ti} in any pool zone i at any time t: DM_{ti} is a limit of technical, logistical, financial, or any other nature.

Since the lower level variables remain unchanged, the new model, labeled as $PR3(x, s, EA, EP, y, u, v, z, q; \Pi^c, \Pi^e, DC)$, can be expressed by equations (3.5a)-(3.5h), along with the already shown equations (3.2a)-(3.2k).

$$\begin{aligned}
\text{Minimize} \quad & h_3\left(x, s, z, EA, EP; \Pi^c, \Pi^e, DC\right) = \\
& z - \sum_{t\in\mathbf{T}} \sum_{i\in\mathbf{J}} \left[\Pi^c_{ti}\left(\min\{EA_{ti}, DC_{ti}\}\right) + \Pi^e_{ti}\left(\max\{0, EA_{ti} - DC_{ti}\}\right)\right] \quad (3.5a) \\
& \quad + \sum_{t\in\mathbf{T}} \sum_{i\in\mathbf{J}} \left[CI_{ti}\left(\max\{0, DC_{ti} - EA_{ti}\}\right) + CR_{ti}EP_{ti}\right];
\end{aligned}$$

Subject to

$$x^L_{ti} \le x_{ti} \le x^U_{ti}, \quad t \in \mathbf{T}, i \in \mathbf{J}; \quad (3.5b)$$

$$s^L_{ti} \le s_{ti} \le s^U_{ti}, \quad t \in \mathbf{T}, i \in \mathbf{J}; \quad (3.5c)$$

$$x^L_t \le \sum_{i\in\mathbf{J}} x_{ti} \le x^U_t, \quad t \in \mathbf{T}; \quad (3.5d)$$

$$x_{ti} = x_{t-1,i} + s_{ti}, \quad t \in \mathbf{T}, i \in \mathbf{J}; \quad (3.5e)$$

$$x_{ti} = EP_{ti} - EA_{ti}, \quad t \in \mathbf{T}, i \in \mathbf{J}; \quad (3.5f)$$

$$0 \le EA_{ti} \le DM_{ti}, \quad t \in \mathbf{T}, i \in \mathbf{J}; \quad (3.5g)$$

$$0 \le EP_{ti} \le DM_{ti}, \quad t \in \mathbf{T}, i \in \mathbf{J}; \quad (3.5h)$$

where variable z is the corresponding component of the optimal solution of the lower level problem represented by (3.2a)-(3.2k) Note that, this time, the NGSC has an actual motivation not to leave natural gas in the pipeline (and create unnecessary positive imbalances): from the mathematical point of view, and for a fixed EP_{ti},

making x_{Ni} unjustifiably larger would reduce the magnitude of EA_{Ni}, preventing the NGSC from gaining $[\Pi^c_{Ni}(\min\{EA_{Ni}, DC_{Ni}\}) + \Pi^e_{Ni}(\max\{0, EA_{Ni} - DC_{Ni}\})]$. This amount is typically larger than any revenue $[\delta_i y_i^2 - r_i y_i]$ the NGSC could obtain from the POC by leaving gas in the pipeline (since buying, selling, and generally, trading gas is not a business activity for the POC, thus it is not interested in buying and offers a low price, only as means of balancing the network.)

The NGSC is then encouraged not to leave lots of gas into to the pipeline instead of extracting it. This was the only reason to add term $\delta_i(y_i)^2_+$ to equation (3.2h), a modeling one, since it had otherwise no real world meaning. Dropping this term (or equivalently, fixing $\delta_i = 0\,\forall i$,) we obtain problem
$PR3'(x, s, EA, EP, y, u, v, z, q; \Pi^c, \Pi^e, DC)$, which uses equation

$$z = -\sum_{i\in\mathbf{J}} r_i y_i - \sum_{(i,j):i<j} v_{ij} b_{ij} + \sum_{(i,j):i<j} f_{ij}\,(1 - e_{ij})\,u_{ij},$$

instead of equation (3.2h).

We have then removed the piecewise quadratic term in line (3.6h), but one would still want to eliminate most other nonlinearities from the problem.

To achieve this, consider first lemma 1:

Lemma 1. *Let $I \subset \mathbb{R}$. Then*

$$\arg\min_{z\in I} z^2 = \arg\min_{\substack{d\in I \\ -d\le z\le d}} d$$

Proof. Function $f(z) = \sqrt{z}$ is bijective over $[0, +\infty)$, which is the domain of both $f(z) = z^2$, and $f(z) = |z|$. Since $|z| = \sqrt{z^2}$, we have:

$$\arg\min_{z\in I} z^2 = \arg\min_{z\in I} |z|$$

The optimal point obtained by minimizing z^2 is then equivalent to that obtained by minimizing either side of expression:

$$|z| \le d; \quad z, d \in I,$$

which can be expressed in linear terms as $-d \leq z \leq d$. □

This does away with the quadratic z^2 in the lower level objective function. Indeed, the leader needs not to use the exact value the lower level is minimizing, as long as the optimal vector is the same. Lemma 1 delivers a way to calculate such optimal point by putting I as the feasible set for the lower level problem, and minimizing d instead of z^2.

Another of the nonlinearities found in problem $PR3$ has so been removed.

The lower level corresponding to problem $PR3'(x, s, EA, EP, y, u, v, z, q; \Pi^c, \Pi^e, DC)$, considering the last lemma, can be written as:

$$\text{Minimize:} \quad g_1(y, u, v, z, d, q; x) = d \tag{3.6a}$$

Subject to:

$$y_j = x_{Ni} + \sum_{i:i<j} (1 - e_{ij}) u_{ij} - v_{ij} + \sum_{k:k>j} v_{jk} - u_{jk};\ j \in \mathbf{J}; \tag{3.6b}$$

$$\sum_{j:i<j} u_{ij} + \sum_{k:k<i} v_{ki} \leq \max\{0, x_{N,j}\};\ j \in \mathbf{J}; \tag{3.6c}$$

$$u_{ij} \leq \begin{cases} x_{N,i} & \text{if } x_{N,i} > 0 \text{ and } x_{N,j} < 0, \\ 0 & \text{otherwise;} \end{cases} \tag{3.6d}$$

$$v_{ij} \leq \begin{cases} x_{N,j} & \text{if } x_{N,j} > 0 \text{ and } x_{N,i} < 0, \\ 0 & \text{otherwise;} \end{cases} \tag{3.6e}$$

$$\min\{0, x_{N,i}\} \leq y_i \leq \max\{0, x_{N,i}\};\ i \in \mathbf{J}; \tag{3.6f}$$

$$-M(1 - q) \leq y_i \leq Mq;\ i \in \mathbf{J}; \tag{3.6g}$$

$$z = -\sum_{i \in \mathbf{J}} r_i y_i - \sum_{(i,j):i<j} v_{ij} b_{ij} + \sum_{(i,j):i<j} f_{ij} (1 - e_{ij}) u_{ij} \tag{3.6h}$$

$$-d < z < d; \tag{3.6i}$$

$$y_i, z \in \mathbb{R},\ i \in \mathbf{J}; \tag{3.6j}$$

$$u_{ij}, v_{ij} \geq 0;\ i, j \in \mathbf{J}; \tag{3.6k}$$

$$q \in \{0, 1\}. \tag{3.6l}$$

Problem $PR3(x, s, EA, EP, y, u, v, z, q; \Pi^c, \Pi^e, DC)$ is a mixed integer, bilevel optimization problem with non-smooth upper level objective function, lower level logical and quadratic constraints, and a lower level quadratic function. Problem $PR3'(x, s, EA, EP, y, u, v, z, q; \Pi^c, \Pi^e, DC)$, defined by (3.5a)-(3.5h), (3.6a)-(3.6l) is one step closer to obtaining a linear model, dropping the quadratic terms from the lower level objective function and constraints. In the next section, we will provide tools for solving the problems so far formulated using theoretical and computational tools intended for linear programming.

3.2 Bilevel Linear Reformulation[1]

This section describes a method to remove the *min* and *max* operators in the objective function (3.5a), replacing the upper level problem shown in section 3.1.4 with an almost linear problem, and the corresponding lower level problem with a linear programming problem, which is an inexact equivalent problem.

Reducing the independent levels to linear problems will eventually led us to a formulation like:

$$\text{Minimize: } c^1\mathbf{x} + d^1\mathbf{y} \tag{3.7}$$

$$\text{Subject to: } A^1\mathbf{x} \leq b^1 \tag{3.8}$$

$$\text{and } y = \arg\min\{d^2\mathbf{y} | G^2\mathbf{y} \leq b^2 - A^2\mathbf{x}\}, \tag{3.9}$$

where A^1, A^2 and G^2 are linear coefficient matrices, c^1, d^1, d^2, b^1 and b^2 are linear coefficient vectors, and variables $\mathbf{x}$ and $\mathbf{y}$ are the decision variables for the upper and lower level, respectively. This structure is called a *bilevel linear optimization problem*, and is formed by a linear upper level problem with respect to every variable except for those controlled by the lower level problem, which form the optimal solutions to another linear programming problem.

Reducing the upper level optimization problem to an almost linear equivalent and the lower level to a linear optimization problem should to boost the convergence speed of the solution methods for the models shown in this section. Solution techniques for linear problems and bilevel linear problems are more deeply

[1]The contents of this section appear in the authors' [55]

researched and usually more effective and widely applied than those used for nonlinear optimization, so being able to use them is of great help in this research.

3.2.1 Upper Level Equivalent Problem

Lemma 2 shows a simpler linear equivalent problem to the upper level of $PR3$. It is linear in every variable but z, which is controlled by the lower level and depends on this level's response.

Lemma 2. *Let $\Pi^c_{ti} - \Pi^e_{ti} + CI_{ti} > 0 \, \forall t, i$; and consider the single-level optimization problem, which is linear with respect to all variables except for z:*

$$\begin{aligned}
\textit{Minimize:} \quad & h_4(x, z, EA, A^1, A^2; \Pi^c, \Pi^e, DC) = \\
& z - \sum_{t \in \mathbf{T}} \sum_{i \in \mathbf{J}} \left[\Pi^e_{ti} A^1_{ti} + \left(\Pi^c_{ti} - \Pi^e_{ti} \right) A^2_{ti} \right] && \text{(3.10a)} \\
& \quad + \sum_{t \in \mathbf{T}} \sum_{i \in \mathbf{J}} \left[CI_{ti} \left(DC_{ti} - A^2_{ti} \right) + CR_{ti} (x_{ti} + EA_{ti}) \right] ; \\
\textit{Subject to:} \quad & \\
& x^L_{ti} \leq x_{ti} \leq x^U_{ti}, \quad t \in \mathbf{T}, i \in \mathbf{J}; && \text{(3.10b)} \\
& s^L_{ti} \leq x_{ti} - x_{t-1,i} \leq s^U_{ti}, \quad t \in \mathbf{T}, i \in \mathbf{J}; && \text{(3.10c)} \\
& x^L_t \leq \sum_{i \in \mathbf{J}} x_{ti} \leq x^U_t, \quad t \in \mathbf{T}; && \text{(3.10d)} \\
& 0 \leq EA_{ti} \leq DM_{ti}, \quad t \in \mathbf{T}, i \in \mathbf{J}; && \text{(3.10e)} \\
& 0 \leq x_{ti} + EA_{ti} \leq DM_{ti}, \quad t \in \mathbf{T}, i \in \mathbf{J}; && \text{(3.10f)} \\
& A^1_{ti} \leq EA_{ti}, \quad t \in \mathbf{T}, i \in \mathbf{J}; && \text{(3.10g)} \\
& 0 \leq A^2_{ti} \leq DC_{ti}, \quad t \in \mathbf{T}, i \in \mathbf{J}; && \text{(3.10h)} \\
& 0 \leq A^2_{ti} \leq A^1_{ti}, \quad t \in \mathbf{T}, i \in \mathbf{J}; && \text{(3.10i)}
\end{aligned}$$

where variable z is the corresponding component of the solution to the lower level problem (3.6a)-(3.6l).

1. *Let $(x^*, z^*, s^*, EA^*, EP^*)$ be an optimal solution to problem (3.5a)-(3.5h), where z^* solves problem (3.6a)-(3.6l). Then there exist A^1, A^2 such that $(x^*, z^*, EA^*, A^{1*}, A^{2*})$ is an optimal solution to problem (3.10a)-(3.10i).*

2. *Let $(x^{**}, z^{**}, EA^{**}, A^{1**}, A^{2**})$ be an optimal solution to problem (3.10a)-(3.10i), where z^* solves problem (3.6a)-(3.6l). Then there exist s^{**}, EP^{**} such that $(x^{**}, s^{**}, z^{**}, EA^{**}, EP^{**})$ is an optimal solution to problem (3.5a)-(3.5h).*

Proof. (a) Let vector $(x^*, z^*, s^*, EA^*, EP^*)$ be feasible solution to problem (3.5a)-(3.5h). Constraints (3.10b),(3.10d), and (3.10e) are trivially satisfied by x^*, EA^*. Putting together (3.5c) and (3.5e), we have that (3.10c) also holds. Similarly, (3.5f) and (3.5h) imply that x^*, EA^* satisfy (3.10f).

Define now the auxiliary variables A^1, A^2 as follows:

$$A^{2*}_{ti} = \min\{EA^*_{ti}, DC_{ti}\}, \tag{3.11}$$

$$A^{1*}_{ti} = \max\{0, EA^*_{ti} - DC_{ti}\} + A^{2*}_{ti} = \max\{0, EA^*_{ti} - DC_{ti}\} + \min\{EA^*_{ti}, DC_{ti}\}. \tag{3.12}$$

If $EA^*_{ti} \geq DC_{ti}$, then $A^{1*}_{ti} = (EA^*_{ti} - DC_{ti}) + DC_{ti} = EA^*_{ti}$. Otherwise, if $EA^*_{ti} \leq DC_{ti}$, then $A^{1*}_{ti} = 0 + EA^*_{ti} = EA^*_{ti}$. Therefore, $A^{1*} = EA$ and constraint (3.10g) is satisfied by A^{1*}. Moreover, since $DC_{ti} \geq 0 \forall t, i$, line (3.5g) and the definition of A^{1*}, A^{2*} make the latter satisfy (3.10h), (3.10i). Thus, vector $(x^*, z^*, EA^*, A^{1*}, A^{2*})$ is feasible for (3.10a)-(3.10i).

Let us next demonstrate the equivalence of the correspondent objective functions.

By the definition of A^{1*}, A^{2*}, the first double sum in (3.5a) can be rewritten as:

$$\sum_{t\in\mathbf{T}}\sum_{i\in\mathbf{J}}\left[(\Pi^c_{ti})A^{2*} + \Pi^e_{ti}(A^{1*} - A^{2*})\right] = \sum_{t\in\mathbf{T}}\sum_{i\in\mathbf{J}}\left[\Pi^e_{ti}A^{1*} + (\Pi^c_{ti} - \Pi^e_{ti})A^{2*}\right].$$

Also, since

$$\max\{0, DC_{ti} - EA^*_{ti}\} = -\min\{0, EA^*_{ti}, DC_{ti}\} = DC - \min\{DC_{ti}, EA_{ti}\}, \tag{3.13}$$

and because of (3.5f), the second double sum in (3.5a) can be expressed as:

$$\sum_{t\in\mathbf{T}}\sum_{i\in\mathbf{J}}\left[CI_{ti}(DC_{ti}-A^{2*}_{ti})+CR_{ti}(x^{*}_{ti}+EA^{*}_{ti})\right]$$

Hence both functions (3.5a) and (3.10a) take the same value at their corresponding feasible solutions $(x^*, s^*, z^*, EA^*, EP^*)$ and $(x^*, z^*, EA^*, A^{1*}, A^{2*})$.

(b) Consider now a feasible optimal solution $(x^{**}, z^{**}, EA^{**}, A^{1**}, A^{2**})$ to problem (3.10a)-(3.10i). Constraints (3.5b),(3.5d), and (3.5g) are trivially satisfied since they are identical to (3.10b), (3.10d) and (3.10e), respectively.

Defining $s^{**}_{ti} = x^{**}_{ti} - x^{**}_{t-1,i}$ and $EP^{**} = x^{**} + EA^{**}$, and making use of (3.10c), (3.10f), constraints (3.5c), (3.5e),(3.5f), and (3.5h) are readily satisfied by variables $x^{**}, s^{**}, EA^{**}, EP^{**}$. Thus the vector $(x^{**}, s^{**}, z^{**}, EA^{**}, EP^{**})$ is feasible for problem (3.5a)-(3.5h).

Now, we will prove that the objective functions (3.5a) and (3.10a) have the same values at their corresponding feasible solutions.

The coefficients for variables A^{1**}_{ti} in (3.10a) are $-\Pi^{e}_{ti}$, which are all nonpositive. Therefore, in a minimization process, variables A^{1**}_{ti} will take their maximum possible value in order to minimize the objective function. Since the growth of variable A^{1**} is only constrained by EA^{**}, then at the optimal point, we will have

$$A^{1**} = EA^{**} = \max\{0, EA^{**} - DC\} + \min\{EA^{*^*}, DC\}. \tag{3.14}$$

Similarly, the coefficients of variables A^{2**}_{ti} in (3.10a) are $\Pi^{e}_{ti} - \Pi^{e}_{ti} + CI_{ti}$, which are positive by the lemma's assumptions. The variables A^{2**}_{ti} will take then their maximum allowed values, which are given by constraints (3.10h) and (3.10i) and by (3.14), as

$$A^{2**}_{ti} = \min\{DC_{ti}, A^{1**}_{ti}\} = \min\{DC_{ti}, EA^{**}_{ti}\}. \tag{3.15}$$

Using both (3.14) and (3.15), we can rewrite the first double sum of the objective function (3.10a) using (3.14) and (3.15) as:

$$\sum_{t\in\mathbf{T}}\sum_{i\in\mathbf{J}}[\Pi^e_{ti}\left(\max\{0, EA^{**}-DC\}+\min\{EA^{**}, DC\}\right)+$$
$$+\left(\Pi^c_{ti}-\Pi^e_{ti}\right)\min\{DC_{ti}, EA^{**}_{ti}\}]=$$
$$=\sum_{t\in\mathbf{T}}\sum_{i\in\mathbf{J}}[\Pi^e_{ti}\max\{0, EA^{**}-DC\}+\Pi^c_{ti}\min\{DC_{ti}, EA^{**}_{ti}\}]\,;$$

which corresponds to the first double sum of the nonlinear problem objective function (3.5a).

As for the second double sum in (3.10a), it can be expressed (using (3.13), (3.15), and the definition of EP^{**}) as:

$$\sum_{t\in\mathbf{T}}\sum_{i\in\mathbf{J}}[CI_{ti}\left(\max\{0, DC_{ti}-EA^{**}_{ti}\}\right)+CR_{ti}EP^{**}_{ti}]$$

Using the last two equalities, we can show that both objective functions have equal values at the corresponding feasible solutions $(x^{**}, s^{**}, z^{**}, EA^{**}, EP^{**})$ and $(x^{**}, z^{**}, EA^{**}, A^{1**}, A^{2**})$.

(c) Let $((x^*, s^*, z^{**}, EA^*, EP^*))$ be an optimal solution for problem (3.5a)-(3.5h) with an optimal value of $h_3(x^*, s^{**}, z^{**}, EA^{**}, EP^{**}; \Pi^c, \Pi^e, DC) = h^{**}_3$. If the feasible solution $(x^*, z^*, EA^*, A^{1*}, A^{2*})$ constructed as in part (a) is not optimal for problem (3.10a)-(3.10i), then there exists a feasible vector $(x^{**}, z^{**}, EA^{**}, A^{1**}, A^{2**})$ such that $h_4(x^{**}, s^{**}, z^{**}, EA^{**}, EP^{**}; \Pi^c, \Pi^e, DC) = h^{**}_4 < h^*_3$. Then by (b), the feasible solution $(x^{**}, s^{**}, z^{**}, EA^{**}, EP^{**})$ to (3.5a)-(3.5h), where $EP^{**} = x^{**} + EA^{**}$, has the objective function value $h^{**}_3 = h^{**}_4 < h^*_3$, which is a contradiction. Therefore, for any optimal solution to the nonlinear problem, we can construct an optimal solution to the linear problem such that their objective functions optimal values are equal. The same argument can be used for the converse statement. This verifies both assertions of the lemma.

□

The upper level problem described in the preceding lemma uses the same amount of variables than problem $PR3'$, because it drops $2NP$ variables $(s, EP,)$ and adds

the artificial variables A^1, A^2. The number of constrains, though, is increased, since the additional artificial variables need more constraints to relate them to the other variables in the problem.

It is also worth to notice that both sets of variables s, EP are removed because of their merely explicative nature: variables s are completely determined by parameters x_0 and the variable x, whereas variables EP are determined by the sum of variables x and EA.

3.2.2 Lower Level Approximate Equivalent Problem

The lower level problem of problem $PR3$ has three nonlinearities that need to be removed:

1. The objective function is quadratic, reflecting the objective of the POC not to maximize the cash-out it obtains, but rather to minimize the absolute value of the cash-out to benefit both parties. The equation for z, in line (3.6h), contains also a piecewise quadratic term.

2. Several constraints include maximum and minimum operators that can only be considered linear in certain solution procedures that use variable x as a parameter while solving the lower level problem (unlike, for example, the Inexact Penalization Algorithm described in section 3.3.)

3. Variable q in the lower level problem is binary, though being unique, setting it as a parameter and solving two problems instead of 1 can effectively remove this nonlinearity.

We have already removed the quadratic terms listed in item 1 by adding the artificial variable d and noting the lack of modeling usefulness of variable δ. The remaining nonlinearities will be removed applying different techniques to obtain a lower level linear equivalent problem as we did with the upper level problem.

The variable q requires a rather complicated way to be removed from this level without loss of operability. Strictly speaking, we can fix q to either value and then solve two problems instead of one problem in the lower level, but with the added benefit of having only continuous variables in the lower level, which is something one would desire. While the prospect of this is already advantageous, there is still

one problem to consider: a direct algorithm would solve the lower level as many times as the upper level objective function is evaluated. This poses a great risk in the computational aspects of the problem, for the more complex the lower level is, the longer time the upper level will require to evaluate its outcome and finish its own optimization. The Inexact Penalization Algorithm proposed below would also have its own set of issues when having to solve two lower level problems instead of one in each function evaluation.

Instead of just fixing the value of variable q, we will give the control thereof to the upper level. This has a downside, as the actual decision over q is not in the hands of the NGSC, but in those of the POC. However, under lenient conditions, we can guarantee that the solutions of both problems are equal.

If the NGSC controls its final day imbalances so that it is impossible for the POC to have either $y_i < 0$ or $y_i > 0$ for all $i \in \mathbf{J}$, then the solution of the problem will remain equal no matter what level is in control of variable q. On the contrary, if the sum of all final day imbalances $x_{Ni} i \in \mathbf{J}$, in control of the NGSC, is small enough, then the POC has an actual choice of setting q to either allowed value. In this case, however, the difference between the solutions with q in the upper level and q in the lower level is considerably small, so this can be considered a good approximation, especially when additional terms—likely having a larger order—of interest are added to the upper level objective function. More on the manipulation of variable q, and the conditions for the equivalence of the problems with q in a different level, can be found in [23].

Finally, we address the existence of maximum and minimum operators in the constraints of problem $PR3'$. If the lower level is solved as an optimization problem parameterized by x, then this problem is linear, as the value of the expression $\max\{x_{N1}, 0\}$ is just a comparison *between parameters.* There is nothing nonlinear in comparing two parameters before solving the problem.

However, when using a different methods to solve the bilevel problem, like the Inexact Penalization Algorithm in section 3.3, is becomes evident that not every algorithm will have the upper level variables fixed when solving the lower level. It is then useful to formulate a fully linear version of the problem, considering the changes to the lower level already stated above.

The lower level problem can be reduced to an inexact linear formulation by adding $2N$ new variables, which render the constraints containing *max* and *min* operator linear. The downside (besides the fact that the new problem can be feasible even if the original is not,) is that the objective function is no longer equivalent to the penalization (i.e., the explicit interest for the POC). Also, this method is only inexactly equivalent, though the structure of the problem makes it easy to estimate a proper penalization constant $\mathbf{M}$, for the magnitude of the original objective function can be readily estimated and the penalization parameter assigned accordingly.

Lemma 3. *For a given vector x_N of final day imbalances, consider the linear programming problem:*

$$\textit{Minimize:} \quad g_4(y,u,v,z,d,\xi,\zeta;x,q) = d + \mathbf{M}\sum_{i\in\mathbf{J}}(\xi_i+\zeta_i) \tag{3.16a}$$

Subject to:

$$y_j = x_{Ni} + \sum_{i:i<j}(1-e_{ij})u_{ij} - v_{ij} + \sum_{k:k>j} v_{jk} - u_{jk};\ j\in\mathbf{J}; \tag{3.16b}$$

$$u_{ij} \le \xi_i, \quad i,j\in\mathbf{J}, i<j; \tag{3.16c}$$

$$u_{ij} \le \zeta_j, \quad i,j\in\mathbf{J}, i<j; \tag{3.16d}$$

$$v_{ij} \le \xi_j, \quad i,j\in\mathbf{J}, i<j; \tag{3.16e}$$

$$v_{ij} \le \zeta_i, \quad i,j\in\mathbf{J}, i<j; \tag{3.16f}$$

$$-\zeta_i \le y_i \le \xi_i, \quad i\in\mathbf{J}; \tag{3.16g}$$

$$\xi_i \ge x_{Ni}, \quad i\in J; \tag{3.16h}$$

$$\xi_i \ge 0, \quad i\in J; \tag{3.16i}$$

$$\zeta_j \ge -x_{Ni}, \quad j\in J; \tag{3.16j}$$

$$\zeta_j \ge 0, \quad j\in J. \tag{3.16k}$$

$$-M(1-q) \le y_i \le Mq;\ i\in\mathbf{J}; \tag{3.16l}$$

$$z = -\sum_{i\in\mathbf{J}} r_i y_i - \sum_{(i,j):i<j} v_{ij}b_{ij} + \sum_{(i,j):i<j} f_{ij}(1-e_{ij})u_{ij} \tag{3.16m}$$

$$-d < z < d \tag{3.16n}$$

$$y_i, z \in \mathbb{R},\ i\in\mathbf{J}; \tag{3.16o}$$

$$u_{ij}, v_{ij} \ge 0;\ i,j\in\mathbf{J}; \tag{3.16p}$$

where $\mathbf{M}$ *is a large enough scalar.*

1. *Let $(y^*,u^*,v^*,z^*,d^*,q;x)$ solve the original problem (3.6a)-(3.6l), with q^* fixed. Then there exists $(\xi*,\zeta^*)$ such that $(y^*,u^*,v^*,z^*,d^*,\xi^*,\zeta^*;x,q^*)$ is an optimal solution to (3.16a)-(3.16p).*

2. *Let* $(y^{**}, u^{**}, v^{**}, z^{**}, d^{**}, \xi^{**}, \zeta^{**}; x, q)$ *solve the linear problem (3.16a)-(3.16p). Then* $(y^{**}, u^{**}, v^{**}, z^{**}, d^{**}, q)$ *is an optimal solution to problem (3.6a)-(3.6l), when* q *is fixed.*

Proof. (a) Let $(y^*, u^*, v^*, z^*, d^*, q^*; x)$ solve the original problem (3.6a)-(3.6l). If we define $\xi_i^* = \max\{x_{Ni}, 0\}, \zeta_i^* = \max\{-x_{Ni}, 0\}, i \in \mathbf{J}$, then it is clear that ξ^* and ζ^* satisfy (3.16h)-(3.16k). Variables $y^*, u^*, v^*, q^*, z^*, d^*$ will trivially satisfy the constraints (3.6b), (3.2g)-(3.2j). If (3.6d), (3.6e) hold true for u^*, v^*, then the later variables will also fulfill (3.16c), (3.16e) and, as y^*, u^*, v^* satisfy (3.6b) and (3.6f), then (3.16d), (3.16f) are also valid for these u^*, v^*.

With ξ^*, ζ^* defined as above, constraint (3.6f) can be rewritten as (3.16g), therefore, as y^* satisfies the first, it will also yield the later. The vector $(y^*, u^*, v^*, z^*, d^*, \xi^*, \zeta^*)$ is then feasible for problem (3.16a)-(3.16p).

In view of lemma 1, the objective value of the linear problem coincides with

$$\begin{aligned} &g_4(y^*, u^*, v^*, z^*, d^*, \xi*, \zeta^*; x, q^*) = \\ &\quad = g_3(y^*, u^*, v^*, z^*, d^*, z^*, q^*; x) + \mathbf{M} \sum_{i \in \mathbf{J}} (\max\{x_{Ni}, 0\} + \max\{-x_{Ni}, 0\}) \end{aligned} \tag{3.17}$$

(b) Consider now an optimal solution $(y^{**}, u^{**}, v^{**}, z^{**}, d^{**}, \xi^{**}, \zeta^{**}; x, q)$ to problem (3.16a)-(3.16p). If $\mathbf{M}$ is large enough, a minimization process will force the variables ξ^{**}, ζ^{**} to take their minimum values in order to minimize their impact to the objective function. Thus, we will have

$$\xi_i^{**} = \max\{x_{Ni}, 0\}, \zeta_i^{**} = \max\{-x_{Ni}, 0\}; i, j \in P. \tag{3.18}$$

The variables ξ_i represent the *amount of gas that can be drawn from zone* i, whereas variables ζ_j represent the *amount of gas that can be deposited into zone* j. If ξ_i^{**}, $\left[\zeta_j^{**}\right]$ are 0, then u_{ij}^{**} $\left[v_{ij}^{**}\right]$ will be equal to 0 because of (3.16c)-(3.16f). Hence, u^*, v^* will then satisfy (3.6d), (3.6e).

With ξ^{**}, ζ^{**} defined in (3.18), constraint (3.16g) can be rewritten as (3.6f). Therefore, if the former is true for y^{**}, the later will also hold.

Let us now prove that u^{**}, v^{**} satisfy (3.6c). If $x_{Ni} \geq 0$, for any $i \in \mathbf{J}$, then expression (3.16g) becomes

$$-\zeta_i^{**} = 0 \leq y_i \leq x_{Ni} = \xi_i^{**}. \tag{3.19}$$

Constraint (3.6b) can be transformed as follows:

$$\sum_{j:j>i} u_{ij} + \sum_{k:k<i} v_{ki} = x_{Ni} + \sum_{j:j<i} (1 - e_{ji}) u_{ji} + \sum_{k:k>i} v_{ik} - y_i; i \in \mathbf{J}. \tag{3.20}$$

By (3.16d), (3.16f), the sums in the right hand side of the equation become 0, which yields:

$$\sum_{j:j>i} u_{ij} + \sum_{k:k<i} v_{ki} = x_{Ni} - y_i \leq x_{Ni} = \xi_i^{**} \tag{3.21}$$

Now, on the contrary, suppose $x_{Ni} \leq 0$ for an arbitrary $i \in \mathbf{J}$. In this case, the left-hand sums in (3.20), when combined with (3.16c), (3.16e), become zero:

$$\sum_{j:j>i} u_{ij} + \sum_{k:k<i} v_{ki} = 0 = \xi_i^{**} \tag{3.22}$$

Lines (3.21), (3.22) show that constraint (3.6c) is satisfied, hence the values $y^{**}, u^{**}, v^{**}, q, z^{**}, d^{**}$ are feasible for problem (3.6a)-(3.6l).

The objective value of the nonlinear problem is related to that of the linear problem as:

$$\begin{aligned} g_3(y^{**}, u^{**}, v^{**}, z^{**}, d^{**}, z^{**}, q; x) = \\ = g_4(y^{**}, u^{**}, v^{**}, z^{**}, d^{**}, \xi^{**}, \zeta^{**}; x, q) - \\ - \mathbf{M} \sum_{i \in \mathbf{J}} (\max\{x_{Ni}, 0\} + \max\{-x_{Ni}, 0\}) \end{aligned} \tag{3.23}$$

(c) We have shown that, for any feasible solution for either problem, one can find a corresponding feasible solution for the other problem with an explicit relationship between both problems' function values. It should also be clear that, if a vector solves one problem, so does its counterpart to the other problem, given the

consideration for q. Indeed, let

$$\kappa = \mathbf{M} \sum_i (\max\{x_{Ni}, 0\} + \max\{-x_{Ni}, 0\});$$

then if the nonlinear problem has an optimal solution with an objective function value σ^* strictly less than $((\tau^*) - \kappa)$, where τ^* is the optimal solution of the linear problem, then by part (a) of this proof, the linear problem has a feasible solution with objective function value $\sigma^* + \kappa < \tau^*$, which contradicts the optimality of τ^*. The same argument can be easily applied to the converse statement in 2.

This verifies both assertions 1 and 2 of the lemma and ends the proof. □

We have now reduced problem $PR3$ from its original of a bilevel mixed integer problem with quadratic and logical constraints and objective functions, to an approximate inexact equivalent bilevel linear problem, in the sense that the modified problem has the form:

$$\begin{aligned} &\text{minimize } c^1\mathbf{x} + d^1\mathbf{y} \\ &\text{Subject to } A^1\mathbf{x} \leq b^1 \\ &\qquad \text{and } y = \arg\min\{d^2\mathbf{y} | G^2\mathbf{y} \leq b^2 - A^2\mathbf{x}\}, \end{aligned}$$

where A^1, A^2 and G^2 are linear coefficient matrices, c^1, d^1, d^2, b^1 and b^2 are linear coefficient vectors, and variables $\mathbf{x}$ and $\mathbf{y}$ are the decision variables for the upper and lower level, respectively. We call it inexact because of its necessity of having appropriately selected parameters to guarantee the equivalence of the optimal solution.

This is very important because of the existence of linear tools for solving either level that can be employed, instead of relying on dedicated nonlinear methods.

The complete formulation of the obtained model, labeled $PR5(x, z, EA, A^1, A^2, y, u, v, \xi, \zeta; q, \Pi^c, \Pi^e, DC)$, is shown next, with q fixed to either 1 or 0, $\Pi^c_{ti} - \Pi^e_{ti} + CI_{ti} > 0 \, \forall t, i$, and $\mathbf{M}$ a scalar large enough:

$PR5(x, z, EA, A^1, A^2, y, u, v, \xi, \zeta; q, \Pi^c, \Pi^e, DC)$:

Upper Level:

$$\begin{aligned}
\text{Minimize:} \quad & h_5(x, z, EA, A^1, A^2; q, \Pi^c, \Pi^e, DC) = \\
& z - \sum_{t\in\mathbf{T}} \sum_{i\in\mathbf{J}} \left[\Pi^e_{ti} A^1_{ti} + (\Pi^c_{ti} - \Pi^e_{ti})\, A^2_{ti}\right] \qquad (3.24a) \\
& \quad + \sum_{t\in\mathbf{T}} \sum_{i\in\mathbf{J}} \left[CI_{ti}\left(DC_{ti} - A^2_{ti}\right) + CR_{ti}(x_{ti} + EA_{ti})\right];
\end{aligned}$$

Subject to:

$$x^L_{ti} \le x_{ti} \le x^U_{ti}, \quad t \in \mathbf{T}, i \in \mathbf{J}; \qquad (3.24b)$$

$$s^L_{ti} \le x_{ti} - x_{t-1,i} \le s^U_{ti}, \quad t \in \mathbf{T}, i \in \mathbf{J}; \qquad (3.24c)$$

$$x^L_t \le \sum_{i\in\mathbf{J}} x_{ti} \le x^U_t, \quad t \in \mathbf{T}; \qquad (3.24d)$$

$$0 \le EA_{ti} \le DM_{ti}, \quad t \in \mathbf{T}, i \in \mathbf{J}; \qquad (3.24e)$$

$$0 \le x_{ti} + EA_{ti} \le DM_{ti}, \quad t \in \mathbf{T}, i \in \mathbf{J}; \qquad (3.24f)$$

$$A^1_{ti} \le EA_{ti}, \quad t \in \mathbf{T}, i \in \mathbf{J}; \qquad (3.24g)$$

$$0 \le A^2_{ti} \le DC_{ti}, \quad t \in \mathbf{T}, i \in \mathbf{J}; \qquad (3.24h)$$

$$0 \le A^2_{ti} \le A^1_{ti}, \quad t \in \mathbf{T}, i \in \mathbf{J}; \qquad (3.24i)$$

Lower Level:

$$\text{Minimize:} \quad g_5(y,u,v,z,d,z,\xi,\zeta;x,q) = d + \mathbf{M}\sum_{i\in\mathbf{J}}(\xi_i+\zeta_i) \tag{3.25a}$$

Subject to:

$$y_j = x_{Ni} + \sum_{i:i<j}(1-e_{ij})u_{ij} - v_{ij} + \sum_{k:k>j} v_{jk} - u_{jk},\ j\in\mathbf{J}; \tag{3.25b}$$

$$u_{ij} \le \xi_i, \quad i,j\in\mathbf{J}, i<j; \tag{3.25c}$$

$$u_{ij} \le \zeta_j, \quad i,j\in\mathbf{J}, i<j; \tag{3.25d}$$

$$v_{ij} \le \xi_j, \quad i,j\in\mathbf{J}, i<j; \tag{3.25e}$$

$$v_{ij} \le \zeta_i, \quad i,j\in\mathbf{J}, i<j; \tag{3.25f}$$

$$-\zeta_i \le y_i \le \xi_i, \quad i\in\mathbf{J}; \tag{3.25g}$$

$$\xi_i \ge x_{Ni}, \quad i\in J; \tag{3.25h}$$

$$\xi_i \ge 0, \quad i\in J; \tag{3.25i}$$

$$\zeta_j \ge -x_{Ni}, \quad j\in J; \tag{3.25j}$$

$$\zeta_j \ge 0, \quad j\in J. \tag{3.25k}$$

$$-M(1-q) \le y_i \le Mq,\ i\in\mathbf{J}; \tag{3.25l}$$

$$z = -\sum_{i\in\mathbf{J}} r_i y_i - \sum_{(i,j):i<j} v_{ij}b_{ij} + \sum_{(i,j):i<j} f_{ij}(1-e_{ij})u_{ij} \tag{3.25m}$$

$$-d < z < d; \tag{3.25n}$$

$$y_i, z \in \mathbb{R},\ i\in\mathbf{J}; \tag{3.25o}$$

$$u_{ij}, v_{ij} \ge 0;\ i,j\in\mathbf{J}; \tag{3.25p}$$

We conclude this section with table 3.9, which summarizes the most important specifics in each of the problems we have herein formulated.

3.3 Solution Methods for the Bilevel Linear Problems

This section presents two algorithms, originally formulated in [58], whose purpose is to solve problem $PR1(x,s,y,u,v,z,q)$ (or rather, to provide an approximation to its solution.) An expansion to these algorithms will be provided so that they can be used to solve $PR5(x,z,EA,A^1,A^2,y,u,v,\xi,\zeta;q,\Pi^c,\Pi^e,DC)$. Both algorithms

Table 3.9. Comparison of the Different Deterministic Models.

Feature	Problem			
	$PR1$	$PR3$	$PR3'$	$PR5$
Upper Level Objective Function:	Linear	Nonlinear	Nonlinear	Linear
Lower Level Objective Function:	Quadratic	Quadratic	Linear	Linear
Upper Level Linear Constraints:	$5NP + 2N$	$10NP + 2N$	$10NP + 2N$	$13NP + 2N$
Upper Level Nonlinear Constraints:	0	0	0	0
Lower Level Linear Constraints:	$P^2 + 2P$	$P^2 + 2P$	$P^2 + 2P + 3$	$3P^2 + 5P + 3$
Lower Level Nonlinear Constraints:	$P^2 + 2P + 1$	$P^2 + 2P + 1$	$P^2 + 2P$	0
Upper Level Real Variables:	$2NP + 1$	$5NP + 1$	$5NP + 1$	$5NP + 1$
Upper Level Integer Variables:	0	0	0	0
Lower Level Real Variables:	$P^2 + 1$	$P^2 + 1$	$P^2 + 1$	$P^2 + 2P + 2$
Lower Level Integer Variables:	1	1	1	0
Times the problem is solved:	1	1	1	2
Exact Solution	Yes	Yes	Yes	No

assume that the upper level has control of variable q, along with the corresponding integrality constraint $q \in \{0, 1\}$, as stated in section 3.2, thus the approximation remarks.

Inexact Penalization Algorithm

The Inexact Penalization Algorithm (IPA) models the lower level of the problem as a Variational Inequality [50]. This is then added as a penalization term to the upper level, thus reducing the problem to a single-level problem with a minimization problem as a part of its objective function.

Consider problem $PR5$, and let $\mathbf{x} = (x, z, EA, A^1, A^2) \in \mathbb{R}^{4NP+1}$, $\mathbf{y} = (y, u, v, z, \xi, \zeta) \in \mathbb{R}^{P^2+2P+1}$. These vectors comprise the decision variables for

both the upper and lower levels the problem. Consider also the sets:

$$\Omega^u(q) = \{\mathbf{x} \in \mathbb{R}^{4NP+1} : (3.24b) - (3.24i) \text{ hold}\}, \tag{3.26}$$

$$\Omega^l(\mathbf{x}, q) = \{\mathbf{y} \in \mathbb{R}^{P^2+2P} : (3.25a) - (3.25l), (3.25o) - (3.25p) \text{ hold}\}, \tag{3.27}$$

each representing a subspace of the feasible space for both levels of problem $PR5$.

Note that the lower level feasible sets omits information regarding variables z and d, since the Inexact Penalization Algorithm incorporates them into the solution process. Also note that the sets Ω^u, Ω^l both depend on the value that variable q takes (the first directly, the second indirectly,) so there will be two feasible upper level sets and consequently, two problems to be solved.

In order to apply the IPA, the upper level objective function remains intact, but the algorithm requires defining function

$$F(\mathbf{y}) = \sum_{i \in \mathbf{J}} r_i y_i + \sum_{(i,j):i<j} b_{ij} v_{ij} - \sum_{(i,j):i<j} f_{ij}(1 - e_{ij}) u_{ij}. \tag{3.28}$$

Function $F(\mathbf{y})$ is equal to the negative of variable z as the latter appears in equation (3.25m.) It is used in the variational inequality

$$\langle -F(\mathbf{y}) \nabla F(\mathbf{y}), \hat{\mathbf{y}} - \mathbf{y} \rangle \geq, \quad \text{for all } \hat{\mathbf{y}} \in \Omega^l(\mathbf{x}, q). \tag{3.29}$$

According to Kalashnikov and Ríos-Mercado [59], the solution to the variational inequality (3.29), parameterized by the gap function

$$G(\mathbf{x}, \mathbf{y}, q) = \max_{(\hat{\mathbf{y}}) \in \Omega^l(\mathbf{x})} \langle -F(\mathbf{y}) \nabla F(\mathbf{y}), \hat{\mathbf{y}} - \mathbf{y} \rangle - \frac{1}{\alpha} \|\hat{\mathbf{y}} - \mathbf{y}\|^2, \tag{3.30}$$

coincides with the solution of the lower level of problem $PR5$. Function $G(\mathbf{x}, \mathbf{y}, q)$ is nonnegative over its domain, and $G(\mathbf{x}, \mathbf{y}, q) = 0$ only if $\mathbf{y}$ is the optimal solution to the lower level problem of problem $PR5$.

