Optimization Methods for Pipeline Transportation of Natural Gas

CONRADO BORRAZ-SÁNCHEZ

Dissertation for the degree of *PhilosophiæDoctor (PhD)*

Department of Informatics
University of Bergen, Norway

October 2010

To my beloved wife, Denyce,
and to my apprehensive parents,
Professor Conrado de Jesus and Magnolia,
for everything you all have done for me in pursuing my dream.

Acknowledgements

I would like here to express my thankfulness to all those people who somehow contributed to make my path towards the achievement of this modest work easier and, indeed, more enjoyable. Nevertheless and foremost I thank life for making dreams come true.

First, I extend my profound and sincere gratitude to my main supervisor, Dag Haugland, for making every meeting a challenge for improvement. There was nothing better than being face to face with an experienced researcher like him to enhance the potential insights I might have had on the research projects at that time. This work is as a result of all discussions we had on the last three years.

I would also like to thank to my co-supervisor, Trond Steihaug, who implicitly taught me through non-linear optimization about how to cope with smooth or non-smooth functions in order to achieve an optimum level. Trond and Dag, besides being remarkable professors, are great researchers who made me comprehend, by means of their day-to-day research work, why the University of Bergen possesses a high standard in terms of the research quality is concerned.

A special thanks goes to Dr. Roger Z. Ríos-Mercado, associate professor at the Universidad Autónoma de Nuevo León, Mexico, who besides being co-author of my first article, has been my mentor and my friend all along the way I've covered into the wonderful world of research.

I would be completely unfair if I did not mention to Dr. Lennart Frimannslund, Dr. Mohamed El Ghami, and Mohammed Alfaki, who if they allow me, I may say that they have been the closest friends I could ever have had here in Bergen. Talking to them with such a freedom, expecting always to have a really smart conversation was undoubtedly very helpful to increase my knowledge in optimization, their mathematical expertise was overwhelming. My most sincere appreciation to the entire Optimization group, Professor Emeritus Sverre Storøy, Joanna Bauer, and Sara Suleiman.

I would like to thank to Ida Holen and Signe Knappskog, current and former Head of the Administration, respectively, for all their valuable help and charismatic friendship for coping with all administrative issues during my PhD studies. My gratitude goes also to Marta Lopez, Department Secretary, who beyond being the person in charge of dealing with bureaucracy, has become an excellent friend of mine... *muchas gracias por todo Marta*.

Finally, I am deeply grateful to the University of Bergen, and particularly to the Department of Informatics, for being granted a research fellowship financed by the Norwegian Research Council, Gassco and Statoil under contract 175967/S30.

Contents

List of Figures

List of Tables

Part I

Overview

CHAPTER 1

Introduction

Three years have come and gone so quickly. I can now say that my long learning road has undoubtedly been as enthralling as challenging. Mountains and non-smooth continuous valleys of knowledge may represent such amazing journey, in which I had obtained –perhaps through exponential efforts– a tiny portion of that valuable knowledge only when reaching the maximum point of the highest mountains.

OVER THE PAST COUPLE OF CENTURIES, fossil fuels, as primary energy sources, have been essential for global economic growth. During the industrial revolution in Europe in the 19th century, coal played a key role in supporting technological progress in agriculture, manufacturing and transport. Since then, petroleum has superseded the position of coal, and is an essential factor in sustaining our very expensive and 'dangerous' lifestyle.

Nowadays, however, the continual and indiscriminate increase in the price of oil, coupled with the significant decline in reserves, as well as the new environmental attitude expressed by various national governments about the existing high levels of air pollution, have led to the exploitation of a cleaner and more economically attractive fuel, namely *the natural gas*.

In contrast to petroleum or coal, natural gas can be used directly as source of primary eco-friendly energy that causes less carbon dioxide and nitrogen oxide emissions (*green house gases*). Thus, natural gas has proven to be a strategic commodity that augments current global energy supplies and, to some extent it alleviates some of the possible consequences of using petroleum and petroleum derivatives.

As stated by Shell [126], a global group of energy and petrochemical companies, *the world is hungry for energy*. As a consequence of this demand, a significant effort to increase world energy reserves has been made during the last decades despite the recessions and economic issues currently faced by worldwide gas producers.

According to *Cedigaz*[1]'s 2009 News Report [19], 2008 witnessed the largest expansion of gas reserves in the world for almost a decade (see Fig. 1.1). This increase may certainly lead to the establishment of a number of sustainable but complex projects in both international and national arenas, which would in turn entail major challenges for the gas industry. A clear example is that of the Islamic Republic of Iran. In 2008, this country increased its national reserves of natural gas to 27.57 trillion cubic meters (Tcm), corresponding to 18% of the world's total natural gas reserves [18]. Iran may now face the challenge of implementing several international projects, possibly with foreign investors, which would imply, among other things, a rapid expansion of its natural gas transportation systems if the country wants to succeed in becoming a leading natural gas exporter in the years ahead.

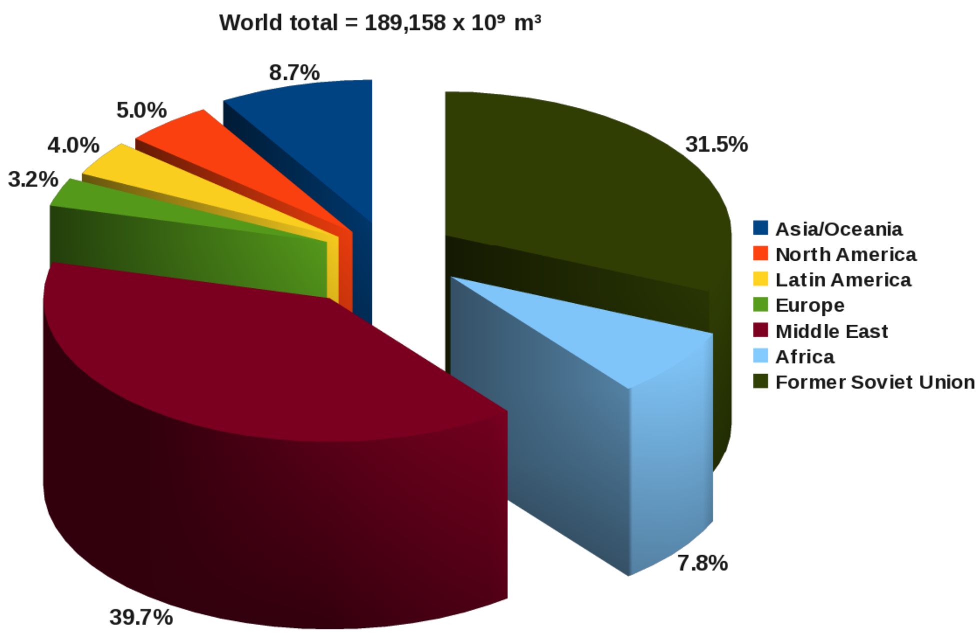


FIGURE 1.1: Global natural gas reserves in 2009 – *CEDIGAZ Source*

In Northern Europe, according to the *Oil&Gas Journal*'s 2010 International News [103], a new plan for the development of *the North Sea Gudrun gas and*

[1] *Cedigaz* – An international association dedicated to natural gas information in the world.

oil field has recently been approved by *the Norwegian Parliament* to Statoil (the leading operator on *the Norwegian Continental Shelf*) and partners. The project is estimated at US$3.6 billion and comprises the drilling of 7 production wells and the construction of a fixed-steel platform in 2011. The wells are expected to go into operation in 2014, thus leading to a steady production growth for the years ahead.

In South America, several multi-national contracts have been signed to transport offshore gas ashore. Chile, for example, secured international funding worth more than US$1 billion in 2007 [21] for constructing the regasification plant "GNL Quintero S.A." to provide natural gas to its central region from 2009 and to overcome some of the difficulties in the energy sector currently facing the country. Similarly, Argentina and Uruguay signed an agreement in 2008 [18] for the construction of a regasification terminal, which is intended to come into operation in 2012.