Using functions $F(\mathbf{y})$ and $G(\mathbf{x}, \mathbf{y}, q)$, we can formulate problem $PR5(x, s, EA, EP, A^1, A^2, y, u, v, z; q, \Pi^c, \Pi^e, DC)$ as the single-level optimization

problem

$$PR5i(\beta): \min_{\substack{\mathbf{x}\in\Omega^u(q)\\ \mathbf{y}\in\Omega^l(\mathbf{x},q)\\ q=\beta}} -F(\mathbf{y}) + \mu G(\mathbf{x},\mathbf{y},q).) \tag{3.31}$$

Problem $PR5i(\beta)$ is solved for both $\beta = 0, 1$, and the best solution from the NGSC point of view will be taken as the approximation to problem's $PR5$ optimal solution.

Direct Algorithms

Besides the Inexact Penalization Algorithm, we will also consider a Direct Method (DM.) This technique will minimize function $h_5(x, s, z, EA, EP, A^1, A^2, q, \omega)$, subject to the constraints given for the upper level; however, every time the upper level objective function is evaluated, the numerical solver will calculate the value of the function as the optimal response of the lower level to the current iteration's values for x, q, z, complying with the lower level constraints. Algorithms 1 and 2 describe this process more thoroughly.

Algorithm 1. *Upper Level Direct Algorithm*

- *Load the corresponding parameters and form the appropriate constraint matrices and vectors.*
- *For each value of* $q \in \{0, 1\}$*, do:*
  - *Solve problem (3.24a)-(3.24i) using the function* `linprog` *and* $\sum_i r_i x_{Ni}$ *as the objective function. If this problem is unfeasible, end and report the unfeasibility. Otherwise, label the resulting preliminary optimal point as* $\hat{x}$*.*
  - *Solve problem (3.24a)-(3.24i) using function* `fmincon`*, with* $\hat{x}$ *as the starting point required by the algorithm. At every function evaluation performed by the optimization function, the value of* z *will be evaluated as dictated by algorithm A2 below, using the currently proposed upper level variables as parameters for the lower level problem. The resulting upper level optimal vector will be labeled* $\mathbf{X}^*$*, and the optimal value,* $\mathbf{Z}^*$

- *Solve the lower level problem using* `linprog` *and* $\mathbf{X}^*$ *as parameters and retrieve the lower level optimal point* $\mathbf{Y}^*$.
- *Report the optimal vectors and values for both levels.*

Algorithm 2. *Lower Level Direct Algorithm*

- *Use the vector* $\mathbf{X}$ *passed by the upper level optimization as a parameter to form the appropriate constraint matrices and vectors.*
- *Use function* `linprog` *to solve problem (3.25a)-(3.25p). If the problem is feasible, report the value of* z *(and not that of the objective function* $g_5(y,u,v,z,q,d,b,c) = d + \mathbf{M}\sum_{t,i}(\xi_{ti} + \zeta_{ti})$*) to the upper level as the result of the algorithm; otherwise, report a big number* $\mathbf{M}$*, which is increased at every function evaluation by a fixed amount (e. g.* $\mathbb{M} \leftarrow \mathbb{M} + \mathbf{m}$*).*

3.3.1 Proposed Numerical Solution Methodology

Problem $PR5$ will be solved using both the Inexact Penalization Algorithm and the Direct Method, coded in Matlab®. Matlab, a high-level computer language and software specialized in mathematical programs, provides a good amount of built-in specialized functions and mathematical oriented programming capabilities. Primarily, we will make use of functions `fmincon`, a constrained nonlinear minimization function which uses either an active-set algorithm or an interior point algorithm to provide a solution to a nonlinear constrained problem; and function `linprog`, which uses either a Linear Interior Point Solver (for large scale problems,) or the Dantzig's Linear Programming Simplex (for medium scale problems.) Both functions are included in the Matlab add-in called Optimization Toolbox ™. [94]

There are, however, two drawbacks when working with Matlab's Optimization Toolbox functions, especially `fmincon`: it was designed to work with continuous, continuously differentiable functions, and (related to this) `fmincon` it has been observed to be very sensitive to the initial point feed the solver.

We plan to study the effectiveness of Matlab's solvers focusing in the following points:

- Procuring a good starting point to Matlab's minimization functions in the upper level.

- Procuring a good starting point to Matlab's minimization functions in the lower level.

- Using a heuristic approach to solve functions G in the IPA.

The first item involves obtaining values for each x_{ti} (which will automatically generate values for s) that will lead to a good, feasible point in the variables' domain. Viable options to consider include: setting all x initial values equal to zero, or set equal to the lower bound, or to the upper bounds, or simply chose a vector of zeros as the starting search point. We attempt to analyze the running times and the convergence (i.e., quality of the obtained optimal value) for several starting point options for problem $PR5$.

The second and third items have very similar implementations. For the second item, we plan to take advantage of the lower level's particular structure once the upper level variables have been determined (something that happens every iteration in the solution algorithm.) Once the values for x are set, the lower level can be seen as a generalized transportation model, and a feasible solution can be proposed using algorithm 3:

Algorithm 3. *Lower Level Feasible Point Algorithm:*

1. *Determine if the problem is solved or unfeasible (all final imbalances are nonnegative or nonpositive.) If it is, go to 9, if not, go to 2.*

2. *Set* $y_i = x_{Ni} \forall i$.

3. *Select a pool zone among the positive imbalances,* i^*.

4. *Select a zone among the negative imbalances,* j^*.

5. *Calculate the gas transported as* $g_t = \min\{y_{i^*}, -y_{j^*}\}$.

6. *Set variable* $u_{ij} = g_t$ *(*$v_{ij} = g_t$ *if* $j < i$*)*

7. *Subtract* g_t *from* y_{i^*} *and add it to* y_{j^*} *(reduce* g_t *by the correct amount* e_{ij} *if* $i < j$ *before the addition.)*

8. *Determine if the problem is solved or unfeasible (all final imbalances are nonnegative or nonpositive.) If it is, go to 9, if not, go to 3.*

9. *Report the final imbalances y_i, the transportation volumes u_{ij}, v_{ti}, and calculate function $F(y, u, v)$*

Algorithm 3 will eventually provide a feasible point (if there is such) for the lower level problem. It can be calculated quickly and easily enough to be inserted in the argument minimization function of `fmincon` and obtain a good, nearest-to-zero value of function $F(y, u, v)$. This in turn can is used as a starting point for a numerical minimization of function G, instead of using, for example, the last day imbalances x_{Ni} as starting points for variables y_i, and zeros for variables u_{ij}, v_{ij}.

As we have said, the third item can be achieved in similarly. Judging by computational economy, one can elect to generate several feasible points $\mathbf{y}_k$, and solve (in an iterative process) the variational inequality (3.29), considering the set of the generated solutions as the whole solution space, instead of using another minimization procedure for function G_5. The obtained point will likely be a good, feasible approximation to the lower level problem, providing (this is to be analyzed) shorter running times.

Once the set of different solution method variations have been defined, we will run several problem instances, using the figures from the Energy Information Administration [28] monthly data for residential natural gas consumption and prices to form matrices DC, Π^c, Π^e, and the problem generating framework discussed in the next section. Once the data from the experimentation has been obtained, results for each method will be evaluated in order to determine the best options among the many given in this chapter.

3.4 Instance Generation for the Deterministic Problems

The pseudocode in algorithm 4 describes the computer routine used in the generation of instances to test problem $PR5$'s performance.

Algorithm 4. *Deterministic Problem Instance Generation*

- *Set the number of days/periods N and pool zones/states P.*

- *Randomly select from the proper data matrices a list of states. Select then the amount of information from those matrices to be considered historical. Form with this data matrices* Π^c, DC*, and calculate suitable values for* Π^e

- *Enter the maximum and minimum parameter reference for initial imbalances* $PM1, Pm1$*. Calculate the initial imbalance vector* IIV *parameter as random independent uniform numbers between* $PM1, Pm1$*.*

- *Enter the maximum and minimum parameter reference for final imbalances* $PM2, Pm2$*. Calculate a feasible final imbalance vector* FIV *as random independent uniform numbers between* $PM2, Pm2$*.*

- *Calculate a feasible imbalances matrix* FIM *for the remaining days as uniform as uniformly distributed intervals between the correspondent entries of* IIV *and* FIV*, with average interval length for pool zone* i *of* $(FIV_i - IIV_i)/P$*.*

- *Calculate the maximum and minimum imbalance matrix as the feasible imbalance matrix* FIM *plus a fixed reference parameter and plus a uniformly distributed random vector.*

- *Calculate the maximum and minimum daily imbalance sum as the feasible daily imbalance sum (from matrix* FIM*) plus a fixed reference parameter and plus a uniformly distributed random vector.*

- *Obtain feasible maximum and minimum imbalance swings from matrix* FIM *and set the maximum and minimum imbalance swing figures to form the correspondent matrices.*

- *Calculate the lower level forward and backward matrices* f, b, e *as 1 plus the triangular upper part of an inverse Poisson randomly generated matrix of suitable* λ *parameter.*

- *Enter a linear cost vector for the lower level objective function.*

The algorithm will generate a batch of problems of any size and with several variations to the parameters. Even with the same reference parameters, the algorithm can generates very different problems due to the various instances where

randomness is employed. Generating the parameters needed for only one level of the problem, we can obtain suitable instances for testing the individual levels of the problem as it is done in chapter 5.

Chapter Summary

In order to obtain a solution to the natural gas optimization model described in chapter 1, we present in this section a bilevel optimization model originally developed by Ríos-Mercado et al [88]. Since the original formulation did not include, from our point of view, important elements of the NGSC-POC subsystem, such as the sales and prices the NGSC is concerned with, we went on to expand the original model, until we obtained a more elaborate and refined bilevel model.

Theoretical work was applied to reduce thus obtained model into an inexact bilevel linear approximation. The process from the original problem $PR1$ to the latest $PR5$ is described in sections 3.1, and 3.2, along with a table that summarizes the main characteristics of the formulations developed previously to problem $PR5$. Finally, the numerical algorithms and solution techniques intended to be used to solve computationally problem $PR5$ are shown in section 3.3.

The deterministic model $PR5$ developed here is intended to be a tool for:

- Providing a starting point to model a stochastic model of the same problem, so that unknown parameters like prices and demands for the operative period can be estimated and acted upon, planning the operations of the NGSC based on stochastic versions of variables x, EA, etc; and
- Being used as an analysis model to evaluate the solution techniques before formulation the more complex stochastic bilevel models.

CHAPTER 4
STOCHASTIC MODELS

The purpose of this chapter is to expound an expansion of problem $PR5$, given in the last chapter, by adding stochastic programming elements to the deterministic formulation. The final model in section 4.4 is a multi-stage stochastic model with recourse, with variables x and EA and the parameters Π^e, Π^c and DC vary according an underlying ternary tree.

We first describe, in section 4.1, the time series analysis and forecast techniques used to obtain an estimation of price and consumption parameters Π^c and DC. Section 4.2 provides regression analysis between the data in matrices Π^c and DC, as well as a method to identify which pool zones can be clustered in sub-markets by means of their demand and price time series. The construction of the scenario tree used in the stochastic models that defines the ways the forecasting of prices and demands will be done is shown in section 4.3.

The data obtained will be used to formulate a bilevel stochastic model. This model is expected to better reflect the possibilities for the NGSC and serve as a better decision tool for the upper level decision maker.

4.1 Time Series Forecast Analysis

We have stated before that the deterministic models would use the American Government EIA database of residential monthly prices and consumptions for each state of the American Union, as well as the District of Columbia and the country taken as a whole, which will be called states too from here onwards. Each month's entry in the EIA's database will be considered here as the data corresponding to one day or period (it may indeed represent any other time unit; we call them days only for personal choice.)

The EIA's database contains monthly data for the 52 states from January 1989 to October 2007.

In order to formulate a stochastic version of problem $PR5$, we need to define the nature of the stochastic parameters that will be in use. The times series used as parameters in the deterministic variant of the model will no longer appear as such

in the stochastic version. Instead, they will be the arguments of the forecasting techniques in this section. The values forecasted are actually estimations of random variables, hence their stochastic nature.

4.1.1 Time Series

Let us begin by describing the Time Series notation we will use throughout this work.

Definition 1. *A time series* $S = \{s_t\}_{t=1}^{\mathbf{T}}, s_i \in \mathbb{R} \forall i$ *is an ordered set of* $\mathbf{T}$ *real values indexed by* t*, called the* time parameter*. [15] The value* s_t *of the series for a given value of* t *is called the* t*-th observation of the series, and the series length is the number of observations, equal to* $\mathbf{T}$ *when the series' first observation occurs at time 1.*

In this work, we may refer to series S as simply $\{s_i\}$ when no ambiguity can arise from the starting or ending observation of the series.

Another useful note we make when talking about time series is the fact that we may refer to certain matrices and vectors as time series and vice-versa. Thus, time series $S = \{s_t\}_{t=1}^{\mathbf{T}}$ can also be shown as *vector* $S = (s_1, s_2, ..., s_i, ..., s_{\mathbf{T}})$.

Equivalently, we may refer to column j of matrix $A = (a_{ij}) \in \mathbb{R}^{\mathbf{I},\mathbf{J}}$ as the time series $\{a_{ij}\}_{i=1}^{\mathbf{J}}$, and vice-versa.

4.1.2 Forecast Modeling

From the point of view of the NGSC, knowledge about future prices and demand faced is needed in order to form the parameter matrices D^c, and Π^c, since this data is not available at day 0 in the upper level decision process. In most cases, the NGSC does not have certainty over those parameters. It does have, though, the capacity to perform mathematical forecasts, predictions of the future prices based in the observed data.

Because of this, it is desirable to provide our stochastic models with means of obtaining these forecasts.

Definition 2. *Let* $S = \{s_t\}_{t=1}^{\mathbf{T}}$ *be a time series of length* $\mathbf{T} \in Z^{++}$*. The* ***forecast operator*** ϕ_γ^κ *is such that* $\phi_\gamma^\kappa S$ *is the time series resulting from forecasting the first*

$\kappa \in Z^{++}$ values of the $0.5+\gamma$ percentile of the prediction band of the historical time series S. The operator can be applied successively, so that $\psi^{\kappa_n,...,\kappa_1}_{\gamma_n,...,\gamma_1} S = \phi^{\kappa_n}_{\gamma_n} \circ \phi^{\kappa_{n-1}}_{\gamma_{n-1}} \circ ... \circ \phi^{\kappa_1}_{\gamma_1} S$.

If we define the vectors $\vec{\kappa} = (\kappa_1, \kappa_2, ..., \kappa_n,)\vec{\gamma} = (\gamma_1, \gamma_2, ..., \gamma_n)$, the cumulative form of operator ϕ^{κ}_{γ}, $\Phi^{\vec{\kappa}}_{\vec{\gamma}}$, is such that

$$\begin{aligned}\Phi^{\vec{\kappa_1}}_{\vec{\gamma_1}} S &= \left(\phi^{k_1}_{\gamma_1} S,\ \phi^{k_2}_{\gamma_2} \circ \phi^{k_1}_{\gamma_1} S,\ ...,\ \phi^{k_n}_{\gamma_n} \circ ... \circ \phi^{k_1}_{\gamma_1} S\right) \\ &= \left(\psi^{k_1}_{\gamma_1} S,\ \psi^{k_2,k_1}_{\gamma_2,\gamma_2} S,\ ...,\ \psi^{k_n...k_1}_{\gamma_n...\gamma_1} S\right). \end{aligned} \tag{4.1}$$

A quick note on the nature of operators $\psi^{\kappa_n,...,\kappa_1}_{\gamma_n,...,\gamma_1} S$ and $\Phi^{\vec{\kappa_1}}_{\vec{\gamma_1}} S$: the first will produce κ_n predictions, using the last $\kappa_1 + \kappa_2 + ...\kappa_{n-1}$ as historic. The second operator will produce $\kappa_1 + \kappa_2 + ...\kappa_n$ predictions, chained together in a logical manner.

Definition 3. *The κ-step, γ prediction from time h, denoted by $\bar{\phi}^{\kappa}_{\gamma}\{S_t\}^h_1$, is the last ($\kappa$-th) observation of the time series $\phi^{\kappa}_{\gamma}\{S_t\}^h_1$. In case $\gamma = 0$ and h coincide with the final observation of series S, we will put simply $\bar{\phi}^{\kappa} S$ and call it the κ-th step prediction of time series S.*

Operator ϕ, as well as its successive and cumulative versions, do not make any reference to the method used to produce the forecast. This may even be an extension of the rows of a certain historical vector representing the time series. As a result, the forecasts operators serve formal modeling purposes only, and not calculations. This latter process will be explained in section 4.1.3

4.1.3 Numerical Predictions

In this section, we present the numerical analysis that will be performed to obtain predictions from the data found in the EIA databases for Natural Gas residential prices and consumptions.

First, we show the two forecast algorithms, as described in [15], that will be used to obtain forecasts $\phi^{\kappa}_0 S$ from a certain time series i.e. the mean expected figures from a time series. Based in the residuals from the predictions, we then produce prediction bands $\phi^{\kappa}_{\alpha} S$.

The ARAR Algorithm

The first step in the ARAR forecasting algorithm is the memory shortening of the series. This basically consists in determining which of the next three alternatives better describe the time series we are using, S, and iteratively modifying series S until case 3 is satisfied.

1. $\tilde{S}_t = S_t - \hat{\zeta}(\hat{\tau})S_{t-\hat{\tau}}$ (S_t is long-memory),
2. $\tilde{S}_t = S_t - \hat{\zeta}_1 S_{t-1} - \hat{\zeta}_2 S_{t-2}$ (S_t is moderately long-memory),
3. S_t is short-memory.

The process to decide whether S_t belongs to case 1, 2 or 3 depends on the values of $\hat{\zeta}$ and $\hat{\tau}$ in the expression:

$$\begin{aligned}(\hat{\zeta}, \hat{\tau}) &= \underset{\tau=1,...,15;\zeta>0}{\arg\min} \left(\mathrm{ERR}(\zeta, \tau)\right) \\ \mathrm{ERR}(\zeta, \tau) &= \frac{\sum_{t=\tau+1}^{n} \left[S_t - \zeta S_{t-\tau}\right]^2}{\sum_{t=\tau+1}^{n} S_t^2}\end{aligned} \tag{4.2}$$

Once those values are calculated, they are evaluated (see the reference) and S is classified as either long, medium-long, or short memory. In case the time series S turns out to belong to cases 1 or 2, we apply again the parameter fitting in 4.2 until we have that S_t belong to case 3—this process, reportedly, does not usually takes more than 3 attempts.

Let $\{C_t\}$ be the memory shortened series obtained above, and let $\bar{C}$ be its sample mean ($(\sum_{t=1:\mathbf{T}} C_t)/\mathbf{T}$.) Construct the new series $\{Z_t\} = \{C_t\} - \bar{C}$, and fit an autoregressive process:

$$Z_t = C_t - \bar{C}_t = \eta_1 Z_{t-1} + \eta_{l_1} Z_{t-l_1} + \eta_{l_2} Z_{t-l_2} + \eta_{l_3} Z_{t-l_3} + z_t, \tag{4.3}$$

where z_t is distributed as white noise with mean 0 and variance σ^2. The lags l_1, l_2, and l_3 are calculated by solving the Yule-Walker equations on η, η_1, η_2, and η_3 [14].

Finally, the h predictions $\bar{\phi}^1 S, \bar{\phi}^2 S, ..., \bar{\phi}^h S$ are calculated by the ARAR method with

$$\bar{\phi}^h S = -\sum_{j=1}^{1+l_3} \left[\xi_j \bar{\phi}^{h-j} S\right] + \eta' \eta\, \bar{S}, \quad h \geq 1, \tag{4.4}$$

where η' is the transpose of the parameter vector η.

In expression (4.4), coefficients ξ_j are calculated using the coefficients η_i and ζ_j found in steps 1 and 2. [15]

The process described before calculates the h-step mean prediction, namely $\bar{\phi}_0^h S$. In order to calculate the prediction bands $\bar{\phi}_\alpha^h S$, we need first to have the h-step MSE estimate:

$$\begin{aligned} MSE[(S_{n+h} - \bar{\phi}_0^h\{S_t\}_1^n)^2] &= \sum_{j=1}^{h-1} \tau_j^2 \sigma^2, \\ \tau_0 = 1; \sum_{j=0}^{n} \tau_j \xi_{n-j} &= 0, \quad n = 1, 2, ...; \end{aligned} \tag{4.5}$$

where σ^2 is the sample variance of series $\{St\}_1^n$. The bands are thereafter calculated assuming a normal distribution for the forecasts with mean $\bar{\phi}_0^h S$ and variance equal to the mean square error for that observation and step.

The Seasonal Holt-Winters Method

The Seasonal Holt-Winters (SHW) algorithm assumes that a seasonal period d is given for the time series $\{S_t\}$, and uses it to calculate series of values that parameterizes the whole time series.

The predictions will be based in the calculations of series of coefficients $\hat{a}_i, \hat{b}_i$, and $\hat{c}_i$, for $i = 1, 2, ..., \mathbf{T}$, produced by equations

$$\hat{a}_{d+1} = S_{d+1}, \tag{4.6}$$

$$\hat{b}_{d+1} = (S_{d+1} - S_1)/d, \tag{4.7}$$

$$\hat{c}_{d+i} = c_i = S_i - (S_1 + \hat{b}_{d+1}(i-1)), \quad i = 1, 2, ..., d+1. \tag{4.8}$$

We then form the h-step predictor from observation n as

$$\bar{\phi}^h\{S_t\}_1^n = \hat{a}_n + \hat{b}_n h + \hat{c}_{n+h}, h = 1, 2, ... \tag{4.9}$$

For $\mathbf{T}, \mathbf{T}+1, ...$, coefficients $\hat{a}_i, \hat{b}_i$, and $\hat{c}_i$ are calculated with the recursions:

$$\hat{a}_{n+1} = \mathbf{a}(S_{n+1} - \hat{c}_{n+1-d}) + (1-\mathbf{a})(\hat{a}_n + \hat{b}_n,)$$
$$\hat{b}_{n+1} = \mathbf{b}(\hat{a}_{n+1} - \hat{a}_n) + (1-\mathbf{b})\hat{b}_n,$$
$$\hat{c}_{n+1} = \mathbf{c}(S_{n+1} - \hat{a}_{n+1}) + (1-\mathbf{c})\hat{c}_{n+1-d},$$

with initial conditions

$$\hat{a}_{d+1} = S_{d+1},$$
$$\hat{b}_{d+1} = (S_{d+1} - S_1)/d,$$
$$\hat{c}_i = S_i - (S_1 + \hat{b}_{d+1}(i-1)), \quad i = 1, 2, ..., d+1.$$

The parameters $\mathbf{a}, \mathbf{b}$, and $\mathbf{c}$ may either be set arbitrarily, or calculated in such a way that the MSE of the one-step predictors for $\{S_t\}_{t=d+2}^{n}$ is minimized i.e.

$$\sum_{t=d+2}^{n} \sqrt{\frac{[S_t - P_{t-1}S_t]^2}{n-d-3}} \to \min_{\mathbf{a},\mathbf{b},\mathbf{c}} \tag{4.10}$$

The MSEs for the h-step predictors need to be found in order to calculate the prediction bands for the forecasting. For every h required, we will take a sample of the h-step prediction of series $\{S_t\}$, calculate the sample mean (μ_h) and sample variance $\hat{\sigma}_h^2$ of these predictions, and use the sample standard deviations and the normal distribution $N(\bar{\mu}_h, \sigma_h^2)$ to calculate the desired α-prediction bands.

Forecast Analysis

The usage of either algorithm described above requires the data series to have certain properties, thus we need to validate and report some assumptions regarding the data we are using.

We have stated that the $0.5+\gamma$ prediction bands $\phi_\gamma^h S$ of series S are calculated using the MSEs $\hat{\sigma}$ as estimators of the series errors sample standard deviations: the h-step γ prediction band is calculated as $\bar{\phi}_0^h S + \hat{\sigma} z_\gamma$, were $z_{0.5+\gamma}$ is the inverse standard normal value for the probability $0.5+\gamma$. This requires the assumption that the h-step errors are distributed $N(0, \hat{\sigma}_h^2)$. We will need to calculate the sample

means $\bar{\mu}_h$ and MSEs $\hat{\sigma}_h$, validate that $\hat{\mu}_h = 0$ through a hypothesis test, and provide proofs (through normal probability plots) that the errors are normally distributed. We must note, however, than assuming normality is commonplace in time series analysis: the software ITSM2000 v7.0 [15] makes this assumption automatically when prompted to plot prediction bands.

Autocorrelation is another important parameter characterizing a time series: a high (near-1) autocorrelation at lag h, indicates that the series involved is separating itself from a white noise process (which is basically an *iid* random time series,) and is correlated with the values h observations before and after it.

Definition 4. *Let S be a time series, $h \in \mathbf{Z}^+$. The **autocorrelation function** ρ of series S at lag h is calculated as*

$$\rho_S(h) = \frac{\mathrm{Cov}(S_t, S_{h+t})}{\mathrm{Cov}(S_t, S_t)}, \tag{4.11}$$

where $\mathrm{Cov}(S_{t_1}, S_{t_2})$ is the covariance of series $\{S_t\}_{t=1}^{n-(t2-t_1)}$ and $\{S_t\}_{t=t_2}^{n}$), seen as random variables.

While autocorrelation is good to have among time series values, it is not when analyzing the series' forecast residuals.

We need to prove that our time series have near-1 levels of autocorrelation. Likewise, proofs of low autocorrelation for the h-step residuals are needed to validate the forecast process.

4.2 Time Series Regression Analysis[1]

Besides forecasting, another aspect of the time series analysis in our work is the study of the price and consumption relation in a given state or pool zone. The existence of an explicit relationship between a price time series and the corresponding demand time series for a given state will greatly help in reducing the information needed and have a leaner modeling. Section 4.3 describes how the regression analysis here relates to our stochastic bilevel problem via the scenario tree.

[1]The contents of this section appear in the authors' [56]

Furthermore, from an economic standpoint, it is convenient to have an expression for price in terms of the demand (output) a firm or group of firms deliver to a particular market. This makes the consumption faced by the NGSC (or any other firm, for the matter) the decision variable, and not the price. Dealing with the demand instead of with the price is good since the former is usually analytically and computationally easier to work with. The Inverse Demand Function (IDF) is a generalized "Price Function", but for the most part is accepted simply as the price corresponding to a certain consumption (this is especially true for markets with low price elasticity.)

Depending of the type and scope of the analysis, the inverse demand function can be modeled differently, based in assumptions regarding continuity, differentiability, range, etc. In its simplest form, a IDF is a non-increasing function on demand, which means that lower values for the demand yield high values for the IDF and conversely. Negative coefficients and negative exponents are then usually included in an IDF expression. [93]

We will make use of a multiple linear regression model to fit the price time series to their respective demand time series. The model is based in the facts that (1) the amplitude of the price time series increases with time; (2) the cumulative average of the price series also increases with time; and (3) demand time series present the same seasonality than the price time series. Research has been profusely done linking natural gas price and demand/consumption. The models reviewed often show:

- A focus in estimating the demand-price elasticity (as well as other econometric data,) and consequently

- Usage of logarithmic expressions for the parameters the models linking price and demand/consumption, something that eases the elasticity analysis [41] and econometrically representing a steady growth market.

These characteristics appear in [8, 4, 66, 63, 110, 6]; however, despite this, we have found that models with pragmatic objectives such as demand estimation do not necessarily rely on logarithmic transformations of the regression variables.

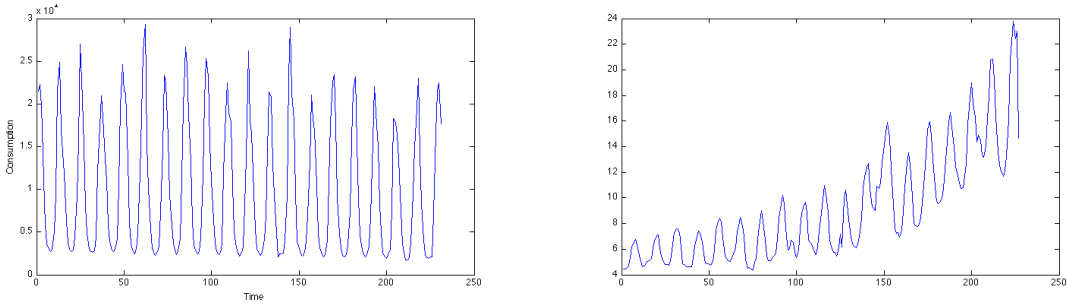
Instead, we have found linear expressions [1, 67] (such as the one employed by the EIA,) and discrete-exponential analysis [105].

The method described in the next section is called Individual Multiple Linear Regression (IMLR,) since a particular set of parameters is calculated for each individual state involved in the regression. Next section will discuss a pooled regression method developed to form *clusters* of states with similar regression parameters. These clusters will form the state sets used in the optimization problems described in section 4.4.

4.2.1 Individual Multiple Linear Regression

Let P be the total number of states; $\mathbf{T} = \{1, 2, ..., m\}, m \in \mathbb{Z}^+$, the number of observations per time series—months, in this case—; $I = \{1, 2, ..., P\}$ the set of states (up to a maximum of 52); $t \in \mathbf{T}$, the (discrete) time parameter; $\{D_{i,t}\}$, the time series corresponding to the natural gas demand in state $i \in I$ at time $t \in \mathbf{T}$; and $\{\Pi_{i,t}\}$, the time series corresponding to the price of natural gas in state i at time t. When no confusion may occur, we will assume $\{\Pi_i\} = \{\Pi_i\}_{t=1}^{m}; t \in \mathbf{T}$, for $i \in I$.

Consider figure 4.1. It shows the monthly consumption and price of natural gas in the state of Missouri, from January 1989 to October 2007. The shapes of the series are important because demonstrate the reasons why we have developed the models below.



(a) Consumption (b) Price

Figure 4.1. Missouri Monthly Prices and Consumption. Source: Energy Information Administration.

The regression model created will fit a curve estimate for $\{\Pi_i\}$ with both t and $\{D_i\}$ as independent variables. It will be formed first by a quadratic polynomial in t

(including the intercept,) such as

$$a_0 + a_1 t + a_2 t^2. \tag{4.12}$$

The amplitude of the price series' wave will be determined by a polynomial with the consumption variable multiplied by an expression like the one in (4.12):

$$D_{ti}^{-1}\left(a_3 + a_4 t + a_5 t^2\right). \tag{4.13}$$

Merging 4.12 and 4.13, we obtain the IMLR model that estimates the price TS's $\{\Pi_i\}$ as a function of demand:

$$\hat{\Pi}_{ti} = a_0^i + a_1^i t + a_2^i t^2 + D_{i,t}^{-1}\left(a_3^i + a_4^i t + a_5^i t^2\right) + e_i^*, \quad \forall t \in T, \forall i \in I, \tag{4.14}$$

where $e_i^* \sim N(\mu_i^*, \sigma_i^*)$.

This expression determines a matrix of parameters estimates $A \in \mathbb{R}^{P\times 6}$. The usual regression analysis (residuals independence, randomness, and normality; parameter significance; and the like) will be provided in the stochastic models' results chapter to support our regression models.

We have modeled the natural gas residential consumption/price scenario with a general formula that represents the main aspects of the market behavior. Consumption is more or less resilient with respect to time, always reaching similar heights and lows, whereas prices raise nonlinearly [62]. Though population grows, advances of technology helps reducing the impact of the number of consumers in the consumption. Raising prices also help in diminishing a marked growth [82, 30]. The increase of prices is usually described as exponential, which is understandable. However, in a limited period of analysis (as it is ours,) we can instead consider a quadratic mean growth rate represented by equation (). This kind of accelerated growth was not seen before the Natural Gas Act took place, bringing about the changes described in chapter 1. [69]

4.2.2 Pooled Price to Demand Regression

The regression fits described in the preceding section are used in the optimization model for three purposes: (1) reducing the number of time series involved in a given problem instance, thus (2) reducing the number of forecasts performed while in the stochastic solution of the problem, and (3) apply the concept of IDF to our problem, characterizing the prices as functions of the consumption.

During the analysis, the question of whether we could use the same regression function for number larger than one was posed. Such action would reduce even further the amount of data used in the problem and would open the chance for us to characterize state classes, groups, or clusters. In this way we can define optimization processes and/or economical market analysis in the formed clusters of states with similarly behaved sets of parameters.

We propose to split the P price-consumption pairs of time series into groups, with the members of each group sharing a common set of regression parameters. The pooled data from the groups would then be regressed at the same time, creating *pooled regressions.*

More interestingly, the existence of a common regression function among a number of states or pool zones will also validate the assumptions in section 4.3, justifying a common behavior among different time series corresponding to several pool zones when prices and demands all go higher or lower than the mean estimates for the pool zones involved.

Let $\mathbf{I} = \{I_1, I_2, ..., I_K\}$, be a partition of the state-set I, and consider the model:

$$\begin{gathered}\hat{\Pi}_{i,t} = a_0^k + a_i^k t + a_2^k t^2 + D_{i,t}^{-1}\left(a_3^k + a_4^k t + a_5^k t^2\right) + e_k^{**}, \\ \forall t \in \mathbf{T}, \forall i \in I_k, k = 1, 2, ..., K,\end{gathered} \tag{4.15}$$

where $e_i^{**} \sim N(\mu_i^{**}, \sigma_i^{**})$.

Note that this model—called Pooled Multiple Linear Regression (PMLR) model—has K sets of regressions for every parameter except for the intercept a_0^i—which is allowed to be different for each state—whereas (4.14) has P. On the other hand, it is expected and understandable that grouping the time series into a single regression will reduce the goodness/significance of the regression. There

might also be the possibility that some series cannot be "reasonably" matched with none of the others.

The objectives while forming partition **I** are:

a) Select an appropriate number of groups q to partition the set I. A number of classes between 10% and 20% of the total number of states is be deemed acceptable in an *a priori* estimation.

b) Find a partition with high values for the R^2 statistics in each of the regressions performed in each group (the higher the value is demanded to be, the most likely the number of groups q will be increased.)

c) Maximize the mean R^2 value among the K regressions (finding the global maximum mean value of R^2 for a given size K of the partition is a computationally intensive task and will not be attempted.)

4.2.3 Dendrogram Grouping Method

Dendrograms are planted binary trees formed so that every node represents either a vector (in this case, some IMLR parameter vector a_i,) or a group comprising all the vectors in the nodes that branch from the original node. The leaves of the dendrogram (i. e. those nodes not having other nodes branching from them) are exclusively formed by the parameters vectors resulting from a previously performed IMLR using model 4.14. Two nodes branch from the same node if they are considered closest among themselves than with any other node (either a leave or another node.)

The term "closest" is interpreted according to one of three criteria, called a "linkage function" [95]:

- **Closest Neighbor**: Two nodes ν_1 and ν_2 are considered the closest by the Closest Neighbor criterion if the smallest Euclidean distance $\min_{v_1 \in \nu_1, v_2 \in \nu_2} d(v_1, v_2)$, is the smallest among all pairs of nodes.

- **Farthest Neighbor**: Two nodes ν_1 and ν_2 are considered the closest by the Closest Neighbor criterion if the largest Euclidean distance $\max_{v_1 \in \nu_1, v_2 \in \nu_2} d(v_1, v_2)$, is the smallest among all pairs of nodes.

- **Average Distance** Two nodes ν_1 and ν_2 are considered the closest by the Average Distance criterion if the average Euclidean distance $\sum_{v_1 \in \nu_1} \sum_{v_2 \in \nu_2} d(v_1, v_2)$, is the smallest among all pairs of nodes.

After the dendrogram is formed, it is "cut" from the root down forming (sub-)dendrograms with the branches that resulted from the cut; the height of the cut is determined according to one of several choices (number of sub-dendrograms produced, maximum allowed membership for the sub-dendrogram, etc.) The vectors forming each leave that belongs to a given sub-dendrogram will pool their regression data together and form one group for the regression.

This grouping method will be called the Dendrogram Grouping Method (DGM.)

In our problem, we will form a dendrogram using the parameter vectors $a^i = (a^i_0, a^i_1, ..., a^i_5)$ obtained from the IMLR in section 4.2.1 as observations (leaves,) using the three linkage functions described above, and a dendrogram cut method of maximum allowed number of clusters (groups) to define the groups that will pool in their regression data to perform the Pooled Multiple Linear Regression. The resulting pooled regressions will be called, respectively, Closest Neighbor Dendrogram Regression (CNDR,) Farthest Neighbor Dendrogram Regression (FNDR,) and Average Distance Dendrogram Regression (ADDR.)

4.2.4 Heuristic Grouping Method

The Heuristic Grouping Method (HGM) is intended to deliver a high mean R^2 value for a given size of the partition of the state set I. It based on the consideration that a state should be a member of a group i only if, once pooling its data with each of the groups, i has a higher R^2 value than any other group.

Besides the parameters and sets described in section 4.2.1—$P, \mathbf{T}, K, I$, etc.—, we use here the parameters m_{groups}, the initial (minimum) number of classes allowed; M_{groups}, the maximum groups allowed; $R_{min} \in (0, 1)$, the minimum desired value of R^2 for the regressions in each class I_k; as well as λ, the number of attempts to make the partition. The algorithm is given in two parts: algorithm 5 is the algorithm that forms the groups, whereas algorithm 6 repeats algorithm 5 to obtain several solutions and picks the best of them.

Algorithm 5. *Heuristic Grouping Algorithm*

1. *Set the values for the parameters: Activeset = I; Assignset* $= (a_1, a_2, ..., a_P)$*;* $a_i = 0 \quad \forall i \in I$*;* $K = m_{groups}$*;* $\boldsymbol{I} = \{I_1, I_2, ..., I_K\}$*, with* $I_k = \emptyset$ *for* $k = 1, 2, ..., K$*.*

2. *Randomly pick* K *different values (states) from among the entries of Activeset and store them in the ordered set* $Initset = \{\iota_1, ..., \iota_K\}$*. These are the group leaders.*

3. *Assign each group leader* ι_k *to its corresponding group in Initset:*

$$I_k \leftarrow I_k \cup \iota_k, \quad k = 1, 2, ..., K.$$

4. *Delete the leaders from Activeset,*

$$Activeset \leftarrow Activeset \backslash Initset.$$

5. $Assignset_{\iota_k} \leftarrow k, \quad k = 1, 2, ...K$

6. *Randomly pick a new member from Activeset, and store it in* θ*.*

7. *For* $k = 1, 2, ..., K$*, estimate* $\hat{\Pi}_{i,t}$ *from the regressors* $D_{i,t}$*;* $\quad \forall i \in \{I_k \cup \theta\}, t \in T$*, according to model (4.15). Calculate the values of* R^2 *for each regression and store them in their respective entry in the ordered set* $\rho = \{\rho_1, \rho_2, ..., \rho_K\}$*.*

8. *If* $(\max\{\rho_1, ..., \rho_K\} < \beta) \wedge (K < M_{groups})$*, create a new group:*

   (a) $K \leftarrow K + 1$*.*

   (b) $\boldsymbol{I} = \boldsymbol{I} \cup I_K$*.*

   (c) $\varsigma = K$*.*

   Otherwise, make $\varsigma = \gamma \ni \rho_\gamma = \max\{\rho_1, ..., \rho_K\}$*.*

9. *Assign the new state to its selected set* ς*:*

$$I_\varsigma \leftarrow I_\varsigma \cup \theta.$$

10. *Remove the chosen state θ from the possible new candidates in Activeset:*

$$Activeset \leftarrow Activeset \backslash \theta.$$

11. *Record the assignment:*

$$Assignset_\theta \leftarrow \varsigma.$$

and clear θ

12. *If $Activeset \neq \emptyset$, go to step 6, else*

13. *Perform the regressions for all members of $\mathbf{I}$ according to model (4.15), and calculate the mean of the c regressions.*

Algorithm 5 will calculate a good partition $\mathbf{I}$. One would like, though, to repeat this process several times to come up with a possibly better partition, one having the highest mean R^2 value we are able to find.

Algorithm 6. *Iterative Heuristic Grouping*

1. *Set the pertinent parameters values.*

2. *Set $l = 1$, $ER = 0^{1\times\lambda}$.*

3. *Use Algorithm 5 to create a Partition $\boldsymbol{I}^l$ and its correspondent mean R^2 value, and store the latter in ER_l.*

4. *If $l < lambda$, go to step 3, else:*

5. *Select the partition $\boldsymbol{I}^l$ such that $ER_l = \min\{ER_1, ER_2, ..., ER_\lambda\}$.*

Algorithm 6 iterates several times the steps in algorithm 1 in order to create different partition, each of which is expected to have different number of classes c. Here, the criterion for choosing the best of such partitions is made using solely R^2. Decisions can also be made based in regression significance, parameter significance, etc.

Example. Let us say that we have 5 states, $S1, S2, ..., S5, S6$, with a threshold for R^2 values of $R_{min} = 0.85$ and a maximum allowed group number of $M_{groups} = 4$.

Algorithm 5 will randomly select, say, states $S2, S3, S4$, to form three initial groups $\{S2\}, \{S3\}, \{S5\}$. It will then randomly choose a fourth state from among the remaining ones (e.g. $S1$,) pool its consumption and price time series with those of each formed group and form temporary groups $\{S2, S1\}, \{S3, S1\}, \{S5, S1\}$. A multiple linear regression on each pair is done, and we will then come up with R^2 values for each group of 0.86, 0.75, and 0.64. Thus we select group 1 for state 1 (since it has the highest R^2 value above or chosen threshold 0.85,) and repeat the process for the remaining states.

Again, we randomly pick a state, $S6$, pool consumption and price data and form temporary groups $\{S2, S1, S6\}, \{S3, S6\}, \{S5, S6\}$. They deliver R^2 values of 0.43, 0.72 and 0.84. Since there are no values above 0.85, we open a new group and have $S6$ assigned to it. Finally, we pick the last unassigned state $S4$, pool and perform the pooled regression. The groups $\{S2, S1, S4\}, \{S3, S4\}, \{S5, S4\}, \{S6, S4\}$ deliver each a R^2 value of 0.73, 0.32, 0.80, and 0.79, respectively. Again, there are no values above the threshold, but this time we have no more extra groups allowed, so we pick the group with the highest R^2 value, regardless of the threshold, and have the final list from algorithm 5 as $\{S2, S1\}, \{S3\}, \{S5, S4\}, \{S6\}$.

4.2.5 Pooled Regressions Analysis

Both pooled regression grouping methods, DGM and the HGM have their own advantages and disadvantages, which will be evaluated in turn. Aspects of each method that need to be studied include:

- Replicability: there is little use for an algorithm that cannot provide the same response for every instance run on. The DGM delivers the exact same grouping for a given set of regression parameters' vector, but the HGM may provide random groups, whose similarity needs to be evaluated.

- Homogeneity: while is natural to expect some single-state or two-state groups, a good grouping algorithm is desired to deliver several groups of similar size, and not one or two groups containing over 20 states, and the rest of the sates ending up isolated (though this can be the actual situation, nevertheless.)

- Running time: an algorithm is always desirable to have the shortest running time when computationally implemented.

- Statistical Significance: it is not desirable for a grouping method to ignore statistical evidence and forcefully put a state in a group whose common regression parameter vectors do not hold statistical significance for that state.

4.2.6 Clustering Similitude

In order to compare the similitude of two partitions, we define here a suitable evaluation method.

Let $\mathbf{I} = \{I_1, I_2, ..., I_K\}, \mathbf{J} = \{J_1, J_2, ..., J_L\}$, be two arbitrary partitions of the state set, I as described in section 4.2.2; and let $I_i = \{I_1^i, I_2^i, ..., I_{k^i}^i\}, i = 1, ..., K$, and $J_j = \{J_1^j, J_2^j, ..., J_{l^j}^j\}, j = 1, ..., L$ be their respective partition sets.

Define then function

$$\mathbf{a_{I,J}}(I_i) = \begin{cases} 1 & \text{if } k^i = 1 \text{ and } l^j = 1 \text{ for } m \in I_i \text{ and } m \in J_j, \\ 0 & \text{otherwise} \end{cases} \qquad I_i \in I;$$

Function $\mathbf{a_{I,J}}$ identifies whether a group I_i contains a single state in partition $\mathbf{I}$ *and* if that state is also single-grouped in partition J. If it is so, then the state in group I_i has been found incompatible with other states twice, by whatever methods formed partitions $\mathbf{I}, \mathbf{J}$.

If a state is not individually forming a group, then we need to seek if it shares its group with another state in two partitions. For every pair of states, we will evaluate if they share a group in a given partition using function

$$\mathbf{b_J}(m, n) = \begin{cases} 1 & \text{if } m, n \in J_j \text{ for any } j; \\ 0 & \text{otherwise} \end{cases} \qquad m, n \in I.$$

In the case that function $\mathbf{a}$ has a value of 1, we say that we have a (one-state) coincidence. Equivalently, if function $\mathbf{b}$ returns a 1, we say that we have a (two-state) coincidence

To measure the number of two-state coincidences that the states in a group in one partition have in another partition, we will use function

$$C2(I_i, \mathbf{I}, \mathbf{J}; q) = \mathbf{a}_{\mathbf{I},\mathbf{J}}(I_i) + (1 - \mathbf{a}_{\mathbf{I},\mathbf{J}}(I_i)) \left(\frac{\sum_{m \in I_i} \sum_{n \in I_i, n \neq m} [q\mathbf{b}_{\mathbf{J}}(m,n) + (1-q)]}{2} \right),$$

for $I_i \in I, q = \{1, 0\}$.

If the parameter q is 1, then function $C2$ counts the number of two-state coincidences that state couples in group I_i have when comparing them with the groups they belong to in partition $\mathbf{J}$. If $q = 0$, then we are simply counting the total number of possible coincidences the states in group $I_i \in \mathbf{I}$. Note that function $C2$ is not necessarily symmetric; $C2(I_i, \mathbf{I}, \mathbf{J}, q)$ is not necessarily the same as $C2(I_i, \mathbf{J}, \mathbf{I}, q)$.

To determine the similitude of two partitions, we will use an expression that, roughly speaking, counts the number of coincidences found in two partitions and divides it by the number of total possible coincidences, given the sizes of each partition. While there are arguably many ways to measure the similitude between partitions with a different number of families, this method was chosen because of its normality: it will always return a 1 when both partitions are equal, and will always return a 0 when there are no coincidences between two partitions.

The similitude function used, $Sim : I^2 \times I^2 \rightarrow [0, 1]$ is defined as:

$$Sim(\mathbf{I}, \mathbf{J}) = \frac{\sum_{I_i \in \mathbf{I}} C2(I_i, \mathbf{I}, \mathbf{J}, 1) + \sum_{J_j \in \mathbf{J}} C2(J_j, \mathbf{J}, \mathbf{I}, 1)}{\sum_{I_i \in \mathbf{I}} C2(I_i, \mathbf{I}, \mathbf{J}, 0) + \sum_{J_j \in \mathbf{J}} C2(J_j, \mathbf{J}, \mathbf{I}, 0)}. \tag{4.16}$$

Chapter 6 provides data for each of the items above. Based in the evidence encountered, we expect to provide an enhanced grouping method, refining some characteristics of either algorithm e.g. randomly sampling the observations included in the process of forming the heuristic groups to reduce calculation time; using the DGM as a template or base to form the initial assignment of the HGM, which is done randomly in algorithm 5; or using the groups obtained in every run of algorithm 5 in algorithm 6 as the template for the starting assignment of the next run of algorithm 5.

4.3 Scenario Tree Generation

This section explains the nature of the scenario tree that the stochastic formulation is based upon. Grossly speaking, a scenario tree in a stochastic programming setting explains the form in which the variation of the stochastic parameters occurs, and the timing when information is being updated and new decisions are taken.

Definition 5. *Let $\mathbb{A} = \{\mathbf{K}, \mathbf{A}\}$ be a planted tree such that every path $\mathbb{K} = \{k_1, ..., k_s, ..., k_S\} \subset \mathbf{K}$ from the root node k_1 to any leaf node k_S has exactly S nodes. Then:*

- *The tree $\mathbb{A}$ is said to have S stages;*
- *All the nodes within a distance of μ to the root are said to belong to, or to be in, stage $\mu + 1$; the set of nodes in stage s is identified as $\mathbf{K}^s$.*
- *Any path $\mathbb{K}$ is called a scenario of the tree, and we say that $\mathbb{K}$ passes through nodes $k_1, ..., k_s, ..., k_S$ when $\mathbb{K} = k_1, ..., k_s, ..., k_S$;*
- *If ν is the number of leaves in tree $\mathbb{A}$, then $\mathbf{A}$ is said to have ν scenarios.*
- *A node k_i is said to be the predecessor of node k_j if and only if k_i and k_j share an arc and $i < j$. Function $a : \mathbf{K} \backslash k^0 \rightarrow \mathbf{K}$, where k_0 is the root node, maps node $a(k)$ into its predecessor.*

In figure 4.2 we can see a three-staged scenario tree, which has nine scenarios. Scenario 1 passes through nodes 1, 2, and 5; scenario 2, through nodes 1,2, and 6, and so on. Each node represents a given number of periods, which is identical for every node in a particular stage but not necessarily for nodes in different stages. For example, if the operative period is one calendar month, we can talk of a week as a stage, so that the nodes for the first stage (i.e. the root) contains the possible outcome of a certain random variable each day of the first week; the nodes in the second stage represent the possible outcomes of that random variable each day during the second week, etc. Every scenario passes through the root, and only one scenario passes through each leaf, so that for every scenario $\mathbb{K}$, the set $\mathbb{K} \cup \mathbf{K}^s$ is the uniquely-determined *characteristic leaf* through which $\mathbb{K}$ passes.

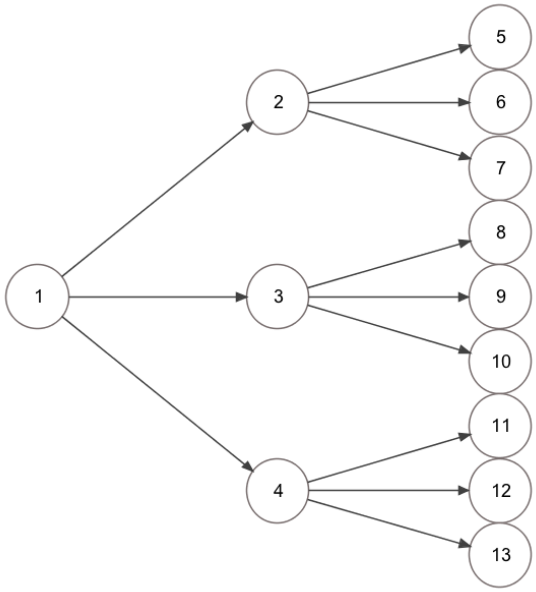


Figure 4.2. A 3-Staged, Ternary Scenario Tree.

Each scenario represents a possible outcome of a series of random variables across the operative period. The latter the periods, the more uncertainty we have; this is why the first stages have fewer nodes than the latter stages. The probability for each scenario to occur is equal to the probability for its characteristic leaf to occur. A tree like that in definition 5, along with a certain set of probabilities for each scenario, is called a *scenario tree*:

Definition 6. *Let* $\mathbb{A} = \{\mathbf{K}, \mathbf{A}\}$ *be a planted tree such that for every path* $\mathbb{K} = \{k_1, ..., k_s, ..., k_S\} \subset \mathbf{K}$ *from the root node* k_1 *to any leaf node* k_S *has exactly* S *nodes. Let* $p : \mathbf{K} \rightarrow [0, 1]$ *be a function such that:*

- $\sum_{i:k_i \in \mathbf{K}^s} p(k_i) = 1 \, \forall s$*;*
- $\sum_{i,j:a(k_i)=a(k_j)} = p(a(k_j.)$

Then $\mathbf{A}, p$ *is called a* scenario tree*, and each scenario* $\mathbb{K}$ *has a probability of occurrence of* $p(k)$*, where* $k = \mathbb{K} \cap \mathbf{K}^S$ *is the characteristic leaf of scenario* $\mathbb{K}$*. The nodes in* $\mathbf{K}$ *are called* event nodes*.*

Each node has a probability of occurrence equal to the sum of the probabilities of each scenario passing through it: since the root belongs to every scenario, its probability of occurrence is 1; on the contrary, since only one scenario passes through each leaf, a leaf's probability and that of the scenario passing through it are equal.

While definition 6 represents the conditions for a probability function p to be allowed to form a scenario tree, it says nothing regarding the way in which that function is defined. The way used in this work to define function p depends in the scenario tree forecast formation method described in the next section.

4.3.1 Scenario Tree Underlying the Bilevel Problem

This section explains the way in which the scenario tree concept is applied to the bilevel optimization problem we discussed in chapter 3, paving the way for the definition of the stochastic model related to problem $PR5$.

The parameters that may vary in the NGSC-POC subsystem are only those related to price and consumption. If they are instead fixed, the there is no stochasticity and the problem is deterministic, with $PR5$ abstracting it. For each of those time series (one for each of variables Π^c_{ti}, and DC_{ti},) and for each of the P pool zones, there is a scenario tree. All these trees have equal number of stages and nodes in each stage. The stochastic formulation replaces the need for the non-contractual term corresponding to prices Π^e with other ways to manage these extra sales, thus the lack of consideration for those time series in this chapter.

The scenario tree, as shown in figure 4.2, consists of several event nodes [75], each of which contains N periods and also each branches out to other nodes. The data correspondent to those trees is based in actual, historical time series (one per tree,) which are used to make forecast that will render the information of every node. The first stage will use only actual information in its forecasts, while the second through the last stages will use information derived from all past stages to obtain forecasted information.

For every node-stage, the data corresponding to prices/consumptions for any of the scenarios passing through said stage are calculated as certain α prediction percentile based on a forecast made with the corresponding data up to the last node

in the tree. The percentile is calculated as a function of the sample error σ of the time series n-th step prediction. Thus, the mean expected forecast corresponds to an α of 0.5, or equivalently, a 0 σ forecast error band from the forecasted values. The possible outcome of prices higher (lower) than the mean forecast corresponds to α taking a value higher (lower) than 0.5, or equivalently, to a forecast error band of $\mathbf{a}\sigma$, where $\mathbf{a}$ is a positive (negative) number whose exact relationship with α depends on the distribution the n-th step forecast error has for a specific time series. In this paper, we are considering that a normal distribution is acceptable; the software ITSM2000 uses the assumption that the forecast errors are normally distributed with a standard forecast error equal to the sample forecasted error. [14]

Information Reduction

The tendency to show high, mean, or low forecasts for either prices or consumptions is considered to be consistent in every node; that is, if the first day of one node showed prices in the α forecast percentile, the second through the last day of that node will also show such high prices. Note that, in a ternary scenario tree, every stage will have three nodes branching from each node in the previous stage, and those nodes correspond to the high/low/mean cases we have talked about.

In the first period of every node, however, the possible outcome (high, mean or low prices/consumptions) can change across every state and pool zones. Recalling figure 4.2, the first node has mean forecast based only in historical, factual information. Nodes 2 to 4 in the second stage, though, have both factual information, and the information forecasted in node 1, interpreted as historic; node 2 corresponds to higher-than-mean forecasts upon the data; node 3, to mean forecasts; and node 4, lower-than-mean forecasts. The process is evidently iterative. Nodes 5 though 7 have information that is both actual and forecasted as historic, and are each a higher-than-average, average, and lower-than-average forecast based upon that information. The same goes for every other node.

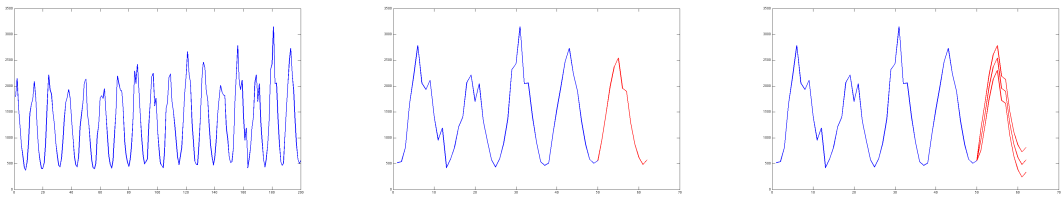
Consider the time series for the monthly consumption of natural gas in the state of Alaska, USA, from January 1989 to September 2005 (200 observations,) shown in figure 4.3(a). This is a rather well-behaved time series with noticeable 12-period seasonality. The expected 12-step forecasts using the Seasonal Holt-Winters (SHW)

method is shown next in figure 4.3(b), cropped to the last observations and with the forecasted values in red for better visibility. These forecasted data corresponds to the first node of the scenario tree (figure 4.2.)

For each of nodes 2, 3, and 4, we obtain 12 new observations by taking the historical time series used in the forecast at node 1, *adding* the forecast obtained in that node as new historical observations, and then forecasting the 0.25, 0.5, and 0.75 percentiles, for example, corresponding to a low, mean and high predictions, and calculated by obtaining the n-th prediction error parameters for an assumed normal distribution. The resulting three time series can be seen in figure 4.3(c), sharing the blue section among them. We have now a three-leaf scenario tree, formed by the nodes 1,2,3, and 4, and underlying the time series shown.

For nodes 5-13, we make this process iterative, by assuming the three time series shown in figure 4.3(c) (formed by a historical part, a mean forecast, and three percentiles, one for each series,) as historical and branching out at their endpoints as done with nodes 2-4. In this way, we end up with nine time series and the corresponding nine-leaved scenario tree in figure 4.2 underlying the series in figure 4.4. All of the series in this figure share the initial observation, whereas only groups of three share the second section.

If we have two parameters (price and consumption) and P pool zones or states, then there are $2P$ different scenario trees like the one showed before underlying a stochastic problem. In a one-staged setting, there are only $2P$ node combinations possible, which is reasonable. However, if we have more than one stage, the number of node combinations grows exponentially. For example, with a 2 stage, P pool zone



(a) Alaska's historical consumption (b) Alaska's Average Forecasted Consumption (c) Alaska Consumptions 0.25, 0.5 and 0.75 Prediction Percentiles

Figure 4.3. Scenario Generation Through Forecasting.

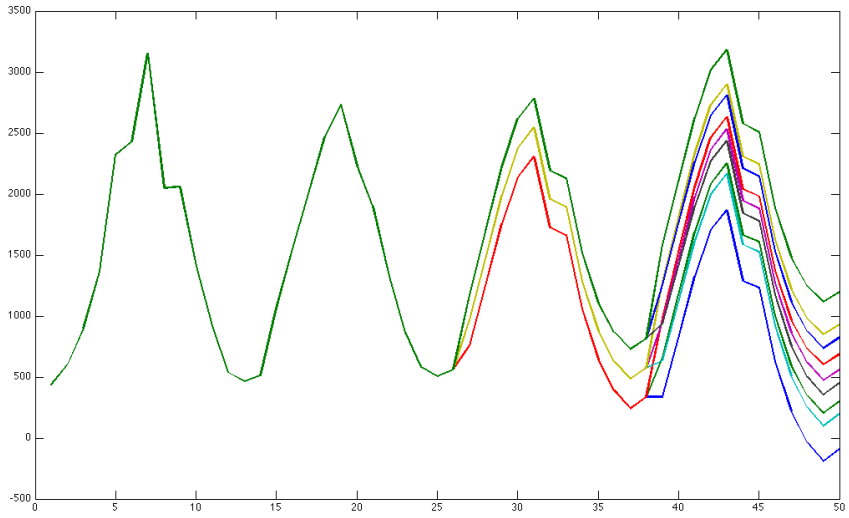


Figure 4.4. Nine Partially Overlapping Time Series Resulting From a 3-Staged Scenario Tree.

problem, the first stage would have $3^{2^P} = 9^P$ nodes, while the second would have $9^{P9^P} = 9^{2P^2}$. The general formula for the number of nodes for a S staged problem with P pool zones is $\sum_{s=1}^{S} 9^{sP^s}$! The number of variables the stochastic problem needs to use for each of the nodes is enormous, since we require several scalars for each day in each node.

In order to reduce the number of variables in each problem, we will use the next two assumptions:

Assumption 13. *There exists a relationship between the demands and the prices of any given pool zone in any given node; that is, in every node, both prices and demands will behave consistently regarding their high, mean or low price/demand outcomes.*

Assumption 14. *The high/mean/low outcome for either prices or demands is common across all pool zones.*

The first item is guaranteed by the obtaining of a regression formula relating price and consumption for each zone. A preliminary analysis of the EIA time series

proposed for use shows that when demand rises (in winter months,) prices decreases accordingly (which is a fairly common behavior for most goods except for a few, e.g., Giffen goods [17].) The existence of a regression function like the one in section 4.2.1 makes the existence of this relationship precise enough to validate this assumption.

The second assumption is verified by the obtaining of a pooled regression like the one developed in section 4.2.2. The specific form of the regression function guarantees not only that the same relationship between price and consumption holds for every pool zone, but also that this relationship is the same at equal times: if demands go higher than the forecasts in one zone, the existence of a pooled regression function like the one proposed guarantees that all other pool zones will have increasing consumptions, and that the prices will increase or decrease in the same way.

These two assumptions allow us to use a single scenario tree for all the pool zones and both parameters in the model, reducing the total number of nodes to only $\sum_{s=1}^{S} 3^{s-1}$.

Probabilities of the Scenario Tree

For a ternary tree to be considered as a scenario tree, we need to pair its graph with a suitable function that assigns a probability to each node of the tree. This section shows the way in which such probabilities are assigned and the image of the probability function is formed.

Each node in the tree is then assigned a probability of occurrence, represented by the evaluation of a certain probability function p. This probability function p must of course comply with definition 6, but the form in which is formed responds exclusively to the ternary tree we employ.

The idea behind the assignment of probabilities is that those nodes product of mean forecast information have a greater probability of occurrence than nodes product of high or low outcomes. For the first stage, being single-node, the probability for this node is 1. For the second stage, the probability of each of the tree branching nodes n_2, n_3, n_4 is, respectively, $\beta^h, \beta^m,$ and β^l, for the high, mean and low outcomes. Though the values $\beta^h, \beta^m,$ and β^l could be assigned arbitrarily, we will make them so that these postulates are held:

- The mean forecast probability of occurrence β^m is larger than either of the other two probabilities. We must consider this as we defineβ^h, β^l.

- The high forecast probability β^h is greater or equal to 1 minus the percentile for high values, $\beta^h \geq 1 - \alpha^h$. If α^h corresponds to the 0.75 percentile, then we want β^h to be somewhere around 0.3. If we want a more explicit method, we can use the sample forecast error band and use a normal (or possibly, a lognormal) probabilities table to determine the exact right tail probability that contains the percentile.

- In a similar manner, we want the lower probability β^l to be higher than the low outcome percentile band α^l e.g. for an α^l of 0.2, we can set β_l to 0.3 or 0.25. Again, we can use a normal table to determine a better assignment of probabilities to a given percentile α^h.

Assigning a probability to a node n is always done multiplying the probability of the predecessor $p(a(n))$ times the corresponding $\beta^h, \beta^m,$ or β^l, depending on n being a high, mean or low outcome node.

Therefore, the probability for a node in stage s has then the form

$$\prod_{i=1:s} \gamma_i, \gamma_i \in \{\beta^h, \beta^m, \beta^l\} \forall i$$

.

For example, the probability of occurrence of a scenario containing only higher than mean forecast nodes in a S staged scenario tree will be $(\beta^h)^S$; in a similar manner, the probability for a scenario with only mean forecast nodes will be $(\beta^m)^S$, and so on.

The entire image of function p as described in definition 6 is formed by applying this criterion to the entire scenario tree.

4.4 Stochastic Optimization Models

This section displays the final mathematical optimization model in this dissertation work. The stochastic optimization modeling shown here is strongly based in the models found in section 3.1.1, using parameters obtained as noted in

section 4.1, and build upon a scenario tree like the one described in section 4.3. It can be described as a multistage bilevel linear stochastic optimization problem with recourse, which means that there are several information update times (multistage,) it has two levels, both of them linear (except for the lower level decision variables in the upper level,) it has stochastic variables and parameters, and some of those stochastic variables influence the decisions made at the beginning of the process (recourse.)

4.4.1 Stochastic Optimization Notes

A Stochastic Optimization Problem is a problem in which certain parameters (constraint bounds or objective coefficients) are random variables whose distribution may or may not be known. At least one variable is affected by this parameter variation, so that the optimal value of said variable is different for different realizations of the random parameter. The decision maker is therefore forced to consider this variation in its model, decide the most advantageous way of action against uncertainty.[60]

In our case, all stochasticity is limited to the parameters D^c and Π^c in the upper level objective function (which is the reason why we specifically state their value when naming problem $PR5(q, D^c, \Pi^c, \Pi^e; x, EA, A^1, A^2, y, u, v, z, d, \xi, \zeta)$.) All other parameters in the model are deterministic; either because they have been previously agreed as fixed by both levels, or because their nature is not subject to randomness.

The stochastic optimization problem described in section relies on the following assumptions, additional to those stated in section 1.4

Assumption 15. *The operative process can be divided into equal-length stages: every node in every stage is therefore equal.*

Assumption 16. *The required demand, and the forecasted price for each node can be forecasted using any of the two methods in section 4.1.3; they are the only random parameters in the model.*

Assumption 17. *At the beginning of every stage, demands/prices will be forecasted as any of the three percentile choices as described in section 4.3.1.*

Assumption 18. *All pool zones behave similarly regarding their high/mean/low demand and prices outcomes. Further, demands and prices for a given pool zone also behave according to a functional relationship obtained through regression.*

Assumption 19. *There is enough certainty in the first forecast to guarantee that the mean forecast demand and prices for the first stage are acceptable; there is then only one node at the root of the scenario tree.*

Assumption 20. *The n-th step forecast error can be considered to be normally distributed for every pool zone, every node, and both parameters analyzed.*

Assumption 21. *The NGSC can draw more gas than the demand faced for a given node, but this amount cannot be marketed by the NGSC.*

Most of these assumptions have already been explained. Assumption 21, however *contradicts* assumption 8, prohibiting the NGSC to sell gas beyond its contractual obligations. This serves a modeling purpose, since allowing this term to be profitable for the NGSC would disrupt the first stage/second stage framework of the stochastic formulation.

4.4.2 Notation

The models presented below make use of the notation shown in tables 4.1 to 4.6; the parameters for the upper level differ slightly to those used in the deterministic problem, but for the most part the problem is the same. Some symbols are also different from the deterministic counterpart, because of the need to differentiate certain features only present in the stochastic problem.

Table 4.1. Sets Used in Chapter 4.

N	Number of time periods at each node in the process; $N \in Z^{++}$.
P	Number of pool zones; $P \in Z^{++}$.
K	Number of event nodes in the process; $K \in Z^{++}$.
S	Number of stages in the process; $S \in Z^{++}$.
$\mathbf{T}$	Set for time periods in any given node; $\mathbf{T} = \{1, 2, ..., N\}$.
$\mathbf{J}$	Set for Pool Zones; $\mathbf{J} = \{1, 2, ..., P\}$.
$\mathbf{K}$	Set for event nodes; $\mathbf{K} = \{1, 2, ..., K\}$.
$\mathbf{K}^i$	Set for nodes in stage i, i=1,...,S.

Table 4.2. Upper Level Parameters Used in Chapter 4.

x^L_{kti}, x^U_{kti}	Lower and upper bounds for the daily imbalances on day t at node k, in pool zone i; $i \in \mathbf{J}$, $t \in \mathbf{T}$, $k \in \mathbf{K}$.
x^L_{kt}, x^U_{kt}	Lower and upper bounds for the sum of the daily imbalances on day t at node k; $t \in \mathbf{T}$, $k \in \mathbf{K}$.
sw^L_{kti}, sw^U_{kti}	Bounds on balance swing before day t at node k's start, in pool zone i; $t \in \mathbf{T}$, $i \in \mathbf{J}$, $k \in \mathbf{K}$.
x_{0i}	Imbalance at the beginning of day 1 at node 1, in pool zone i; $i \in \mathbf{J}$.
D_{kti}	Expected demand on day t, at node k, in pool zone i; $t \in \mathbf{T}$, $i \in \mathbf{J}$, $k \in \mathbf{K}$.
Π_{kti}	Unit price for each unit of gas extracted/sold (contracted gas) at time t at node k in zone i; $t \in \mathbf{T}$, $i \in \mathbf{J}$, $k \in \mathbf{K}$.
CI_{kti}, CR_{kti}	Recourse cost and booking capacity cost per gas unit on day t at node k, in pool zone i; $t \in \mathbf{T}$, $i \in \mathbf{J}$.
p_k	Probability of node k to occur in any scenario; $k \in \mathbf{K}$.

Table 4.3. Lower Level Used in Chapter 4.

e_{ij}	Amount of gas lost when moving one unit from pool zone i to pool zone j; $i, j \in \mathbf{J}$, $i < j$.
f_{ij}, b_{ij}	Forward haul cost and backward credit for moving one unit of gas from pool zone i to pool zone j; $i, j \in \mathbf{J}$, $i < j$.
r_i	Cash-out penalization coefficients in pool zone i; $i \in \mathbf{J}$.

Table 4.4. Upper Level Decision Variables Used in Chapter 4.

x_{kti}	Imbalance on day t at node k, pool zone i; $t \in \mathbf{T}$, $i \in \mathbf{J}$, $k \in \mathbf{K}$.
sw_{kti}	Imbalance swing for day t's start in node k, in pool zone i; $t \in \mathbf{T}$, $i \in \mathbf{J}$, $k \in \mathbf{K}$.
EA_{kti}	Amount of gas actually extracted on day t at node k, in pool zone i; $t \in \mathbf{T}$, $i \in \mathbf{J}$, $k \in \mathbf{K}$.
EP_{kti}	Amount of gas planned to be extracted (i.e. booked pipeline capacity) on day t at node k, in pool zone i; $t \in \mathbf{T}$, $i \in \mathbf{J}$, $k \in \mathbf{K}$.
xa_{ktj}	Amount of gas actually extracted and sold during day t at node k, in pool zone i; $t \in \mathbf{T}$, $i \in \mathbf{J}$, $k \in \mathbf{K}$.
xd_{ktj}	Amount of demand D_{kti} not met during day t at node k, in pool zone i; $t \in \mathbf{T}$, $i \in \mathbf{J}$, $k \in \mathbf{K}$.

Table 4.5. Lower Level Decision Variables Used in Chapter 4.

y_i	Final imbalance in pool zone i, $i \in \mathbf{J}$.
u_{ij}	Volume of gas moved from pool zone i to pool zone j, $i, j \in \mathbf{J}$, $i < j$.
v_{ij}	Gas credited from pool zone j to pool zone i; $i, j \in \mathbf{J}$, $i < j$.
z	Total cash-out for the Natural Gas Shipping Company.

Table 4.6. Artificial Variables Used in Chapter 4.

q	Binary variable equal to 1(0) if final imbalances y_i are all non-negative (non-positive.) In the special case when $y_i = 0; i \in \mathbf{J}$ we accept $q = 1$.

Notice that we require a larger number of sets in this model, and all of them are characterized by bold typeface. Also, there is now only one demand and one price parameter matrices, since we no longer allow an out-of-contract gas sale.

Nodes are numbered, so that the parameter p_k is the image of function p when evaluated in node n_k, that is, $p_k := p(n_k)$.

Imbalance swing variables and bounds change from s to sw; imbalances, extracted amounts and booked/planned amounts are still represented by EA, EP. Variables xa and xd are used exclusively in this formulation, and they have no direct

counterpart or meaning in the deterministic problem, although their nature is similar to that of auxiliary variables A^1 and A^2.

Since costs CI and CR are deterministic, they are equal for every node at a given stage s, that is, $CR_{kti} = CR_{k'ti}, \quad k, k' \in \mathbf{K}^l, l = 1, ..., S$. The same occurs with variable EP, as explained by line (4.17h).

Because of the ternary nature of the tree, the parameters K and S are related by expression

$$K = \sum_{i=0}^{S-1} 3^i.$$

As a final remark, we will use the node imbalance matrix x_k, which is defined by fixing one node k and taking all the days and pool zones in it to form an imbalance matrix, that is, $x_k = (x_{kti})_{t=1,i=1}^{N,P}$ when modeling the lower level in order to simplify the notation.

4.4.3 Stochastic Models[2]

Equations (4.17a)-(4.17i) describe the upper level part of a bilevel problem, which is stochastic and based in a ternary scenario tree such as that in figure 4.2. This particular stochastic formulation is node-based instead of scenario based [75]; this means that we will use one set of variables for each node, and not one for each scenario. Proper variable identification through the *predecessor function a* will considerably reduce the number of variables used (as compared to a scenario formulation,) as well as removing the need to include *non-anticipativity constraints*. [60]

Much like we did in the deterministic formulation, the original model is only for descriptive purposes. The actual model, shown later, will be a linear, modified version of this one that has no redundant variables and eliminates nonlinearities through a variable replacement.

[2]The contents of this section appear in the authors' [54]

Minimize: $h_6(x, sw, EA, EP, xd, z; q) =$

$$\sum_{k\in\mathbf{K}} p_k \left[\sum_{t\in\mathbf{T}}\sum_{i\in\mathbf{J}} (CI_{kti}xd_{kti} - \Pi_{kti}\min\{EA_{kti}, D_{kti}\} + CR_{kti}EP_{kti})\right] \tag{4.17a}$$

$$+ \sum_{k\in\mathbf{K}^S} p_k z$$

Subject to:

$$x^L_{kti} \leq x_{kti} \leq x^U_{kti}, \quad k \in \mathbf{K}, t \in \mathbf{T}, i \in \mathbf{J}; \tag{4.17b}$$

$$sw^L_{kti} \leq sw_{kti} \leq sw^U_{kti}, \quad k \in \mathbf{K}, t \in \mathbf{T}, i \in \mathbf{J}; \tag{4.17c}$$

$$x^L_{kt} \leq \sum_{i\in\mathbf{J}} x_{kti} \leq x^U_{kt}, \quad k \in \mathbf{K}, t \in \mathbf{T}; \tag{4.17d}$$

$$x_{kti} = \begin{cases} x_{a(k,)N,i} + sw_{kti} & \text{if } t = 1 \\ x_{k,t-1,i} + sw_{kti} & \text{if } t \neq 1 \end{cases}, \quad k \in \mathbf{K}, t \in \mathbf{T}, i \in \mathbf{J}; \tag{4.17e}$$

$$x_{kti} = EP_{kti} - EA_{kti}, \quad k \in \mathbf{K}, t \in \mathbf{T}, i \in \mathbf{J}; \tag{4.17f}$$

$$xd_{kti} = \max\{0, D_{kti} - EA_{kti}\}; \quad k \in \mathbf{K}, t \in \mathbf{T}, i \in \mathbf{J}; \tag{4.17g}$$

$$EP_{kti} = EP_{k'ti}; \quad k, k' \in \mathbf{K}^S, t \in \mathbf{T}, i \in \mathbf{J}. \tag{4.17h}$$

$$EA_{kti}, xd_{kti} \geq 0; \quad k \in \mathbf{K}, t \in \mathbf{T}, i \in \mathbf{J}; \tag{4.17i}$$

where variable z is controlled by the lower level in response to each upper level scenario in the exact same manner as in problem $PR5$. Problems like (4.17a)-(4.17i) are known as *multi-staged stochastic optimization problems with recourse*. The term "multi-staged" refers to the existence of more than 1 stage: variables corresponding to the first stage can't be changed in the future, so whatever decision they represent is fixed and unchanged as the process develops. In our case, the main first stage-variables of the problem are the booked volumes EP: once notified to the POC before the beginning of the operative period, these amounts cannot be altered by the NGSC. It has to conform to them throughout the whole operative period.

On the contrary, the variables in the latter stages are the *recourse* variables, that is, those variables whose value can be changed in the future in response to the

different scenarios faced. Extraction volumes EA, and their derived imbalance volumes and sold volumes, x and xa, are recourse variables since the NGSC will decide upon them once the time arrives.

Recourse variables are thus named because they can "change the course" or impact of the first stage variables in the optimum. If the chosen EP turns out to be unsuitable for the NGSC (too low or too high in a particular zone and day) then the NGSC can decide upon EA accordingly to minimize the costs derived from the miscalculation.

In the objective function, the term $h_6(x, s, EA, EP, xd; y, u, v, z, q)$ is a function depending on variables from both levels. It represents negative revenues for the NGSC. The term $\Pi_{kti} \min\{EA_{kti}, D_{kti}\}$ is the revenue from the gas extracted and sold in every pool zone and time period, at every node, while $CI_{kti}xd_{kti}$ is the cost for the unmet demand, which can be interpreted either as a penalization cost from the clients, or the cost for the purchase of just enough supply from a third party to satisfy the contracted demands.

The term $CR_{kti}EP_{kti}$ represents the cost the NGSC incurs when booking capacity in the pipeline at time t, node k and pool zone i; this term is particularly important for the non-triviality of the solutions: if it were free to allocate capacity for the NGSC, the later would respond by maximizing variables EP and selling all possible gas at prices Π.

Notice that if we have more than one stage, then $\sum_{k \in \mathbf{K}} p_k > 1$. This means that the first term in the objective function is not an expected value *over the nodes* but rather *over the scenarios*. The non-anticipativity constraints (which prevent decision being taken based upon information not already obtained) have here been implicitly stated by using the node formulation instead of a scenario formulation. If we separate the scenarios in each stage and assign them their corresponding probabilities, then the first node would have a probability of occurring of 1, since all scenarios pass through it, and so on.

Line (4.17g) defines the unsatisfied demand variables, and constraint (4.17h) represent the first-stage nature of variable EP, in the sense that every node in a given stage must share the same values for that variable. All other constraints have similar meaning than their deterministic counterparts.

In the computational solution of the stochastic bilevel problem, we work not with the model above, but the almost linear equivalent given by lemma 4. As explained in the previous chapter, the computational solution of a linear problem is considerably faster than that of a nonlinear problem, so having a (almost) linear equivalent version of a problem will greatly aid us in our purpose of solving the stochastic model given above.

Lemma 4. *Consider the following optimization problem, which is linear with respect to every term except for variable z, which is controlled by lower level in response to this upper level:*

Minimize: $h_7(x, EA, xa, z; \Pi, D, q) =$

$$\sum_{k\in\mathbf{K}} p_k \left[\sum_{t\in\mathbf{T}} \sum_{i\in\mathbf{J}} CI_{kti} \left(D_{kti} - xa_{kti}\right) - \Pi_{kti} xa_{kti} + CR_{kti} \left(x_{kti} + EA_{kti}\right) \right] \tag{4.18a}$$

$$+ \sum_{k\in\mathbf{K}^S} p_k z$$

Subject to:

$$x^L_{kti} \leq x_{kti} \leq x^U_{kti}; \quad k \in \mathbf{K}, t \in \mathbf{T}, i \in \mathbf{J}; \tag{4.18b}$$

$$x^L_{kt} \leq \sum_{i\in\mathbf{J}} x_{kti} \leq x^U_{kt}; \quad k \in \mathbf{K}, t \in \mathbf{T}; \tag{4.18c}$$

$$sw^L_{k1i} \leq x_{k1i} - x_{a(k,)N,i} \leq sw^U_{k1i}; \quad k \in \mathbf{K}, i \in \mathbf{J}; \tag{4.18d}$$

$$sw^L_{kti} \leq x_{kti} - x_{k,t-1,i} \leq sw^U_{kti}; \quad t = 2, ..., N, k \in \mathbf{K}, i \in \mathbf{J}; \tag{4.18e}$$

$$xa_{kti} \leq D_{kti}; \quad k \in \mathbf{K}, t \in \mathbf{T}, i \in \mathbf{J}; \tag{4.18f}$$

$$xa_{kti} \leq EA_{kti}; \quad k \in \mathbf{K}, t \in \mathbf{T}, i \in \mathbf{J}; \tag{4.18g}$$

$$x_{kti} + EA_{kti} = x_{k'ti} + EA_{k'ti}; \quad k, k' \in \mathbf{K}^S, t \in \mathbf{T}, i \in \mathbf{J}. \tag{4.18h}$$

$$EA_{kti}, xa_{kti} \geq 0; \quad k \in \mathbf{K}, t \in \mathbf{T}, i \in \mathbf{J}; \tag{4.18i}$$

1. *Let $(x^*, sw^*, EA^*, EP^*, xd^*, z^*)$ be an optimal solution to (4.17a)-(4.17i). Then, there exists xa^* such that (x^*, EA^*, xa^*, z^*) is an optimal solution for (4.18a)-(4.18i) with the same optimal objective value.*

2. *Let $(x^{**}, EA^{**}, xa^{**}, z^{**})$ be an optimal solution of (4.18a)-(4.18i). Then there exist sw^{**}, EP^{**}, xd^{**} such that $(x^{**}, sw^{**}, EA^{**}, EP^{**}, xd^{**}, z^{**})$ solves (4.17a)-(4.17i) with the same objective function value.*

Proof. (a) Let $(x^*, sw^*, EA^*, EP^*, xd^*)$ be an optimal solution to (4.17a)-(4.17i). If x^* satisfies (4.17b) and (4.17d), then (4.18b) and (4.18c) are trivially true for this x^*. Next, because (4.17f) and (4.17h) are valid for (x^*, EA^*, EP^*), then (4.18h) clearly holds for (x^*, EA^*). Further, if (x^*, sw^*) satisfies (4.17e), (4.17c), then (4.18d) and (4.18e) are evidently true for x^*.

Now, define $xa^*_{kti} = \min\{D_{kti}, EA^*_{kti}\}, k \in \mathbf{K}, t \in \mathbf{T}, i \in \mathbf{J}$, and notice that xa^* yields (4.18f), (4.18g) and (4.18i). Combining this and (4.17g) into the upper level objective function of the original problem, we come to the equality:

$$h_6(x^*, sw^*, EA^*, EP^*, xd^*) = h_7(x^{**}, EA^{**}, xa^{**}). \tag{4.19}$$

(b) Let $(x^{**}, EA^{**}, xa^{**}, z^{**})$ be an optimal solution of (4.18a)-(4.18i). If x^** satisfies (4.18b) and (4.18c), then it is easy to see that (4.17b) and (4.17d) also holds for x^{**}. Next, if (4.18h) is valid for (x^{**}, EA^{**}), then we can define $EP^{**} = EA^{**} + x^{**}$ so that $(x^{**}, EA^{**}, EP^{**})$ satisfies (4.17f), (4.17h).

Furthermore, if (4.18d), (4.18e), are true for (x^{**}), we can define sw^{**} as

$$\begin{aligned} sw^{**}_{k1i} &= x^{**}_{k1i} - x^{**}_{a(k)Ni}, \quad k \in \mathbf{K}, i \in \mathbf{J}, \text{ and} \\ sw^{**}_{kti} &= x^{**}_{kti} - x^{**}_{k,t-1,i}, \quad t = 2, ..., N, k \in \mathbf{K}, i \in \mathbf{J}. \end{aligned} \tag{4.20}$$

Then, sw^{**} will satisfy (4.17e), (4.17c).

Now take xa^{**} and consider the coefficient of objective function h_7 for xa_{kti} in (4.18a), that is, $-CI_{kti} - \Pi_{kti}$. Since no variable other than EA^{**}—together with the parameter value D—limits the growth of xa^{**}, then we can state that, in the minimization process, the variable xa_{kti} will naturally achieve the maximum value allowed by constraints (4.18f) and (4.18g), which is the minimum of EA_{kti} and D_{kti}. This means that in the optimal solution of (4.18a)-(4.18i) one has $xa = \min\{D_{kti}, EA_{kti}\}$. Defining $xd^{**}_{kti} = D_{kti} - xa^{**}_{kti}, k \in \mathbf{K}, t \in \mathbf{T}, i \in \mathbf{J}$, yields xd^{**} satisfying (4.17g) and (4.17i). Therefore, if we substitute $xa^{**} = D - xd^{**}$ into the

objective function of the nonlinear problem, (4.18a) we obtain:

$$h_6(x^{**}, sw^{**}, EA^{**}, EP^{**}, xd^{**}, z^{**}) = h_7(x^{**}, EA^{**}, xa^{**}, z^{**}) \tag{4.21}$$

(c) We have proved so far that, starting from an optimal solution of one problem, a feasible solution for the other problem can be constructed such that their corresponding objective function values will be equal. Hence, it is readily seen that any optimal solution for one problem will give a correspondent optimal solution for the other problem with the same optimal value. Indeed, if any problem had an optimal solution with the objective function value κ^* strictly less than the optimal value of the other problem, τ^*, then, by either (4.19) or (4.21), this latter problem would also have a feasible solution with the objective function value $\kappa^* < \tau^*$, thus denying the optimality of the value τ^* and bringing us to a contradiction. This establishes both assertions of the lemma and implies the equivalence of problems (4.17a)-(4.17i) and (4.18a)-(4.18i) □

Once we have defined the upper level almost linear stochastic problem we will solve, we need only to state the lower level problem; since there are no randomly varying parameters in this lower level, the model is conceptually the same as in the deterministic model (once linearized as described in lemma 3) developed, with some differences in the symbols used. For the sake of completeness, the Bilevel Stochastic Lower Level Model is re-stated here in lines (4.22a)-(4.22p). The bilevel problem formed when putting lines (4.22a)-(4.22p) together with the upper level problem (4.18a)-(4.18i) is labeled $SPR2$.

$$\text{Minimize:} \quad g_7(y,u,v,z,d,\xi,\zeta;x_{kN},q) = d + \mathbf{M}\sum_{i\in\mathbf{J}}(\xi_i+\zeta_i) \tag{4.22a}$$

Subject to:

$$y_j = x_{kNi} + \sum_{i:i<j}\left[(1-e_{ij})u_{ij} - v_{ij}\right] + \sum_{k:k>j}\left[v_{jk}-u_{jk}\right],\ j\in\mathbf{J}; \tag{4.22b}$$

$$u_{ij} \leq \xi_i; \quad i,j\in\mathbf{J}, i<j; \tag{4.22c}$$

$$u_{ij} \leq \zeta_j; \quad i,j\in\mathbf{J}, i<j; \tag{4.22d}$$

$$v_{ij} \leq \xi_j; \quad i,j\in\mathbf{J}, i<j; \tag{4.22e}$$

$$v_{ij} \leq \zeta_i; \quad i,j\in\mathbf{J}, i<j; \tag{4.22f}$$

$$-\zeta_i \leq y_i \leq \xi_i; \quad i\in\mathbf{J}; \tag{4.22g}$$

$$\xi_i \geq x_{kNi}; \quad i\in J; \tag{4.22h}$$

$$\xi_i \geq 0, \quad i\in J; \tag{4.22i}$$

$$\zeta_j \geq -x_{kNi}; \quad j\in J; \tag{4.22j}$$

$$\zeta_j \geq 0; \quad j\in J. \tag{4.22k}$$

$$-M(1-q) \leq y_i \leq Mq;\ i\in\mathbf{J}; \tag{4.22l}$$

$$z = -\sum_{i\in\mathbf{J}} r_i y_i - \sum_{(i,j):i<j} v_{ij}b_{ij} + \sum_{(i,j):i<j} f_{ij}(1-e_{ij})u_{ij} \tag{4.22m}$$

$$-d < z < d; \tag{4.22n}$$

$$y_i, z \in \mathbb{R};\ i\in\mathbf{J}; \tag{4.22o}$$

$$u_{ij}, v_{ij} \geq 0;\ i,j\in\mathbf{J}; \tag{4.22p}$$

The hierarchical problem described by the equations (4.18a)-(4.18i), (4.22a)-(4.22p), which we will call $SPR2$ is a stochastic bilevel linear problem with recourse, and it can be solved, once reduced to the general form shown in (3.7), using general bilevel linear programming techniques.