Hence, the current position of natural gas as a primary non-renewable energy source (second to oil) leads to the conclusion that the analysis, design and improvement of its processes, including transportation, play a significant role for both private and public sectors while offering a number of challenges to the scientific research community.

Here, the robustness and profitability offered by the study and application of an interdisciplinary field such as *optimization*, also referred to as *mathematical programming*, are essential to effectively tackle many of the difficulties associated with natural gas from extraction until delivery to the end-costumers by means of transportation pipeline systems, for example.

We can think of *optimization* as a science in constant development, with a narrow link between *computer science* and *applied mathematics*. As is the case in many scientific areas, detailed definitions of the theory can fill several large volumes. The same applies to *optimization* and as such, this would exceed the expectations of this work. Nonetheless, in order to provide a frame of reference in which to base the contributions of this thesis, Chapter 3 provides an overview and a brief formal introduction of this magnificent field of scientific research.

A number of well-developed methods have been already applied to many processes used in connection with natural gas. However, the favorable increase in global natural gas reserves described above, coupled with the expansion of gas consumption in both public and private sectors, as well as the exponential expansion of the transmission networks during the last decades, have posed several challenging problems that require more accurate models and more sophisticated and advanced algorithms. Hence, the design, application and assessment of the optimization methods proposed in this work are required in order to tackle the three research projects that constitute this PhD thesis. A summary of these three research projects is presented next.

1.1 My PhD research frame

My PhD research frame comprises three research projects directly related to the optimization of pipeline network systems for the transport and storage of natural gas. The main goal of this thesis is to present the research conducted in each of these projects. Thus, the major issues covered by the thesis are the study of the mathematical formulations and difficulties of the corresponding optimization problems, as well as the several approaches that help to a better understanding of the topic for the future research. The title and a brief description of each project, and the way how they were tackled during my PhD studies are presented as follows.

PROJECT 1 is entitled:

MINIMIZING COMPRESSOR FUEL COST ON LARGE PIPELINE TRANSPORTATION NETWORKS FOR NATURAL GAS

This project can basically be described as follows: In gas transportation networks, designed for either gathering, transmitting or distributing natural gas, compressor stations are used to supply the energy necessary to overcome frictional pressure losses while maintaining the gas streams flowing through the network. However, a significant proportion of the transported gas (estimates vary between 3% and 5%) is consumed by series-parallel banks of compressors installed in the network before the gas arrives at the reception units. Keeping this consumption to a minimum is a task that not only represents large financial value to the industry, but also has an important environmental dimension. This results in the *fuel cost minimization problem* (FCMP).

This project is based on an in-depth mathematical study of the NLP model suggested by Wu et al. [144]. Consequently, the thesis presents three solution methodologies in order to tackle the complexities associated with the mathematical model. These approaches are applied to a wide range of network topologies with real-data of 9 different compressor units provided by the industry.

The first methodology is based on a hybrid meta-heuristic approach that includes a reduction technique proposed by Ríos-Mercado et al. [120] as a preprocessing phase. As essential part of this approach, a discretized subproblem is solved by a non-sequential dynamic programming method proposed by Carter in [15].

This solution approach is documented in Paper I (see Table 1.1). The findings presented by the paper are the result of my Master studies done under the supervision of Roger Z. Ríos-Mercado at the Universidad Autónoma de Nuevo León (UANL), Mexico. The extended numerical experiments concerning the multi-local search procedure introduced in the paper are the contribution done during my PhD studies. The experiments were done via remote connection with the facilities in Mexico.

This study led to the proposed approaches that followed my PhD research under the supervision of Dag Haugland at the University of Bergen. In 2009, the paper was awarded the most prestigious university-wide award granted by UANL within the engineering and technological research area.

This approach has two main advantages: a) It is applicable to linear, tree-shaped and cyclic networks, and b) it can easily handle the non-linearity and non-convexity of the model. However, Carter's technique requires the network to be sparse, and even small instances can easily be found where this method fails.

To overcome the limitation of Carter's technique, we enhance the dynamic programming (DP) technique invented by Bellman [7] into a solution methodology that handles a more general class of transportation network. The problem in question has a structure similar to a *frequency assignment problem* in wireless communication, and for such problems a tree-decomposition approach [77] has been applied in combination with a DP technique. We investigate the possibilities of adapting this approach to FCMP, and verify the method both theoretically and experimentally.

Even though the previous approach is applicable to a wide set of network topologies, the output and the running time depend on how fine the discretization is. Hence, a more sophisticated algorithm based on an adaptive discretization approach is proposed. This third methodology can be considered as an extension of the previous approach where the bounding techniques introduced in Paper I play a crucial role. This heuristic approach outperforms the efficiency of the previous approach and can effectively be applied to large networks.

PROJECT 2 is entitled:

MATHEMATICAL MODELING OF NETWORK FLOW PROBLEMS UNDER GAS VOLUMETRIC PROPERTIES VARIATION

This project focuses on the mathematical modeling of the pipeline resistance in transmission systems. The aim is to estimate in a more accurate way the flow capacity in pipelines based on the intrinsic variation of the volumetric properties of the natural gas mixture during the transmission process. In particular, this study concerns the influence of the variability of the gas specific gravity (g) and compressibility (z-factor) on maximum flows at downstream nodes of the network.

The literature on optimization models for pipeline gas transportation does not seem to be very rich on models with variable specific gravity or compressibility, and most works focus on models for transient flow. Interested readers are referred to the works presented by Abbaspour and Chapman [1] and Chaczykowski [20], and the references therein.

In contrast, the idea of modeling arc capacities as decision dependent functions is already well established in the optimization literature. In an early work on minimization of compressor fuel cost, Wong and Larson [141] suggested to model the pipeline capacity by means of the well-known Weymouth equation [105]. The same principle is followed by more sophisticated works, as those presented by Carter [15], Ríos-Mercado et al. [120], De Wolf and Smeers [30, 31], Borraz-Sánchez and Ríos-Mercado [11, 13], Bakhouya and De Wolf [6], Kalvelagen [69], and Borraz-Sánchez and Haugland [10].

All the cited works neglect the fact that the parameter in the Weymouth equation depends not only on pipeline characteristics, but also on thermodynamic and physical gas properties. This includes temperature, specific gravity (relative density) and compressibility. In instances where the network elements show no or only modest variation in these properties, it is sound modeling practice to represent them by global constants. This does however not seem to be the case in all real-life instances.

In this work, a non-convex NLP model for *maximizing flow in gas transmission networks while considering the variability of g and z-factor* is introduced. Basically, the specific gravity balance in the network is estimated by the incoming gas flows at junction points, where we assume an isothermal gas pipeline system in steady-state. The compressibility is in turn computed as a function of specific gravity, temperature and average pressure values by the *California Natural Gas Association method* [28].

Due to the non-linearity and non-convexity of the model, the proposal of a suitable method to overcome such complexities is also required. Hence, a heuristic algorithm based on an approximate model is proposed. The results obtained by applying BARON [131] (a global optimizer) and MINOS [97] (a local optimizer) on a GAMS [49] formulation are compared to the results achieved by the heuristic approach on a wide set of test instances.

PROJECT 3 is entitled:

OPTIMAL LINE-PACK MANAGEMENT IN GAS TRANSPORTATION NETWORKS

This project can be described as follows: Gas pipelines do not only serve as the transportation links between producer and consumer, but also represent potential storage units for safety stocks. Due to the compressible nature of dry gas, large reserves can be stored inside the pipe for subsequent extraction when flow capacities elsewhere in the system break down. Since it is likely that such unpredictable events do occur, keeping a sufficient level of line-pack therefore becomes critical to the transporter.