4.4.4 Problem Comparison

We finish this section with table 4.7, which compares the originally formulated stochastic problem (4.17a)-(4.17i), (4.22a)-(4.22p), labeled $SPR1$, the stochastic bilevel linear model $SPR2$, and the bilevel linear deterministic model, $PR5$.

Table 4.7. Comparison of the Different Stochastic and Deterministic Models Using the Notation Given in Tables 4.1-4.6.

Feature	Problem		
	$SPR1$	$SPR2$	$PR5$
Upper Level Objective Function:	Nonlinear	Linear	Linear
Lower Level Objective Function:	Linear	Linear	Linear
Upper Level Linear Constraints:	$(9P + 2)NK + (K - S)NP$	$(8P + 2)NK + (K - S)NP$	$13SNP + 2SN$
Upper Level Nonlinear Constraints:	KNP	0	0
Lower Level Linear Constraints:	$3P^2 + 5P + 3$	$3P^2 + 5P + 3$	$3P^2 + 5P + 3$
Lower Level Nonlinear Constraints:	0	0	0
Upper Level Real Variables:	$5KNP + 1$	$3KNP + 1$	$5SNP + 1$
Upper Level Integer Variables:	0	0	0
Lower Level Real Variables:	$P^2 + 2P + 2$	$P^2 + 2P + 2$	$P^2 + 2P + 2$
Lower Level Integer Variables:	0	0	0
Times the problem is solved:	2	2	2
Approximate Solution?	Yes	Yes	Yes

4.5 Solution Methods for the Stochastic Bilevel Problem

The bilevel problem (4.18a)-(4.18i), (4.22a)-(4.22p), though conceptually different than the deterministic variant $PR5$, relies on constraint matrices and objective function vectors that have a very similar structure than those of $PR5$ Therefore, all of the algorithms and methods described in section 3.3, such as the Inexact Penalization Algorithm, the Direct Algorithm, and the Lower Level Feasible Point Algorithm can be applied with little to no modification to problem $SPR2$.

It easy to see that, though having the same number of days in the operative period and the same number of pool zones, the stochastic problem $SPR2$ has an exponentially increasing number of decision variables and their respective constraints, as we can see with the inclusion of the parameter K in the accounting of the variables for the stochastic problems. The number of event nodes K, as shown before, is an exponential function of the number of stages, S. It will be then considerably more difficult (and computationally slower) to solve a stochastic problem than a deterministic problem with a similar number of days and pool zones (which .

The generation of problem instances to solve and test the stochastic problem and its solution methods is similar to that explained for the deterministic variants in algorithm 4, with little changes to accommodate the differences found in the stochastic formulation.

Chapter Summary

This chapter described the theoretical tools, models and assumption required to reformulate the deterministic optimization problems described in chapter 3 as stochastic optimization problem.

We first describe the forecast methods, assumption and algorithms needed to make use of the time series of natural gas consumption and prices as parameters in the stochastic models. The time series are also used in a regression analysis to produce a functional relationship between consumption and price, as well as a pooled regression that guarantees similar behavior across pool zones. The series are expected to satisfy the statistical analyses proposed and be usable as valid parameters.

The stochastic optimization models here apply the notion of a scenario tree, which is adapted here to our needs, specifying the desired qualities of a ternary scenario tree and the exact way in which this structure serves as the basis of the parameter scheme used.

Finally, we present the stochastic models expanded from the deterministic optimization problems from the last chapter. First, an explicative model is produced, upon which theoretical manipulations are done to obtain a simpler

problem that is more tractable, computationally speaking, in a manner similar to that performed in the deterministic analogous situation. The model thus obtained is paired with the lower level model already obtained in the last chapter and the variable and problem structure is compared to the deterministic problem $PR5$. The same solution techniques are deemed acceptable for use with the new formulations (with some obvious modifications to the matrices and objective functions,) and the problem generation is also observed to be similar enough as to employ the same basic notions of the algorithm used to produce the deterministic instances, once it has been modified to account for the parameter's stochasticity.

CHAPTER 5
DETERMINISTIC MODELS: NUMERICAL RESULTS

This chapter presents the numerical analysis and computing results obtained from the models proposed in chapter 3. Numerical results corresponding to the individual level's solutions are given in sections 5.1 and 5.2, whereas the complete bilevel problem solution reports are shown in section 5.3.

The experiments here reported test the deterministic models in chapter 3, and are focused in the three objectives:

- Verify the equivalence of the upper level (almost) linear problem with the correspondent nonlinear version, as well as the performance of the solution methods for each formulation.

- Verify the equivalence of the lower level linear problem with the correspondent nonlinear version, as well as the performance of the solution methods for each formulation.

- Verify the equivalence of the bilevel linear problem $PR5$ with the bilevel nonlinear problem $PR3'$, as well as the performance of the solution methods for each formulation.

The first two objectives are important because of the usage of lemmas 2 and 3 in the solution of our bilevel problem. Both linearization lemmas rely on the fact that the artificial variables will converge towards certain values in the optimal solution. Sections 5.2 and 5.1 detail experimental numerical solutions for each problem, focusing on how the numerical methods provide the values for these variables.

Additionally, the lower level must also prove that, for an arbitrary feasible problem, a solution point delivered complies with the complementarity of the artificial variables ξ, ζ. If these variables are not complementary, then the lower level inexact linear formulation proposed by problem $PR5$ is not guaranteed to be equivalent to the nonlinear formulation in $PR3'$; this is also covered in section 5.2

Finally, section 5.3 presents the comparison of the direct solutions of the bilevel linear approximation $PR5$, and the solutions delivered by the Inexact Penalization

Method. The running times and comparisons of these experiments are provided, as well as a discussion of the findings.

All instances in this chapter were solved in an Intel® Core™ 2 Quad processor running at $2.66MHz$, with $4GB$ of RAM. The optimization software used is Matlab®R2008b. We employed baseline Matlab functions and the Optimization Toolbox™, a set of functions specifically designed by the authors of the software to carry on optimization processes.

5.1 Upper Level Model Comparison

The purpose of this section is to compare the equivalence of the upper level (almost) linear formulation with its correspondent nonlinear formulation, as it is theoretically demonstrated by lemma 2. The running times that each of the methods requires obtaining a solution for the tested problem instances are also studied.

For the assertion proved by the lemma to be correct, the artificial variables A^1, A^2 must have an optimal value equal to their respective maximum limit, which is

$$A^{1**} = EA^{**} = \max\{0, EA^{**} - DC\} + \min\{EA^{*}{}^{*}, DC\}$$

for A^1, and

$$A^{2**}_{ti} = \min\{DC_{ti}, A^{1**}_{ti}\} = \min\{DC_{ti}, EA^{**}_{ti}\}$$

for variable A^2.

Variable A^1_{ti}, represents the amount of fuel *that can be sold either within or outside of the contracted amount*, at the opportunity price of Π^e_{ti}. Since the coefficient of this variable is nonpositive, a minimization process will likely increase this variable to the maximum value allowed by the constraints, that is, the value of EA_{ti}.

On the other hand, variable A^2_{ti}, if the hypothesis of the lemma ($\Pi^c_{ti} - \Pi^e_{ti} + CI_{ti} > 0 \,\forall t, i,$) is valid, should be equal to the minimum between EA_{ti} and DC_{ti}, that is, the *the maximum amount of fuel* that can be extracted and sold at a contracted price of Π^c_{ti}.

5.1.1 Nature of the Upper Level Solution

In order to illustrate the way the results are obtained when experimenting with the upper level model of $PR3'$, we will first consider one problem instance, labeled D001. We solved this instance using the same formulation in equations (3.5a)-(3.5h), except for the fact that we will eliminate z (or equivalently, fix $z = 0$) from the objective function in 3.5a: indeed, we are not interested right now in the lower level, only in the behavior of the upper level formulations. The solver used is function `fmincon` , a constrained nonlinear Matlab function that uses as input a custom-built objective function, custom-built nonlinear inequalities and equalities constraints function, and linear inequalities, equalities and bounds matrices and vectors, as well as a starting point. For this particular model, the nonlinear constraint function is null, and the starting point used is a vector of zeros. [94]

The results of solving problem instance D001 using the modified model $PR3'$ are shown in tables 5.1 to 5.3. It was verified that $EP = EA + x$. The obtained optimal solution value is 4,032,907.26, and the running time required to reach to this solution is 29.28 seconds.

Table 5.1. D001, $PR3'$: Optimal Results for Variable x.

Imbalances x			
Pool Zone			
1	2	3	4
143.00	124.00	-66.00	-154.00
122.00	94.00	-28.00	-167.00
116.00	83.00	-40.00	-129.00
105.00	78.00	-26.00	-147.00
98.00	57.00	-32.00	-140.00
106.89	49.00	-60.89	-102.00
91.00	40.00	-25.00	-118.00
85.00	30.00	-21.00	-104.00
80.00	25.00	-39.00	-66.00
75.00	17.00	-51.00	-28.00
77.00	13.00	-71.00	-1.00
88.00	4.00	-33.00	-31.00

Table 5.2. D001, $PR3'$: Optimal Results for Variable EA.

Actual Extraction EA			
Pool Zone			
1	2	3	4
4,346.52	2,129.98	2,223.07	1,302.31
10,210.99	4,297.11	2,705.65	3,538.10
18,589.74	9,671.01	7,574.22	7,557.05
44,736.06	21,214.34	15,890.02	16,315.79
31,128.28	16,576.47	16,613.82	10,818.28
36,998.44	16,245.11	13,508.08	14,430.91
27,871.52	12,798.88	12,645.34	8,382.01
12,863.95	7,636.79	6,355.33	3,166.47
9,234.81	3,528.76	3,083.37	2,101.94
5,290.27	2,311.15	2,289.73	1,971.44
3,511.54	2,228.42	2,089.02	1,450.43
2,611.46	1,699.75	1,654.01	1,257.55

Table 5.3. D001, $PR3'$: Optimal Results for Variable EP.

Notified/Booked Extraction EP			
Pool Zone			
1	2	3	4
4,489.52	2,253.98	2,157.07	1,148.31
10,332.99	4,391.11	2,677.65	3,371.10
18,705.74	9,754.01	7,534.22	7,428.05
44,841.06	21,292.34	15,864.02	16,168.79
31,226.28	16,633.47	16,581.82	10,678.28
37,105.33	16,294.11	13,447.19	14,328.91
27,962.52	12,838.88	12,620.34	8,264.01
12,948.95	7,666.79	6,334.33	3,062.47
9,314.81	3,553.76	3,044.37	2,035.94
5,365.27	2,328.15	2,238.73	1,943.44
3,588.54	2,241.42	2,018.02	1,449.43
2,699.46	1,703.75	1,621.01	1,226.55

Let us solve again instance D001, this time using the upper level model of problem $PR5$ (sans the z in the objective function.) We obtain the results of tables 5.4 to 5.6.

The figures in the tables allow us to corroborate the equivalence of both solutions and to compare the running times when solving the linear formulation $PR5$ with those obtained using formulation $PR3'$.

Model $PR5$ is solved using Matlab Optimization Toolbox function `linprog` , which takes all of its arguments (objective functions, constraints, right-hand sides, and bounds) in the form of regular matrices and vectors and uses them in a linear programming routine. [94]

Table 5.4. D001, $PR5$: Optimal Results for Variable x.

Imbalances x			
Pool Zone			
1	2	3	4
143.00	124.00	-66.00	-154.00
122.00	94.00	-28.00	-167.00
116.00	83.00	-40.00	-129.00
105.00	78.00	-26.00	-147.00
98.00	57.00	-32.00	-140.00
89.00	49.00	-43.00	-102.00
91.00	40.00	-25.00	-118.00
85.00	30.00	-21.00	-104.00
80.00	25.00	-39.00	-66.00
75.00	17.00	-51.00	-28.00
77.00	13.00	-71.00	-1.00
88.00	4.00	-33.00	-31.00

The optimal imbalance matrices for both formulations are identical.

The values of artificial variable A^1 have reportedly grown to their maximum allowed values as we can see from their correspondence with the figures for variable EA. The implicit values for EP for the linear approach are thus also equal to those of the nonlinear approach.

The reported optimal solution value for the linear approach is 4,032,940.14, slightly higher than that of the nonlinear case at 4,032,907.26. Nevertheless, and given the large amount of sizable parameters, we cannot discard rounding errors when considering these optimal values. Therefore, we accept both methods as equally correct for this problem instance in spite of this minimal discrepancy.

Table 5.5. D001, $PR5$: Optimal Results for Variable x.

Artificial Variable A^1			
Pool Zone			
1	2	3	4
4,346.52	2,129.98	2,223.07	1,302.31
10,210.99	4,297.11	2,705.65	3,538.10
18,589.74	9,671.01	7,574.22	7,557.05
44,736.06	21,214.34	15,890.02	16,315.79
31,128.28	16,576.47	16,613.82	10,818.28
37,016.33	16,245.11	13,508.08	14,430.91
27,871.52	12,798.88	12,645.34	8,382.01
12,863.95	7,636.79	6,355.33	4,120.28
9,234.81	3,528.76	3,083.37	2,101.94
5,290.27	2,311.15	2,289.73	1,971.44
3,511.54	2,228.42	2,089.02	1,450.43
2,611.46	1,699.75	1,654.01	1,257.55

One remarkable difference, though, lies in the running times for both approaches. The linear formulations takes only 0.35 seconds to be solved; this is significantly faster the running time for the nonlinear approach by 28.23 seconds, i.e., almost 81 times.

This first example provides us with encouraging evidence: the linear approach appears as not only theoretically correct (by lemma 2,) but also numerically sound. Besides this, its running time is undeniable better than that of the nonlinear approach.

5.1.2 Upper Level Solution Reports

It we report the results obtained after running 25 problem instances of various sizes. The obtained solutions and the running times required are shown, and a comparison thereof is carried on.

Each problem instance was generated using the problem generating algorithm 4, considering only the parameters involving the upper level. All instances were solved without considering variable z in the upper level objective function in either $PR3'$ or $PR5$ formulation.

Table 5.6. D001, $PR5$: Optimal Results for Variable x.

Artificial Variable A^2			
Pool Zone			
1	2	3	4
2,852.21	1,306.35	1,256.86	979.19
6,446.13	2,642.26	1,865.89	2,383.37
14,318.77	5,853.01	4,301.26	5,280.05
28,478.38	12,024.79	11,263.38	10,803.12
20,107.89	10,163.31	10,890.39	7,727.44
21,807.01	10,551.89	9,546.70	8,534.39
17,987.35	8,724.58	8,222.52	6,133.70
7,712.52	4,421.02	4,426.64	2,800.81
5,205.29	2,415.92	2,087.77	1,442.69
3,054.24	1,578.12	1,545.21	1,133.37
2,453.76	1,587.48	1,282.76	921.37
2,611.46	1,277.23	1,148.76	946.71

We created 10 problem instances with 12 days and 4 pool zones, 10 problem instances with 8 days and also 8 pool zones, and 5 larger problem instances with 30 days and 10 pool zones. All the results are presented in tables 5.7 to 5.9. Larger values for the optimal solutions are considered better, as are smaller values for the running times. The column labeled "Quotient" under the Running Time heading is obtained by dividing the running time for the nonlinear case by the running time for the linear case. A high figure indicates a better performance of model $PR5$.

In table 5.7, we see that small problem instances show not too different results in terms of objective function values. Furthermore, in every case in which the optimal solution values for both methods differed more than 0.01, the linear approach presented a better result.

Running times have noticeable differences between each approach. The linear approach obtains its results 80 to almost 300 times faster than the nonlinear approach. These results are consistently below 0.4 seconds while the instance solved the fastest by the nonlinear method is just below 30 seconds. The evidence from this batch of small 12-day, 4-pool zone cases strongly supports the usage of the linear approach over the nonlinear one.

Table 5.7. Optimal Solutions and Running Times for Instances D001 to D010.

	Solution		Running Time		
	Nonlinear	Linear	Nonlinear	Linear	Quotient
D001	4,032,907.26	4,032,940.14	29.28s	0.35s	80.8
D002	5,572,261.09	5,572,280.98	86.74s	0.32s	271.06
D003	10,104,306.08	10,104,306.08	60.28s	0.34s	177.29
D004	3,994,161.24	3,994,161.24	34.61s	0.33s	104.87
D005	9,419,304.01	9,419,304.01	58.83s	0.33s	178.27
D006	1,370,306.84	1,370,307.11	30.80s	0.33s	93.33
D007	9,510,283.50	9,510,283.50	107.59s	0.37s	290.78
D008	6,166,486.24	6,166,486.23	44.59s	0.34s	131.14
D009	3,703,127.55	3,703,127.55	44.19s	0.33s	133.90
D010	5,641,775.34	5,641,775.34	48.16s	0.32s	150.50

Table 5.8 presents analogous figures for the two methods than those in table 5.7. The optimal values for both formulations are much the same, this time with only 2 concrete coincidences (with a 0.01 tolerance for the comparison.) All the other cases have results for the linear approach better than those of the nonlinear approach. The running times reported are also in favor of the linear approach. The gaps between corresponding times are now larger; the average nonlinear time is around 120 seconds, whereas the linear times are all under 0.4 seconds.

Table 5.8. Optimal Solutions and Running Times for Instances D051 to D060.

	Solution		Running Time		
	Nonlinear	Linear	Nonlinear	Linear	Quotient
D051	5,694,350.86	5,694,358.17	127.61s	0.39s	327.20
D052	4,662,195.21	4,662,195.21	85.31s	0.35s	243.74
D053	6,718,600.79	6,718,600.79	76.80s	0.37s	207.56
D054	3,779,139.68	3,779,139.69	117.19s	0.33s	355.12
D055	6,661,749.65	6,661,749.64	114.14s	0.34s	335.70
D056	9,315,454.03	9,315,470.33	193.58s	0.34s	569.35
D057	13,811,198.14	13,811,212.21	206.44s	0.36s	5.73.44
D058	3,694,646.69	3,694,646.69	61.56s	0.36s	171.00
D059	8,922,689.75	8,922,817.48	113.01s	0.33s	342.45
D060	6,874,756.03	6,874,756.03	113.89s	0.37s	307.81

Lastly, table 5.9 shows an apparent shortcoming the nonlinear approach has when dealing with large problem instances. The nonlinear formulation failed to deliver a solution in under 3 hours for any of the five 30-day, 10-pool zone instances. Instance D101 took around 3.8 hours and, after running out of its 3×10^5 allowed function evaluations, still could not match the solution offered by the linear approach in under 1 second.

Table 5.9. Optimal Solutions and Running Times for Instances D101 to D105.

	Solution		Running Time		
	Nonlinear	Linear	Nonlinear	Linear	Quotient
D101	24,165,142.15	33520637.03	13,862.65s	0.54s	25,000+
D102	NA	200,124,718.21	10,800s+	0.61	NA
D103	NA	39,477,570.18	10,800s+	0.28	NA
D104	NA	17,642,276.71	10,800s+	0.27	NA
D105	NA	28,050,008.62	10,800s+	0.23	NA

Judging from the results observed in past experiments, we decided to stop the solution process of the nonlinear formulation of instances D102-D105 in case they exceeded 3 hours of running time, which occurred in every case. The solutions of the linear approach never took longer than one second to be obtained, not even when dealing with the 12,000-decision variable instances. Also, based on the evidence from the past instances, contained in tables 5.7 and 5.8, we have no reason to believe that the linear approach did not deliver correct solutions; this is, if the nonlinear approach had been tested with unlimited time and resources, we consider that it would have reached the same result.

For comparison purposes, we accept that the linear approach is better at solving our optimization problem, providing fast results to the upper level of problem $PR5$, which has been proved equivalent to the upper level model of problem $PR3'$ both theoretically and practically. We will therefore consider the upper level formulation $PR5$ and not the upper level formulation of $PR3'$ as the chosen method for solving the bilevel problem formed when adding up the lower level and variable z to the upper level objective function.

In a final note, we recall that Matlab function `fmincon` is, according to The MatWorks, not particularly suited to solve non-differentiable problems [94]. It is,

though, the only built-in way to solve an optimization problem with nonlinear constraints and/or nonlinear objective function in the software. The results herein presented prove that `fmincon` , while able to effectively solve the nonlinear model problem instances (and in reasonable time for the smaller ones,) behaves poorly once the number of variables is increased.

5.2 Lower Level Solution Methods Comparison

Much like the upper level in the past section, the lower level of the deterministic problem can be solved using either the original formulation $PR3'$, the linear equivalent formulation $PR5$ proposed in lemma 3. This lemma relies on variables ξ, ζ attaining their maximum allowed values in order to have problem $PR5$ being equivalent to problem $PR3'$. As mentioned above, it is important that variables ξ, ζ are complementary in the optimal solution point of problem $PR5$, otherwise, the equivalence of both formulations is not guaranteed.

For the purpose of comparing the solution methods' speed and the equivalence of the obtained optimums, we have generated 26 problem instances using the methodology given in chapter 3, this time using only the steps concerning the lower level parameters and using the initial vector x_{0i} as though it was the final day vector x_{Ni} required by the lower level models.

5.2.1 Nature of the Lower Level Solution

Consider table 5.10; it summarizes the numerical results for a single sample problem instance solution, $C001$ from problem $PR3'$, solved by the nonlinear constrained optimization function `fmincon`. The objective function and the logistic constraints were passed as an explicit function to the solver `fmincon` , whereas the remaining problem information (namely, the linear constraints and bounds) was passed as regular vectors and matrices.

While simple, problem instance C001 permits us to examine the structure of the lower level solutions with clarity. The values of x_{Ni} are entered as parameters, but y, u and v are the decision variables obtained of once the model is solved. We can see that y_i always coincides with x_{Ni} in sign, which is a requirement. Also, each y_i is in all cases nonnegative, which is in accordance with the value of $q = 1$.

Table 5.10. Results for Instances C001, q=1, Nonlinear Optimization Function.

Final Day Imbalances				
x_{Ni}	190.00	67.00	124.00	8.00
Final Imbalances				
y_i	181.58	67.00	124.00	-0.00
Forward Haul				
u_{ij}	0.00	-0.00	0.00	8.42
	0.00	0.00	0.00	0.00
	0.00	0.00	0.00	0.00
	0.00	0.00	0.00	0.00
Backward Credit				
v_{ij}	0.00	0.00	0.00	0.00
	0.00	0.00	0.00	0.00
	0.00	0.00	0.00	0.00
	0.00	0.00	0.00	0.00
Running Time:	1.164,59s			
z	-43,281.47			

Matrices u and v are strictly upper triangular, with every entry in or below the main diagonal being zero. The fact that there is only one entry in u and none in v in the optimal solution means that the natural gas is being moved downstream (forward) the pipeline form one single pool zone to another single pool zone. The volume moved is just enough to cover the negative 8 imbalance plus the proper loss dictated by parameter e_{ij}. For any given instance, the upper triangular parts of matrices u, v are, by definition, complementary, that is, $u_{ij}v_{ij} = 0, i < j$.

Now, let us compare this latter solution to the one provided by Matlab function `linprog` when applied to formulation $PR5$. This requires the addition of the decision variables vectors ξ and ζ to the already shown decision variables. The results for this case are displayed in table 5.11.

Table 5.11. Results for Instance C001, q=1, Linear Optimization Function.

Final Day Imbalances				
x_{Ni}	190.00	67.00	124.00	8.00
Final Imbalances				
y_i	181.58	67.00	124.00	-0.00
Forward Haul				
u_{ij}	0.00	-0.00	0.00	8.42
	0.00	0.00	0.00	0.00
	0.00	0.00	0.00	0.00
	0.00	0.00	0.00	0.00
Backward Credit				
v_{ij}	0.00	0.00	0.00	0.00
	0.00	0.00	0.00	0.00
	0.00	0.00	0.00	0.00
	0.00	0.00	0.00	0.00
Positive Imbalances				
ξ_i	190.00	67.00	124.00	0.00
Negative Imbalances				
ξ_i	0.00	0.00	0.00	8.00
Running Time:	0.27498s			
z	-43,281.48			

As we compare the optimal solution values of both problems, we find them to be identical, save for a minimal difference of 0.01 in the values of z, which is really innocuous. Variables ξ and ζ are indeed complementary, and they accurately identify the positive and negative parts (respectively) of the final day imbalances inputted.

The running times, however, are clearly lower than those of the nonlinear case. Indeed, formulation $PR5$ was solved more than four times faster than formulation $PR3'$. Although 0.9 seconds may seem like a meager gain, we remind the reader that this is just a very small instance; as we increase the number of pool zones P,

the number of variables increases quadratically, and the running time is also expected to be increased.

5.2.2 Lower Level Solution Reports

This section summarizes the results for the problem instances designed for this lower level model, all of them produced in the same way as instance C001 described above. The instances tested are broken down in three batches, according to problem size. Instances labeled C001-C010 are small, 4-pool zone problem instances; instances C040-C050 have 9 pool zones; and C101-C105 are large, 16-pool zone problems instances.

Tables 5.12 to 5.14 condense the results for the problem instances C001 to C010. The optimal solution reached by either method, and the running time required, are shown for each case. The quotients for the running times were again calculated, like it was done with the upper level experiments.

For the first batch of problems (table 5.12), both problems reached the same objective value for the same value of q, notwithstanding minor differences attributed to rounding errors and different tolerance bounds. The largest discrepancy is less than 6 units, which is deemed tolerable given that the general order of the optimal solution is of tens of thousands.

Regarding running times, we see that, with two exceptions, they are all around four times faster for the linear approach than for the nonlinear approach. Instances C004 and C008 are actually trivial problems because their final day imbalances x_{Ni} are either all nonnegative or all nonpositive. In these cases, both methods behave similarly, with the nonlinear approach being somewhat faster. Its figures, nevertheless, have the same order than those of the linear approach.

Instances C040 to C050, reported in table 5.13, exhibit the same type of correspondence between both methods optimal solutions than the past batch of experiments. The largest discrepancy is around 11 units, that is, less than 0.004 percent of the solution's value. This case also happens to be the problem instance with the largest ratio between both methods running times.

Reporting zeros is, from the point of view of the lower level, the best solution one could obtain in every problem, something not always possible. In the case of

Table 5.12. Optimal Solutions and Running Times for Instances C001 to C010.

	Solution		Running Time		
	Nonlinear	Linear	Nonlinear	Linear	Quotient
C001	-4,3281.47	-4,3281.48	1.164,59s	0.274,98s	4.235,1
C002	-7,305.00	-7,305.00	1.179,46s	0.278,22s	4.239,3
C003	-12,659.97	-12,661.52	1.131,60s	0.294,21s	3.846,2
C004	-69,736.00	-69,736.00	0.055,05s	0.063,57s	0.086,5
C005	-8,681.80	-8,681.81	1.165,17s	0.293,71s	3.967,0
C006	-10,920.78	-10,926.14	1.158,38s	0.266,79s	4.341,9
C007	3,263.24	3,263.24	1.146,96s	0.294,01s	3.901,0
C008	25,704.00	25,704.00	0.056,33s	0.070,22s	0.080,2
C009	0.00	0.00	1.179,89s	0.252,72s	4.668,7
C010	-8,666.29	-8,666.29	1.150,36s	0.287,39s	4.002,7

problem instances C040 to C050, we have that five out of eleven instances were able to reach this global optimal, which is the minimum value the squared variable z^2 can take. Among the 4-pool zone instances, only C009 reported $z = 0$. We interpret this as having a larger number of pool zones being related to having a larger number of possible choices to set the y, u, v values so that the utmost result is global optimal. Less pool zones imply less choices and a smaller chance to reach the ideal 0 value of the objective function.

With respecting to running times, the nonlinear approach presents figures ranging from 1.75 to around 6.5 seconds, whereas the times for the linear approach are consistently around 0.3 seconds. The quotients calculated vary from 5.68 to 22.68, that is, the linear solution method is between 5 and 22 times faster than the nonlinear one.

Table 5.14 displays the reports for the five 16-pool zone instances tested. Unfortunately, the nonlinear approach failed in every case to reach an optimal solution within the allowed number of function evaluations, which is 100 times the number of decision variables in every instance ($100P^2$.) In not a single case, `fmincon` was able to deliver even a feasible, non-optimal solution to the instances, though this is largely because of how `fmincon` internally works and not necessarily due to a shortcoming of the formulation. On the contrary, the linear approach was able to reportedly solve every problem.

Table 5.13. Optimal Solutions and Running Times for Instances C040 to C050.

	Solution		Running Time		
	Nonlinear	Linear	Nonlinear	Linear	Quotient
C040	0.00	0.00	1.747,73s	0.307.42s	5.685,1
C041	0.00	0.03	1.854,75s	0.271,80s	6.823,9
C042	0.00	-0.06	2.034,59s	0.288,74s	7.046,4
C043	0.00	0.00	1.836,68s	0.279,58s	6.563,4
C044	-5,444.28	-5,444.33	3.877,85s	0.334,43s	11.590,5
C045	0.00	0.00	1.734,01s	0.287,72s	6.026,7
C046	-247.46	-247.77	3.19795s	0.282,42s	11.323,3
C047	0.00	0.00	1.893,81s	0.282,96s	6.692,8
C048	6,502.00	6,504.70	3.297,57s	0.291,53s	11.311,2
C049	4,134.00	4,135.77	6.419,14s	0.285,48s	22.586,4
C050	3,245.35	3,254.86	6.224,63s	0.275,01s	22.634,1

The running times for the linear method are all lower than 0.6 seconds. The nonlinear approach, however, ran out of function evaluations at around 470 seconds in each case, not being able to report feasible solutions even after taking up to one thousand times longer than the linear approach.

We decided not to increase the number of function evaluations for `fmincon` to give it a chance to solve instances C101 to C105 because it was already clear that the linear approach is considerably faster. Also, the structure of the solutions, the evidence from the past problems, and the optimality reports from the optimizer `linprog` , all give us no reasons to believe the reported solutions by the linear approach in table 5.14 are not optimal. Therefore, a comparison against the values of the nonlinear approach like the one done for the former instances is not considered to be required, given the circumstances. Analogously, we believe that statistical tests to corroborate the greater efficiency of the linear approach are not needed.

For informative purposes, we tested a single 50-pool zone problem instance under the linear $PR5$ formulation. The running time it took the linear method to solve this large nontrivial problem instance is just around 100 seconds.

Considering the evidence mentioned above, we conclude that the linear equivalent model that constitutes the lower level of problem $PR5$ is remarkably efficient when compared to solving the nonlinear lower level problem $PR3'$, providing the same

Table 5.14. Optimal Solutions and Running Times for Instances C101 to C105.

	Solution		Running Time		
	Nonlinear	Linear	Nonlinear	Linear	Quotient
C101	MFE!	1,448.83	490.374,85s	0.488,25s	1,004.35
C102	MFE!	-0.02	466.047,57s	0.519,76s	896.75
C103	MFE!	-2,816.30	468.588,69s	0.516,28s	907.62
C104	MFE!	-4,893.51	482.433,72s	0.578,57s	833.83
C105	MFE!	-504.47	442.033,08s	0.485,81s	909.88

quality of results in mere fractions of the running time required by the latter. Hence, there should be no major issues if we use the formulation $PR5$ when solving the complete bilevel linear problem formed by putting together both levels so far analyzed in this chapter.

5.3 Results for the Bilevel Model

This section presents the most important results concerning the deterministic models proposed in chapter 3, namely, the solution of the bilevel optimization problem, labeled $PR5$, using both methods presented in section 3.3: the Direct Method (DM,) and the Inexact Penalization Algorithm (IPA.) The latter was developed in [23, 59] to deal with the original formulation $PR1$.

We have corroborated, based on the evidence from sections 5.1 and 5.2 in this chapter, that formulation $PR5$ provides an accurate and more efficient (from the optimization point of view) way to obtain an optimal solution to formulation $PR3'$. Therefore, we will use $PR5$ here as the working model for both bilevel methods, DM and IPA.

The structure of the solution is exactly the same as the merging of the solution structures for both the upper and lower levels presented in the immediately preceding sections, with matrices x, A^1 and A^2 in the upper level solution, and matrices/vectors y, u, v, z, d, ξ and ζ in the lower levels.

5.3.1 Numerical Implementation Highlights

The algorithms sketched in section 3.3 are applied here. However, in subsection 3.3.1, we talked about three proposed focus point when implementing the bilevel

solution process of model $PR5$: providing good starting point for both levels' optimizers, and a fast, possibly heuristic way to solve function G in the implementation of the IPA.

The upper level starting point can be readily provided by applying the methodology used in section 5.1 of this chapter referring to the model $PR5$. By dropping the term z from $PR5$ upper level objective function (3.24a), and the entire lower level model, we obtain a plain linear programming problem that, even in large instances with 30 days and 10 pool zones, can be solved in less than one second. Comparing the magnitudes of z and the upper level objective function (3.24a), it becomes evident that this simplified upper level problem not only provides a good point to start a nonlinear search, but also one that is speedily obtained, analytically justifiable, and usable in both the DM and the IPA approaches.

Upper Level Starting Point Selection

Remember that solving $PR5$ in a bilevel setting (as opposed to solving only the upper level linear problem explained in 5.1), we are obligated to solve the upper level problem using a nonlinear minimization `fmincon` instead of `linprog` . This is the reason why we need a starting point in this level now but not necessarily so in section 5.1: `fmincon` compulsorily needs a starting point, preferably as near as the expected optimal as possible to avoid convergence issues.

Tables 5.15 and 5.16 summarize the results for ten problems instances, each with 4 pool zones and 4 days, comparing the running times for four starting search points for `fmincon` applied to the upper level:

- using the result from an individual upper level minimization, like it was done in section 5.1, as starting search point (IND)
- setting the starting point equal to the decision variables' lower bounds (LB)
- setting the starting point equal to the decision variables' upper bounds (UB)
- using a vector of zeros as the starting point (ZERO.)

Table 5.15. Running Times for Four Upper Level Starting Point Methods, DM Approach.

Instance	IND	LB	UB	ZERO
A021	15.61s	77.53s	2.66s	4.07s
A022	149.37s	38.37s	117.62s	116.03s
A023	119.96s	119.85s	5.07s	5.03s
A024	143.64s	135.47s	154.42s	153.06s
A025	121.74s	121.65s	50.74s	49.66s
A026	100.12s	88.82s	33.01s	32.79s
A027	103.85s	146.03s	67.37s	66.81s
A028	119.51s	122.36s	29.59s	29.24s
A029	57.96s	78.73s	8.71s	8.62s
A030	86.43s	75.27s	15.65s	15.40s

Using the DM, we obtained similar results for both the IND and LB methods, with no significant statistical difference between them when tested using the Wilcox Signed Rank Test [72] at 0.1 significance. The UB and ZERO methods are, on the contrary, statistically different, but a quick analysis suggest that, while the ZERO times are better, they are so only in very small amounts, so that both methods can indeed be considered equally good and also statistically better than methods IND and LB.

The same occurs with the IPA regarding IND and LB; at 0.1 significance, the Wilcox Signed Rank test indicates equal goodness for both methods. No statistical tests are needed to see that the UB and ZERO starting point approaches are worse than the first two in this solution method.

Regarding the DM optimal values obtained, there is not considerable difference among the starting point choices, for all of them deliver similar optimal values (with two exceptions for the LB method, which had to be rerun to obtain the final figures.) At 0.1 significance, the Wilcox signed rank test supports the UB as better, then the IND, and lastly the LB. Nonetheless, at 0.05 significance, we cannot assert that there is any difference between the methods: are all considered equally good, save for the LB apparent tendency to reach degenerated solutions.

In the case of the IPA, however, considerable different findings were obtained. Both the IND and the LB methods are equally good solving the experimenting

Table 5.16. Running Times for Four Upper Level Starting Point Methods, IPA Approach.

Instance	IND	LB	UB	ZERO
A021	21.35s	44.89s	1,052.05s	995.34s
A022	39.82s	28.01s	1,056.58s	879.23s
A023	36.58s	47.80s	1,053.51s	980.27s
A024	41.52s	36.79s	1,051.95s	1275.08s
A025	87.83s	0.50s	857.70s	560.12s
A026	30.63s	0.52s	888.29s	879.75s
A027	78.56s	64.46s	1,048.42s	795.01s
A028	70.09s	358.93s	904.60s	1016.96s
A029	53.73s	129.87s	898.85s	657.44s
A030	114.47s	0.47s	859.91s	940.37s

instances, and their values approach well those of the IND, again according to 0.1-significance Wilcox signed rank tests. The UB and ZERO approaches, on the contrary, both failed to provide good solutions, often falling in degenerate, bad local optima that rendered their results unusable.

Considering this facts, we decided that, if one or the other method is to be used, the DM may use any of the LB or ZERO approach, with the IND providing similarly qualified results but a little more slowly (here, we have an average absolute difference of around 55 seconds between the IND and the LB and ZERO approaches.) The IPA, on the contrary, should be paired with either the LB or the IND starting point choices, avoiding the other two which apparently pose convergence issues.

If one single choice of starting point is required for both methods, the IND is the best decision in both convergence and speed. Besides, we speculate that the IND approach, given its more refined procuring, is better suited when facing settings not resembling the ones presented here. Consequently, even though there are better choices when judging individually, we will support the usage of the IND upper level starting point selection method in the comparison done below between the IPA and DM bilevel solution techniques.

Lower Level Numerical Solution

Obtaining a good starting point for the lower level is not an issue for the DM: the lower level problem is solved linearly by `linprog` and does not require a starting point-obtaining routine. The IPA, on the contrary, uses function (3.30) to represent the lower level, and this function is not linear (in spite of the linearity of the lower level it uses as a base.)

The first proposal is using the values derived from applying algorithm 3, which iteratively tries to obtain solutions for the lower level and then reports a good one, which is then fed to `fmincon` . The second method is simply using a vector of zeros as a starting point for all of the lower levels' decision variables. Still a third alternative could be to use the same value as the fixed y in function

refgapfunc5, but this was disregarded for its potential for causing of convergence problems.

Simultaneously, we propose the option of not using a nonlinear minimization with `fmincon` at all, but rather use algorithm 3 to propose several solution vectors and use them as though they were the whole feasible space of G In this manner, minimizing G can be done by merely selecting an appropriate vector among those proposed. This removes the need for using `fmincon` , substituting it for a rather fast, yet possible inaccurate method.

The shortcomings of thus using 3 are (1) the fact that we hardly cover the whole space of solutions, particularly in problems with many pool zones, and (2) that there is no way to determine if the proposed solution is at least a local optimum (from the generalized transportation problem point of view.)

Table 5.17 presents the optimal solutions found for problem instances A031 to A040, created to test the lower level solution techniques proposed. The three methods are: the heuristic plus `fmincon` approach (H+,) the zero vector approach (ZERO,) and the heuristic only approach (H.)

The running times corresponding to these tests are given in table 5.18

It is clear that the solutions for both the H+ and H methods are similarly good, with the later delivering values a little better in a shorter time. Using a zeroes vector is by far the worst approach. It seems to have a tendency for getting trapped in bad solutions, in spite of having running times similar to those of the H+ approach.

Table 5.17. Optimal Solutions for Three Lower Level Solution Methods, IPA Approach.

Instance	H+	ZERO	H
A031	16,407.3832	13,459.382,3	16,793.043,1
A032	1,108.8059	41.869,0	1,355.794,8
A033	14,405.1517	2,818.818,7	14,057.771,1
A034	15,511.4222	2,749.335,5	16,127.569,9
A035	15,608.0004	1,5681.139,2	15,429.667,0
A036	4,971.5174	6,020.584,9	5318.153,6
A037	2,952.0746	59.4561	2,749.033,2
A038	8,331.4528	8,167.000,0	8,798.300,8
A039	1,7416.6706	3,711.255,7	16,868.033,5
A040	11,309.6357	10,814.726,5	11,398.959,9

Table 5.18. Running Times for Three Lower Level Solution Methods Using, Approach.

Instance	H+	ZERO	H
A031	137.94s	2,068.59s	10.70s
A032	25.37s	13.65s	1.60s
A033	211.79s	24.31s	2.26s
A034	225.89s	16.61s	32.48s
A035	61.97s	197.75s	2.82s
A036	54.11s	412.25s	2.02s
A037	96.78s	77.59s	13.44s
A038	107.31s	281.89s	12.66s
A039	146.62s	26.16s	3.15s
A040	223.69s	346.44s	6.08s

Judging from running times only, the H approach is the best option. However, this method is only a heuristic, and not an actual minimization algorithm, with all the shortcomings this can imply for untested cases. Still, for instances with a small number of pool zones, it should be safe to endorse the usage of this method. Otherwise, the H+ approach has similarly good results and can be recommended whenever running times are not a concern.

5.3.2 Structure of the Bilevel Solution

Using the upper level starting point IND and G function solution method H+, we now solve the bilevel optimization problem $PR5$. For documentary purposes, at the same time we evaluate the effectiveness of both the DM and the IPA approaches when working with two different upper level functions, z and the expression in (3.24a), tough it is only the latter the one that matters in the system. This important as the IPA approach was developed in the sources with function z in mind as the only term in the upper level objective function, whereas the DM method here is applied only after linearization techniques were performed to improve upon its effectiveness.

For the first problem only, and for each type of upper level objective function and value of variable q, we present: the final day imbalances $x_{N,i}$, the final day extraction amounts E^a_{Ni}, the final imbalances y_i, and the corresponding objective function values, z or h_5. This is all reported in tables 5.19 to 5.22.

Table 5.19. Instance A001: N=8, P=4, Objective Function z, $q = 1$

	IPA				DM			
Zone	1	2	3	4	1	2	3	4
x_{Ni}	-20.98	70	-29.02	150	0	70	100	0
E^a_i	25.7	20.80	29.02	115.3	963.45	915.76	931.36	963.17
y_i	0	70	0	100	0	70	100	0
	z	-1,060	Time	600.6s	z	-1,160	Time	0.6s

Table 5.20. Instance A001: N=8, P=4, Objective Function z, $q = 0$

	IPA				DM			
Zone	1	2	3	4	1	2	3	4
x_{Ni}	-127.86	-105.44	98.73	134.32	-145.88	-97.63	92.58	149.97
E^a_{Ni}	1043.14	449.99	1758.28	786.44	965.27	915.24	877.85	815.27
y_i	-0.25	0	0	0	-0.46	-0.61	0	0
	z	-664.46.47	Time	15.77s	z	-669.91	Time	65.44s

Table 5.21. Instance A001: N=8, P=4, Objective Function h_5, $q = 1$

	IPA				DM			
Zone	1	2	3	4	1	2	3	4
x_{Ni}	-0.01	69.89	0.06	0.07	-71.52	50	-8.19	29.71
E^a_{Ni}	3000	2930.11	2999.94	2999.93	3000	2950	3000	2970.29
y_i	0	69.89	0.06	0.07	0	0	0	0
	h_5	-1,275,547	Time	30.6s	h_5	-1,286,188	Time	226.8s

Table 5.22. Instance A001: N=8, P=4, Objective Function h_5, $q = 0$

	IPA				DM			
Zone	1	2	3	4	1	2	3	4
x_{Ni}	-19.97	-4.77	-19.78	36.30	-123.60	43.04	-49.77	-39.63
E^a_{Ni}	2666.21	2671.03	2975.30	2937.59	3000	2956.96	3000	3000
y_i	-3.45	-4.77	0	0	-80.56	0	-49.77	-39.63
	h_5	-1,116,394	Time	145s	h_5	-1,286,878	Time	144s

Solution times for instance A001 do not favor either approach: the DM worked faster in one case, and was practically tied in another, while the IPA proved to be remarkably faster in the other two instances. As for the objective functions' optimal values, the DM provided slightly better results than the IPA in the four tables. We obtained better optimal values with the DM, two of them are found faster than a similar but suboptimal solution was encountered by the IPA.

5.3.3 Deterministic Bilevel Problem Solution Reports[1]

Table 5.23 exhibits the results obtained for a batch of 5 problem instances, labeled A005-A009, each having 9 pool zones and 20 days.

We present a comparison of the optimal solution found by each approach, the status of the procured solution, and the running time needed to get to the reported optima. The optimal value of the binary variable q is also displayed.

The figures listed under objective functions z and h_5 are obtained independently for each problem, i.e., the second and third column of table 5.23, in the rows

[1]The results in this section appear in the author's [55]

corresponding to problem A005, represent values for two different runs of a single problem instance, each independently minimizing a different objective function.

The solution status codes used in the tables are: *OPT*: a local optimal solution was found; *MFE*: the maximum number of function evaluations reached, a feasible point is returned; *MFE!*: the maximum number of function evaluations reached, but no feasible point is returned.

In cases where the upper level objective function is z, the IPA attained a better solution than the DM in four of the five problem instances. This took, however, a larger amount of time to reach a solution in four of the instances, with problem instances A008 and A009 being the ones less efficiently solved (more than five hours of running time for value merely 6% better than the one obtained by the DM.)

As for when the upper level objective function is h_5, i.e., the objective function in (3.3), the IPA failed to deliver a feasible solution for problem A009 after running out of function evaluations; this even took longer than what the DM needed to provide an optimal point. Regarding the other instances, the optimal values were similarly good in both approaches, with problem A007 giving the IPA the greatest advantage in the objective function value, besting the DM optimal value by 7%. When it comes to running times, in every instance with h_5, the DM performed considerably better, delivering all of the results between one hour and one hour and a half, whereas the IPA took between one and a half hours to four and a half.

The status of the solutions was also very different between both methods. While the IPA reported reaching an optimal point found in all nine cases, the DM only reported reaching 6 optimal values; the remaining three, while not recognized by the optimizer as local optima, proved to be feasible and good suboptimal solutions. All these were cases had h_5 as the objective function.

It is worthwhile to notice that, while IPA was developed with the usage of z in mind, the DM was modeled explicitly for dealing with functions h_5 (and/or h_3.) Also, notice the case z of problem $A005$; though the IPA's optimal value is three times more negative than the DM's, the analysis of the optimal vectors indicate that they are similar; the large difference between the values of z can be explained by the magnitude of the parameters used in that function, which are accordingly large.

We conclude this section referencing to the outcome of a large-sized instance, A500, experimented upon with both the DM and the IPA methods, just as the smaller instances before. The problem instance consists of 30 days and 10 pool zones, with the rest of the parameters obtained using the problem instance generating algorithm 4. The results are rather large to be presented in a table, so we only sketch them out here.

The problem reportedly had an optimal solution with $q = 0$, 7 negative final day imbalances and 3 positive ones when solved by the MD approach. The lower level is feasible for this arrangement and eliminates the 3 positive figures in the final imbalances configuration. There are in total 3 backward volume moves (corresponding to variable v) and no forward volume moves (variable u.) The lower level minimization objective function (z) has an optimal solution value of 8,896.71, whereas the upper level optimal value is 16,313,885.26. Running time was 4974.47, around 1.4 hours. A manual local search was carried on, and it indicates that this is indeed a local optimum.

On the other hand, we were unfortunately not able to solve instance A500 using the IPA approach. After 10 hours of running time, the IPA failed to deliver a feasible solution to the problem (even when substituting the H+ approach for the H lower level solution approach in section 5.2) and was thus stopped.

Further tests using the DM method and the LB, UB and ZERO starting search point options support the fact that the solution originally reported (obtained with the IND approach,) was not only a local optimum, but also a good one: one of the starting point options (LB) reported a worse local optimum, while the other two (UB, ZERO,) delivered the same solution point as the one obtained with the IND option. Of all these three starting point options, only the LB required a similar running time than the IND at 4,915.17s (against the 4,974.47s from the IND,) whereas the other two required more than 5,300 seconds.

Chapter Summary

This chapter presents several results concerning the solutions obtained after numerically experimenting on the models formulated in chapter 3. Numerous

aspects and sub-problems of these models were tested and the outcomes and conclusions thereof are given in their respective sections.

Individual optimizations were first performed with the models corresponding to both levels of the bilevel models developed. These tests prove that, for both cases, the linearization lemmas theoretically demonstrated are not only correct, but also helpful in the process of procuring a numerical solution to the problems. The linear models were solved considerably faster than the nonlinear models, and their solutions are reportedly equivalent. Large instances, however, could not be solved by the nonlinear approach described here, though the linear approach worked properly.

Based on this, we decided to use formulation $PR5$ when solving the bilevel problem.

The results obtained by solving the complete bilevel problem are detailed next; they refer to model $PR5$ and to either the DM or the IPA numerical solution methods described in section 3.3. First, we addressed the points highlighted in section 3.3.1 regarding obtaining starting points and lower level solution for the IPA approach. We report statistical evidence that favors some of the options against the others and concluded about their possible usage in actual bilevel tests.

The final set of tests are concerned with comparison between the IPA and DM solution methods, performed according to the methodology built throughout the chapter and the considering two different upper level objective functions. Evidence is in favor of the DM approach, which takes advantage of the linearity of both problems, unlike the more refined but nonlinearly solved IPA method, which is unable to solve the largest instance tested. The DM reached similarly good solutions in only a fraction of the time the IPA required in most of the instances tested.

Once the test were carried on, we support the decision of implementing the stochastic version of model $PR5$ (described in section 4.4.1) using the DM approach, and the upper level starting point options IND or LB.

Table 5.23. Results for Problems A005-A009.

A005				
Method	DM		IPA	
O. F.	z	h_5	z	h_5
Optimal Value	-376.63	-13,467,608	-961.72	-14,164,572
Solution Status	*OPT*	*MFE*	*OPT*	*OPT*
Running Time	1,937.4s	4,962s	5,510.95s	7,635.01s
$q = 1$				
A006				
Method	DM		IPA	
O. F.	z	h_5	z	h_5
Optimal Value	60.55	-5,696,531	56.20	-5,534,923
Solution Status	*OPT*	*MFE*	*OPT*	*OPT*
Running Time	590.15s	3,737.4s	2,662.73s	16,530.21s
$q = 0$				
A007				
Method	DM		IPA	
O. F.	z	h_5	z	h_5
Optimal Value	0	-35,240,591	0	-37,885,387
Solution Status	*OPT*	*MFE*	*OPT*	*OPT*
Running Time	360s	4,842.79s	3,791.03s	7,805.36s
$q = 1$				
A008				
Method	DM		IPA	
O. F.	z	h_5	z	h_5
Optimal Value	-1,578	-13,466,564	-1,673.78	-13,234,066
Solution Status	*OPT*	*MFE*	*OPT*	*OPT*
Running Time	56.04s	4,279.12s	2,404.67s	7,533.16s
$q = 1$				
A009				
Method	DM		IPA	
O. F.	z	h_5	z	h_5
Optimal Value	3,452.97	-13,651,190	3,211.33	NA
Solution Status	*OPT*	*OPT*	*OPT*	*MFE!*
Running Time	57.10s	3,590.4s	20,844.21s	5,638.95s
$q = 0$				

CHAPTER 6
STOCHASTIC MODELS: NUMERICAL RESULTS

This chapter presents the experimenting results concerning the various tools, methods and algorithms needed to create the multi-level stochastic optimization problem with recourse $SPR2$. The aforementioned methods were described and theoretically studied in chapter 4.

The chapter is divided in three sections. Section 6.1 presents the commonplace test for the time series forecasting procedures, as well as a comparison of the two forecasting methods proposed. Section 6.2 shows our efforts for developing a regression analysis in the price and consumption time series. This is done with the intention of reducing the information needed for any given stochastic problem instance. The most important part of the chapter is section 6.3, which shows the numerical experiments devised to solve stochastic model $SPR2$.

6.1 Time Series Analysis and Forecasting

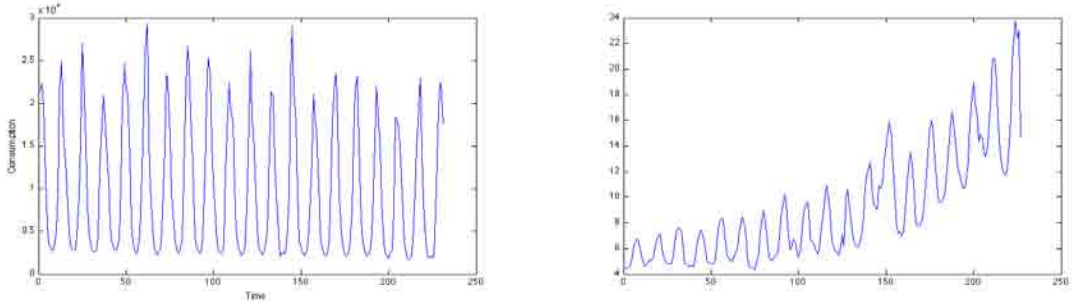
The time series used in this work were obtained from the Energy Information Administration database for residential monthly prices and consumptions per state of the US. The series are each composed of 227 observations corresponding to the months from January 1989 to October 2007.

Unlike most of the tools studied in this and the past chapters, the results in this section are completely dependent of the data selected. Depending on the figures used in, one could or could not come by the same results as those shown here. The analysis in this and the next sections justifies the usage of the particular set of data used in section 6.3; other problem instances should validate their data accordingly.

First, let us introduce the general shape of the time series used in this chapter. Figure 6.1 shows two randomly selected time series, one for consumption and another one for prices of the monthly residential gas figures pertaining to the state of Missouri. Notice that the consumption time series presents a more or less steady behavior, with relatively equal peaks and nadirs, and a definite 12-lag period that represents the seasonal increase(decrease) of gas consumption during the winter(summer) periods. Contrarily, the price time series depicted, while sharing

the same 12-lag seasonality has a clear growing trend of the mean prices, described by some authors even as exponential.

While each state has distinct consumption/price figures, they all seem to approximately adhere to the behavior patterned above. Such features are the reasons that prompted us to develop the models in section 4.2.1.



(a) Consumption (b) Price

Figure 6.1. Monthly Consumption and Price of Natural Gas in the State of Missouri, from January 1989 to October 2007. Source: Energy Information Administration.

6.1.1 Correlation Analysis

We begin showing a correlation analysis for a sample of the time series used. Consider again figures 4.1: they clearly describe autocorrelated observations with their respective 12-lag predecessors. Therefore, can expect a waving behavior in the correlogram, and a large peak in the 12 mark of the periodogram for a given consumption series.

The monthly consumption for the state of Nevada is portrayed in figure 6.2, whereas figure 6.3 shows the corresponding autocorrelation (correlogram) and partial autocorrelation graphs. We can also see in the correlogram a noticeable seasonality at lags 12, 24,..., etc. Furthermore, the maximum absolute values in this figure have a slow decrease over the lags, hinting at a non-stationarity of the series.

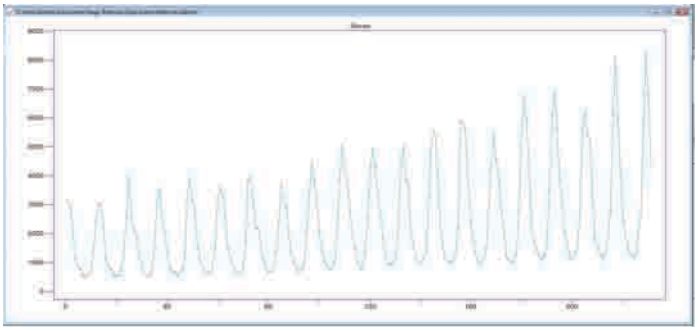


Figure 6.2. Nevada Monthly Gas Consumption

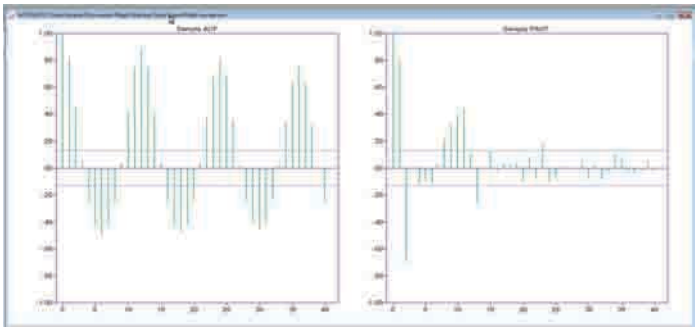


Figure 6.3. Autocorrelation and Partial Autocorrelation of the Nevada Time Series

Figure 6.4 illustrates the periodogram of this consumption series. The peak around frequency $2\pi/12 \approx 0.52$ is dominant, which is easily explained by the waving behavior in figure 6.2. There are small harmonics around 1.1 and 1.65 (corresponding to lags 24 and 36,) though they are not significant when compared with the main frequency.

This analysis suggests [15] not using an ARMA process to model the series, considering the strong non-stationarity of the series original form; there are then three possible alternatives: (1) fitting an ARMA process (and probably obtaining a relatively large number of parameters and a very problematic model,) (2) transform the series (using variance stabilizing and differencing transformation) up until the point where we have a stationary process and then fit the ARMA model, or (3) use a technique that is adequate to forecast seasonal series.

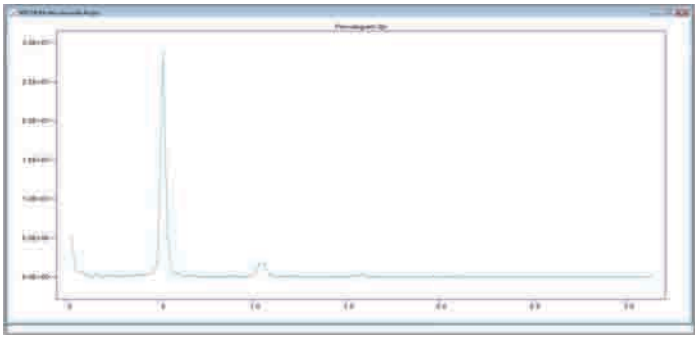


Figure 6.4. Periodogram of the Nevada Time Series

Let us now analyze the price time series for the randomly selected state of selected state of Connecticut. They appear in figure 6.5.

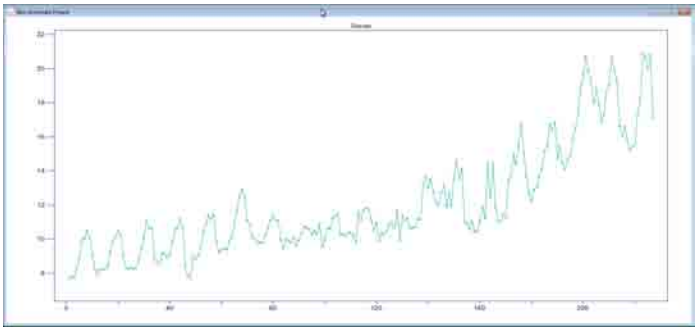


Figure 6.5. Connecticut Monthly Gas Price

What is seen in figures 6.6 and 6.7 does not differ much from the already shown consumption analyses. Despite the increase of the cumulative mean, periodicity remains at 12 lags with minor 24 and 36 harmonics.

Due to space constraints we will not show all 52 states series' plots, correlograms and periodograms. We have, nevertheless, verified that the same assertions remain true for the large majority, with only a handful of states showing ill-behaved time series (mostly for the price figures.)

Let us turn now our attention to some numerical calculations for some series' autocorrelation. A high, near-1 autocorrelation value at lag h, indicates that the

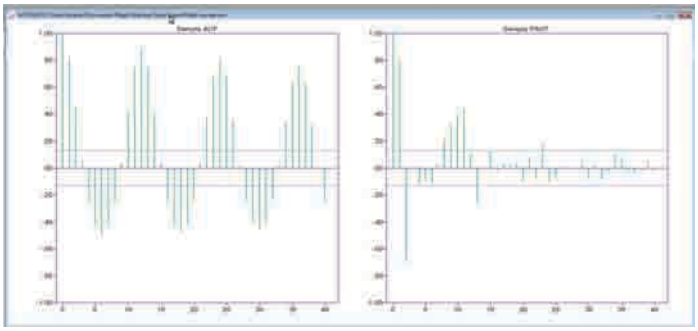


Figure 6.6. Autocorrelation and Partial Autocorrelation of the Connecticut Time Series

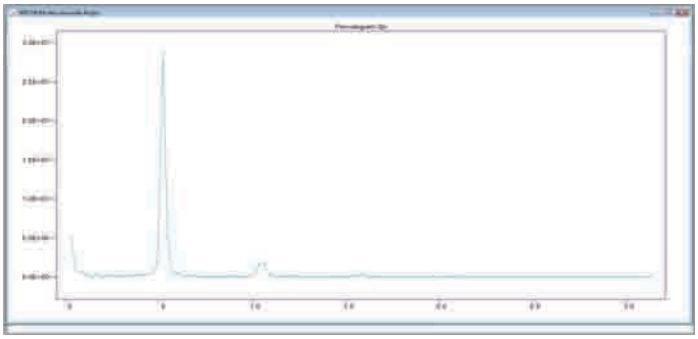


Figure 6.7. Periodogram of the Connecticut Time Series

series involved is separating itself from a white noise process (which is basically an *iid* random time series,) and is correlated with the observations h periods before or after it. Tables 6.1 and 6.2 show the values of the sample autocorrelation function at lags $1 - 14$ for the time series of prices and consumptions, respectively, for the country average and the randomly selected states of Michigan, Pennsylvania, and Vermont.

Table 6.1. Autocorrelation of the Price Series

	Autocorrelation												
lag	1	2	3	4	5	6	7	8	9	10	11	12	13
US	0.97	0.90	0.83	0.76	0.70	0.67	0.67	0.68	0.72	0.76	0.79	0.79	0.76
MI	0.96	0.88	0.78	0.70	0.63	0.60	0.60	0.63	0.69	0.75	0.80	0.82	0.78
PA	0.95	0.86	0.74	0.63	0.56	0.52	0.52	0.56	0.62	0.70	0.77	0.79	0.76
VT	0.96	0.88	0.79	0.71	0.66	0.63	0.62	0.64	0.67	0.71	0.74	0.75	0.71

Table 6.2. Autocorrelation of the Consumption Series.

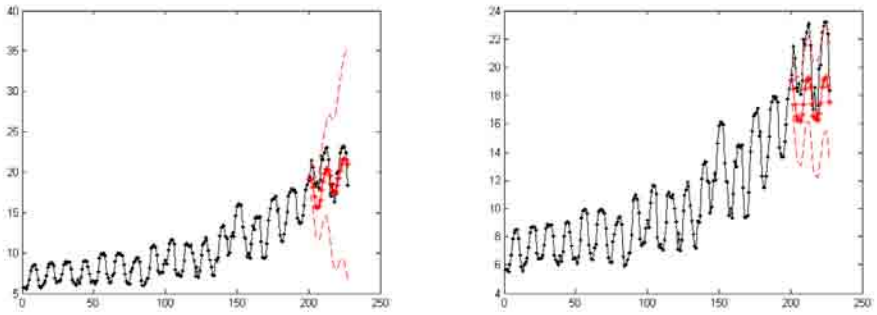
	Autocorrelation												
lag	1	2	3	4	5	6	7	8	9	10	11	12	13
US	0.97	0.90	0.83	0.76	0.70	0.67	0.67	0.68	0.72	0.76	0.79	0.79	0.76
MI	0.96	0.88	0.78	0.70	0.63	0.60	0.60	0.63	0.69	0.75	0.80	0.82	0.78
PA	0.95	0.86	0.74	0.63	0.56	0.52	0.52	0.56	0.62	0.70	0.77	0.79	0.76
VT	0.84	0.47	0.0014	-0.43	-0.73	-0.83	-0.73	-0.44	-0.01	0.44	0.78	0.91	0.78

6.1.2 Forecast Analysis

This set of analyses refers to the forecasting techniques described in 4.1.2, namely, the ARAR algorithm and the Seasonal Holt Winters (SHW) Algorithm. Both of them are considered adequate alternatives to deal with highly seasonal time series like the ones used here.

We decided to utilize these two methods because traditional techniques for removing seasonality like polynomial fitting and series differencing failed to deliver good stationary series upon which we could use more commonplace forecasting techniques like the ARMA or Holt-Winters approaches.

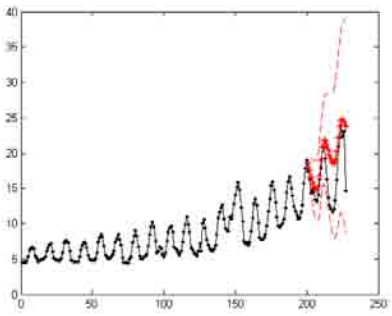
Figures 6.8 to 6.10 show the results obtained after applying both the ARAR and SHW forecasting methods to the states of Alabama, Missouri (for prices), and Texas (for consumption.) All of them show reasonable forecasting abilities.



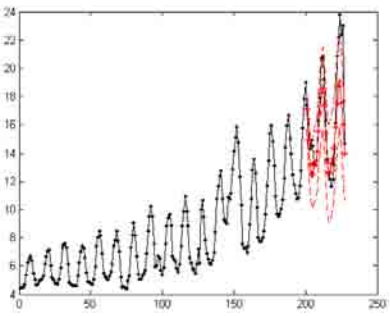
(a) Alabama Price ARAR (b) Alabama Price SHW

Figure 6.8. Alabama Price Forecasts.

We have no concerns regarding the predictions of the consumptions, as both methods seem to deliver good forecast of the demand and similarly tight prediction bands. On the contrary, the price time series are trickier because of their more

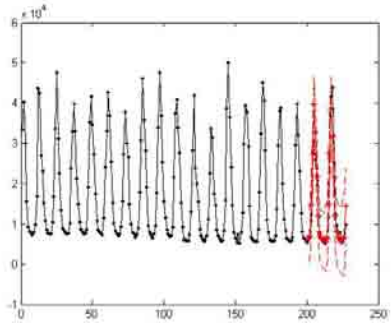


(a) Missouri Price ARAR

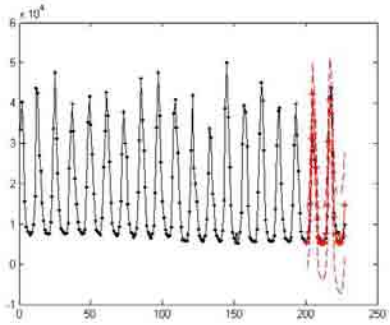


(b) Missouri Price SHW

Figure 6.9. Missouri Price Forecasts.



(a) Texas Consumption ARAR



(b) Texas Consumption SHW

Figure 6.10. Texas Consumption Forecasts

erratic behavior and the increasing trend. In the case of the state of Alabama, the ARAR method gives a better approximation, while the SHW method underestimated the series in almost all of the observations. On the contrary, the SHW follows the series peaks well, but misses the nadirs in the Mississippi series, whereas the ARAR method follows the wave but 'equalizes it,' delivering a gentler curving.

Similar results occur for the majority of the time series analyzed, with sometimes poor, sometimes good price forecasts for both methods, yet highly proper consumption forecasts also for both methods.

Besides whatever accuracy it may have, the SHW provides faster computations; if considering, for example, a rolling horizon optimization with multiple forecasts

instances, this could be important point in favor of the SHW forecasts. Nevertheless, and since this alternative is not considered in this work, we will only take the speed of the SHW algorithm lightly in favor of it, but no further comparison will be performed.

In conclusion, the consumption time series are easily forecasted by both the ARAR and the SHW methods, but the price time series are less accurately predicted. Therefore, no advantage can be given so far to any forecasting method.

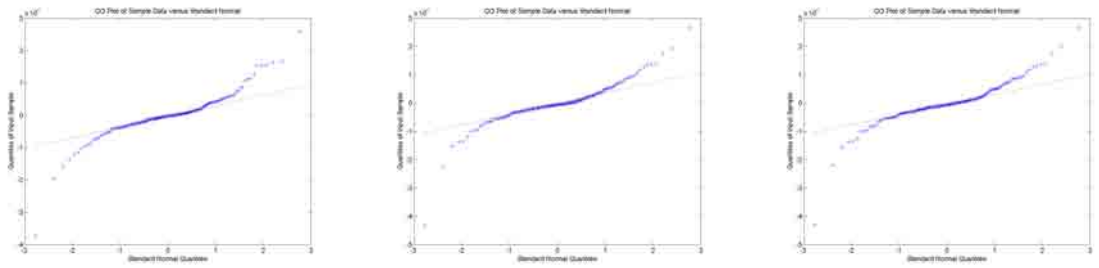
Prediction Bands and Normality of Errors

In order to obtain the prediction bands required to form the scenario trees in chapter 4, we need first to find the distribution of the h-step prediction errors (residuals) for every state and type of data. Proving the normality of the forecast residuals (at least an approximate normality) would greatly help in the calculus of the bands. As we have experienced with the time series analysis software ITSM2000 v7.0, normality is usually assumed when calculating prediction bands.

Figures 6.11 to 6.14 show the consumption and price normality plots for the 1-, 3-, and 8-step predictions for the states of New York and Mississippi using both the ARAR and the SHW forecasting algorithms.

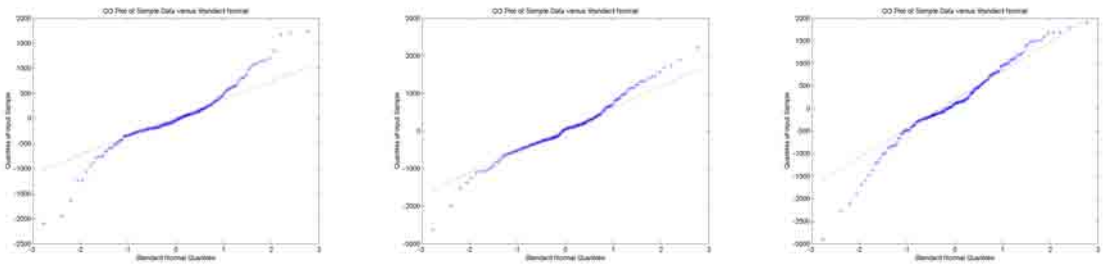
The plots in tables 6.11 do not seem to have high normality: there are significant departures from the line in all the series in the bottom-left part, with top-right tails also separating themselves, though a little more smoothly.

On the contrary, the SHW table 6.12 have less-pronounced departures from the normal line at the top-right tails (slightly more noticeable in the 1-step graph,) and also arguably less pronounced tails at the bottom-left part.



(a) 1-step Predictions (b) 3-step Predictions (c) 8-step Predictions

Figure 6.11. Normal Plots for the State of New York's Consumption Using the ARAR Algorithm.



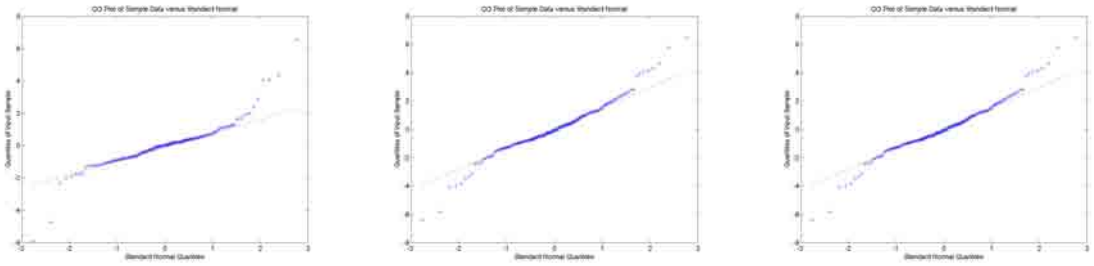
(a) 1-step Predictions (b) 3-step Predictions (c) 8-step Predictions

Figure 6.12. Normal Plots for the State of Mississippi's Consumption Using the SHW Algorithm.

The normal probability plots of the price time series residuals, as seen in tables 6.13 and 6.14, present arguably better shapes. While the lower tails are still steeply decreasing, the upper tails are much better behaved in the case of the ARAR method (except for the 1-step case,) and even more so with the SHW method, where no significant departures from normality apparently exists in the upper tails.

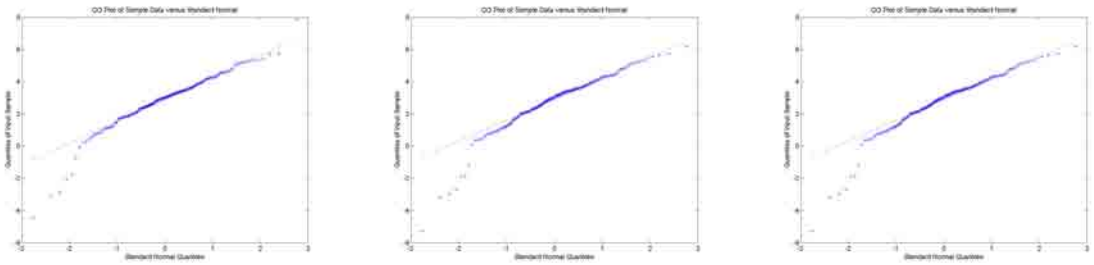
We tried using a using logarithmic transformation of the data thus fitting a log-normal distribution. Experimentation failed to provide us with any noticeably improvement of normality.

Besides justifying a good forecast technique selection, the normality of the forecasts residuals is important because we assume it when calculating the prediction bands, using the mean forecast error/residuals to form the sample error/variance of the method.



(a) 1-step Predictions (b) 3-step Predictions (c) 8-step Predictions

Figure 6.13. Normal Plots for the State of New York's Prices Using the ARAR Algorithm.



(a) 1-step Predictions (b) 3-step Predictions (c) 8-step Predictions

Figure 6.14. Normal Plots for the State of Mississippi's Prices Using the SHW Algorithm.

The prediction bands will be used, as explained in sections 4.1.3 and 4.3.1, are used to generate the branching data of the scenario tree for the stochastic optimization problem in chapter 4. Inaccurate normality results imply that the scenarios shown may not be completely correspondent with reality.

Still, considering the rather accurate consumption forecasts and fair price forecasts, we see that the predictions for consumption and prices, added to the not so bad normality of the residuals, guarantee that even if inaccurate, the forecasts bands would be no more different (from the scenario generation point of view) than those calculated using another, more fitting forecast model.

Hence, we accept as proficient both forecast methods and will make use of them when generating the underlying scenario tree as described in section 4.3. The Seasonal Holt-Winters method is (so far) a little better than the ARAR method,

providing faster and more normal residuals. This advantage, nonetheless, is in no way meaningful enough to discard the usage of the ARAR forecasting technique.

Forecast Residuals Autocorrelation

We finish this section presenting the autocorrelation analysis for the forecast residuals for both the ARAR and the SHW methods. We expect the residuals to be as random and lowly autocorrelated as possible.

Figures 6.15 and 6.16 exemplify the correlograms for the autocorrelation and partial autocorrelation for the 1-step forecast residuals applying the SHW method to the state of Iowa's monthly consumption, and for the 4-step residuals applying the ARAR method to the state of North Carolina. While not perfectly independent, the autocorrelation is not all that different from a white noise process, with only a few values outside the 0.95 error bands in either correlogram. There is a small, yet noticeable seasonality, but overall the randomness of the residuals is not questioned.

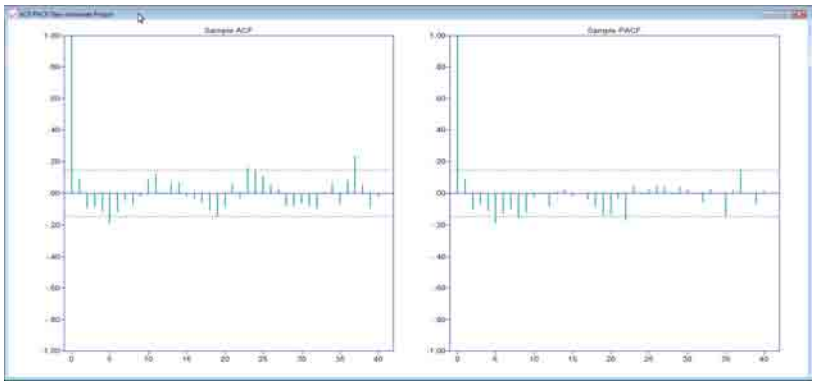


Figure 6.15. 1-Step SHW Residuals Correlograms for Iowa's Consumption.

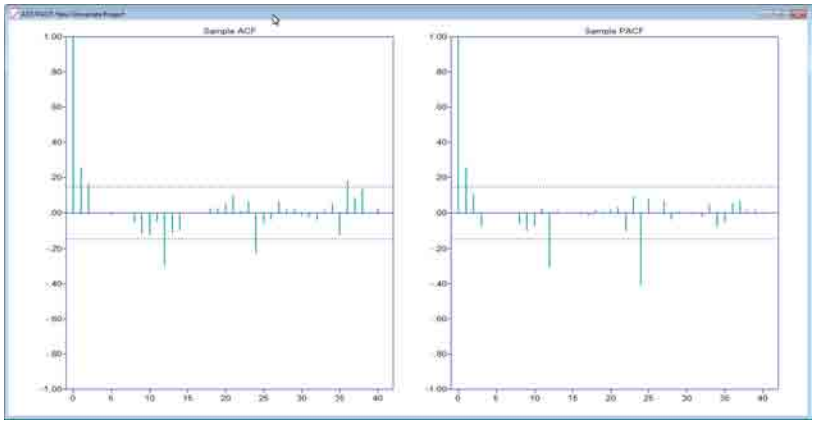


Figure 6.16. 4-Step ARAR Residuals Correlograms for North Carolina's Consumption.

As for the price time series, figure 6.17 displays how the application of the ARAR algorithm produces a very good, non-correlated set of residuals at a 1-lag. Unfortunately, the seasonal Holt-Winters delivered, for the 4-step prediction of the state of North Dakota (figure 6.18,) a rather problematic set of residuals. There is no reason to doubt of the randomness of the Oregon data, but not so much for the North Dakota series. While the latter does not shows considerable issues, it definitely gives us reasons to doubt of the residuals' randomness.

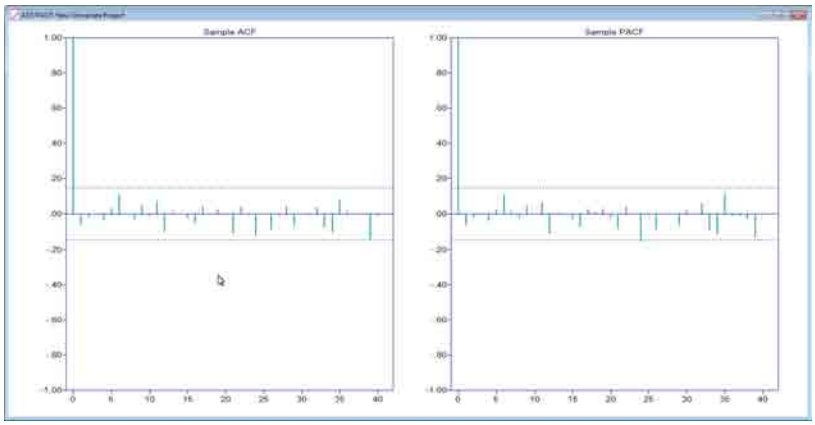


Figure 6.17. 1-Step ARAR Residuals Correlogram for Oregon's Prices.

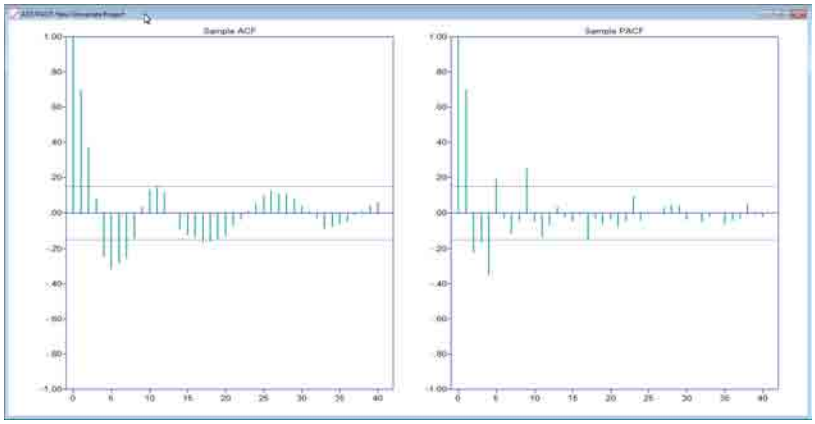


Figure 6.18. 4-Step SHW Residuals Correlogram for North Dakota's Prices.

The correlograms for other prediction steps, states, and both methods are similar to the ones above. The ARAR method provides less randomness than the SHW, but even in the worst cases (from analysis performed for up to 12-step predictions,) the SHW shows no worse randomness than the cases above.

Based on the evidence procured in the analyses presented in this section, as well as the corresponding ones for the series but not explicitly presented, we have no reason to incline ourselves towards either method. But seem to be fair, with small issues that may or may not be overcome by using the other method. Furthermore, and more importantly, the shortcomings encountered do not appear to render the methods unsuitable for our purposes.

6.2 Regression Analysis[1]

.

This section introduces the regression analysis performed as means to reduce the information needed in the stochastic optimization problem. Our purposes are to (1) find an explicit regression function between the price and the consumption in every state, so that only one time series (and one forecast operation) per pair is needed in an optimization problem; and (2) use such regression formula to determine which states have similar parameters thus grouping them into clusters.

[1]Some of the results in this section have appeared in the authors' [56]

6.2.1 Individual Multiple Regression Analysis

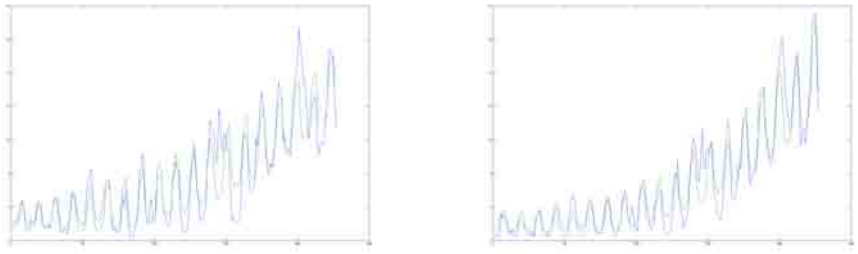
The Individual Multiple Linear Regression (IMLR) formula:

$$\hat{\%}_{ti} = a_0^i + a_1^i t + a_2^i t^2 + D_{i,t}^{-1}\left(a_3^i + a_4^i t + a_5^i t^2\right) + e_i^*, \qquad t \quad T, \;\; i \quad I, \tag{6.1}$$

was applied to the historic monthly residential price and consumption figures from the Energy Information Administration [28], obtaining 52 sets of regression parameters. The coe cients of determination R^2 for this data came up between 0.74 in California and 0.94 in Arkansas, with a sample mean of 0.9 and a sample standard deviation of 0.05. The coe cients for the regressions were nearly all significant for the a_0, a_1 and a_3 parameters; only 20 for the a_4 and 13 for the a_5. No significance was given in any case to the a_2 coe cient. These results were all obtained using a 0.95 confidence regression test.

All regressions appeared as significant, with every corresponding p-value under 0.001.

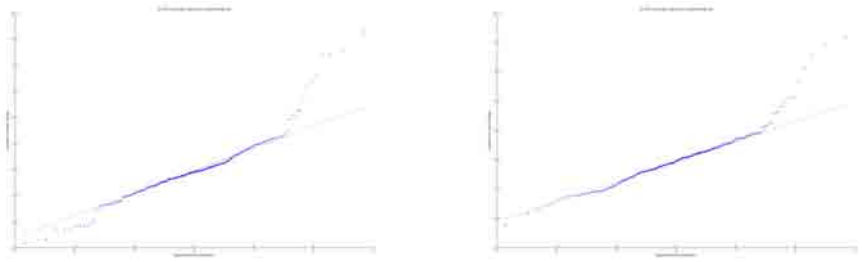
Figure 6.19 shows the regression fits for two randomly selected states, Illinois and Nebraska. Lines in green approximate quite well the original time series (in blue.)



(a) Illinois IMLR fit (b) Nebraska IMLR fit

Figure 6.19. IMLRF for Two States.

The normal plots for the residuals are presented in figure 6.20. As we can see, we do not have a high normality in the regressions for Illinois and Nebraska, but still the majority of the observations fall around the line. The scatter plots, not presented here, show that, while there are continuous portions of the residual series above and below the 0 mark, certain randomness is maintained overall.



(a) Illinois IMLR normal plot (b) Nebraska IMLR normal plot

Figure 6.20. IMLR Fits for Two States.

While the regressions residuals are not very random or normal, these shortcomings are not all that serious for the objectives of this work. Moreover, the fits are quite good in the majority of the cases, with some exceptions where either the consumption or the price time series behave particularly erratic.

6.2.2 Dendrogram Pooled Multiple Linear Regression

This section displays the results obtained when using the Dendrogram Grouping Method in section 4.2.3 to cluster the states and then applying regression using the data pooled from the whole cluster.

Figure 6.21 contains colored maps illustrating Matlab's Dendrogram Clustering function applied to the data obtained above in section 4.2.1. They summarize the clusters formed for the Closest Neighbor Dendrogram Regression (CNDR,) Farthest Neighbor Dendrogram Regression (FNDR,) and Average Distance Dendrogram Regression (ADDR) (all of which are explained in chapter 4.) All three partitions were obtained defining the number of groups to be exactly 14 (or equivalently, selecting the point at which the dendrogram is cut so that it forms 14 sub-dendrograms.)