Managing the line-pack in a gas transportation network basically means optimizing the refill of gas in pipes in periods of sufficient capacity, and optimizing the withdrawal in periods of shortfall. Some attempts, although few, have been made in the direction of mathematical planning models for this problem (see [17] and [47]).

In this project, a multi-period model is proposed to tackle *the line-packing problem*. The model has non-linear constraints and both continuous and integer decision variables, and qualities thus as a mixed-integer non-linear programming (MINLP) model. In the project, we conduct an extensive numerical experimentation to evaluate the computability of the model. This experimental phase is based on a GAMS formulation for the MINLP model while applying the global optimizer BARON [131].

To sum up, based on a thorough research on these projects, this thesis demonstrates the need for studying and modeling complex optimization problems, including (non-convex) NLP and MINLP optimization models. Particularly, it shows the suitability of applying optimization methods, including exact and heuristic methods to natural gas transportation problems in steady-state. In addition, this thesis also presents several sets of test instances based on imaginary and real gas network systems that challenge the proposed solution techniques.

Details on the research conducted in each project are presented and discussed formally in Chapters 4–6. The scientific contribution of this thesis is presented next.

1.2 Contributions of the thesis

The contributions of this thesis are given by the five scientific papers presented in Table 1.1. The relation between the papers and the three research projects described above is shown in Table 1.2.

Table 1.1: Contributions of the thesis.

Paper #	Title, authors and publication channel
I	IMPROVING THE OPERATION OF PIPELINE SYSTEMS ON CYCLIC STRUCTURES BY TABU SEARCH *Borraz-Sánchez, Conrado & Ríos-Mercado, Roger Z.* Published in: *Journal of Computers & Chemical Engineering, Vol. 33 (1), pp. 58-64. (2009)*
II	A TREE DECOMPOSITION ALGORITHM FOR MINIMIZING FUEL COST IN GAS TRANSMISSION NETWORKS *Borraz-Sánchez, Conrado & Haugland, Dag* Published in: *IEEE conference proceedings 2009.* The cie39 International Conference on Computers & Industrial Engineering. ISBN 978-1-4244-4136-5. pp. 244-249. (2009)
III	MINIMIZING FUEL COST IN GAS TRANSMISSION NETWORKS BY DYNAMIC PROGRAMMING AND ADAPTIVE DISCRETIZATION *Borraz-Sánchez, Conrado & Haugland, Dag* Article in press: *Journal of Computers & Industrial Engineering.* doi: 10.1016/j.cie.2010.07.012 (2010)
IV	OPTIMIZATION METHODS FOR PIPELINE TRANSPORTATION OF NATURAL GAS WITH VARIABLE SPECIFIC GRAVITY AND COMPRESSIBILITY *Borraz-Sánchez, Conrado & Haugland, Dag* Preprint submitted to *Journal of the Spanish Society of Statistics and Operations Research: TOP.* (2010)
V	MODELING LINE-PACK MANAGEMENT IN NATURAL GAS TRANSPORTATION PIPELINE SYSTEMS *Borraz-Sánchez, Conrado & Haugland, Dag* Preprint submitted to *the INFORMS Computing Society 2011 Conference.* (2010)

Table 1.2: Relationship between the scientific paper and the research projects conducted.

Project No.	Key statements	Contributions
1	*Compressor stations* – Fuel Cost Minimization Problem (FCMP)	PAPER I, II AND III
2	*Gas specific gravity and compressibility* – Flow Maximization on pipeline systems	PAPER IV
3	*Short-term basis storage of natural gas* – Gas line-packing optimization on transportation pipeline networks	PAPER V

1.3 Outline of the thesis

The next five chapters present an overview of the major issues addressed in this work, including the formal discussion of the findings presented in Section 1.1.

Chapter 2 provides a formal introduction of two essential components associated with the optimization problems addressed in this thesis: *natural gas* (Section 2.1) and *transportation pipeline systems* (Section 2.2). The chapter provides a brief description of the main characteristics of both components, stressing the points at which this thesis contributes.

Chapter 3 presents an overview of the theoretical framework related to the research area studied in this thesis: *Optimization*. This chapter discusses the key issues when solving real-world decision-making problems, particularly related to the optimization of natural gas transport by means of pipeline systems.

Chapter 4 focuses on Project 1 on the operability of compressor stations, particularly when solving *the fuel cost minimization problem* (FCMP). The chapter introduces a (non-convex) NLP model and provides an analysis of the main challenges to be faced. Furthermore, three different solution methodologies are introduced and discussed to tackle the FCMP (Papers I, II, and III).

Chapter 5 focuses on Project 2 and presents a (non-convex) NLP model for a more accurate estimation of the maximum flow in transmission pipeline systems while taking into account the variation of gas volumetric properties. The chapter particularly offers a complete study on the role that the variability of gas specific gravity and compressibility plays in the correct estimation of the maximum resistance of pipelines (Paper IV). An extensive numerical evaluation of a heuristic approach, a local optimizer, and a global optimizer is also presented. The experiments are conducted on a wide range of test instances.

Chapter 6 focuses on Project 3 on the mathematical modeling of a strategy (line-pack) to hedge against events that may occur in a gas transmission system. Here, the design of a MINLP formulation to tackle the line-packing problem (Paper V) is discussed in detail.

Finally, the concluding remarks are given in Chapter 7. All papers cited in Table 1.1 are given in separate attachments.

Chapter 2

Natural Gas – *From the wellhead to the end consumer*

From prehistoric times, an exothermic chemical reaction known as *combustion* has been a key piece to generate heat and light for surviving. Two elements are required for the combustion to take place: an *oxidant agent* and a *fuel*. An oxidant agent is basically a chemical compound that transfers oxygen atoms, e.g., the *air* or *fluorine*[1]. The fuel is any source with the ability of releasing energy when it is burned while its chemical structure is transformed, such as *natural gas*.

This chapter provides both a formal introduction to natural gas, as well as a brief description of the main stages involved with the transportation of natural gas from the point at which it is extracted from the wellheads until it is delivered to the end-consumers, stressing the stages where this thesis contributes.

2.1 A closer look at natural gas

Natural gas (NG) is a colorless, odorless and tasteless fossil fuel, which is characterized as one of the cleanest and safest energy sources worldwide. Like the coal, uranium (nuclear energy) and oil and its derivatives such as gasoline, diesel fuel, propane, etc., NG is a non-renewable resource that is expected to be widely expanded in the decades to come. It is considered a very safe energy source when

[1] *Fluorine* is a highly dangerous element and the most chemically reactive of all elements. However, an advantage of this element is that it may also be used as oxidant agent to burn hydrocarbons without requiring a spark as energy input.

transported, stored and used since its range of flammability is limited, i.e., it requires both a correct mixture of air and fuel to burn somewhere, and an ignition temperature of approximately 1100 degrees Fahrenheit.

NG is a mixture consisting mainly (70-95%) of methane (CH_4, a *covalent bond*[1] composed of one carbon atom and four hydrogen atoms, see Fig. 2.1). Methane was discovered and isolated by Alessandro Volta, Italian physicist, known for his pioneering work in electricity, particularly for the development of the first *electric cell* (the so-called *voltaic pile*) in 1800.

NG also contains heavier gaseous hydrocarbons such as ethane (C_2H_6), propane (C_3H_8), normal butane (n-C_4H_{10}), isobutane (i-C_4H_{10}), pentane (C_5H_{12}), among other higher molecular weight hydrocarbons.

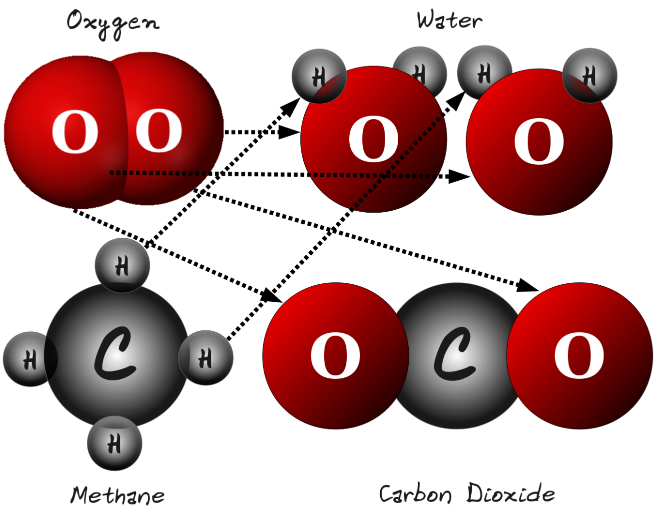


FIGURE 2.1: Models of molecules of oxygen (O_2), water (H_2O), methane (CH_4) and carbon dioxide (CO_2) – *NG typical molecules*

At the moment of its extraction from the ground, NG contains impurities or contaminants that have to be removed before being used as a consumer fuel (with the exception of the *mercaptans*[2] which sometimes are kept or added for safety reasons). Among these impurities we can find *acid gases*, such as hydrogen sulfide (H_2S), carbon dioxide (CO_2), mercaptans (methanethiol – CH_3SH, and ethanethiol – C_2H_5SH), nitrogen (N_2), helium (He), and water vapor (H_20).

Two terms are typically used in the literature concerning the *acid gas* components: a *sour gas* and a *sweet gas*. The former is associated to natural gas that contains significant quantities of sulfur compounds and hydrogen sulfide. The

[1] *Covalent bond* – Chemical bonding defined by the sharing of electrons pairs between atoms.
[2] *Mercaptans* – Strongly odorous compounds of carbon, hydrogen and sulfur.

latter is related to natural gas that is relatively free of sulfur compounds. The first term is still in debate since some authors may also refer to a sour gas as that gas containing CO_2 but no sulfur compounds.

During combustion, NG produces a great amount of heat while releasing very small amounts of sulfur dioxide (SO_2) and nitrogen oxides (NO_x, referred specifically to NO and NO_2) unless such compounds are removed before burning it. A typical characteristic of NG combustion in the air is a blue flame (see Fig. 2.2).

FIGURE 2.2: NG undergoing combustion – *The characteristic blue flame*

In contrast to coal and oil, NG produces lower levels of CO_2, CO, water vapor and particulate matter (PM, also referred to as fine particles, are tiny substances of solid or liquid matter suspended in the gas [138]). Table 2.1 shows the fossil fuel emission levels provided by *U.S. Energy Information Administration* [132] (EIA) in pounds of pollutants per Billion Btu[1] of energy input of fossil fuels.

Table 2.1: Fossil fuel emission levels (pounds per Billion Btu of energy input).

	Fossil fuels		
Contaminant	Natural gas	Oil	Coal
CO_2	117,000	164,000	208,000
CO	40	33	208
NO_x	92	448	457
SO_2	1	1,122	2,591
Particulates	7	84	2,744
Mercury	0.0	0.007	0.016

Source: EIA - Natural Gas Issues and Trends 1998

[1] *Btu – British thermal unit of energy.* 1 Btu ≈ 251.9 calories ≈ 1055 joules, is equal to the amount of heat required to raise the temperature of 1 pound of liquid water by 1 degree Fahrenheit at its maximum density, which occurs at a temperature of 39.1 degrees Fahrenheit. (The heat output of computer devices is often expressed in Btus.)

Note that even though NG is considered non-toxic, there may exist a potential asphyxiation hazard when mixed with air due to reduced oxygen content in the atmosphere.

2.1.1 Natural gas history in a nutshell

Natural gas has been observed since ancient times. The humankind's encounters with natural gas can likely be traced back from ancient Mesopotamia, or the cradle of civilization that is known today as the Middle East [64]. However, it was until a few centuries ago that countries such as China, Great Britain, USA, among others, started using natural gas as a means of maintaining their need for energy [93]. Moreover, it was the construction of a sustainable gas infrastructure, including storage, preprocessing and transport facilities, that allowed natural gas become commercially available throughout the world.

Mokhatab et al. mention in [93] that the Chinese drilled the first known NG well in 211 B.C. A few centuries later (around 500 B.C.), they would employ crude bamboos as a means to transport NG. Among other applications, they would use NG as energy source to boil sea water in order to desalinate it, i.e., to get drinkable water.

In Europe, even though the British discovered natural gas in the middle of the 17th century, it was until the late 18th century (around 1785) when they started trading NG obtained from coal seams for lighting houses and streets [122]. In North America, a few decades later (about 1816), the Americans followed the same strategic idea and began using NG for lighting the streets of Baltimore, Maryland. Few years later, in 1821, William A. Hart would be the first one to succeed in digging a $27ft$ wellhead in Fredonia, New York, USA [129].

In contrast to the almost exclusive use of natural gas in lighting houses and streets at the late 19th century, the *Bunsen burner*, named after Robert Bunsen in 1885 [68], was a key factor to show the immense scope that it could be given to the use of NG [81].

Undoubtedly, an intriguing factor that caused a faster expansion of the use of NG around the world in the last decade of the 19th century, was the fact that many cities began replacing their gas lamps with electric lamps. Thus, the gas industry was required to look for new markets, perhaps far away from their usual customers.

Natural gas had been clearly eclipsed by electricity. However, at that time the real problem was certainly the lack of a pipeline infrastructure to transport and distribute natural gas, as well as the lack of facilities to store it. In addition, events such as World War II would hinder the development or growth of NG as primary energy resource in the early 20th century.

After World War II, the technological progress in e.g., pipeline manufacturing, metallurgy and welding, enabled the gas industry to reemerge again. Thus, gas transport companies would start building up and expanding their pipeline systems. The fast and steady growth of gas industry would finally entail the construction of various gas facilities, including processing and storage plants, as well as a number of sustainable projects around the world since the late 20th century.

2.1.2 Uses

Currently, natural gas is used as fuel and raw material in petrochemical manufacturing such as hydrogen, ammonia (used in a range of fertilizers), ethylene and sulfur. It also provides power to a large set of gas appliances, such as gas furnaces, water heaters, gas grills, cooking stoves, etc., including heating and lighting in homes.

Due to its environmental qualities, coupled with its low cost, natural gas in its compressed (CNG) form has become a good alternative to be used to fuel vehicles (NGVs, see Fig. 2.3). In [64], Ingersoll claims that we are approaching the time when the transition from gasoline (and diesel) vehicles to NGVs is an unavoidable phase. However, although this technology is not recent (it goes back to 1930), it is still an ongoing process due to the limitations of CNG distribution to and at fueling stations, among other practicalities. (See [64] for a better survey.)

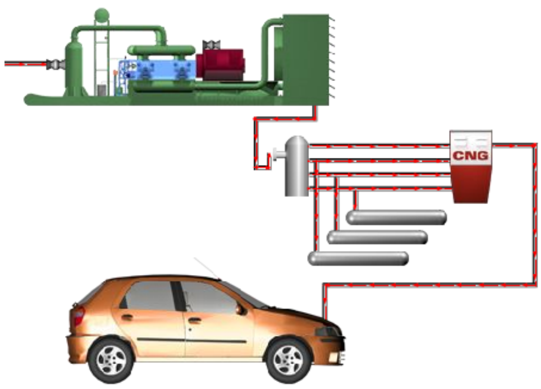


FIGURE 2.3: Natural gas vehicles – *(NGV)*

2.1.3 Distinguishing NG from NGL, CNG, LNG, ANG, and LPG

Special attention may be required to distinguish NG from other terms widely used in the literature such as NGL, LNG, CNG, ANG, and LPG.

The first term, NGL (*Natural Gas Liquids*) basically refers to all those gaseous hydrocarbons heavier than methane (previously mentioned) that are extracted in liquid state during the processing of NG for its consumption.