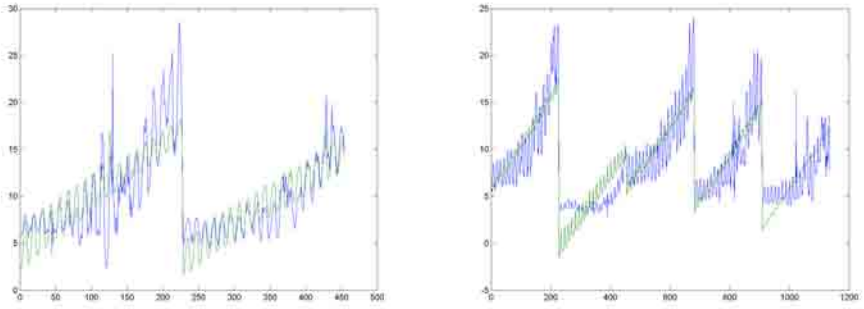
Grouping the states as in subfigure 6.21(a) delivers a very large group (of 38 states!) which would doubtfully deliver a good pooled regression fit. The R^2 for this group, though, is 0.789. Still, the p-values for all pooled regressions are below 0.01. Figure 6.22 shows the second group and some of the members of the fifth group of this partition, since they are the only groups with more than 1 member. All other pooled regressions are trivially equivalent to the individual cases.

(a) CNDR Clustering. (b) FNDR Clustering. (c) ADDR Clustering.

Figure 6.21. Dendrogram Clustering Using Different Linkage Functions.

The average group-R^2 for this method is 0.86, with a sample standard deviation of 0.07. The average state R^2, however, is only 0.74, with a standard deviation of 0.22.

Note that requiring a number of groups different from 14 will deliver differently results. For example, selecting 20 groups we obtain a large group of 19 states, three groups of 5 states, another three of 2 states, and the rest with one state.

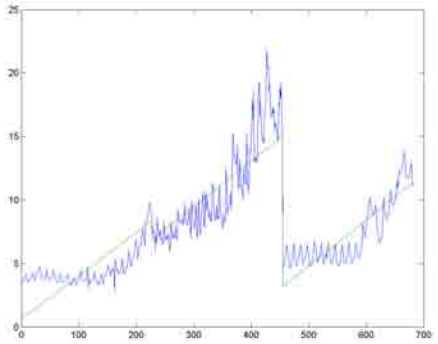


(a) Second Group Fit (b) Fifth Group, 5-State Sample Fit

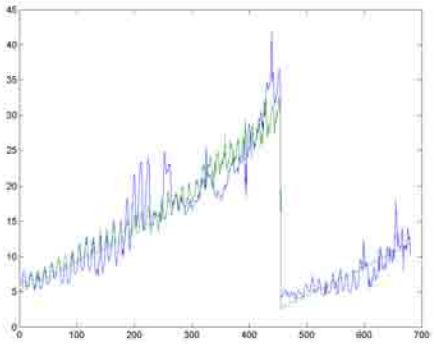
Figure 6.22. Dendrogram pooled fits using CNDR, two sample groups.

The FNDR approach delivers a more balanced partition. We have one group for each size among 11, 10, 9, 8; and lesser 1- or 2-State groups. Figure 6.23 shows the fits for the four largest groups for this case.

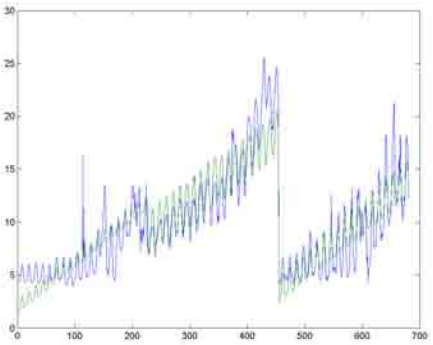
The average group-R^2 for this method is 0.84, with a standard deviation of 0.08; but the average state-R^2 is 0.74 mean. The state-R^2 standard deviation is 0.27, larger than in the CNDR.



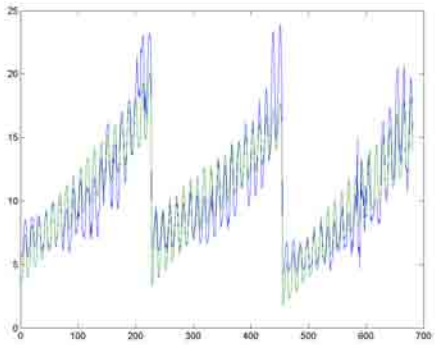
(a) First Group 3-State Sample.



(b) Second Group 3-State Sample.



(c) Seventh Group 3-State Sample.



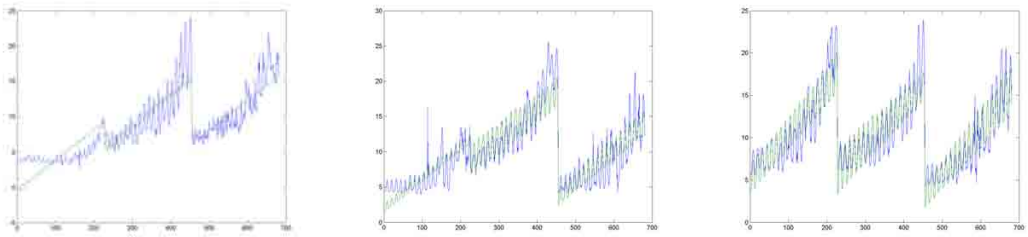
(d) Ninth Group 3-State Sample.

Figure 6.23. Dendrogram pooled fits using the FNDR, four sample groups.

Lastly, we have the ADDR clustering method. This produces a very similar solution than that of the FNDR, with 3 large groups of 8, 10 and 20 states, and the rest of the groups produced having less than 4 members.

The average group-R^2 value for this method is 0.85, with a group-R^2 standard deviation of 0.6. Of all the groups, this one appears slightly better regarding the individual states R^2 values, with an average state-R^2 of 0.76 and standard deviation of 0.22.

An attempt was done to normalize the coe cients (i.e., dividing $a_i^j / \max_j\{a_i^j\}, i = 0, ...5; j = 1 : 52$) before applying the linkage method. The results are not exhibited because they appear to have no major difference with those



(a) Fifth Group 3-State Sample. (b) Sixth Group 3-State Sample. (c) Eight Group 3-State Sample.

Figure 6.24. Dendrogram pooled fits using the ADDR, three sample groups.

already obtained for the non-standardized linkage above: while the groups are not identical, they do not present a different distribution of the group sizes.

6.2.3 Heuristic Pooled Multiple Linear Regression

In figure 6.25 we can see the colored maps obtained after applying the Heuristic Clustering (algorithm 6) using $= 2, 5$ and 10. We allowed a maximum number of groups $M_classes = 14$, and a new group threshold $= 0.85$. The resulting regression sets will be labeled as iHR, where , stands for the number of iterations in algorithm 6.

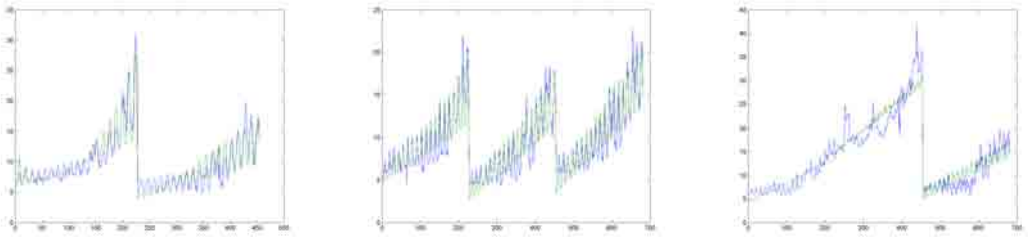
(a) $\lambda = 2$ (b) $\lambda = 5$ (c) $\lambda = 10$

Figure 6.25. Heuristic Grouping Method using different iterations in A2.

We can note from 6.25 that the heuristic method is not outwardly prone to repeatability, at least not when applied to the data chosen. Indeed, different runs with the same input deliver largely different partitions. We expect, however, that this will change if we somehow select the initial random vector in a logical manner, instead of selecting it randomly.

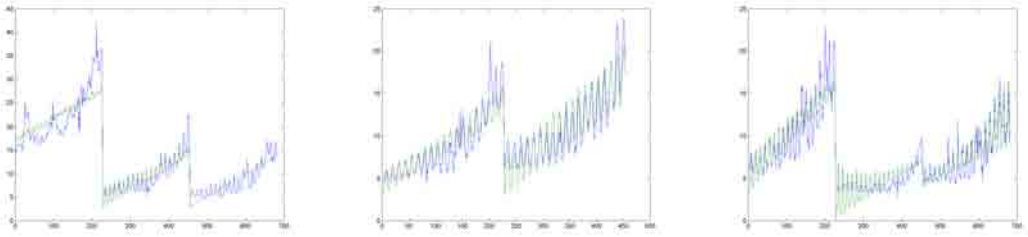
The partitions described by the maps above show groups that are more homogeneous in size than those seen the dendrogram approach. Though usually a large group still exists, the remaining states cluster in mid-sized groups, while five to ten states remain in single- or double-membership groups.

Because of this, we have chosen to show here three randomly selected groups and three or less states of that group to exemplify the fits for the three partitions. Figures 6.26-6.28 the plots for the Heuristic Grouping Method with $= 2, 5$ and 10. Each figure shows a set of graphs pertaining to a particular group—groups are not necessarily presented in their totality due to their size—, with the original price time series and the grouped fit obtained portrayed.



(a) 7th Group (b) 11th Group (c) 12 Group

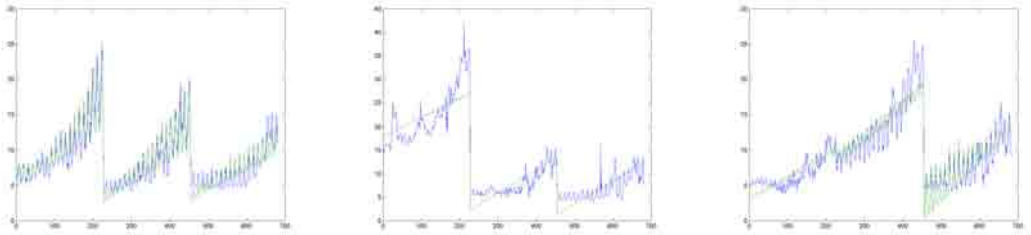
Figure 6.26. Heuristic Pooled Regression Fits for Three Random Groups and the Fist 3 States of Those Groups, $= 2$.



(a) 2nd Group (b) 6th Group (c) 10th Group

Figure 6.27. Heuristic Pooled Regression Fits for Three Random Groups and the Fist 3 States of Those Groups, $= 5$.

As we can see, there are cases in which the pooled fits are rather good, whereas other cases show a marked lack of proper approximation. We may very well face the



(a) 2nd Group (b) 4th Group (c) 13th Group

Figure 6.28. Heuristic Pooled Regression Fits for Three Random Groups and the Fist 3 States of Those Groups, $= 10$.

situation that several time series cannot be reasonably matched in partitions with less than a certain number of states: the heuristic method would then be forced to find the best partition possible, even if this delivers results not quite good.

6.2.4 Grouping Methods Remarks

Let us now compare the advantages and disadvantages for each method, as well as proposed means to correct and improve on them.

Dendrogram Grouping Method

Advantages

- Fast. The clustering method for the various examples presented here required only between 0.33 and 0.5 seconds.
- Replicable. Using the same data and the same linkage method, the resulting partition is always the same.

Disadvantages

- Unbalanced. There are usually many single state groups, whereas the rest of the states are hedged in a few large groups. Defining different partition sizes may deal away with some of the larger groups; additional ways to determine the goodness/homogeneity of a given grouping has to be defined and implemented.

- Statistically Ignoring. States may share a group even if their regression coe cients a_j^i are not equally significant in their respective regression analysis. Modifying the regression model (for example, applying logarithm transformations) until we find a model with nearly all sets of parameters significant would certainly contribute to improve the statistical meaning of the dendrogram clusters, as long as it does not affect regression significance.

Heuristic Grouping Method

Advantages

- It delivers a way to guarantee a minimum value for the R^2 coe cient of determination in every group, as long as we are willing to admit a certain number of groups.
- It can give more importance to individual regression R^2's than it does to pooled regressions R^2's, or vice-versa.
- All the states' regression coe cients estimates a_k^i in a given group have the same statistical significance.

Disadvantages

- Slow. It takes a single iteration of algorithm 5 about 3 minutes to complete. By sampling the regression observations (for example, randomly taking a half of the prices and consumptions for each state) we expect that the pooling speed is improved without sacrificing statistical significance.
- Random. Even a high number of iterations cannot guarantee similar partitions. The selection of a good seed partition (instead of relying on randomly chosen group leaders every time) should contribute to reduce the randomness.

6.2.5 Heuristic Method with Random Sampling

This section describes one of the proposed ideas for improving the running times required by the heuristic algorithm. We hypothesize that, when forming the groups,

selecting just a sample of the observation (both those belonging to the states already in the group, and those belonging to the new candidate state) will reduce the time needed perform the pooled regression and to calculate the (sample) determination coe cient for the possible new group. The determination coe cient corresponding to the full range of observations will be calculated and reported once all the groups are determined, and the effectiveness of this sampling proposal will be done based in this number and in the observed fit graphs.

Table 6.3 shows the results of 20 runs of the heuristic algorithm 5, each run corresponding to one row in the table. We considered three sample sizes: all the observations (1-sample), half the observations (0.5-sample) and one-tenth of the observations (0.1-sample.) For the 1-sample case, the average group-R^2 is given in the first column. For the 0.5-sample and the 0.1 sample cases, we report both the average sampled group-R^2 (i.e., the value of R^2 obtained from the sample and based upon which the groups were formed) and the average full range-R^2 (the one calculated only after each group was completely determined.)

The running times corresponding to variables each of the 60 runs, 20 per sample size, are given in table 6.4. As we can see, the column corresponding to the 0.5-sample has running times lower yet still similar to those corresponding to the 1-sample. On the contrary, the 0.1-sample greatly reduces the time it takes for each run of the heuristic algorithm, almost to half the time required by the 1-sample in some instances.

Although the improvement in times is good, and the values for R^2 are also similar for all the methods, the fact is that the fits for the sampled regressions largely vary in their goodness. Figure 6.29 shows the 1-sample (in green,) 0.5-sample (in red,) and 0.1 sample (in cyan) regression fits for the randomly selected states of Minnesota and North Carolina. While the approximation of all three fits in the North Carolina series is very good, the 0.1-sample is poor in the case of Minnesota.

In both cases, the 0.5 sample performs well but not considerably so. However, looking at the times for the 0.5-sample heuristic clustering, we conclude that there is no reason why should we sacrifice the accurateness of the 1-sample for such a meager save in running time.

Table 6.3. Sampled Heuristic Regression R^2 Values.

1-Sample	0.5-Sample		0.1-Sample	
R^2	R^2	Sample R^2	R^2	Sample R^2
0.8929	0.8883	0.8924	0.8911	0.9132
0.8882	0.8870	0.8929	0.8905	0.9169
0.8915	0.8858	0.8894	0.8879	0.9061
0.8907	0.8937	0.8965	0.8853	0.9197
0.8838	0.8922	0.8954	0.8942	0.9166
0.8926	0.8845	0.8847	0.8873	0.9118
0.8824	0.8859	0.8922	0.8848	0.9171
0.8928	0.8949	0.9004	0.8843	0.9118
0.8864	0.8854	0.8908	0.8849	0.9126
0.8803	0.8906	0.8924	0.8906	0.9227
0.8826	0.8904	0.8929	0.8894	0.9294
0.8827	0.8924	0.8979	0.8826	0.9042
0.8906	0.8947	0.8997	0.8626	0.8990
0.8836	0.8913	0.8962	0.8775	0.9118
0.8862	0.8919	0.9002	0.8832	0.9211
0.8899	0.8862	0.8922	0.8810	0.9112
0.8748	0.8844	0.8891	0.8884	0.9194
0.8865	0.8894	0.8933	0.8851	0.9153
0.8871	0.8954	0.9032	0.8893	0.9204
0.8888	0.8968	0.8997	0.8832	0.9079

Judging from the evidence presented in this section, we conclude that sampled regression is not a good resource for reducing the times required by the heuristic method. Maybe if the series included were better behaved and more similar to each other, the accurateness of the sample regression would be increased enough to be justified by the running time savings.

6.2.6 Dendrogram and Heuristic Method Combination

This section round up the regression analysis part of this work discussing the idea of merging both methods, the Dendrogram Grouping Method, and the Heuristic Grouping Method. This is made attempting to benefit from the speed and

Table 6.4. Sampled Heuristic Regression Running Times.

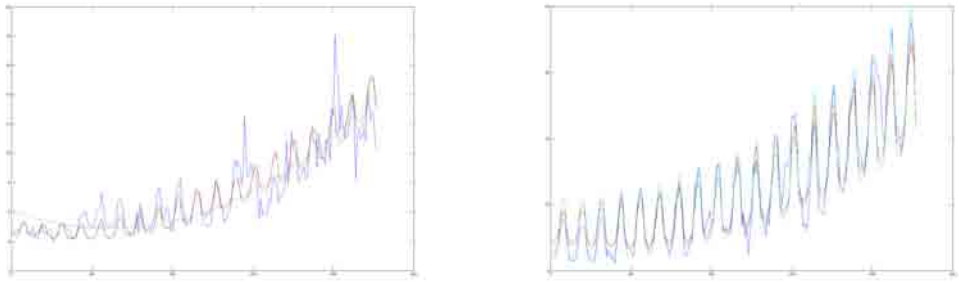
1-Sample	0.5-Sample	0.1-Sample
6.3894	5.7585	3.9058
6.5560	5.9869	3.9531
6.1558	5.3123	3.9382
6.5839	5.5920	4.3128
6.5048	5.4861	4.0752
6.5597	5.8046	3.9957
6.4806	5.4373	3.8577
5.9848	5.5187	3.7808
6.4239	5.5724	4.0592
6.3148	5.8141	3.9694
6.2539	5.4147	4.1343
6.1767	5.2691	4.5270
6.5269	5.4437	3.9792
6.3780	5.7587	3.7899
6.1298	5.8282	4.2254
6.3939	5.5937	3.6094
6.3923	5.4617	4.2027
6.4591	5.5808	4.4910
6.3414	5.4348	3.9557
6.3831	5.1930	4.3500

predictability of the DGM, forming an initial partition, and the ability of the HGM to provide a statistically significant and a more homogeneous partition.

We first form a DGM partition using the ACDC method and demanding 15 groups, which we have observed delivers fairly homogeneous groups; then, we randomly select a state from each group formed and use it as a group leader for the heuristic method.

We will make 10 runs of the HGM with full information (1-sample) and five attempts each (that is, $= 5$,) allowing only 5 extra groups (to a possible maximum of 20 groups) and using a R^2 threshold equal to the minimum group-R^2 value obtained by the dendrogram regression.

Table 6.5 shows the values for the *Sim* function in line 4.16, for each of the 10 partitions produced and also for the dendrogram seed-partition.



(a) Minnesota Sampled Fits (b) North Carolina

Figure 6.29. Sampled Regression Fits for Two States.

Table 6.5. Dendrogram-Heuristic Groupings

	Heuristic Groups									
Group	HC1	HC2	HC3	HC4	HC5	HC6	HC7	HC8	HC9	HC10
ADDC	0.22	0.24	0.22	0.22	0.30	0.23	0.24	0.22	0.21	0.22
HC1		0.30	0.36	0.33	0.33	0.36	0.39	0.40	0.35	0.47
HC2			0.41	0.45	0.44	0.34	0.33	0.36	0.46	0.30
HC3				0.49	0.47	0.52	0.51	0.51	0.31	0.50
HC4					0.42	0.36	0.39	0.44	0.29	0.42
HC5						0.52	0.45	0.39	0.32	0.40
HC6							0.47	0.39	0.29	0.37
HC7								0.63	0.42	0.45
HC8									0.56	0.46
HC9										0.43

To have an idea of the meaningfulness of the values of function *Sim*, tables 6.6 and 6.7 condense the partitions HC7 and HC8, which have the highest similitude function value in table 6.5. In each table, a state has been boldfaced if it shares a group with at least another state in the other table, or if it appears alone in both tables. In total, 38 states appear reasonably matched in both partitions, with some groups like the Missouri–Indiana–Virginia–Maryland, or the larger Arkansas–Alabama–(...)–Nevada, being identical in both tables.

All the heuristic groups delivered higher similitude values when compared among themselves than when compared with the ADDC partition. This is encouraging, as

Table 6.6. HC7 Partition

Illinois , **Ohio**, Rode Island, **Delaware, Mississippi**, **Oregon**
New York
Texas
Michigan, **West Virginia**, **Nebraska**, Utah, **Connecticut**
New Jersey, New Mexico
Vermont, **DC**, **New Hampshire**
North Carolina, **Oklahoma**, **Iowa**, **Kansas**, **Colorado**
Georgia
Arkansas, Alabama, Florida, Minnesota, Tennessee, Kentucky, Washington, Wisconsin, Louisiana, Arizona, Nevada
Pennsylvania
California, South Carolina
Massachusetts, **South Dakota, North Dakota, Wyoming, Montana**
Missouri, Indiana, Virginia, Maryland
United States, Idaho, Maine
Alaska, Hawaii

Table 6.7. HC8 Partition

Illinois
Ohio, **Oregon**, **DC**, **New Hampshire**
New York
Texas, New Mexico
Michigan, Rode Island, **South Dakota, Wyoming**, Idaho
New Jersey, South Carolina
Vermont, Maine
North Carolina, **Iowa**, **Colorado**
Georgia
Arkansas, **Tennessee**, **Florida**, **Kentucky**, **Kansas**, **Nevada**, **Wisconsin**, **Washington**, **Minnesota**, **Louisiana**, **Arizona**, **Alabama**, **Oklahoma**
Pennsylvania
California, **Alaska, Hawaii**, **Montana, North Dakota**, Utah
Massachusetts, **West Virginia, Connecticut, Nebraska**
Missouri, Virginia, Maryland, Indiana
United States, **Delaware, Mississippi**

it means that the heuristic method is actually modifying in a useful manner what was initially made by the dendrogram method. The similitude values, while may not seem overly attractive at a first glance, are clearly good when compared to those in table 6.8, which was performed using a random vector instead of the ADDC seed-partition for each run of the heuristic algorithm (but otherwise the same running parameters.) Whereas the combined method has an average similitude value of 0.63, using only the heuristic method with randomly chosen group leaders delivers an average similitude value of 0.44. A Wilcox signed rank test at 0.001 significance rejects the null hypothesis of equality of the means in favor of a larger mean similitude value for the partitions created using the combined method.

Table 6.8. Random Start Heuristic Groupings

	Heuristic Groups								
Group	HC12	HC13	HC14	HC15	HC16	HC17	HC18	HC19	HC20
HC11	0.21	0.27	0.36	0.38	0.35	0.44	0.29	0.30	0.31
HC12		0.27	0.15	0.36	0.28	0.24	0.28	0.22	0.34
HC13			0.22	0.32	0.31	0.22	0.21	0.44	0.37
HC14				0.32	0.21	0.27	0.32	0.32	0.22
HC15					0.35	0.30	0.33	0.43	0.23
HC16						0.29	0.30	0.31	0.24
HC17							0.27	0.31	0.38
HC18								0.25	0.27
HC19									0.40

As for the R^2 values, while the combined approach indeed improved the average group R^2 value of the seed-partition created with the DGM, these R^2 are not significantly different from those obtained with using only the heuristic method. Both the combined and the heuristic approaches deliver group-R^2s of around 0.9, with very little variance and similar maximum and minimum group-R^2 for both cases.

From the evidence here, we conclude that using a somehow balanced DGM partition as a starting point, we can improve the statistical significance of the regression and, in general, the homogeneity of the groups formed by the HGM. While the combined approach does not significantly improves R^2 values obtained

using only the HGM, there is statistical evidence that the similitude is improved upon the DGM seed-partition after applying the combined method.

6.3 Stochastic Bilevel Optimization Results[2]

Concluding this chapter is exposition of the numerical results corresponding to the stochastic bilevel optimization model in section 4.4.3. The model $SPR2$ is an expansion over the deterministic formulation developed in chapter 3. This optimization problem is solved using the entire information and evidence obtained in this and the preceding chapter.

In sections 5.1 and 5.2, we already proved (theoretically) that the bilevel linear reformulation of problem $PR3'$, which was called $PR5$, is indeed equivalent to the problem, and that it provides solutions considerably faster than using nonlinear solving methods. We thus decided in favor of model $PR5$, which has (almost) linear problems in both levels. For the exact same reasons, we will prefer $SPR2$ over $SPR1$ when working the bilevel stochastic setting.

Section 5.3 described two computational solution techniques for the bilevel deterministic problem, indicating that the direct approach, once applied to a bilevel linear problem, had solutions *at least as good as the Inexact Penalization Approach* given in the references. The running times for the direct method, however, were considerably faster when dealing with large problems. Since the stochastic model (4.18a)-(4.18i), (4.22a)-(4.22p) is similar in structure to problem $PR5$, the direct approach is chosen for solving the stochastic bilevel problem instances.

In the forecasting part of this chapter (section 6.1,) we concluded that neither the Seasonal Holt-Winters nor the ARAR methods is significantly better than the other when applied to the data we are using, so both methods can be applied to our ends. For no more reason than the slightly faster computation times, we will use the Seasonal Holt-Winter method to produce the forecasts needed in this section.

The regression analysis performed in section 6.2 validates the usage of a given state's consumption time series to represent, by means of a regression formula, that state price time series. Furthermore, the pooled regression analysis justifies the formation of clusters or groups of states within which the same regression

[2]The results in this section have appeared in the authors' [54]

relationship can be applied. These two facts permits us to use a single ternary scenario tree as the one described in section 4.3 for a given stochastic problem instance, instead of an exponential combination of trees. Because of its fairly straightforward and computationally cheap implementation, and the obvious improvements in group homogeneity and results repeatability, the combined Dendrogram-Heuristic clustering method was chosen to produce partitions of the state set.

Based in the specifics above, we generated a batch of 20 problem instances, labeled *BXXX*, using a modified version of the problem generating algorithm in section 3.4 that uses SHW predictions to characterize a scenario tree as shown in 4.3.1, built inside groups generated by the combined Dendrogram-Heuristic clustering method. The time series data used is again drawn from the o cial figures of the American Energy Information Administration.

6.3.1 Nature of the Solution Structure

The optimization will report:

- one expected value calculated over all scenarios
- one $SN \quad P$ matrix for the variable EP
- one running time
- the optimal value of q
- K blocks of solution variables (one per scenario.) Each one of these blocks consists of
  - Three $N \quad P$ matrices, one for each of the upper level decision variables x, s, EA
  - one P-sized vector for the final imbalances y, and
  - two $P \quad P$ matrices for the backward/forward moves of gas u, v.

Scenarios that share one or more nodes will have equal values for x and EA in those nodes (i.e. all scenarios share the first $N \quad P$ values of x and EA, one third of the scenarios share the first $2N \quad P$ values of x and EA, and so on.)

Table 6.9 gives the decision variables x and EA for the problem instance B001, which has 12 days divided in 3 stages, and 4 pool zones. The final imbalances y_i for scenario 1 (correspondent to an all-decreasing consumption forecast and the corresponding regressed prices) match the final day imbalances x_{kNi}, delivering a final value revenue for the NGSC of $4,390,721.13$. Running time was $245.77s$, and the optimal value for q is 1.

Table 6.9. Optimal Values of Variables x, EA for Scenario 1 of Problem B001

x				EA			
133	-46	-67	181	1,070.32	968.27	1,201.77	767.22
109	-43	-58	157	1,330.72	1,471.92	1,454.22	1,096.95
89	-33	-55.89	134	2,560.86	1,986.79	2,435.16	2,678.91
87	-33	-23.98	107	6,237.95	2,378.29	5,511.81	5,693.08
94.04	-16.05	-3.98	106.55	10,257.52	2,541.01	7,004.89	7,982.71
94.10	-1.29	21.71	84.73	9,349.71	1,944.97	5,966.54	7,387.38
98.99	15.91	56.16	79.22	6,730.28	1,875.78	4,605.23	5,863.75
94.80	39.96	90.15	62.91	4,301.35	1,270.21	2,777.24	3,610.51
97.48	61.11	116.12	55.69	2,568.65	863.31	2,025.76	1,879.41
111.49	83.97	137.52	55.01	1,964.41	609.81	1,538.82	1,373.32
123.88	94.66	147.61	57.66	1,839.48	474.52	1,348.99	1,325.32
118	98	147	64	1,832.54	572.38	1,293.27	1,306.49

6.3.2 Solution Reports

This section summarizes the results for the nine problem instances tested. The figures in table 6.3.2 are, for each problem: the optimal value for the upper level (ULOV), the mean optimal lower level value (mLLOV), the running time for the problem (RT), and the optimal value for the variable q.

Running times are varied, even among similarly sized problems. Overall, half of the 20 instances tested delivered results in less than $1000s$. Problem B002 was exceptionally fast to solve at less than one minute, whereas the largest instance tested took nearly two and a half hours to be solved.

Besides the stochastic problem solutions, another interesting features to look at is the comparison of the Stochastic Solution (SS) shown above, the Perfect Information Solution (PIS, obtained when knowing beforehand the values of the

Table 6.10. Solution reports for problems B001-B020.

Instance	N	P	S	ULOV	mLLOV	RT	q
B001	12	4	3	891,190.48	29.01	$245.77s$	1
B002	8	5	2	4,449,570.40	-60	$21.45s$	0
B003	12	5	3	8,413,667.18	68.34	$929.99s$	1
B004	12	5	3	7,943,942.17	-53.73	$3,319.53s$	0
B005	12	5	3	1,306,501.04	-7.86	$421.23s$	0
B006	12	5	3	510,335.27	-2.13	$120.43s$	0
B007	12	5	3	2,123,587.05	-87.48	$281.89s$	0
B008	12	5	3	2,753,643.38	0	$650.31s$	0
B009	12	5	3	1,853,811.99	-6	$398.76s$	0
B010	12	5	3	3,973,184.60	47.30	$706.78s$	1
B011	28	4	4	2,663,283.93	-56.74	$7,893.23s$	0
B012	8	5	2	3,298,333.98	0.00	2,321.69s	0
B013	8	5	2	3,417,055.13	0.00	2,078.14s	0
B014	8	5	2	2,477,883.48	0.00	1,891.12s	0
B015	8	5	2	3,667,407.35	-2.04	701.21s	0
B016	10	7	2	4,742,822.11	-49.00	5,002.13s	0
B017	10	7	2	2,862,635.34	38.00	2,548.44s	1
B018	10	7	2	1,901,243.90	0.00	4,518.93s	1
B019	10	7	2	2,413,298.05	37.34	4,950.69s	0
B020	10	7	2	3,437,553.53	10.00	4,735.71s	1

parameters previously considered stochastic,) and the Expected Value Solution (EVS,) obtained using the expected values of the stochastic parameters in a single-staged optimization problem.

Both the PIS and the EVS are procured by solving a $SPR2$ problem with a single scenario, which is more or less similar to the deterministic problem $PR5$. The PIS is the best possible solution: it can only be attained by having a perfect forecast method; the EVS is the result of solving a very simply designed stochastic problem using only mean predictions. The results for these values are summarized in table 6.11. Notice the considerably low revenues predicted by the EVS.

The values in columns three and four, SS and EVS, are just the apparent solution obtained by these approaches. This is not the value that the NGSC would obtain if it uses either the SS or the EVS optimal points to plan its production. The values

Table 6.11. Revenues Attained by the PSI, the SS and the EVS

Instance	PIS	SS	EVS
B001	892,046.450	891,190.480	321,597.810
B002	4,809,743.730	4,495,704.000	4,447,902.350
B003	7,565,976.720	8,413,667.180	299,320.010
B004	7,314,748.100	7,943,942.170	277,613.600
B005	1,450,506.940	1,306,501.040	64,176.150
B006	687,882.410	510,335.270	33,709.760
B007	4,009,499.020	2,123,587.050	75,496.910
B008	4,753,643.380	4,464,049.990	107,444.870
B009	3,660,097.860	1,853,811.990	69,717.000
B010	3,973,184.600	5,732,921.020	148,418.230
B011	2,860,385.920	2,663,283.930	10,941.390
B012	3,339,391.970	3,298,333.980	594,800.360
B013	3,456,927.710	3,417,055.130	613,688.100
B014	2,346,027.840	2,477,883.480	404,106.510
B015	3,770,818.280	3,667,407.350	729,237.780
B016	5,544,639.960	4,742,822.110	1,009,750.530
B017	3,498,196.160	2,862,635.340	484,217.120
B018	3,620,877.780	1,901,243.900	364,856.310
B019	4,007,071.210	2,413,298.050	438,157.220
B020	2,714,532.160	3,437,553.530	484,318.740

for the Stochastic Solution Implementation (SSI) and the Expected Value Solution Implementation (EVSI), shown in table 6.12, represent the actual gain from the usage of the SS or EVS solutions.

The PIS optimal value is the highest that can be achieved in table 6.12, as opposed to table 6.11 in which the SS or EVS values can underestimate or overestimate the PIS value. That is, the implementation values SSI and EVSI can only be as high as the PIS value, and the closest they are to it, the better the former solutions are considered.

To compare the PIS and the SS and EVS solutions, we also introduce the relative error ratios, SSRE and the EVSRE. Shown in the fourth and fifth columns, these relative errors are calculated as the difference between the PIS value and the SS

Table 6.12. Revenues from the Implementation of the Obtained Solutions

Instance	PIS	SSI	SSRE	EVSI	EVSRE
B001	892,046.450	690,430.120	0.226	225,820.630	0.747
B002	4,809,743.730	3,941,490.700	0.181	4,193,267.120	0.128
B003	7,565,976.720	6,775,471.650	0.104	243,360.910	0.968
B004	7,314,748.100	6,679,739.870	0.087	254,197.160	0.965
B005	1,450,506.940	1,191,949.530	0.178	23,320.520	0.984
B006	687,882.410	440,987.770	0.359	8,356.560	0.988
B007	4,009,499.020	2,836,230.010	0.293	71,259.130	0.982
B008	4,753,643.380	3,439,549.720	0.276	103,326.530	0.978
B009	3,660,097.860	2,563,110.730	0.300	56,860.690	0.984
B010	3,973,184.600	3,636,830.260	0.085	142,463.870	0.964
B011	2,860,385.920	1,281,300.450	0.552	20,145.870	0.993
B012	3,339,391.970	3,248,556.140	0.027	410,164.810	0.877
B013	3,456,927.710	2,350,283.110	0.320	430,266.640	0.876
B014	2,346,027.840	1,885,270.860	0.196	223,830.330	0.905
B015	3,770,818.280	2,995,647.930	0.206	504,094.130	0.866
B016	5,544,639.960	3,928,716.040	0.291	466,874.490	0.916
B017	3,498,196.160	378,844.380	0.892	22,034.310	0.994
B018	3,620,877.780	3,190,119.320	0.119	394,682.310	0.891
B019	4,007,071.210	2,860,895.910	0.286	460,516.230	0.885
B020	2,714,532.160	2,355,637.710	0.132	297,257.550	0.890

(EVS) value, divided by the PIS value, that is,

$$\text{SSRE} = \frac{(\text{PIS} - \text{SSI})}{\text{PIS}}, \quad \text{EVSRE} = \frac{(\text{PIS} - \text{EVSI}|)}{\text{PIS})},$$

for each one of the 20 instances. A ratio near 0 means that the approximation of the PIS value by the SSI (EVSI) is good; on the contrary, a ratio near 1 implies that the SSI (EVSI) are considerably poor when implemented in a given situation.

Of the 20 SSREs, only two are above 0.5 (the SSI solution is less than half that of the PIS,) with the highest relative error at 0.89 in problem B017. Only one of the EVSREs values displayed is below 0.75. The only instance in which the EVSI provides a good approximation of the PIS value is B002, and even in this case the figure is very similar to that obtained by the SSI. Results employing the ARAR

forecasting method instead of the Seasonal Holt-Winters method deliver similar results, heavily underestimating the PIS value.

It becomes readily apparent that the solving the problem forecasting only the mean predictions heavily underestimates the possible future gains, and, since the solution vector is obtained optimizing under this underrated gains scenario, the implementation also ends up being lacking, at least for this data and model selection. Comparing the relative errors for the EVS and the SS, it safe to state that the usage of a stochastic framework considerably improves over the simpler approach of solving a deterministic-like variant obtained by solely forecasting the mean predictions.

Chapter Summary

This chapter reports the numerical results pertaining to the three main parts of this work that employ stochastic variables, namely, time series forecasting, time series regression analysis, and stochastic optimization.

The time series for monthly prices and consumptions per state in the US were tested with two different forecasting methods. No conclusive evidence was found favoring one or the other, but the Seasonal Holt-Winters algorithm was found to a certain extent faster and with more normal residuals.

Using regression analysis and dendrogram theory, we created two grouping methods that cluster the states showing similar regression parameters into sets that behave likewise, each one based in different concepts and exhibiting equally different strengths and weaknesses. Obtaining a good pooled regression function is of great aid in reducing the amount of information needed while forming the scenario trees for the stochastic problems. It ultimately leads to less variables in the upper level and less evaluations of the lower level in the solution of the bilevel optimization problem. The combination of the clustering methods later proposed, permits us to gain homogeneity in the partition, significance of the regressions and, to a certain extent, repeatability of the results.

Lastly, we solved twenty instances of the bilevel stochastic optimization problem with recourse, which has been already linearized in chapter 4. Except for a reduced number of cases, the stochastic formulation delivered fairly good approximations of

the Perfect Information Solution values, with only two of the twenty instances presenting Stochastic Solution Relative Errors below half the PIS value. On the contrary, the EVS appear as overwhelmingly surpassed by our stochastic model, with only one instance providing a relative error below 0.85. Indeed simply forecasting the expected values for each stochastic parameter has a considerable lack of accuracy and robustness. The better the approximation, the better the solution tools this chapter described would be when used as decision tools for NGSC in their daily operations.

CHAPTER 7
CONCLUDING REMARKS

Culminating this dissertation work, the current chapter collects the conclusions, discussions, and general highlights drawn from all the other chapters of the document. Sections 7.1 summarizes the most important points in the introductory chapters 1 and 2, and how these impact the research we herein present. Sections 7.2 shows the major specifics regarding the deterministic mathematical optimization models in chapter 3. It also discusses the results obtained after experimentation in chapters 5 concerning this deterministic cases. In turn, section 7.3 does the same for the stochastic models, summing up the theoretical exposition of chapter 4 and the experimental findings attained in chapter 6.

7.1 Background

Natural gas markets were affected by laws and regulations around the world, which had the intention of fostering a new type of economic dynamics for the fuel. Natural gas has a rapidly developing industry, in virtue of its availability, clean burning, and potential for both energy production and household usage. While many new scenarios were born form the FERC resolutions 636 and 637, and the European Union Regulation (EC) 1775, the most important concern for us in this work is the study of the new set of relationships between the Pipeline Operating Companies (POCs) and the Natural Gas Shipping Companies (NGSCs.)

Because of the functional prohibition from acting as NGSCs that the POCs now face, they are essentially relegated to act as mere transporters of gas, but never as marketers. Only NGSCs independent form the POCs can buy and sell gas from the wells and to the public. While generally thought as beneficial for all sectors, this new market setting proved to be more complicated for the POCs, who have to find additional means to guarantee the operative balance of their lines, which are forcefully opened to a number of NGSCs in the area.

In this work, we study a situation with only one NGSC and one POC interacting with the usage and balance of a pipeline. The NGSC injects and extracts natural gas into the system based on a prearranged contract, but is allowed for a certain

imbalance tolerance. To discourage the NGSC from creating large imbalances in the system, the POC imposes the former with a cash-out penalization. The NGSC then has to decide how whether such penalization is a good tradeoff compared to the profits gained from the gas over-extracted, or not.

We thus propose a bilevel optimization scheme to model this situation. In this model, the NGSC is the leader or upper level decision maker, and the POC is the follower or lower level decision maker. The NGSC controls a set of variables that represent planned and actual extraction amounts, imbalances created, etc. It controls these variables so as to sell the largest possible amount of gas each day without incurring in large penalizations. The POC controls variables regarding the rearrangement of the imbalances in all pool zones. The goal of the POC in the lower level is to obtain a reconfigured system with a small total imbalance, in which the cash-out to/from the NGSC is minimal.

We have not reportedly encountered a bilevel model of this specific subsystem of the natural gas supply chain, save for those created within our own research group. Most natural gas network models apparently focus in the operation of the pipelines, trying to find and control the parameters that govern the network to obtain operations with minimal costs. Other network models applied to natural gas more akin to the ones developed here, consider the system with considerably different assumptions, like a different number of NGSC in the same problem, or the possibility of selling gas at a futures' market.

Beyond providing a deterministic model, we went on and formulated a bilevel stochastic problem for our NGSC-POC subsystem. The price and demand faced are now randomly distributed and we make use of real world data to estimate, forecast and regress the figures we need. Models in the literature regarding natural gas price/consumption we have found are usually dedicated to econometric analysis of the market, many of them with the ultimate purpose of assessing, for example, the elasticity of a given price/consumption setting. More practical purposes, such as forecasting future behaviors, present models with different characteristics, like a considerable different regression formula, or the inclusion of extraneous considerations, like the number of appliances, or the seasonal temperature changes.

The problem statement given in section 1.4 clearly shows the type of model we are developing, and the most general assumptions required for it to be an adequate abstraction.

7.2 Deterministic Models' Remarks

Chapter 3 shows the theoretical work performed to develop a deterministic model that represents the setting explained above.

We begin with a nonlinear-mixed integer bilevel optimization problem, originally formulated in the work by Ríos-Mercado, Secomandi and Buraparte [88]. This model, while the first to abstract the NGSC-POC subsystem in such way, has many shortcomings, as it does not consider important interests for the parties involved.

Throughout the chapter, we display the expansion and reformulation of this original model, adding new terms and constraints to improve its meaningfulness, accordance with the real world situation, and usability for the NGSCs. This new features, however, had the unavoidable consequences of adding up complexity to the models. Lemmas 2 and 3 explain two models that, as it is theoretically demonstrated, have equivalent solution vectors to each model in the modified upper level model and are considerably simpler. Applying these lemmas in a bilevel setting we obtained problem $PR5$, which is a bilevel linear stochastic problem. The linearization lemmas rely on artificial variables that increase the number of constraints, but that also considerably decrease the complexity of the problem, allowing us to solve the independent level with the efficient tools of numerical linear optimization.

In order to computationally implement and solve the deterministic models formulated, we state two solution techniques developed in [59], adapting them to the new models here created. Other aspects of the bilevel optimization are described, such as the problem generation procedure, the choice of starting points, and the like. Means to address/correct/implement these are lastly proposed.

In chapter 5, we observed evidence that strongly supports the usage of linearization lemmas and the models therein exposed when working numerically. Both linear problems, when optimized individually, provided accurate solutions

equaling the optimal values of their correspondent nonlinear formulations, in just fractions of the times required by the latter.

After analyzing our options to solve specific parts of the complete problem and concluding about their employment, we compare the bilevel optimization problem, pitting the Direct Method and the Inexact Penalization Method, each with two different upper level objective functions.

The results show that, for small to medium-sized problems, both methods provide similar solutions, with the Direct Method having considerably shorter running times. This is arguably because of the greater advantage this method takes of the linearizations performed. For larger problems, the IPA often failed to obtain a good feasible solution in reasonable times.

The chapter concludes in favor of the bilevel linear models and the direct solution methods, with further comments on the smaller details arising in the deterministic bilevel solution process.