Concerning CNG, LNG and ANG, they represent methods applied for natural gas storage. CNG and LNG are commercially available, whereas ANG is still in a development phase (see [53]). CNG refers to the *compressed natural gas* in gaseous state, which has been exposed at extremely high pressure (200-250 bars[1]). CNG is primarily used for on-board gas storage of *natural gas vehicles* (also referred to as NGVs). In turn, LNG (*liquefied natural gas*) refers to natural gas that has been cooled about minus 260 degrees Fahrenheit while the *condensation* has taken place. LNG imposes a higher cost of production and storage compared to CNG. The motivation for carrying out such procedures of liquefaction of NG can be noted as followed.

As known, NG is typically transported through pipeline systems. However, for those regions with a high production potential where the construction of connection pipeline systems is too expensive or simply not viable, LNG may become a profitable alternative to store and transport natural gas as liquid by means of tanker ships. An inherent advantage of liquefying NG is that it greatly increases its density, i.e., the NG volume in its liquid state at atmospheric pressure is about 600 times less than in its gaseous form [88]. However, the liquefaction facilities at the shipping points, the tankers and cryogenic containers to transport and to keep the gas in its liquid state, as well as the *regasification terminals* at the delivery points require a large investment that in many cases turns out to be too high to make LNG a viable option. (Better surveys on LNG can be found in, e.g. [14] and [88].)

A viable example of the use of LNG is that of countries such as Algeria, Nigeria, Malaysia, Indonesia, Qatar, among others, where the exportation of natural gas to their neighbors can be easily done through pipelines, whereas it would turn out to be impossible exporting natural gas similarly to major consumers such as U.S. or Japan. Hence, LNG would turn out to be an excellent alternative to be applied as a big scale gas marine transportation.

[1] *bar:* Unit of pressure ≡ *atm pressure* on Earth at sea level. (1 bar = 14.5 psi = 100 kPa)

In addition, since typically there is no pipeline infrastructure beyond 500 miles offshore, a major buildup of LNG facilities throughout the world has recently become an important issue for the gas industry [93].

Note that, as mentioned in [88], NG, NGL, CNG and LNG differ from LPG (*Liquefied Petroleum Gas*, incorrectly called propane), since this latter is a term used for the product of crude oil distillation formed mainly of propane (C_3H_8) and butane (C_4H_{10}) in a liquid state. LPG contains much more energy than NG. 1 ft^3 of NG contains about 1,000 Btu, whereas 1 ft^3 of LPG contains about 2,500 Btu.

2.1.4 Deposits and formations of natural gas

NG was formed millions of years ago. Nonetheless, its transformation process is still not well understood.

According to the book *"Geology, the science of a changing Earth"* by Allison and Palmer [3], the theory on the creation of NG that is shared by chemists, geologist and other scientists, is that diverse strains of bacteria from plants and tiny sea animals were buried thousands of feet underground for millions of years. As time went by, pressure and heat exerted by multiple layers of sand and rock, coupled with possible addition of hydrogen from deep-seated sources, contributed to turn this decaying matter into petroleum and natural gas. Moreover, *Cap rock*, a term used for layer of non-porous rock, was in charge to prevent the rising of natural gas to the surface, thus keeping it trapped underground.

The great variety of underground formations, such as *shale formations*, *sandstone beds*, *coal seams*, among others, represent to the gas industry different complexities related to both the nature of physical properties and composition of natural gas, as well as the location of the underground formation. Often, NG flows along with oil to the surface from the same underground formation.

In USA for instance, the rock formations of clay where natural gas was trapped about 350 million years ago (in the Devonian Period), referred to as the *Devonian shales*, announced the first *shale formations* to be explored and commercially drilled in New York [60]. This wellhead produced just a few thousand of cubic feet per day of gas for 35 years, but it led to a new and fruitful energy source in the country.

Shale rock formations have been characterized by Geologists for holding substantial amounts of natural gas. However, since these formations are found several thousand feet underground, besides being relatively impermeable, they were considered unreachable until the advent of new drilling techniques, which made it possible to recover the gas. The result is a dramatic increase in estimated natural gas reserves worldwide.

Sandstone beds are unconventional gas deposits. Deposits known as *tight sand lenses*, due to natural gas has been trapped in tiny spaces, are included in these underground formations. These deposits are composed of tight holes that do not allow the gas to flow easily, thus requiring the application of special drilling techniques to crack the dense rock structure.

Coalbed methane gas (see Fig. 2.4) is a valuable potential source of gas that is found in all coal deposits.

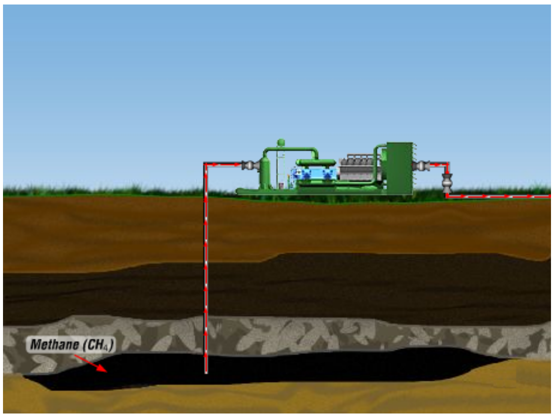


FIGURE 2.4: Coalbed methane – *Underground formation*

Another type of gas deposit that should be also taken into account is that formed by *gas hydrates* (discovered by Sir Humphry Davy in 1811 [29]), which are perhaps, from a gas transportation perspective, less friendly compounds but that promise to be a potential energy resource in the decades to come. They are found in deep ocean beds or in cold areas of the world.

A gas hydrate is an ice-like solid compound that currently causes problems both in the gas transmission process (pipeline plugging) under cold conditions, as well as in the processing and storage plants of natural gas (see e.g. [9], [84] and [146]). They are formed by the combination of water with small molecules called "hydrate formers", which are mainly all compounds of natural gas.

The hydrates are a subset of compounds known as *clathrates* or *inclusion* compounds. We can think of a clathrate as a non-chemical compound in which a molecule of one substance (typically gases) is enclosed in a structure built from molecules of another substance, i.e., it is trapped in a lattice of polyhedral water "cages" (see Figs. 2.5 and 2.6). This leads to the conclusion that, since the clathrates behave as gas concentrators, the amount of methane surrounded by these cages in ocean sediments may become quite significant in a very conservative estimate. (A better survey on gas hydrates can be found in Sloan's book [128], where a formal presentation of these crystalline compounds is provided in detail).

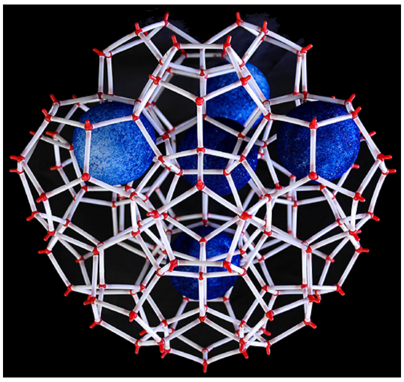


FIGURE 2.5: Typical structure of a clathrate hydrate – *136 water molecules*

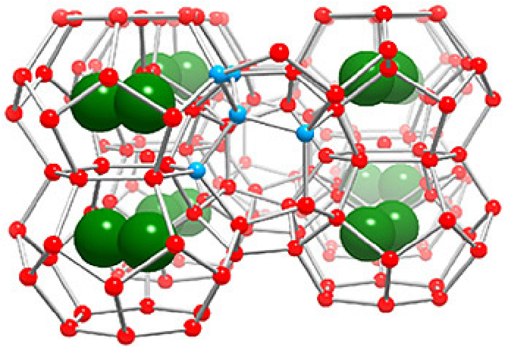


FIGURE 2.6: A more complex structure of a clathrate hydrate

Finally, a new and exciting gas formation has been developing in the last decade as a promising source of energy called *Biogas*. Biogas is provided by high technological devices known as *digesters*. Astonishingly, it is no longer necessary to wait millions of years for natural gas being created underground. Now, from food waste and garbage, biogas can provide electricity by means of mobile 'digesters' in just one day. The proverb *One man's trash is another man's treasure* could be the slogan that sums up this new, amazing technology.

In the next section, an overview of the natural gas industry, including its main activities, processes and stages, as well as its greatest challenges, is presented.

2.2 Gas industry

Natural gas, due to its cleanliness and easiness to be transported, is currently considered by national and transnational corporations all over the world as a good candidate for doing business. Nevertheless, the whole process that entails having natural gas ready for its consumption as a clean and efficient source of energy reflects a quite long road.

The main stages involved with natural gas from its extraction until its delivery to end customers, including their major characteristics and most important challenges, are presented in subsequent sections.

2.2.1 A glance at the scheme

The long road that natural gas typically covers from its origin to its final scheduled destination comprises a number of complex tasks (see Fig. 2.7). Activities such as exploration, extraction, processing, storage, transportation and distribution of natural gas are among these tasks that certainly offer interesting challenges in the everyday activities of the gas industry.

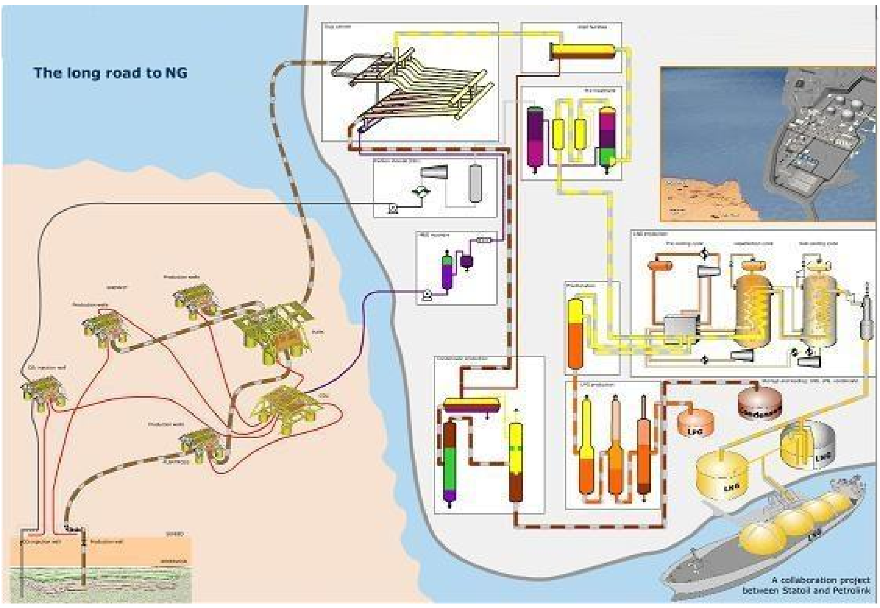


FIGURE 2.7: The long road of NG – *A very challenging task*

Due to the obvious complexity that would entail conducting all these physical processes together, the vast task undoubtedly has to be separated. Sayings such as *"grasp all, lose all"* and *"do not bite off more than you can chew"* seem to represent the philosophy followed by mostly all gas enterprises. Hence, not surprisingly, a common factor is that small and medium corporations make profits by means of facing and mastering only one specific activity. For example, while some companies may be in charge of exploring the ground looking for gas bearing formations that may lead to substantial profits, other companies may be in charge of processing the gas to make it ready for its consumption. Typically, the former companies also drill the wellhead and extract the gas once a potential area has been mapped.

On the other hand, transnational companies are usually in charge of natural gas transportation through cities, states or even countries. They may feed their pipeline systems with natural gas coming from various sources, including preprocessing plants, storage facilities or wellheads, in order to deliver it at certain discharge points, such as local distribution centers and large customers. Moreover, local distribution centers deliver natural gas to a number of end consumers, including residential, commercial and industrial consumers. Regardless of the services provided, a company certainly has to face optimization issues in order to be efficient and to maintain success in business.

2.2.2 Gas production - *Getting gas from the ground*

Before natural gas is ready for being transported as source of energy, three major stages are conducted, namely a) exploration of the ground for potential gas deposits, b) drilling a well for gas extraction, and c) processing the gas at the refinery for transportation and consumption.

2.2.2.1 Exploration stage

In the broadest sense, looking for potentially economic underground gas formations is the first step that gas producers face before starting to think about investing in perforations and land rights.

In early days, the only but inefficient way to find gas deposits was looking for seepages emitted from underground. Due the advent of the high technology, this process has become more accurate over the past years. Geologists and geophysicists can now use several extremely advanced methods to seek for potential gas deposits (*shale formations*, *sandstone beds* or *coal seams*).

The common factor of gas exploration methods is the gathering of data from rock formations to be interpreted later. Due to the complexity of this interpretation, uncertainty and trial-and-error are often involved to make educated guesses.

Geologists start examining the surface structure to determine areas that may serve as cap rock about underground formations. They can use the samples of rock obtained from previously digging gas wells to make inferences about the porosity, permeability, age or sequence of layers of a particular area. Once an area is mapping for possible gas formation, further tests can be conducted to recover more detailed data by means of advanced tools.

Among these tools, we can find the *seismic*[1] *exploration* [127] (the biggest breakthrough in gas and oil exploration) for subsurface researches of depths as great as 150km, but more useful for depths up to 10km. The idea is to produce seismic waves with small explosions in order to record the vibrations of the earth when these waves are reflected back to the surface by the different layers of rock. This seismic data acquisition will allow to infer what kind of layers exist underground and at what depth by means of both data processing (to improve the resolution and quality of the seismic data) and data interpretation (to recognize plausible geological patterns in the seismic image). *China National Petroleum Corporation* is currently the world's leading company in onshore geophysical operations using seismic explorations [147].

2.2.2.2 Extraction stage - *Drilling techniques*

Once an area has been mapped as a potential site containing a large amount of gas underground, it is time to start drilling a well.

The extraction stage is a very complex process that entails a great deal of responsibility because it brings out noise pollution and may put drinking water supplies at risk. This procedure basically consists of drilling the well and installing both a concrete and metal casing into the hole, and a collection pump above it. In addition, some special device or container may be required for incidental fluid that could come out of the gas well during production.

Note also that natural gas is often found in oil deposits and hence gas extraction becomes sometimes a side-operation of oil extraction.

A number of different drilling and fracturing techniques are currently available for gas extraction from the gas-bearing formation underground. Their use depends upon the hole size, equipment availability, geological specifications, and expertise of the well driller. For example, for reaching shallow gas deposits, a percussion drilling may be used. This involves raising and dropping a heavy metal

[1] *Seismology* is the study of how seismic waves (energy) move through the earth's crust and interact with the different layers of rock.

bit repeatedly into the ground. For reaching deep deposits, a rotary drilling technique is the most acceptable one, which involves the use of diamond studded drill bits and multiple sections of pipe.

The drilling procedure can be done in two ways, namely vertical and directional drilling. The vertical procedure basically concerns drilling straight down until the gas deposit and fracturing the rock around 100-feet along the hole.

The directional gas well drilling, also called horizontal, deviated or slant drilling, is a more complex procedure that has been expanded in recent years by gas producers. This procedure involves the use of modern technology to turn up to a 90 degree transitional curve in order to reach a reservoir that can not be reached directly beneath the well-head. The aim is to drill in different directions from the same well site in order to expand the search area thousands of feet down and thus achieving access to more of the shale than a traditional vertical well could provide. This is a significant improvement for surface owners.

A directional drilling may have several variants. For instance, in order to reach a deposit bearing formation that may be hundreds or even thousands of feet away horizontally from the wellhead at the surface, the driller might slant the drilling bit at a slight angle off vertical from the beginning. An other variant may be to drill down vertically until the top of a shale layer is reached thousands feet below the earth's surface and then, to bend the well bore sideways across the shale layer.