7.3 Stochastic Models' Remarks

There are three main sections in chapter 4, each one presenting varied aspects of the stochastic models studied. Also, each section has a counterpart in chapter 6, in which the experiments regarding the former chapter are detailed.

Both chapters have their opening parts concerned with the time series formed by data from the Energy Information Administration and the forecast models used on them. The well-known ARAR and Seasonal Holt-Winters (SHW) prediction methods, taken from [15], are summarized for their later use.

After the proper numerical analysis is carried on, both the ARAR and the SHW forecast procedures showed strengths and weaknesses. While their predictions seem equally good when compared to the figures they estimate, the residual analysis shows less-than-desirable normality and log-normality, as well as some lack randomness. Because of the intended use of the residuals in the scenario tree generations, and the quality of the predictions, the shortcomings of the forecast methods are not deemed serious enough and the ARAR and SHW techniques are accepted.

Still looking at the parameters' time series, we developed a price/consumption regression model in the hopes of reducing the amount of information needed in a given stochastic problem, needing only to use one time series and then regress the other. All 52 individual regression models are satisfactory and statistically significant.

The regression analysis is taken forth with the inclusion of pooled regression analysis. Using two different methods created here, we classified the states in a number of groups or clusters that define a partition of the state set. In each element of the partition, the states contained are expected to share a single regression formula and therefore present a rather similar behavior. The first of the clustering methods (Dendrogram Grouping) we developed is based in the similitude of the individual multiple linear regression parameter vectors. This method is fast, but it may deliver very unbalanced partitions, and the formed groups do not necessarily have similar statistical significance.

The second of the two clustering methods (Heuristic Grouping) is based in a greedy formation of groups that attempts to maximize the groups' pooled R^2 value. This method is slower but it can guarantee a minimum R^2 value and has the same statistical significance for every member in a given group.

Combining the clustering techniques, we managed to improve on statistical meaning and homogeneity (provided by the heuristic grouping method) and on replicability (to some degree, given by the dendrogram approach.) The experiments tested showed that the partitions obtained using the heuristic grouping are more similar among each other when the partition's group leaders are selected based in a dendrogram grouping, than when the heuristic grouping is formed after a random selection of group leaders. While replicability is improved, the R^2s values (which are already high enough) are not apparently increased.

The bilevel linear model, already proved advantageous in the deterministic case, is slightly modified to obtain stochastic formulations of the bilevel problem. There is first an explicative stochastic problem, which is latter modified (much like its deterministic analog, and using similar lemmas,) to obtain a bilevel linear stochastic optimization problem with recourse, which is numerically optimized in a setting similar to that of model $PR5$.

Results for the stochastic case were obtained after testing 20 instances. Besides the Stochastic Solution (SS), we also calculated the Perfect Information Solution (PIS, the best possible revenue, only attainable with flawless forecasts,) and the Expected Value Solution (EVS, a simple minded solution considering only the centermost scenario for each tree.) The revenue obtained for implementing the SS and the EVS, as well as their relative error values when compared with the perfect information solution are also given. Of all the instances, only two saw the stochastic implementation value below half of the PIS; on the contrary, the relative errors of the EVS implementation are, with a single exception, above 0.74. The SS is therefore deemed better than the alternative EVS, which lacks heavily in usefulness.

All stochastic instances were generated using the procedure previously stated, with SHW forecast performed on states belonging to groups defined by the dendrogram/heuristic combinations.

7.4 Final Notes and Future Research

While the work here expounded, from our point of view, a sizable improvement over the sources inspiring this work, much needs to be done before the models can be considered complete. Some research opportunities discovered during the continuous experimentation in this area include:

- Analyze a NGSC-POC cash-out setting in which there are more than one NGSC acting in a single pipeline; or
- Considering the case in which imbalance creation is discouraged not by cash-out penalization, but through volume auctioning among several NGSCs or other ways.
- Include gas sales for the NGSC other than those for spot prices.
- Expand the deterministic and stochastic models to include further terms of interest to either the NGSC or the POC in the objective functions/constraints of the optimization problem.
- Study the possible existence of linear equivalent models for each expansion proposed, in view of the clear computational advantages these have.

- Propose alternative analytical/heuristic solution methodologies for the bilevel optimization
- Code the entire framework in a computer language other than Matlab, to estimate the convenience of the latter compared, for example, with lower level languages.
- Considering the usage of other types of times series data, with characteristics different to those employed, and evaluating the suitability of the regression and forecasts models herein proposed, or the implementation/development of new approaches.
- Apply transformations to the time series data to obtain better autocorrelations, residual normality, independence, and randomness in both the regression and the forecasting models.
- Introduce GRASP/Taboo search procedures into the Heuristic Grouping Method to fine-tune the construction of the state set partition and possibly increasing repeatability, while decreasing the running times.
- Implement a rolling horizon approach to the stochastic problem to better reflect the dynamics of the decision process.

With these focus points, we expect that a continued further development of the research detailed in this work can eventually position it as a valid, accurate, and useful framework in the day-to-day decision making process of the parties involved in the NGSC-POC subsystem of the natural gas supply chain.

BIBLIOGRAPHY

[1] Administration, E. I. Natural gas model description. Retrieved August 9th from http://www.eia.doe.gov/ emeu/steo/pub/document/textng.html.

[2] Administration, E. I. Eia - annual energy outlook 2009 - natural gas demand.

[3] Arnold, L. Stochastic Differential Equations: Theory and Applications. *New York et al* (1974).

[4] Asche, F., Nilsen, O. B., and Tveteras, R. Natural gas demand in the european household sector. *The Energy Journal* (Aug 2008).

[5] Balestra, P., and Nerlove, M. Pooling Cross Section and Time Series Data in the Estimation of a Dynamic Model: The Demand for Natural Gas. *Econometrica 34*, 3 (1966), 585–612.

[6] Baltagi, B., Bresson, G., and Pirotte, A. Comparison of forecast performance for homogeneous, heterogeneous and shrinkage .... *Economics Letters* (Jan 2002).

[7] Bard, J. F. *Practical Bilevel Optimization: Algorithms and Applications*. Kluwer Academic Publishers, 1998.

[8] Beierlein, J., Dunn, J., and Jr., J. The demand for electricity and natural gas in the northeastern united states. *The Review of Economics and Statistics 63* (Aug 1981), 403–408.

[9] Bioenergy Feedstock Information Network. Energy conversion factors. Retrieved July 6, 2008 from http://bioenergy.ornl.gov/papers/misc/energy_conv.html.

[10] Boltz, W. Report on the compliance with regulation 1775/2005. Tech. Rep. Ref: E07-TRA-02-03, European Regulators Group for Electricity and Gas, February 2007. XII Madrid Forum, Spain.

[11] Borraz-Sánchez, C. Improving the Operation of Pipeline Systems— on Cyclic Structures by Tabu Search!

[12] Borraz-Sanchez, C., and Rios-Mercado, R. A non-sequential dynamic programming approach for natural gas network optimization. *WSEAS Transaction on Systems 3*, 4 (June 2004).

[13] Borraz-Sanchez, C., and Rios-Mercado, R. Z. A Hybrid Meta-heuristic Approach for Natural Gas Pipeline Network Optimization. *LECTURE NOTES IN COMPUTER SCIENCE 3636* (2005), 54.

[14] Brockwell, P., and Davis, R. *Time Series: Theory and Methods*. Springer, 1991.

[15] Brockwell, P. J., and Davis, R. A. *Introduction to time series and forecasting*. Springer New York, 2002.

[16] Brown, R., Marx, B., and Corliss, G. Mathematical Models for Gas Forecasting.

[17] Call, S., and Holahan, W. *Intermediate microeconomics*. Belmont, Calif.: Wadsworth, 1983.

[18] Central Intelligency Agency. CIA world factbook. Tech. rep., Central Intelligency Agency, 2008.

[19] Chaton, C., Creti, A., and Villeneuve, B. Some economics of seasonal gas storage. *Energy Policy 36*, 11 (2008), 4235–4246.

[20] Cremer, H., and Laffont, J. J. Competition in gas markets. *European Economic Review 46*, 4-5 (2002), 928–935.

[21] DeBuck, P. Heading for the slippery way: Eu gas regulation with eu gas regulation with comitology procedure. In *EU Energy Law Seminar* (2006).

[22] Dempe, S., Kalashnikov, V. V., and Perez, G. A. Mixed-integer bilevel programming: Application to an extended gas cash-out problem. In *International Business and Economics Research Conference* (Las Vegas, NV., USA, October 2006), R. P. Clute, Ed., The Clute Institute for Academic Research.

[23] Dempe, S., Kalashnikov, V. V., and Rios-Mercado, R. Z. Discrete bilevel programming: Application to a natural gas cash-out problem. *European Journal of Operational Research 166*, 2 (2005), 469–488.

[24] Di Pillo, G., and Roma, M. *Large-Scale Nonlinear Optimization*. Springer, 2006.

[25] Doane, M., and Spulber, D. Open access and the evolution of the u. s. spot market for natural gas. *Journal of Law and Economics 37*, 2 (Oct 1994), 477–517.

[26] Energy Information Administration. FERC order 636: The restructuring rule. Retrieved June 21, 2008 from http://www.eia.doe.gov/oil_gas/natural_gas/analysis_publications/ngmajorleg/ferc636.html.

[27] Energy Information Administration. FERC policy on system ownership since 1992. Retrieved June 21, 2008 from http://www.eia.doe.gov/oil_gas/natural_gas/analysis_publications/ngmajorleg/fercpolicy.html.

[28] Energy Information Administration. Natural gas summary. Retrieved November 2008 from http://eia.doe.gov, 2008.

[29] Environmental Protection Agency. The impacts of FERC order 636 on coal mine gas project development. Retrieved June 21, 2008 from http://www.epa.gov/cmop/docs/pol004.pdf.

[30] EuroGas. Eurogas long term outlook to 2030, 2009.

[31] European Regulation Group for Electricity and Gas. Guidelines for good practice for gas balancing (ggpgb): An ergeg public consultation paper. Tech. Rep. Ref: E06-GFG-17-04, ERGEG, 2006.

[32] European Regulators Group for Electricity and Gas. Gas Balancing. An ERGEG conclusions paper. Tech. Rep. E06-GFG-17-03, ERGEG, April 2006.

[33] Flores-Villarreal, H. J., and Ríos-Mercado, R. Z. Computational experience with a GRG method for minimizing fuel consumption on cyclic natural gas networks. *Computational Methods in Circuits and Systems Applications*, 90–94.

[34] Flores-Villarreal, H. J., and Rios-Mercado, R. Z. Efficient operation of natural gas pipeline networks: Computational finding of high quality solutions. In *International Applied Business Research Conference* (2003).

[35] Freise, J. The anatomy of a natural gas price spike - past and future. The Oil Drum, Aug 2009.

[36] Gassco. Retrieved November 17, 2008 from http://www.gassco.no/wps/wcm/connect/Gassco-EN/gassco/Home.

[37] GEODE. GEODE position paper on the 3rd EU energy package. Tech. rep., GEODE, Barcelona, November 2007.

[38] George, H. *Progress and Poverty*. Cosimo Inc, 2005.

[39] Gomez-Villalva, E., and Ramos, A. Risk management and stochastic optimization for industrial consumers.

[40] Gomez-Villalva, E., and Ramos, A. An algorithm for the mid-term forecast and scenario generation of natural gas and fuel oil prices. *IEEE Transactions on Power Systems* (2004), 00124–2004.

[41] Gowdy, J. Industrial demand for natural gas: Interindustry variation in new york state. *Energy economics* (Jan 1983).

[42] Gutiérrez, R., Nafidi, A., and Gutiérrez Sánchez, R. Forecasting total natural-gas consumption in Spain by using the stochastic Gompertz innovation diffusion model. *Applied Energy 80*, 2 (2005), 115–124.

[43] Horst, R., and Pardalos, P. M. *Handbook of global optimization*. Kluwer Academic Publishers, 1995.

[44] Hugh, D., Hammonds, S., and Whithington, B. Gas balancing rules in europe; a report for CREG. Report, NERA Economic Consulting and TPA Solutions, December 2005.

[45] Huntington, H. Federal price regulation and the supply of natural gas in a segmented field market. *Land Economics 54*, 3 (Aug 1978), 337–347.

[46] Hwang, C., Williams, J. L., and Fan, L. T. *Introduction to the Generalized Reduced Gradient Method*. Institute for Systems Design and Optimization, Kansas State University, 1972.

[47] Ignizio, J. P. *Introduction to Expert Systems: The Development and Implementation of Rule-based Expert Systems*. McGraw-Hill Companies, 1991.

[48] IHS Engineering. Ec proposes new legislation for european energy policy. Retrieved June 21, 2008 from http://engineers.ihs.com/news/eu-en-energy-policy-9-07.html, 2007.

[49] International Energy Agency. Key world energy statistics. Tech. rep., International Energy Agency, 2007. ©OECD/EIA, 2007. *Key World Energy statistics*.

[50] Isac, G., Bulavskii, V. A., and Kalashnikov, V. V. *Complementarity, Equilibrium, Efficiency and Economics*. Kluwer Academic Publishers, 2002.

[51] Juris, A. Market Development in the UK Natural Gas Industry. *World Bank Group, Washington DC* (1998).

[52] Juris, A. The Emergence of Markets in the Natural Gas Industry. *World* (1998).

[53] Kalashnikov, V. V., Kalashnikova, N. I., and Perez, G. A. Natural gas cash-out problem with price predictions. In *International Applied Business Research Conference* (Mazatlan, Mex., March 2007), R. P. Clute, Ed., The Clute Institute for Academic Research.

[54] Kalashnikov, V. V., Kalashnykova, N. I. ., Pérez, G. A., and Tomasgard, A. Natural gas imbalances stochastic bilevel optimization. *European Journal of Operational Research* (2009). Pending Review.

[55] Kalashnikov, V. V., Kalashnykova, N. I., and Pérez, G. A. New algorithms to solve the natural gas cash-out bilevel problem. *Annals of Operations Research* (2009). Pending Review.

[56] Kalashnikov, V. V., Matis, T. I., and Perez, G. A. Time series analysis applied to us natural gas consumption and price. *Energy Economics* (2009). Pending Review.

[57] Kalashnikov, V. V., and Rios-Mercado, R. Z. A penalty-function approach to a mixed-integer bilevel programming problem. In *Proceedings of the 3rd International Meeting on Computer Science, Aguascalientes, Mexico* (2001), vol. 2, pp. 1045–1054.

[58] Kalashnikov, V. V., and Ríos-Mercado, R. Z. An algorithm to solve a gas cash-out problem. In *Proceedings of the International Business and Economics Research Conference (IBERC'2002), Puerto Vallarta, Mexico* (2002).

[59] Kalashnikov, V. V., and Ríos-Mercado, R. Z. A natural gas cash-out problem: A bilevel programming framework and a penalty function method. *Optimization and Engineering 7*, 4 (2006), 403–420.

[60] Kall, P., Wallace, S., and Kall, P. *Stochastic programming*. Citeseer, 1994.

[61] Keyaerts, N., Meeus, L., and D'haeseleer, W. Analysis of balancing-system design and contracting behaviour in the natural gas .... *status: published*.

[62] Kohl, K. Natural gas supply. 3 ways to play the natural gas rebound. *Energy and Capital* (Aug 2009).

[63] Krichene, N. World crude oil and natural gas: a demand and supply model. *Energy Economics 24*, 6 (2002), 557–576.

[64] Laarhoven, P. J. M. v., and Aarts, E. *Simulated Annealing: Theory and Applications*. D Reidel Pub Co, 1987.

[65] Leach, J. *A Course in Public Economics*. Cambridge University Press, 2004.

[66] Lin, W., Chen, Y., and Chatov, R. The demand for natural gas, electricity and heating oil in the united states. *Resources and energy* (Jan 1987).

[67] Liu, B. Natural gas price elasticities: Variations by region and by sector in the usa. *Energy economics* (1983).

[68] Lyness, F. Gas demand forecasting. *Journal of the Royal Statistical Society. Series D (The Statistician) 33*, 1 (Mar 1984), 9–21.

[69] MacAvoy, P., and Moshkin, N. The new trend in the long-term price of natural gas. *Resource and energy economics* (Jan 2000).

[70] Maggioni, F., Allevi, E., Bertocchi, M. I., Innorta, M., and Vespucci, M. T. A gas retail stochastic optimization model by mean reverting temperature scenarios. In *Communications to SIMAI Congress, ISSN* (2007), vol. 9015.

[71] Marcotte, P., and Zhu, D. L. Exact and inexact penalty methods for the generalized bilevel programming problem. *Mathematical Programming 74*, 2 (1996), 141–157.

[72] Mendenhall, W., Wackerly, D., and Scheaffer, R. *Mathematical statistics with applications*. Duxbury, 2002.

[73] Menon, E. S. *Gas Pipeline Hydraulics*. CRC Press, 2005.

[74] Mercado, R. Z. R., Kalashnikov, V. V., and de Cátedra, I. La optimización binivel: Estudio de un caso en la industria de gas natural. *Ingenierías 6*, 20 (2003), 47.

[75] Midthun, K. T. *Optimization models for liberalized natural gas markets.* PhD thesis, Norwegian University of Science and Technology, 2007.

[76] Monge, G. *Mémoire sur la théorie des déblais et des remblais.* De l'Imprimerie Royale, 1781.

[77] Morales, Y. V., and Ríos-Mercado, R. Z. Preprocesamiento Efectivo de un Problema de Minimizacion de Combustible en Sistemas de Transporte de Gas Natural. *Revista Ingenieria de Sistemas Volumen XIX* (2005).

[78] NaturalGas.org. Background. Retrieved July 4, 2008 from http://www.naturalgas.org/overview/background.asp, 2004.

[79] NaturalGas.org. History. Retrieved June 21, 2008 from http://www.naturalgas.org/overview/history.asp, 2004.

[80] NaturalGas.org. The market under regulation. Retrieved June 21, 2008 from http://www.naturalgas.org/regulation/market.asp, 2004.

[81] NaturalGas.org. Natural gas - from wellhead to burner tip. Retrieved July 11, 2008 from http://www.naturalgas.org/naturalgas/naturalgas.asp, 2004.

[82] Nelder, C. Natural gas price forecast. the future of natural gas: It's time to invest. *Energy and Capital* (Aug 2009).

[83] Nelder, J. A., and Mead, R. A Simplex Method for Function Minimization. *The Computer Journal 7*, 4 (1965), 308.

[84] Neumann, A., and von Hirschhausen, C. Long-term contracts for natural gas supply–an empirical analysis. *9th ISNIE Conference September 22nd-24th* (Jan 2005).

[85] Neumann, A., and von Hirschhausen, C. ... revisited–an empirical analysis of producer-importer relations in the natural gas .... *Electricity Policy Research Group 06/10 and MIT ...* (Jan 2006).

[86] Oildom PC. Interruptible gas sales plunge under ferc order 636. *Pipeline & Gas Journal* (December 2004).

[87] Ríos-Mercado, R. Z., Kim, S., and Boyd, E. A. Efficient operation of natural gas transmission systems: A network-based heuristic for cyclic structures. *Computers and Operations Research 33*, 8 (2006), 2323–2351.

[88] Ríos-Mercado, R. Z., Secomandi, N., and Buraparate, V. A mixed-integer bilevel programming model for the problem of minimizing cash-out penalty costs of a natural gas shipping company. Tech. rep., Working paper, Postgraduate Program in System Engineering, UANL, San Nicolas de los Garza, Mexico, 1999.

[89] Smeers, Y. Gas Models and three difficult objectives.

[90] Soto, A. FERC order 636 & 637. Retrieved June 21, 2008 from http://www.aga.org/Legislative/issuesummaries/FERCOrder636637.html, 2008.

[91] StatoilHydro. Retrieved November 17 from http://www.statoilhydro.com/en/Pages/default.aspx.

[92] Stoner, T. W. FERC 636 is changing tax planning methods in the natural gas industry. *Pipeline & Gas Journal* (April 1995). Retrieved July 2008 from http://www.allbusiness.com/personal-finance/individual-taxes/501187-1.html.

[93] Talluri, K. T., and Van Ryzin, G. J. *The theory and practice of revenue management*. Springer, 2004.

[94] The Mathworks. Optimization toolbox: Nonlinear optimization. Retrieved December 1st, 2008 from http://www.mathworks.com/access/helpdesk/help/toolbox/optim/, 2008. ©1984-2008 The Matworks, Inc.

[95] The Mathworks. Statistics toolbox: Linkage. Retrieved December 3rd, 2008 from http://www.mathworks.com/access/helpdesk/help/toolbox/stats/, 2008. ©1984-2008 The Matworks, Inc.

[96] Tobin, J. Natural Gas Transportation–Infrastructure Issues and Operational Trends. *Energy Information Administration, Washington DC (www. eia. doe. gov)* (2001).

[97] Tobin, J., Shambaigh, P., and Mastrangelo, E. Natural gas processing: The crucial link between natural gas production and its transportation to market. Tech. rep., Energy Information Administration, 2006.

[98] Tomasgard, A., Rømo, F., Fodstad, M., and Midthun, K. Optimization models for the natural gas value chain. *Geometric Modelling, Numerical Simulation, and Optimization: Applied Mathematics at SINTEF',*

Springer, chapter Optimization Models for the Natural Gas Value Chain (2007).

[99] United Nations Conference on Trade Development. Characteristics. Retrieved July 5 from http://www.unctad.org/infocomm/anglais/gas/characteristics.htm#origin.

[100] United States Court of Appeals, District of Columbia Circuit. Ugi utilities inc. vs. FERC. Tech. rep., United States Court of Appeals, District of Columbia Circuit, Argued Jan. 26, 1998. Decided June 5, 1998.

[101] United States National Transportation Safety Board. *Pipeline Accident Report: Rupture of Piney Point Oil Pipeline and Release of Fuel Oil Near Chalk Point, Maryland, April 7, 2000*. DIANE Publishing, 2002.

[102] Uraikul, V., Chan, C. W., and Tontiwachwuthikul, P. Development of an expert system for optimizing natural gas pipeline operations. *Expert Systems With Applications 18*, 4 (2000), 271–282.

[103] Viet, N., Mandziuk, J., Politechniki, P., and Warsaw, P. Neural and Fuzzy Neural Networks in Prediction of Natural Gas Consumption. *NEURAL PARALLEL AND SCIENTIFIC COMPUTATIIONS 13*, 3/4 (2005), 265.

[104] Villalobos-Morales, Y., Cobos-Zaleta, D., Flores-Villarreal, H. J., Borraz-Sanchez, C., and Ríos-Mercado, R. Z. On NLP and MINLP formulations and preprocessing for fuel cost minimization of natural gas transmission networks. In *Proceedings of the 2003 NSF Design, Service and Manufacturing Grantees and Research Conference*.

[105] Vondráček, J., Pelikán, E., and Konár, O. A statistical model for the estimation of natural gas consumption. *Applied Energy* (Jan 2007).

[106] Wellisz, S. H. Regulation of Natural Gas Pipeline Companies: An Economic Analysis. *The Journal of Political Economy 71*, 1 (1963), 30.

[107] Winston, W. L. *Operations Research: Applications and Algorithms*. Duxbury Press, 1994.

[108] Wong, P., and Larson, R. Optimization of natural-gas pipeline systems via dynamic programming. *Automatic Control, IEEE Transactions on 13*, 5 (1968), 475–481.

[109] Wu, S., Ríos-Mercado, R. Z., Boyd, E., and Scott, L. R. Model Relaxations for the Fuel Cost Minimization of Steady-State Gas Pipeline Networks. *MATHEMATICAL AND COMPUTER MODELLING 31* (2000), 197–220.

[110] Yoo, S.-H., Lim, H.-J., and Kwak, S.-J. Estimating the residential demand function for natural gas in seoul with correction for .... *Applied Energy* (Jan 2009).

APPENDIX A
MATLAB CODE

This appendix presents the most important pieces of Matlab code, including scripts and functions, that were developed during the course of this research.

Because of the large amount of derivative portions of code, we have selected only the most important functions to be shown here. Small functions, self-indicative functions, as well as processes not innovated in this work (but independently implemented by us, e.g. the forecasting algorithms) are not shown for the sake of brevity.

A.1 Deterministic Optimization

This section reports the linear and nonlinear function coded to obtain the solutions to each level. The bilevel solution for each problem is easily obtained by adding the lower level minimization function to function argument in function `fmincon`, which would replace `linprog` in the linear functions.

A.1.1 Problem Generation

This code shows the Matlab script that generates a text output suitable to be saved as a *.m* file and be run later as a problem instance.

Most optimization parameters are contained in the `O` data structure, whereas those parameters corresponding to the prices and demands contained in the `F` data structure.

When expanding this algorithm to the stochastic case, all new information is stored in the structure `F`, whose members appear considerably different than those shown here.

The problem instances for the individual levels are obtained by only asking for the parameters involved in those levels.

```
%This script will generate a problem of the type AXXX.
display(blanks(4)');
display('global cascade staircase');
display('warning off all');
display('clear O')
display(blanks(2)');
```

```
days = 2;       %enter the number of days
display(['O.N = ',num2str(days,'%2.0f'),';','     %number of total days in the
    process']);
states = 8;       %enter the number of pool zones
    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
display(['O.P = ',num2str(states,'%2.0f'),';','     %number of states to be used']);
display(blanks(2)');
%enter the minimum and maximum reference parameters for initial imbalances
    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
mrpii = -100;
Mrpii =120;
%this will calculate the randomly generated initial vector
initial = mrpii + rand(1,states).*(Mrpii-mrpii);
%dispay the initial imbalance
display('O.XO = ...');
printmat(initial',10,'%10.0f');  %printmat is simply a function that formats a matrix
     so that it can be understood in an .m file
%enter the minimum and maximum reference paramter for the last day feasible
    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
%imbalances
mrpfi = -50;
Mrpfi = 150;
%this will calculate the randomly generated final vector
final = mrpfi + rand(1,states).*(Mrpfi-mrpfi);
step = 2*(final-initial)/days;
%feasible vector for the deterministic case
xfeas = repmat(1,days,states);
xfeas(1,:)= initial + rand(1,states).*step;
for i=2:days
    xfeas(i,:) = xfeas(i-1,:)+rand(1,states).*step;
end
display(blanks(2)');
%set the parameters for thightness of the daily upper/lower bounds
    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
dbparam1=20;
dbparam2=15;
dub = xfeas + dbparam1 + dbparam2*(rand(size(xfeas)));
dlb = xfeas - dbparam1 - dbparam2*(rand(size(xfeas)));
display('O.UB =...');
printmat(dub,3,'%10.0f');
display(blanks(1)');
display('O.LB =...');
printmat(dlb,3,'%10.0f');
%calculate the total sum for the feasible vector
xfeasum = sum(xfeas')';
%set the parameters for the tightness of the total daily upper/lower bounds
    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
tbparam1=50;
tbparam2=74;
```

```
tub = xfeasum + tbparam1 + tbparam2*(rand(size(xfeasum)));
tlb = xfeasum - tbparam1 - tbparam2*(rand(size(xfeasum)));
display(blanks(2)');
display('O.TUB =...');
printmat(tub,15,'%10.0f');
display(blanks(1)');
display('O.TLB =...');
printmat(tlb,15,'%10.0f');
%set the tolerance parameter for the swing bounds, lowerswing must be negative
    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
upperswing=10;
lowerswing = -30;
sfeas = repmat(1,days,states);
sfeas(1,:)= xfeas(1,:)-initial;
for i=2:days
    sfeas(i,:) = xfeas(i,:)-xfeas(i-1,:);
end
usb = repmat(max(max(sfeas)),days,states)+upperswing;
lsb = repmat(min(min(sfeas)),days,states)+lowerswing;
display(blanks(2)');
display('O.LSB =...');
printmat(lsb,3,'%10.0f');
display(blanks(1)');
display('O.USB =...');
printmat(usb,3,'%10.0f');
display(blanks(1)');
display('O.M = 1000000;');
display('O.try = ...');
printmat(xfeas,4,'%10.2f');
display('clear i F');
display(blanks(2)');
%This will select the random states for the problem
selected = selectrand((1:52), states);
%Enter the maximum amount of historic data
    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
lasthist= [180];
display(['F.SD=',num2str(lasthist,'%10.0f'),';']);
display(['states = [ ',num2str(selected,'%10.0f'),'];']);
display(blanks(2)');
Matrix;
display(blanks(1)');
display('F.DC = ...');
printmat(ResD(lasthist+1:lasthist + days, selected) , states, '%15.2f');
display('F.PC = ...');
printmat(Res(lasthist+1:lasthist + days, selected), states, '%15.2f');
display('F.CI = ...');
pencosts = Res(lasthist+1:lasthist + days, selected) .* (3*rand(days, states));
printmat(pencosts, states , '%10.2f');
display('F.CR= ...');
```

```
printmat(  Res(lasthist+1:lasthist + days, selected) .* (rand(days, states)), states
    , '%15.2f');
display(['F.PE = ...'  ]);
printmat(  (0.6+0.1*rand(days,states)).*(Res(lasthist+1:lasthist + days, selected)+
    pencosts ) , states , '%15.2f');
display('F.DM = ...');
printmat( (1.3+0.5*rand(days,states)).*ResD(lasthist+1:lasthist + days, selected)
    ,states , '%15.2f' )
display(blanks(1)');
%lower level matrices
display('O.E = ...');
printmat(leaks(states),6,'%10.4f');  %function leaks forms a strictly upper
    triangular matrix based in the inverse poisson distribution
display('O.E = sparse(O.E);');
display('O.F = ...');
printmat(triu(poissinv(rand(states),repmat(    12   ,states,states))+1,1),5,'%10.4f')
    ;
display('O.F = sparse(O.F);');
display('O.B = ...');
printmat(triu(poissinv(rand(states),repmat(    10   ,states,states))+1,1),5,'%10.4f')
    ;
display('O.B = sparse(O.B);');
display(blanks(2)');
%Some calculations are more straightforward with F, B, and E being lines rather than
    matrices:
display('O.Fl = matrix_to_line_(O.F,O.P);
display('O.Bl = matrix_to_line_(O.B,O.P);');
display('O.El = matrix_to_line_(O.E, O.P);');
display(blanks(2)');
%enter the linear cost coefficient parameters for the lower level
    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
rsmin=15;
rsmax = 25;
rs = rsmin + rand(1,states).*(rsmax-rsmin);
display(['O.R =[',  num2str(rs,'%12.0f'),'];   %The linear cost coeffcient  for gas
    sold/bought);']);
display(blanks(1)');
display('O.A = 0.5;');
display('O.M = 1000000;');
display(blanks(2)');
 %Cascade is a 1's and 0's matrix in the shape of many series of stairs cascading
     each time from a lower height.');
display('cascade=sparse([eye(O.P-1);zeros(1,O.P-1)]);
display('staircase=sparse([0;1;zeros(O.P-2,1)]);
%Staricase is a 1's and 0's matrix in the shape of one staricase of steps each time
    shorter.');
display('for i=1:O.P-2');
display('    cascade=sparse([   [eye(O.P-1-i);zeros(1+i,O.P-1-i)] , cascade]);');
```

```
display('    staircase=sparse([staircase,[zeros(i+1);ones(1,i+1);zeros(O.P-2-i,i+1)
    ]]);');
display('end');
display(blanks(2)');
display('clear Res ResD states i');
display(blanks(5)');
clear all
```

A.1.2 Upper Level Nonlinear Minimization

Function `MD_NONLINEAR_UPPER` performs a nonlinear minimization of the upper level in model $PR3'$. It uses subfunctions `minimize_Z_` and `ulof` as arguments for the nonlinear minimizer `fmincon`.

```
 function [point, value, time] = MD_NONLINEAR_UPPER(O,F)
%function MD_NONLINEAR_UPPER(O,F) takes the optimization data structures 'O' and 'F'
    and solves the problem therein described using a direct nonlinear minimization
    method. It reports the optimal pints and values for both values of the binary
    desicion variable 'q'.
tic     %start the timer
[point,value]=minimize_Z_(O,F);          %Calling (sub)function minimize_Z_ to solve
    the problem in 'O, F.'
time = toc;              %Stops the timer
display('************************************************');
display('************************************************');
display('************************************************');
display('Time (s) = ');
display(num2str(time,'%11.2f'));
beep;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [point, value] = minimize_Z_(O,F)
%function 'minimiza_Z' takes the data in the structures 'O, F' and solves the problem
     by calling a straightforward nonlinear minimzation routine
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            %                          Constraints for Z
            %   these constraints are to be used in the minimization of variables x,
                z, A1, A2
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            %constraints xL<x<xU are included in the bounds
            %cns sL<s<sU:
            AQ=[];   %AQ represent the linear inequalities matrix for the upper level
                 function
            AQ=[diagonalize([  ones(1,O.P);-ones(1,O.P)], O.N), zeros(2*O.N,3*O.N*O.P
                )];       %equivale a XLt<=sum(xti)<= XUt,
            AeQ = [...
```

```
            zeros(O.P, 4*O.P*O.N );...  %the x_1 part of constraint x_1 = x_0 +
                s_1
            -eye(O.P*(O.N-1)), zeros(O.P*(O.N-1), 3*O.P*O.N + O.P)...  %the x_i
                part of x_i + s_i
            ];
        AeQ = AeQ + [speye(O.P*O.N),-speye(O.P*O.N), zeros(O.P*O.N, 2*O.P*O.N)];
              %the s_i part of x_i = x_i-1 + s_i
        AeQ = [AeQ;...
            eye(O.P*O.N), zeros(O.P*O.N), eye(O.P*O.N),-eye(O.P*O.N)...    %
                constraint x = EP - EA
            ];
        BQ=[];                        %BQ represents the linear inequality
            constrains right hand side for the upper level function
        for i=1:O.N
            BQ=[BQ,O.TUB(i),-O.TLB(i)];
        end
        BeQ = [...
            O.X0',...
            zeros(1,O.P*(O.N-1)),...
            zeros(1,O.P*O.N)];
        UQ=[reshape(O.UB',1,O.P*O.N),reshape(O.USB',1,O.P*O.N),reshape(F.DM',1,O.
            N*O.P), reshape(F.DM',1,O.N*O.P)  ]; %upper bounds of the decision
            variables
        LQ=[reshape(O.LB',1,O.P*O.N),reshape(O.LSB',1,O.P*O.N),zeros(1,2*O.N*O.P)
              ];   %lower bounds of the decision variables
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %                        End of  Constraints for Z
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        display('************************************************');
        display('************************************************');
        display('************************************************');
        display('Optimization Results: ');
        display('************************************************');
%fmincon is optimized over the named subfunction ulof(...)
[point,value] = fmincon(@(x)(ulof(x,O,F)),zeros(1,4*O.N*O.P), AQ,BQ , AeQ,BeQ, LQ, UQ
    ,[],optimset('TolCon',0.001, 'Display', 'off'));
if     all(AQ*point'<= BQ'+0.001)*all(AeQ*point'<= BeQ'+0.001) && all(AeQ*point'>=
    BeQ'-0.001)*all(point>=LQ-0.001)*all(point<=UQ+0.001)
    display(blanks(3)')
    display('X = ')
    display( num2str(reshape(point(1:O.P*O.N),O.P,O.N)','%11.2f'));
    display('E^A = ')
    display( num2str(reshape(point(2*O.P*O.N+1:3*O.P*O.N),O.P,O.N)','%11.2f'));
    display('E^P = ')
    display( num2str(reshape(point(3*O.P*O.N+1:4*O.P*O.N),O.P,O.N)','%11.2f'));
    display('Profit = ')
    display( num2str(-value,'%11.2f'))
else
    display(blanks(3)')
```

```
    display('Solution Reported not feasible!')
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function ret = ulof(X,O,F)
ea = X(2*O.P*O.N+1:3*O.P*O.N);
ep = X(3*O.P*O.N+1:4*O.P*O.N);
ret = -reshape(F.PC',1,O.P*O.N)*min(ea, reshape(F.DC',1,O.P*O.N))'...
    -reshape(F.PE',1,O.P*O.N)*max(0,ea - reshape(F.DC',1,O.P*O.N))'...
    +reshape(F.CI',1,O.P*O.N)*max(0,reshape(F.DC',1,O.P*O.N) - ea)'...
    +reshape(F.CR',1,O.P*O.N)*ep';
```

A.1.3 Upper Level Linear Optimization

The upper level optimization can also be achieved linearly using function `MD_LINEAR_UPPER`, which uses its own version of the subfunction `minimize_Z_`.

```
function [point, value, time] = MD_LINEAR_UPPER(O,F)
%function MD_LINEAR_UPPER(O,F) takes the optimization data structures 'O, F' and
    solves the problem therein described b using a direct linear minimization method.
     It reports the optimal pints and values for both values of the binary desicion
    variable 'q'.
tic     %start the timer
[point,value]=minimize_Z_(O,F);         %Calling function minimize_Z_ to solve the
    problem in 'O' with the binary var. q=1.
time = toc;             %Stops the timer
display('************************************************');
display('************************************************');
display('************************************************');
display('Time (s) = ');
display(num2str(time,'%11.2f'));
beep;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [point, value] = minimize_Z_(O,F)
%function 'minimiza_Z' takes the data in the structure 'O' and the value of the
    binary desicion variable 'q' and solves the problem by calling a stright-forward
    minimzation routine
global penalizer record
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            %                Constraints for Z
            %   these constraints are to be used in the minimization of variables x,
                z, A1, A2 %
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            %constraints xL<x<xU are included in the bounds
            %cns sL<s<sU:
```

```
            AQ=[];   %AQ represent the linear inequalities matrix for the upper level
                 function
            for i=1:O.N-1
                new=[];
                new=sparse([zeros(O.P,O.P*(i-1)),-eye(O.P),eye(O.P),zeros(O.P,O.P*(O.
                    N-i-1)),zeros(O.P,2*O.N*O.P)]);  %Equivale a xti-1 + xti <= sti
                AQ=[AQ;new;-new];
            end
            AQ=[AQ;[diagonalize([  ones(1,O.P);-ones(1,O.P)], O.N), zeros(2*O.N,2*O.N
                *O.P)];...       %equivale a XLt<=sum(xti)<= XUt,
                zeros(O.N*O.P,O.N*O.P),-eye(O.N*O.P),eye(O.N*O.P) ;...      %equivale
                    a -A1+ A2<=0
                eye(O.N*O.P), eye(O.N*O.P), zeros(O.N*O.P,O.N*O.P);...
                                                %equivale a  x_ti + A1_ti <= 0
                -eye(O.N*O.P), -eye(O.N*O.P), zeros(O.N*O.P,O.N*O.P)];
                                              %equivale a -x_ti - A1_ti <=0
                BQ=[];                     %BQ represents the linear inequality
                    constrains right hand side for the upper level function
                for i=1:O.N-1
                    BQ=[BQ,O.USB(i+1,:),-O.LSB(i+1,:)];
                end
                for i=1:O.N
                    BQ=[BQ,O.TUB(i),-O.TLB(i)];
                end
                BQ=[BQ,zeros(1,O.N*O.P), reshape(F.DM', 1, O.N*O.P), repmat(0,1,O.N*O
                    .P)];
                UQ=[min(O.USB(1,:)+O.XO',O.UB(1,:))];           %UQ contains the
                    Upper bounds  for the upper level function
                for(i=2:O.N)
                    UQ=[UQ,O.UB(i,:)];
                end
                UQ=[UQ,reshape(F.DM',1,O.N*O.P), reshape(F.DC',1,O.N*O.P)  ];
                LQ=[];          %LQ contains the Lower bounds  for function 'Q=-F+M*G
                    '.
                LQ=[max(O.LSB(1,:)+O.XO',O.LB(1,:))];
                for(i=2:O.N)
                    LQ=[LQ, O.LB(i,:)];
                end
                LQ=[LQ, zeros(1,2*O.N*O.P)];
                Of = [reshape(F.CR',1,O.N*O.P) , -reshape(F.PE',1,O.N*O.P) + reshape(
                    F.CR',1,O.N*O.P),reshape(F.PE' - F.PC' - F.CI',1,O.N*O.P)];
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            %             End of  Constraints for Z
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
display('************************************************');
display('************************************************');
display('************************************************');
display('Upper level Optimization Results: ');
display('************************************************');
```

```
%We minimize function the problem using lingprog(...):
[point,value] = linprog(Of, AQ, BQ, [],[], LQ, UQ,[],optimset('TolCon',1));
if all(AQ*point<= BQ'+0.001)*all(point'>=LQ-0.001)*all(point'<=UQ+0.001)
    display('X = ')
    display( num2str(reshape(point(1:O.P*O.N),O.P,O.N)','%11.2f'));
    display('E^A = ')
    display( num2str(reshape(point(O.P*O.N+1:2*O.P*O.N),O.P,O.N)','%11.2f'));
    display('E^P = ')
    display( num2str(reshape(point(O.P*O.N+1:2*O.P*O.N),O.P,O.N)'+reshape(point(1:O.P
        *O.N),O.P,O.N)','%11.2f'));
    display('A^2 = ')
    display( num2str(reshape(point(2*O.P*O.N+1:3*O.P*O.N),O.P,O.N)','%11.2f'));
    display('Profit = ')
    display( num2str(-value -reshape(F.CI'.*F.DC',1,O.N*O.P)*ones(1,O.N*O.P)' ,'%11.2
        f'))
else
    display(blanks(3)')
    display('Solution Reported not feasible!')
end
```