Due to the compactness of the shale rock, the trapped gas frequently finds difficult to seep into the hole towards the surface; hence, fracturing techniques have to be applied by gas producers. The drillers typically fracture the rock by adding a water and sand mixture into the hole at very high pressure that would lead to open millions of tiny cracks in the rock, enabling the gas to seep out through the pipe.

The disadvantages of this drilling procedure are that it requires a bigger drilling rig, a bigger drilling pit, and a temporary pond to impound up to 1 million gallons of water for the fracturing process, leading to the concern of the pollution of the groundwater or damage to other underground resources given that 40% to 60% of the water that flows back to the surface would also contain the chemicals added to it by the driller.

2.2.2.3 Processing stage

Once the drilling stage is successfully completed, the well goes into production. This involves bringing the gas up to the surface from the subterranean deposit by, e.g., using a decompression process called *retrograde condensation* [78]. Subsequently, the gas is transported as a raw material to a collecting point by means of a gathering pipeline system (see Section 2.2.3.1).

As mentioned before, a gathering pipeline system is typically connected to all adjacent wells. The goal is to take the raw gas extracted from all these underground sources to the next step: *a processing stage*. This is accomplished by pumping the gas directly to a processing plant before using it as a clean, safe fuel. Note that whenever it is needed, the gas may be pumped into an underground facility for future processing and use.

Natural gas processing plants, also referred to as fractionators, are used to remove the impurities or contaminants, such as CO_2, H_2O, H_2S, He, mercury, among others, found in the raw natural gas (see Section 2.1).

Conducting this processing phase basically possesses one particular goal: to satisfy the quality standards imposed by major transportation pipelines, including transmission and distribution companies. These restrictions usually vary with respect to the current markets that the pipelines serve. Furthermore, transporting extra heavy material might reduce significantly the pipeline capacity.

On the other hand, the quality constraints imposed on natural gas also take into consideration the preventive measures on the pipeline system design, i.e., they are forced to preserve the lifespan of the pipelines. For instance, particulate matter (PM) or liquid water, which are typically found in raw natural gas, may cause erosion, corrosion or other damage to the pipeline. Moreover, when natural gas is not sufficiently dehydrated, the formation of methane hydrates do occur, causing a severe damage to both the processing plant, and gas transmission pipeline.

At the processing plant, the raw natural gas is submitted to various treatment processes which depend upon gas composition, quality requirements specified by the clients, means of transportation being used, among others. The most common processes that produce what is known as 'pipeline quality' dry natural gas, can be briefly depicted as follows.

First, acid gases (H_2S and CO_2) are removed by *an amine treating*, also known as *gas sweetening*, or by *a polymeric membrane treating*, a newer technology that has gained increasing acceptance. Note that these acid gases can also be submitted to *a sulfinol process* for future use as by-products. Natural gas is then stripped of all of the extra moisture when going through *a glycol unit* and *PSA unit*, i.e., *a dehydration process*. Next, by filtering the gas through *activated carbon*, and later submitting it to *an absorption process*, mercury and nitrogen are thus removed. Subsequently, the application of *a cryogenic distillation* removes the associated heavier hydrocarbons, known as 'natural gas liquids' (NGLs). NGLs, which are also put separately in the market as valuable by-products, include ethane, propane, butane and pentane. Finally, since raw natural gas is odorless, a distinctive odorant (mercaptans) that smells like strong sulfur is added for security reasons prior to distribution.

2.2.3 Gas transportation

Natural gas transportation is a fundamental activity conducted by the gas industry that basically consists of moving gas from one location to another by any appropriate means, including pipeline systems.

Over the years, a number of factors such as the advent of metallurgical improvements and welding techniques, or the exponential increase of pipeline networks all over the world, among others, have made the gas transportation process more economically attractive. In consequence, a wide area of opportunities for increasing operating profit margins has arose.

While various means might be applied to transport natural gas, it is well-known that pipelines represent the most economical means to transport large quantities.

Pipelines can be used either offshore or onshore, with a remarkable difference in terms of security and construction prices. Building up pipeline systems under the sea is highly costly and technically demanding, a lot more than onshore. For example, according to *Gazprom*[1] [32], while the costs of construction of the onshore pipeline system on Russian and German territory is around € 6 billion, *the 1220 km long (41 in) Nord Stream*[2] *pipeline* [101] is expected to cost around € 8.8 billion. Hence, when financial, political or environmental issues arise, gas transportation operators look for different alternatives to perform this task. For example, by means of tanker ships, natural gas can be transported as LNG or MLG *(medium conditioned liquefied gas,* see [14]); or as CNG by way of flatboats.

In the present work, we only consider the optimization of natural gas transportation by means of pipeline systems.

Three major types of pipeline systems can be classified depending on their purpose, namely gathering, transmission and distribution systems. *Gathering pipeline systems* collect raw natural gas extracted from production wells in order to subsequently bring it to processing plants. *Transmission pipeline systems* bring

[1] *OAO Gazprom* (Open Joint Stock Company) is the Russian state-owned energy monopoly established as the largest extractor of natural gas in the world.

[2] *Nord Stream* is a gas submarine pipeline to link Russia and the European Union via the Baltic Sea.

the gas thousands of miles across the world from the sources toward the market. *Distribution pipeline systems* are those found in a large number of communities, which distribute the gas to residential homes and business sectors.

The main differences among these systems are the physical properties of the pipe used, such as diameter, stiffness, material, etc., and the specifications of the maximum and minimum upstream and downstream pressures. A more detailed description of each one of these systems is presented next.

2.2.3.1 Gathering systems

A gathering system (see Fig. 2.8) is a network of interconnected pipelines with the main goal of transporting large quantities of natural gas from wellheads to processing or storage facilities. At the processing plant, NGLs are separated from NG, whereas at the storage facility the gas is kept for future processing and use. Gas gathering lines also serve to transport natural gas from the production or storage facilities to major transportation lines for its final distribution.

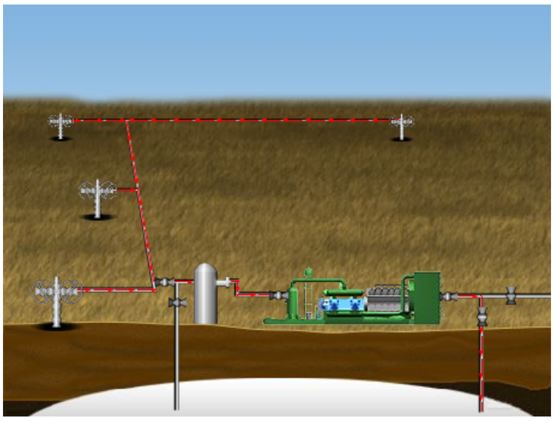


FIGURE 2.8: A gathering pipeline system – *4 wellheads and a preprocessing plant*

These systems consist of pipelines with small diameter and a couple of hundred meter length. They are typically buried about $1m$ and protected by special coatings. Several methods, including *a cathodic protection system*[61] (CPS), are commonly applied for controlling corrosion in the pipes. A CPS refers to the application of enough electric current to the pipe from some outside source in order to make it work as a cathode of an electrochemical cell, while the exterior special coating works as an anode.

A complex gathering system may consist of thousands of miles of pipes, connecting more than 100 wells located in a specific area. Submarine pipelines, which collect gas from the deep water production platforms, are considered as one of the more complex gathering systems.

2.2.3.2 Transmission systems

Natural gas transmission systems are composed of a number of different devices, such as valves, regulators, pipelines, compressors, pressure gauges, junction nodes, among others.

These systems are characterized by long pipelines with large diameter. Larger amounts of natural gas are typically transported by means of compressor stations installed at strategic points, from the production and storage facilities to market regions, including large-volume customers, specific storage facilities, and distribution centers. Note that a large-volume customer may receive similar volumes of gas as a distribution center. This category includes factories, power plants, petrochemical facilities, and institutional users.