A.1.4 Lower Level Nonlinear Minimization

The lower level minimization, carried on using function `fmincon`, is shown next, in the body of function `experimenting_minimize_D_nonlin( O, q)`, using the subfunctions `objective` and `nonlinear`, which contain the objective function and the nonlinear constraints, respectively.

```
function value = experimenting_minimize_D_nonlin( O, q)
%Function experimenting_minimize_D_nonlin(O,q) minimzes the lower level in the data
    structure 'O', for the specified value of variable 'q' using the nonlinear
    optimizer 'fmincon', and the subfunctions 'objective()' and 'nonlinear()' to
    represent the lower level nonlinear obejective function and constraints.
tic;
global cascade staircase
xN=O.xN';
%check first if the problem is already solved
if all(( 2*q-1  )*xN>=-0.001)
    display(blanks(5)');
    display('Easy!!!!!!!!!!');
    value=-O.R*xN';
    experimental_buildyuvzd(O.P, [xN, zeros(1,O.P^2-O.P)]',value); %this is simply a
        formatted report of the variables
    display('Time (s) = ');
    display(num2str(toc,'%11.5f'));
    beep;
    return
end
```

```
%check too if the problem is unfeasible
if all(( 2*q-1  )*xN<=0.001)
    display(blanks(5)');
    display('Unfeasible!!!!!!!!!');
    display('Time (s) = ');
    display(num2str(toc,'%11.5f'));
    beep;
    return
end
%we thus begin forming the restriction matrices for the lower level problem
%the string of variables passed to the optimizer will be formed by
%y,u,v,z
%cns_y=xuv
% [eye(O.P),cascade-(staircase.*repmat(1-O.El,O.P,1)),-cascade+staircase,repmat(0,O.P
    ,1)];
%cns_uv<x
%[zeros(O.P), cascade, staircase,zeros(O.P,1)]
%cns_u<x, v<x wil be put in the nonlinear terms
%constraints x<y<x will be out in the bounds
%constraints -M(1-q)<y<Mq  will be put in the bounds.
%cns_z=yuv
%[O.R, -O.Fl .* (1-O.El), O.Bl, 1]
%the inequalities matrix  and rhs will then be fomed by:
AQ=[...
    [zeros(O.P), cascade, staircase,zeros(O.P,1)]
    ];
%whereas its right-hand side becomes:%cns z<d
BQ=[...
    max(0,xN)
    ];
%The equalities matrix wil be formed by:
AeQ=[...
    [eye(O.P),cascade-(staircase.*repmat(1-O.El,O.P,1)),-cascade+staircase,repmat(0,O
        .P,1)];
    [O.R, -O.Fl .* (1-O.El), O.Bl, 1]
    ];
%whereas its right-hand side becomes:
BeQ=[xN,0, ...
    ];
%The lower bounds are:
LB=[...
    max(-O.M*repmat((1-q),1,O.P),min(0,xN)),...  %y>-M*(1-q)
    repmat(0,1,O.P^2-O.P),...   %u>0,v>0
    -inf
    ];
%And the upper bounds are:
UB = [...
    min(O.M*repmat(q,1,O.P),max(0,xN)),...   %y<Mq
    repmat(inf, 1,O.P^2-O.P),... %u,v \in R
```

```
    repmat(inf, 1, 1)     %z,d \in R
    ];
%so, the minimization function would be:
[ point,value] = fmincon(@(y)objective(y), zeros(1,O.P^2+1), AQ,BQ, AeQ, BeQ, LB, UB,
     @(y)nonlinear(y,xN,O.P));
experimental_buildyuvzd(O.P, point', point(end))  %again, just a formatted report of
    the results
display('Time (s) = ');
display(num2str(toc,'%11.5f'));
beep;
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
function ret = objective(x)
ret = x(end)^2;
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [ineq, eq] = nonlinear(y,x, P)
k=1;
vector = zeros(1,P^2-P);
for i=2:P
    for j=1:i-1
        if x(j) >= 0 && x(i) <= 0
            vector(k) = y(P+k) - x(j);
        else
            vector(k)  = y(P+k);
        end
        k=k+1;
    end
end
for i=2:P
    for j=1:i-1
        if x(i) >= 0 && x(j) <= 0
            vector(k) = y(P+k) - x(i);
        else
            vector(k)  = y(P+k);
        end
        k=k+1;
    end
end
ineq = vector;
eq = [];
```

A.1.5 Lower Level Linear Minimization

The lower level minimization, performed linearly though the usage of Matlab function `linprog` is shown in function `experimenting_minimize_D_ll( O, q)` next:

```
function [poitn, value] = experimenting_minimize_D_ll( O, q)
%Function experimenting_minimize_D_ll(O,q) minimzes the lower level in the data
    structure 'O', for the specified value of variable 'q' using the linear optimizer
     'linprog'
tic;
global cascade staircase
xN=O.xN';
%check if the problem is already solved
if all(( 2*q-1  )*xN>=-0.001)
    display(blanks(5)');
    display('Easy!!!!!!!!!!');
     value=-O.R*xN';
    experimental_buildyuvzd(O.P, [xN, zeros(1,O.P^2-O.P)]',value);   %this is simply
        a formatted report of the results
    display('Time (s) = ');
    display(num2str(toc,'%11.5f'));
    beep;
    return
end
%check too if the problem is unfeasible
if all(( 2*q-1  )*xN<=0.001)
    display(blanks(5)');
    display('Unfeasible!!!!!!!!!');
    display('Time (s) = ');
    display(num2str(toc,'%11.5f'));
    beep;
    return
end
%we now form the restriction matrices for the lower level problem
%the string of variables passed to the optimizer will be formed by
%y,u,v,z,d,xi,zeta.
%cns_y=xuv
% [eye(O.P),cascade-(staircase.*repmat(1-O.El,O.P,1)),-cascade+staircase,repmat(0,O.P
    ,1),repmat(0,O.P,1),repmat(0,O.P,2*O.P)];
%cns_u<xi
%[repmat(0,(O.P^2-O.P)/2,O.P),eye((O.P^2-O.P)/2),repmat(0,(O.P^2-O.P)/2),repmat(0,(O.
    P^2-O.P)/2,1),repmat(0,(O.P^2-O.P)/2,1),-cascade',repmat(0,(O.P^2-O.P)/2,O.P)];
%cns_u<zeta
%[repmat(0,(O.P^2-O.P)/2,O.P),eye((O.P^2-O.P)/2),repmat(0,(O.P^2-O.P)/2),repmat(0,(O.
    P^2-O.P)/2,1),repmat(0,(O.P^2-O.P)/2,1),repmat(0,(O.P^2-O.P)/2,O.P),-staircase'];
%cns_v<xi
%[repmat(0,(O.P^2-O.P)/2,O.P),repmat(0,(O.P^2-O.P)/2),eye((O.P^2-O.P)/2),repmat(0,(O.
    P^2-O.P)/2,1),repmat(0,(O.P^2-O.P)/2,1),-staircase',repmat(0,(O.P^2-O.P)/2,O.P)];
%cns_v<zeta
%[repmat(0,(O.P^2-O.P)/2,O.P),repmat(0,(O.P^2-O.P)/2),eye((O.P^2-O.P)/2),repmat(0,(O.
    P^2-O.P)/2,1),repmat(0,(O.P^2-O.P)/2,1),repmat(0,(O.P^2-O.P)/2,O.P),-cascade'];
%cns_y<xi := y-xi<0
%[eye(O.P),repmat(0,O.P,(O.P^2-O.P)/2),repmat(0,O.P,(O.P^2-O.P)/2),repmat(0,O.P,1),
    repmat(0,O.P,1),-eye(O.P),repmat(0,O.P)];
```

```
%cns_zeta<y := -y-zeta<0
%[-eye(O.P),repmat(0,O.P,(O.P^2-O.P)/2),repmat(0,O.P,(O.P^2-O.P)/2),repmat(0,O.P,1),
    repmat(0,O.P,1),repmat(0,O.P),-eye(O.P)];
%constraints -M(1-q)<y<Mq  will be put in the bounds.
%cns_z=yuv
%[O.R, -O.Fl .* (1-O.El), O.Bl, 1,0,repmat(0,1,2*O.P)]
%cns -d<z := -z-d<0
%[repmat(0,1,O.P^2), -1,-1,repmat(0,1,2*O.P)]
%cns z<d
%[repmat(0,1,O.P^2), 1,-1,repmat(0,1,2*O.P)]
%constraints xi>XN, xi>0, zeta>-xN, zeta>0, u>0, v>0 will be put in the bounds
%the inequalities matrix  and rhs will then be fomed by:
AQ=[...
    [repmat(0,(O.P^2-O.P)/2,O.P),eye((O.P^2-O.P)/2),repmat(0,(O.P^2-O.P)/2),repmat
        (0,(O.P^2-O.P)/2,1),repmat(0,(O.P^2-O.P)/2,1),-cascade',repmat(0,(O.P^2-O.P)
        /2,O.P)];...%cns_u<xi
    [repmat(0,(O.P^2-O.P)/2,O.P),eye((O.P^2-O.P)/2).*diag(1-O.El),repmat(0,(O.P^2-O.P
        )/2),repmat(0,(O.P^2-O.P)/2,1),repmat(0,(O.P^2-O.P)/2,1),repmat(0,(O.P^2-O.P)
        /2,O.P),-staircase'];  ...  %cns_u<zeta
    [repmat(0,(O.P^2-O.P)/2,O.P),repmat(0,(O.P^2-O.P)/2),eye((O.P^2-O.P)/2),repmat
        (0,(O.P^2-O.P)/2,1),repmat(0,(O.P^2-O.P)/2,1),-staircase',repmat(0,(O.P^2-O.P
        )/2,O.P)];...%cns_v<xi
    [repmat(0,(O.P^2-O.P)/2,O.P),repmat(0,(O.P^2-O.P)/2),eye((O.P^2-O.P)/2),repmat
        (0,(O.P^2-O.P)/2,1),repmat(0,(O.P^2-O.P)/2,1),repmat(0,(O.P^2-O.P)/2,O.P),-
        cascade'];...%cns_v<zeta
    [eye(O.P),repmat(0,O.P,(O.P^2-O.P)/2),repmat(0,O.P,(O.P^2-O.P)/2),repmat(0,O.P,1)
        ,repmat(0,O.P,1),-eye(O.P),repmat(0,O.P)];...%cns_y<xi := y-xi<0
    [-eye(O.P),repmat(0,O.P,(O.P^2-O.P)/2),repmat(0,O.P,(O.P^2-O.P)/2),repmat(0,O.P
        ,1),repmat(0,O.P,1),repmat(0,O.P),-eye(O.P)];...%cns_zeta<y := -y-zeta<0
    [repmat(0,1,O.P^2), -1,-1,repmat(0,1,2*O.P)] ;...%cns -d<z := -z-d<0
    [repmat(0,1,O.P^2), 1,-1,repmat(0,1,2*O.P)]; ...%cns z<d
    [repmat(0,2*O.P,O.P^2+2), -eye(O.P*2)] %xi>=xN, zeta>=-xN
    ];
%whereas its right-hand side becomes:%cns z<d
BQ=[repmat(0,1,4*(O.P^2-O.P)/2+2*O.P+2 ), ...
      -xN,xN...
      ];
%The equalities matrix wil be formed by:
AeQ=[...
    [eye(O.P),cascade-(staircase.*repmat(1-O.El,O.P,1)),-cascade+staircase,repmat(0,O
        .P,1),repmat(0,O.P,1),repmat(0,O.P,2*O.P)];...  %cns_y=xuv
    [O.R, -O.Fl .* (1-O.El), O.Bl, 1,0,repmat(0,1,2*O.P)];...%cns_z=yuv
];
%whereas its right-hand side becomes:%cns z<d
BeQ=[xN,0, ...
];
%The lower bounds are:
LB=[...
    -O.M*repmat((1-q),1,O.P),...  %y>-M*(1-q)
```

```
    repmat(0,1,0.P^2-0.P),...    %u>0,v>0
    -inf, -inf,...    %z,d \in R
    repmat(0,1,2*0.P)...  % xi>0,zeta>0
     ];
%And the upper bounds are:
UB = [...
    0.M*repmat(q,1,0.P),...    %y<Mq
    repmat(inf, 1,0.P^2-0.P),... %u,v \in R
    repmat(inf, 1, 2),...       %z,d \in R
    repmat(inf, 1, 2*0.P)...    %xi, zeta \in R
    ];
%The objective function vector is:
Of= [repmat(0,1,0.P^2),0,1,10000000*repmat(1,1,2*0.P)];
%so, the minimization function would be:
[ point,value] = linprog(Of, AQ, BQ, AeQ, BeQ, LB, UB,UB);
%check if the artificial variables are indeed complementary
if (point(0.P^2+3:0.P^2+2+0.P)'*point(0.P^2+3+0.P:end) >= -0.001 )&& (point(0.P^2+3:0
    .P^2+2+0.P)'*point(0.P^2+3+0.P:end) <= 0.001 )
    experimental_buildyuvzdxzeta(0.P,point)
else
    display(blanks(2));
    display('Unable to obtain complementarity vectors xi, zeta!!!!!!!!');
end
display('Time (s) = ');
display(num2str(toc,'%11.5f'));
beep;
```

A.2 Stochastic Optimization

This section presents the most important functions concerning the stochastic optimization models solved. While the solver is very similar to that of the deterministic case, the functions and scripts that deal with the upper level stochastic parameters are considerably different.

A.2.1 Scenario Tree Generation

Function `scenariotreestoch(Mat, F, met)` calculates an array of structures that represent the prices or demands forecast in the scenario tree structure, as well as a vector of probabilities for each scenario faced. It will be used in the scripts defining the problem instances to produce the remaining parts of the structure `F` not created by the problem generator algorithm.

```
function [ret, probs1]= scenariotreestoch(Mat, F,met)
```

```
%Function scenariotreestoch(Mat,F,met) takes data matrix 'Mat', the optimization data
     structure 'F' and the method-specifier 'met', and returns an array of scenario
    tree forecasts structures 'ret',  and  a vector 'probs1'.
%check if the forecasting method is given in integer or character format
if ~ischar(met)
    if met == 1
        met = 'ARAR';
    else
        met = 'HW';
    end
end
%define the probability vector for each branch of the tree.
probs1=F.p;
for i=1 :F.stages-2
    probs=[];
    for j=1:length(probs1)
        probs=[probs,F.p*probs1(j)];
    end
    probs1=probs;
end
%the next nested 'for' loops will build an array of stuctures, one per state, whose
    memebers are forecasts matrices, performed using the first matrix, and whose
    columns corresponds to the data for each scenario of the scenario tree.
ret= [];
for istate  = 1: length(Mat(1,:))
    WS = Mat(1:F.SD,istate);
    for istage= 1:F.stages
        endpos = length(WS(1,:));
        WS1=[];
        for ipos = 1:endpos
            WS1=[WS1,point_adv(WS(:,ipos),F,istage,met)];
        end
        %the following if statement makes the tree single rooted; for
        %several routes, single out the ELSE argument
        if istage == 1
            WS = WS1(:,2);
        else
            WS=WS1;
        end
    end
    ret(istate).mat=WS(F.SD+1:end,:);
end
%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%
function ret1 = point_adv(S,  F, stage,met)   %this functions performs ternary
    frocecasts and the corresponding extension of the source matrix.
ret2= forecast_ARHW2(S, 1, F.SD+(stage-1)*F.stagelength,F.stagelength,12,met,F.band
    ,0);
```

```
ret1=[repmat(S,1,3);ret2];
```

A.2.2 Stochastic Optimization

Function `MD_Stoch(O,F,q)` takes the optimization structures `O,F` and solves the stochastic problem they define, obtaining the expected value solution of this problem. It calls a lower level solution function `minimize_D_prev` similar to that of the deterministic analogue, slightly modified to include the several leaves' final day imbalances and their probabilities.

```
function [point, value, time] = MD_Stoch(O,F,q)
%function MD_Stoch takes the optimization data structures 'O, F' and solves the
%problem therein described by using a direct minimization method in a
%multistage stochastic optimization problem with recourse variables.
%It reports the optimal pints and values for both values of the binary
%desicion variable 'q'.
global penalizer record
tic     %start the timer
switch q
    case  1
        [point,value]=minimize_Z_(1,O,F);               %Calling function minimize_Z_
             to solve the problem in 'O' with the binary var. q=1.
    case  0
        [point,value]=minimize_Z_(0,O,F);       %Calling function minimize_Z_ to
            solve the problem in 'O' with the binary variable q=0.
    case 2
         [point1,value1]=minimize_Z_(1,O,F);            %Calling function minimize_Z_
              to solve the problem in 'O' with the binary var. q=1.
         [point0,value0]=minimize_Z_(0,O,F);       %Calling function minimize_Z_ to
             solve the problem in 'O' with the binary variable q=0.
         if value1 <= value0
             point = point1;
             value = value1;
             q=1;
         else
             point = point0;
             value=value0;
             q=0;
         end
end
time = toc;             %Stops the timer
display('*****************************************************');
display('*****************************************************');
display('*****************************************************');
display('Penalization = ');
display(num2str(penalizer,'%11.2f'));
% display('Record = ');
```

```
% display(num2str(record,'%11.2f'));
display('Time (s) = ');
display(num2str(time,'%11.2f'));
beep;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [sol, final_value] = minimize_Z_(q,O,F)
%function 'minimiza_Z' takes the data in the structure 'O' and the value of
%the binary desicion variable 'q' and solves the problem by calling a
%straight-forward minimzation routine
global penalizer record
                        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            %                           Constraints for Z
            %   these constraints are to be used in the minimization of variables x,
                Ea, xp%
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            %constraints xL<x<xU are included in the bounds
            %In-node imbalance swing bounds inequality matrix M2
            M2=[];
            for k=1:O.K
                M1=[];
                for n=2:O.Np
                    M1=[M1;[repmat(0,O.P,(n-2)*O.P),[-eye(O.P),eye(O.P)],repmat(0,O.P
                        ,(O.Np-n)*O.P)]];
                end
                M2=blkdiag(M2,M1);
            end
            %M2=[M2, repmat(0,length(M2(:,1)),2*O.Np*O.K*O.P)];
            %cross-node imbalance swing bounds inequality matrix
            if O.St>1
                M3=[];
                for k=1:O.K-1
                    M3=blkdiag(M3,[ eye(O.P),repmat(0,O.P,(O.Np-1)*O.P)]);
                end
                M3 = [repmat(0,length(M3(:,1)),O.Np*O.P),M3];
                M4 = [];
                for k=1:O.Ka
                    M4=blkdiag(M4, [repmat(0,3*O.P,(O.Np-1)*O.P),[-eye(O.P);-eye(O.P)
                        ;-eye(O.P)] ]);
                end
                M4=sparse([[M4,repmat(0,3*O.Ka*O.P,(O.K-O.Ka)*O.Np*O.P)]]);
                M5=M3+M4;
            else
                M5=[];
            end
            %total imbalance limits
```

```
M6=sparse([]);
for k=(1:O.K*O.Np)
    M6=blkdiag(M6,sparse(repmat(1,1,O.P)));
end
%initial imbalance swing bounds
M7=[[eye(O.P);-eye(O.P)],zeros(2*O.P,(O.K-1)*O.Np*O.P+(O.Np-1)*O.P)];
%recourse variable to extraction constraints
M8 = [ zeros(O.P*O.Np*O.K),-speye(O.P*O.Np*O.K),speye(O.P*O.Np*O.K) ];
%equal stage booking equalities
M9 = sparse([]);
for i=1:O.St
   M9 = blkdiag(M9, [speye(O.Np*O.P*(3^(i-1)-1)),zeros(O.Np*O.P*(3^(i-1)
       -1),O.Np*O.P) ] + [zeros(O.Np*O.P*(3^(i-1)-1),O.Np*O.P),-speye(O.
       Np*O.P*(3^(i-1)-1)) ] );
end
M10 = [M9,M9,zeros(size(M9))];
AQ = [[[M7;M5;-M5;M2;-M2;M6;-M6],zeros(2*O.Np*O.P*O.K+2*O.Np*O.K,2*O.Np*O
    .P*O.K)];M8];
AeQ = [M10];
clear M1 M2 M3 M4 M5 M6 M7 M8 M10
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Right-hand sides
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%in-node imbalance swing inequiality right-hand-side
N1=[];
N2=[];
for k=1:O.St
    %A= reshape(WU( (k-1)*O.Np+2:k*O.Np,: )',O.P*(O.Np-1),1    )' ;
    N1=[N1,repmat(reshape(O.UWB( (k-1)*O.Np+2:k*O.Np,: )',O.P*(O.Np-1),1
           )' ,1,3^(k-1) )] ;
    N2=[N2,repmat(reshape(O.LWB( (k-1)*O.Np+2:k*O.Np,: )',O.P*(O.Np-1),1
           )' ,1,3^(k-1) )] ;
end
N3=[N1,-N2];
%cross-node imbalance swing inequaity right-hand side
if O.St>1
    N4=[];
    N5=[];
    for k=2:O.St
        %A= reshape(WU( (k-1)*O.Np+2:k*O.Np,: )',O.P*(O.Np-1),1    )' ;
        N4=[N4,repmat(reshape(O.UWB( (k-1)*O.Np+1,: )',O.P,1    )' ,1,3^(
            k-1) )] ;
        N5=[N5,repmat(reshape(O.LWB( (k-1)*O.Np+1,: )',O.P,1    )' ,1,3^(
            k-1) )] ;
    end
    N6=[N4,-N5];
else
```

```
    N6=[];
end
%total imbalance right hand side
N7=[];
N8=[];
for k=1:O.St
    %A = WTU((k-1)*O.Np+1:k*O.Np);
    N7=[N7,repmat( O.TUB((k-1)*O.Np+1:k*O.Np)',1, 3^(k-1) )  ];
    N8=[N8,repmat( O.TLB((k-1)*O.Np+1:k*O.Np)',1, 3^(k-1) )  ];
end
N9=[N7,-N8];
%initial imbalance swing right hand side
N10=[O.UWB(1,:)+O.X0',-O.LWB(1,:)-O.X0'];
%Recourse to extraction right hand side
%hyerarchy of the indexes is P,Np,K
N11=[];
for i=1:O.St %counter for stages
    for j = 1:3^(i-1)   %counter for nodes in the stage
        for k=1:O.Np  %counter for days in the node/stage
            for l=1:O.P   %counter for zones
                N11 = [N11,F.DCT(l).mat(O.Np*(i-1)+k,1+ (j-1)*3^(O.St-i)
                    )];
            end
        end
    end
end
BQ=[N10,N6,N3,N9, zeros(1,O.Np*O.P*O.K)];
BeQ =  zeros(1,length(M9(:,1)));
clear N1 N2 N3 N4 N5 N6 N7 N8 N9 N10 M9
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Bounds
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Imbalance upper and lower bounds
U1=turnNP2KNpP(O.UB,O);
L1=turnNP2KNpP(O.LB,O);
%Extraction upper and lower bounds
U2=inf*ones(1,O.K*O.Np*O.P);
L2=zeros(1,O.K*O.Np*O.P);
%recourse upper and lower bounds
U3 = max(0.001,N11);
L3 = zeros(1,O.K*O.Np*O.P);
UQ = [U1,U2,U3];
LQ = [L1,L2,L3];
clear U1 U2 U3 L1 L2 L3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Objective function linear part
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%sale prices
Of1 = [];
for i=1:O.St %counter for stages
    for j = 1:3^(i-1)   %counter for nodes in the stage
        for k=1:O.Np  %counter for days in the node/stage
            for l=1:O.P   %counter for zones
                Of1 = [Of1,F.PCT(l).mat(O.Np*(i-1)+k,1+ (j-1)*3^(O.St-i)
                    )];
            end
        end
    end
end
%Recourse costs
Of2 = turnNP2KNpP(F.CI,O);
%booking costs
Of3 = turnNP2KNpP(F.CR,O);
%probabilities of each scenario
pmat=[];
for i=1:O.St
    new=[];
    wide = 3^(O.St-i);
    times = 3^(i-1);
    for j=1:times
        new = [new,repmat( sum(F.pt( 1+(j-1)*wide:j*wide ))  , O.Np,wide)
            ];
    end
    pmat = [pmat;new];
end
Of4=[];
for i=1:O.St %counter for stages
    for j = 1:3^(i-1)   %counter for nodes in the stage
        for k=1:O.Np  %counter for days in the node/stage
            for l=1:O.P   %counter for zones
                Of4 = [Of4,pmat(O.Np*(i-1)+k,1+ (j-1)*3^(O.St-i) )];
            end
        end
    end
end
Of = repmat(Of4,1,3).*[Of3, Of3, -Of2-Of1 ];
clear Of1 Of3 Of4
%%%%%%%%%%%%%%%%%%%%%%%%%
%                       End of  Constraints for Z
%%%%%%%%%%%%%%%%%%%%%%%%%
   %%%%%%%%%%%%%%%%%%%%%%%%%
%                       Begining point routine
%               we will use the matrices to solve a more constrained
    problem
```

```
            %               in order to provide the solver with a feasible starting
                point,
            %               the following is also shown in function '
                trick_minimize_D_ll_'
            %%%%%%%%%%%%%%%%%%%%%%%%%
              %here, we solve a linear version of the upper level problem, such that
                  xN<=0, or xN>=0, thus the lower level is trivial
              %this porvides us with a starting point for the double level
                  optimization forecoming
              display('*************************************************');
              display('*************************************************');
              display('*************************************************');
              disp(blanks(3)');
              display('Starting point optiomization results:');
              disp(blanks(1)');
              [starting_point,upper_bound, flaglinear] = linprog(  Of     , AQ,BQ,AeQ
                  ,BeQ,LQ, UQ);
              if flaglinear <0
                  error('    Problem is unfeasible');
              else
                  display('    Starting Point Obtained: Upper Problem is feasible')
              end
              disp(blanks(1)');
              %%%%%%%%%%%%%%%%%%%%%%%%%
              %       End of  Starting Point Routine
              %%%%%%%%%%%%%%%%%%%%%%%%%
              penalizer = 100;
              record= [];
              display('***********************************************');
              display('***********************************************');
              display('***********************************************');
              display('Upper level Optimization Results: ');
              disp(blanks(1)');
%First, we minimize function 'minmize_D_ll_', obtain the value for 'x'
[point1,ul_value] = fmincon(@(x)(minimize_D_prev_(x,q,O,F.pt)+Of*x'), starting_point
    ', AQ, BQ, AeQ,BeQ, LQ, UQ,[],optimset('TolCon',1));
%here, we retrieve the lower level values not delivered by fmincon
sol = minimize_D_prev_rep(point1,q,O,F);
final_value = ul_value + Of2*N11';
clear Of2 N11
report_stoch(sol,final_value,O,F,'all');
%the reported point is formed by the optimal 'x' and the optimal 'yuv' vectors found
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

A.3 Time Series Analysis

This section presents the time computer functions that involve the use and calculations of time series regression analysis.

Dendrogram Grouping Method

The dendrogram grouping method first obtains the IMLR regressions for all states and then forms groups using the statistics toolbox function `linkage`. The pooled regression is performed with the data from the states of all groups.

```
function [full, G, D] = AlgorithmDdg(DM, PM, Min_g, Max_g, linkmet, normalize)
%function AlgorithmDdg(DM, PM, Min_g, Max_g, linkmet, normalize) returns an
%array of structures with length(States) structures, that is the grouped
%regression parameters for the States in the list States. Perform the
%additional cluster formation using linakge function
% %form the simple regression matrix of coefficents 'single'    --- nearest
% distance 'complete'  --- furthest distance 'average'   --- unweighted
% average distance (UPGMA) (also known as group average) 'weighted'  ---
% weighted average distance (WPGMA) 'centroid'  --- unweighted center of
% mass distance (UPGMC) (*) 'median'    --- weighted center of mass
% distance (WPGMC) (*) 'ward'      --- inner squared distance (min variance
% algorithm) (*)
%first let us count the number of arguments
if nargin<=4
    error('Not enough parameters; enter a demand matrix, a price matrix, a minimum
        number of groups, a maximum allowed number of groups, a linkage method, and (
        optional) whether or not normalize the ressults')
end
if nargin<=5
    normalize = 0;
end
%create the list of names for every state.
fullnames = char('United States', 'Alabama', 'Alaska', 'Arkansas', 'Arizona', '
    California', 'Colorado', 'Connecticut', 'Delaware', 'DC', 'Florida', 'Georgia', '
    Hawaii', 'Idaho', 'Illinois', 'Inidiana', 'Iowa', 'Kansas', 'Kentucky', 'Loisiana
    ', 'Maine', 'Maryland', 'Massachusets', 'Michigan', 'Minesota', 'Mississippi', '
    Missouri', 'Montana', 'Nebraska', 'Nevada', 'New Hampshire', 'New Jersey', 'New
    Mexico', 'New York', 'North Carolina', 'North Dakota', 'Ohio', 'Oklahoma', '
    Oregon', 'Pensilvania', 'Rode Island', 'South Carolina', 'South Dakota', '
    Tennesse', 'Texas', 'Utah', 'Vermont' , 'Virginia', 'Washington', 'West Virginia'
    , 'Wisconsin', 'Wyoming');
%Set the general parameters
G.groups=Min_g;
G.states=length(DM(1,:));
G.assign = zeros(1,G.states);
G.span=length(DM(:,1));
% obtain the IMLR regression data
```

```
tic;
tind = (1:length(DM(:,1)))';
for i = 1 :length(DM(1,:))
    full(i).number = i;
    full(i).state = fullnames(i,:);
    full(i).price = PM(:,i);
    full(i).demand = DM(:,i);
    [b,bint,r,rint,stats]=regress(full(i).price,  [ones(tind(end),1),tind, tind.^2,(
        full(i).demand).^-1,( (full(i).demand).^-1).*tind,(tind.^2).*(full(i).demand)
        .^-1]);
    full(i).sr_b = b;
    full(i).sr_sign =(~(~all (bint' < 0)'.*~all (bint'  > 0)').*~all(abs(bint')
        <0.001)' )';
    full(i).sr_bint = bint;
    full(i).sr_R = stats(1);
    full(i).sr_F = stats(2);
    full(i).sr_p = stats(3);
    full(i).sr_err = stats(4);
    full(i).sr_r = r;
    full(i).sr_pest =  [ones(tind(end),1),tind, tind.^2,(full(i).demand).^-1,( (full(
        i).demand).^-1).*tind,(tind.^2).*(full(i).demand).^-1]*b;
end
clear DM PM
%form the parameter matrices
bmatrix = zeros(G.states,6);
for i=1:G.states
    bmatrix(i,:) = full(i).sr_b';
end
if normalize ==1
    bmatrix = bmatrix./repmat(max(bmatrix), length(bmatrix(:,1)), 1);
end
%form the clusters
Art1=clusterdata(bmatrix,'linkage',linkmet,'maxclust',Max_g);
G.dr_groups= max(Art1);
for i=1:G.dr_groups
    D(i).x = [];
    D(i).y=[];
    D(i).size = sum( Art1 == i);
    D(i).list = [];
    D(i).members=[];
end
%initialize and form the pooled regression data
for i= 1:G.states
    ind = Art1(i);
    full(i).dr_group = Art1(i);
    %for j=1:D(ind).size
        D(ind).x = [D(ind).x;full(i).demand];
        D(ind).y = [D(ind).y;full(i).price];
        D(ind).list = [ D(ind).list ,i];
```

```
        D(ind).members = [D(ind).members;full(i).state];
    %end
end
%perform the grouped regression and store that data in the group structures
%D
for i=1: G.dr_groups
    tind=(1:length(D(i).x))';
    unique=[];
    for ki=1: D(i).size
        unique = blkdiag(unique,ones(G.span,1));
    end
    [b1,bint1,r1,rint1,stats1] = regress(D(i).y,   [unique,tind, tind.^2,(D(i).x)
        .^-1,( (D(i).x).^-1).*tind,(tind.^2).*(D(i).x).^-1]);
    %Update the group's R value and variables
    D(i).b =b1;
    D(i).bint = bint1;
    D(i).r = r1;
    D(i).R=stats1(1);
    D(i).F=stats1(2);
    D(i).p=stats1(3);
    D(i).err=stats1(4);
    D(i).sign = (~(~all (bint1' < 0)'.*~all (bint1'  > 0)').*~all(abs(bint1')<0.001)'
         )';
    tind1=(1:length(D(i).y))';
    D(i).pest =  [unique,tind1, tind1.^2,(D(i).x).^-1,( (D(i).x).^-1).*tind1,(tind1
        .^2).*(D(i).x).^-1]*D(i).b;
end
%Assign their respective values to the groups
for i=1 : G.groups
    tind1=(1:length(D(i).y))';
    unique=[];
    for ki=1: D(i).size
        unique = blkdiag(unique,ones(G.span,1));
    end
    pest2 =  [unique,tind1, tind1.^2,(D(i).x).^-1,( (D(i).x).^-1).*tind1,(tind1.^2)
        .*(D(i).x).^-1];
    if isempty(pest2)
        display('ERROR');
    end
    for j=1:length(D(i).list)
        ind = D(i).list(j);
        full(ind).dr_b = [D(i).b(j); D(i).b(end-4:end)];
        full(ind).dr_bint = D(i).bint;
        full(ind).dr_r = D(i).r;
        full(ind).dr_R = D(i).R;
        full(ind).dr_F=D(i).F;
        full(ind).dr_p = D(i).p;
        full(ind).dr_err = D(i).err;
        full(ind).dr_sign = D(i).sign;
```

```
        full(ind).dr_pest =   pest2(      (j-1)*G.span +1 : j*(G.span)   ,:)*D(i).b;
    end
end
for i=1:G.states
    G.assign(i) = full(i).dr_group;
end
%record the running time
G.dr_time=toc;
beep;
```

Heuristic Grouping Algorithm

The heuristic grouping algorithm presented here first calculates regression data for each state, and then groups the states according to the schematics in algorithms 5 and 6.

```
function [full, G1, C1] = Algorithm_Heur(DM, PM, Min_g, Max_g, Threshold, Weight,
    Iter, start_vector, sample_size)
%function Algorithm(DM, PM, Min_g, Max_g, Threshold,Weight, Iter, start_vector,
    sample_size) returns an array of
%structures with length(States) structures, that is the grouped regression
%parameters for the States in the list States.  start_vector is a vector with n group
     leaders
%First, we create the list of names for every state.
fullnames = char('United States', 'Alabama', 'Alaska', 'Arkansas', 'Arizona', '
    California', 'Colorado', 'Connecticut', 'Delaware', 'DC', 'Florida', 'Georgia', '
    Hawaii', 'Idaho', 'Illinois', 'Inidiana', 'Iowa', 'Kansas', 'Kentucky', 'Loisiana
    ', 'Maine', 'Maryland', 'Massachusets', 'Michigan', 'Minesota', 'Mississippi', '
    Missouri', 'Montana', 'Nebraska', 'Nevada', 'New Hampshire', 'New Jersey', 'New
    Mexico', 'New York', 'North Carolina', 'North Dakota', 'Ohio', 'Oklahoma', '
    Oregon', 'Pensilvania', 'Rode Island', 'South Carolina', 'South Dakota', '
    Tennesse', 'Texas', 'Utah', 'Vermont' , 'Virginia', 'Washington', 'West Virginia'
    , 'Wisconsin', 'Wyoming');
%the next if blocks will check the arguments of the function and assign defaults if
    needed.
if nargin <=4
    display('Not enough input parameters. Enter at least a Demand Matrix, a Price
        MAtrix, a minimum number of gropus, a maximum nonber of groups allowed, and a
         threshold.')
    return
end
if nargin <= 5
    Weight = 0.5;
    display('Using the default weight, 0.5!');
end
if nargin <= 6
    Iter = 5;
    display('Using the default number of iterations, 5!');
```

```
end
if nargin <=7
    start_vector = [];
end
if nargin <=8
    sample_size = length(DM(:,1));
end
if ~isempty(start_vector) && length(start_vector) ~= Min_g
    error('The start vector must match the starting minimum number of groups!')
end
if sample_size > 0 && sample_size <=1
    sample_size = ceil(sample_size*length(DM(:,1)));
end
if sample_size > length(DM(:,1)) || sample_size  == 0
    display('Sample size is larger than data, reseting....!');
    sample_size = length(DM(:,1));
end
if all(size(DM) ~=size(PM))
    error('Price and Demand Matrices are not of the same dimensios!')
end
%initialize the definitions of the 'G' report structure
G.groups=Min_g;
G.states=length(DM(1,:));
G.assign = zeros(1,G.states);
G.span=length(DM(:,1));
% obtain the IMLR data
tic;
tind = (1:length(DM(:,1)))';
for i = 1 :length(DM(1,:))
    full(i).number = i;
    full(i).state = fullnames(i,:);
    full(i).price = PM(:,i);
    full(i).demand = DM(:,i);
    [b,bint,r,rint,stats]=regress(full(i).price,  [ones(tind(end),1),tind, tind.^2,(
        full(i).demand).^-1,( (full(i).demand).^-1).*tind,(tind.^2).*(full(i).demand)
        .^-1]);
    full(i).sr_b = b;
    full(i).sr_sign =(~(~all (bint' < 0)'.*~all (bint'  > 0)').*~all(abs(bint')
        <0.001)' )';
    full(i).sr_bint = bint;
    full(i).sr_R = stats(1);
    full(i).sr_F = stats(2);
    full(i).sr_p = stats(3);
    full(i).sr_err = stats(4);
    full(i).sr_r = r;
    full(i).sr_pest =  [ones(tind(end),1),tind, tind.^2,(full(i).demand).^-1,( (full(
        i).demand).^-1).*tind,(tind.^2).*(full(i).demand).^-1]*b;
end
clear DM PM
```

```
%record the IMLR time
G.sr_time= toc;
tic;
%begin the group regression
%select the initial groups
act=(1:G.states);
%begin iterations
best_iter_R=-inf;
for k=1:Iter    %main iteration that repeats the heuristic process time and again
    %begin the group regression
    %select the initial groups
    act=(1:G.states);   %all states are initially active
    clear C
    G.groups=Min_g;
    G.assign = zeros(1,G.states);
    G.avrR = 0;
    %chose the iniital group leaders, either accroding to the argument provided, or
        according to a random vector
    if isempty(start_vector)
        start_vector = selectrand(1:52, Min_g);
    end
    %assign the leaders' data to the groups
    for i = 1:Min_g
        a=start_vector(i);
        G.assign(a) =  i;
        C(i).x=[full(a).demand];
        C(i).y=[full(a).price];
        C(i).list=[a];
        C(i).members = full(a).state;
        C(i).size = 1;
        C(i).R=full(a).sr_R;
        C(i).b=full(a).sr_b;
        C(i).bint=full(a).sr_bint;
        C(i).r=full(a).sr_r;
        C(i).F=full(a).sr_F;
        C(i).p=full(a).sr_p;
        C(i).err=full(a).sr_err;
        C(i).sign=full(a).sr_sign;
        C(i).pest=full(a).sr_pest;
    end
act(start_vector) = [];
    %Pick a new group member and enlist it into some group, one at the time
    while length(act)>0
        %select a new group member
        a= ceil(length(act)*rand(1));
        %chose which group will have the least R^2 if the new member is enlisted into
            it
        best_r=-inf;  %the best R value so far
        R= repmat(0,1,G.groups);   %the value for the group regression
```

```
R_indiv = repmat(0,1,G.groups);   %the value for the new individual entering
    a group
for i=1:G.groups
        tind=repmat((1:length(full(act(a)).demand))', C(i).size+1, 1);
    unique=[];
    for ki=1: C(i).size+1
        unique = blkdiag(unique,ones(G.span,1));
    end
    regressor = [unique,tind, tind.^2,([C(i).x;full(act(a)).demand]).^-1,( ([
        C(i).x;full(act(a)).demand]).^-1).*tind,(tind.^2).*([C(i).x;full(act(
        a)).demand]).^-1];
    independent =  [C(i).y;full(act(a)).price];
    sample = sort(selectrand(1:length(independent), sample_size*(C(i).size+1)
        ));
    [b,bint,r,rint,stats] = regress(independent(sample), regressor(sample, :)
        );
    R(i)= stats(1);
    tind1=(1:length(full(act(a)).demand))';
    pest2 =  [unique,tind, tind.^2,([C(i).x;full(act(a)).demand]).^-1,( ([C(i
        ).x;full(act(a)).demand]).^-1).*tind,(tind.^2).*([C(i).x;full(act(a))
        .demand]).^-1];
    pest2 = pest2(end-length(tind1)+1:end,:)*b;
    R_indiv(i)=  1 -      sum(   (  full(act(a)).price - pest2  ).^2  )   /
        sum((full(act(a)).price - mean(full(act(a)).price)).^2) ;
 % plot([independent, regressor*b])     %uncomment if you want plots
  %A weight of 0 means that only group-R^2 are considered, whereas a Weight
      value of 1 means that only individual-R^2 are considered to evaluate
      the groups
  if (1 - Weight)*R(i) + Weight * R_indiv(i)  > best_r
      cind = i;
      best_r = (1 - Weight)*R(i) + Weight * R_indiv(i);
      b1=b;
      bint1=bint;
      r1=r;
      stats1=stats;
  end
  regressor = [];
  independent=[];
end
%determine if the R^2 is bad enough to have its own group
if(best_r <Threshold) && (G.groups<Max_g)
    G.groups = G.groups+1;
    cind = G.groups;
    C(cind).x=[full(act(a)).demand];
    C(cind).y=[full(act(a)).price];
    C(cind).R=full(act(a)).sr_R;
    C(cind).list = [act(a)];
    C(cind).size = 1;
    C(cind).members = full(act(a)).state;
```

```
        C(cind).b = full(act(a)).sr_b;
        C(cind).bint = full(act(a)).sr_bint;
        C(cind).r = full(act(a)).sr_r;
        C(cind).F=full(act(a)).sr_F;
        C(cind).p = full(act(a)).sr_p;
        C(cind).err = full(act(a)).sr_err;
        C(cind).sign = full(act(a)).sr_sign;
        tind1=(1:length(C(cind).y))';
        unique=[];
        for ki=1: C(cind).size
            unique = blkdiag(unique,ones(G.span,1));
        end
        C(cind).pest =  [unique,tind1, tind1.^2,(C(cind).x).^-1,( (C(cind).x)
            .^-1).*tind1,(tind1.^2).*(C(cind).x).^-1]*C(cind).b;
    else
        %Update the group's R value and variables
        C(cind).x=[C(cind).x; full(act(a)).demand];
        C(cind).y=[C(cind).y; full(act(a)).price];
        C(cind).R=best_r;
        C(cind).size = C(cind).size+1;
        C(cind).list = [C(cind).list,act(a)];
        C(cind).members = [C(cind).members;full(act(a)).state];
        C(cind).b =b1;
        C(cind).bint = bint1;
        C(cind).r = r1;
        C(cind).R=stats1(1);
        C(cind).F=stats1(2);
        C(cind).p=stats1(3);
        C(cind).err=stats1(4);
        C(cind).sign = (~(~all (bint1' < 0)'.*~all (bint1'  > 0)').*~all(abs(
            bint1')<0.001)' )';
        tind1=repmat((1:length(full(act(a)).demand))', C(cind).size, 1);
         unique=[];
        for ki=1: C(cind).size
            unique = blkdiag(unique,ones(G.span,1));
        end
        C(cind).pest =  [unique,tind1, tind1.^2,(C(cind).x).^-1,( (C(cind).x)
            .^-1).*tind1,(tind1.^2).*(C(cind).x).^-1]*C(cind).b;
    end
    plot([C(cind).y,C(cind).pest]);
    %Assign the group to the state
    full(act(a)).hr_group = cind;
    %Make the assignments and updeta the list
    G.assign(act(a))=cind;
    act(a)=[];
end
%determine the best iteration;
cum_R = 0;
for(i=1:G.groups)
```

```
        cum_R = cum_R + C(i).R;
    end
     if cum_R > best_iter_R
         best_iter_R = cum_R;
         C1=C;
         G1=G;
         G1.avrR = best_iter_R/G1.groups;
     end
end
%Assign the latest values to the states.
for i=1 : G1.groups
    for j=1:length(C1(i).list)
        ind = C1(i).list(j);
        full(ind).hr_b = [C1(i).b(j); C1(i).b(end-4:end)];
        full(ind).hr_bint = C1(i).bint;
        full(ind).hr_r = C1(i).r;
        full(ind).hr_R = C1(i).R;
        full(ind).hr_group = i;
        full(ind).hr_F=C1(i).F;
        full(ind).hr_p = C1(i).p;
        full(ind).hr_err = C1(i).err;
        full(ind).hr_sign = C1(i).sign;
         tind1=repmat((1:length(full(ind).demand))', C1(i).size, 1);
         unique=ones(G.span,1);
         for ki=1: C1(i).size-1
             unique = blkdiag(unique,ones(G.span,1));
         end
         pest2 =  [unique,tind1, tind1.^2,(C1(i).x).^-1,( (C1(i).x).^-1).*tind1,(
             tind1.^2).*(C1(i).x).^-1];
         if length(pest2)==0
             display('ERROR');
         end
         full(ind).hr_pest =   pest2(      (j-1)*G.span +1 : j*(G.span)   ,:)*C1(i).b;
    end
end
%measure the heuristic time
G1.hr_time=toc;
tic;
```