Due to the continual expansion that these systems have exposed during the last decades, coupled with the hunger for energy posed by industrial and public sectors around the world, a large number of complex real-life situations related to the gas transportation process have challenged significantly the research in optimization for many years. Hence, they are the main focus on the research projects presented in this thesis, which are thoroughly discussed in the next chapters.

2.2.3.3 Distribution systems

The function of a gas distribution system is to locally distribute natural gas among final users for its consumption as a primary source of energy. These systems are divided into *main and service lines*. The main lines are buried under streets and highways; they are usually between 2-16 inches in diameter and carry the gas from the city gates to the service line. The service lines carry the gas from the main lines directly to residential, commercial and small industrial customers. These lines are usually 0.5-1.5 inches in diameter and its pressure is equal to the pressure of the main distribution line to which it is connected.

These systems differ from a transmission system in a number of ways. For example, they consist of interconnected pipelines which are smaller in diameter and length than those found in a transmission line. As opposed to the latter, plastic or cast iron pipes are more frequently used rather than steel pipes. Furthermore, they are typically located downstream of a main transmission line, and represent simpler systems while having no valves, compressors or nozzles.

According to the pipeline safety regulations, a gas distribution line is different from a gathering or transmission line, and works at low pressure, i.e., its downstream pressure is normally controlled at 2-15 bar (30-218 psi) –in older systems the minimum pressure can be as low as 10mbar (0.014 psi).

Wu et al. in [145] present a mathematical model to find optimal characteristics of pipelines when designing a distribution network. They propose an approach based on *the primal-relaxed duel decomposition method* by Visweswaran and Floudas [136].

Gas business establishments, also referred to as *local distribution companies* (LDCs), are state-jurisdictional entities responsible for managing the distribution lines, including billing and metering services, as well as the purchase and sale of natural gas under demand uncertainty conditions and price rate variability (see [25] for a better survey).

When a new gas service connection is required by a residential or commercial customer, LDCs use a well-known technique called *hot tapping*[75] (also referred to as *pressure tapping*). This technique basically refers to the installation of a new line to a pressurized system that is currently in full operation, i.e. a new branch connection from the existing piping or vessels is done online with no interruption of service.

2.2.4 Segments and components of a gas pipeline system

The gas transportation networks, either gathering, transmission or distribution lines are composed of more than just pipes, which makes them a lot more complex regarding the mathematical modeling. A wide variety of facilities and pieces of equipment operate together to let the gas flow through the system, which reflect the various needs for transporting natural gas.

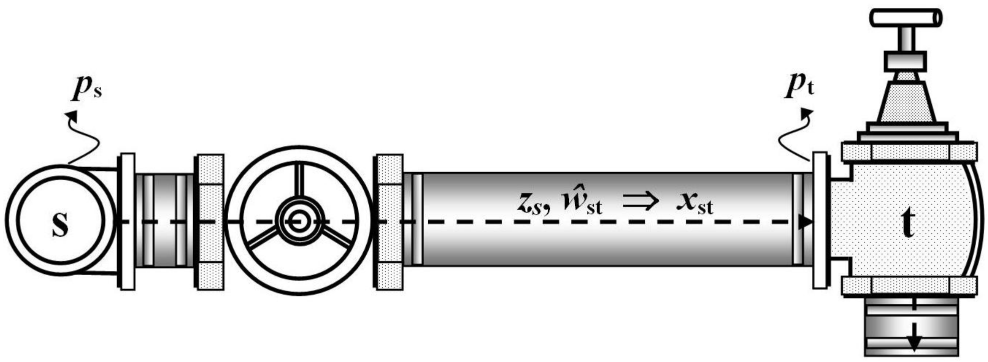


Figure 2.9: Components in a natural gas transmission pipeline – *Natural gas pipeline*

Fig. 2.9 shows the most common components of a pipeline segment, such as the supply point (s) and the sink point (t). At these points we can obtain information of the transmission process, including the upstream pressure of p_s, the downstream pressure of p_t, and the gas flow (x_{st} through the pipeline segment. Furthermore, valves can also be found in pipelines. They act as dams to allow the gas flow through the pipeline. Note that since the elevation difference between the upstream and downstream points is assumed to be zero, the pipeline segments are considered to be horizontal in this work.

2.2.4.1 Monitoring and control facilities

A number of specific facilities for monitoring, management and regulation of natural gas are required in a pipeline system. For example, *the initial injection station*, also known as *the supply* or *inlet station*, is the point at which the system is fed with gas to proceed with the transportation. Processing and storage facilities, as well as main compressor stations are commonly included in this category.

Partial or intermediate delivery stations are facilities where the system operator delivers part of gas being transported.

Final delivery stations, also known as *outlet stations* or *terminals*, are the sink points of natural gas. These stations are basically the connections with, e.g., distribution centers, storage facilities, large-volume or resale customers.

Block valve stations are required to keep in isolation any segment of the network for security, operational or maintenance reasons. These stations are located every certain distance in the network depending on the operational conditions of the system, and the nature of the trajectory of the pipeline. For example, in a transmission system, these stations are installed every 20-30 miles.

Regulator stations are those facilities where the downstream pressure is controlled by the gas pipeline operator. Lower and upper bounds of the inlet and outlet pressures are the essential parameters for the design of a regulator station [65].

A *pigging facility* can also be found along the gas transportation system. These facilities perform certain functions by means of *inspection gauges pigs*, also known as *scrapers* or *Go-devils*. These devices are sent into the pipeline to clean the inside and monitor any rupture, leakage or anomaly that may exist.

City gates, also called *take stations*, are facilities where transmission lines are connected to distribution lines. At these points the downstream pressure is reduced to match the pressure requirements for the distribution line while a sour-smelling odorant is added to the gas for safety reasons to allow for its distribution.

Finally, in order to have access to remote operation of the gas system, a communication center, known as the *Main Control Room* is also required. This center is connected remotely with a large number of field devices, such as flow, pressure and temperature gauges/transmitters, which are installed on specific locations. All data measured by these devices are gathered in a local *Remote*

Terminal Unit (RTU) and transferred in real time to the communication center via satellite channels, microwave links, or any other remote connection. In this center, a computer system known as SCADA *(Supervisory Control And Data Acquisition)* is used to monitor and control all processes. The SCADA system is a human machine interface that allows the operator to monitor the hydraulic conditions of the pipeline and execute commands such as open or close valves, turn on/off compressors, etc.

Compressor stations are essential facilities which are installed at strategic points in the network. The main aim of these components is to provide the propelled force or boost to keep the gas in motion through the pipeline system.

2.2.4.2 Pipelines

Pipelines play a key role in the day-to-day activities while making possible the delivery of available fuels, including natural gas. They are used in a wide range of transportation services. An interesting use of the pipes can be found in, e.g., pubs in the *Veltins-Arena*, a football stadium in the German city of Gelsenkirchen, where several beer distribution lines are interconnected by a 5 km long pipeline system. The aim of these distribution lines is to some extent overcome the demand variability during various stages of a football game. However, the major and most important use of pipelines is undoubtedly the energy transportation, such as oil and natural gas.

In industry and engineering discipline, a pipe is a round-stiff tubular section of the gas system that is made of carbon steel or plastic in function of the inner, outer or nominal diameter and the wall thickness. These measures are imposed by applicable industrial standards, such as ASME/ANSI B36.10/B36.19. The size of a pipe is based on its function and may vary from around 5 cm (2 in) to over 150 cm (60 in) in diameter.

Gas pipelines are usually buried underground about 1-2 meters (3-6 ft) in lands or *rights-of-way* acquired by, or granted to the pipeline company. Whenever burying the pipe becomes less convenient, the strategy is to place the pipeline 5-6 feet above the ground (under strict specifications to withstand environmental conditions) in order to allow for wildlife or any other factor that might damage the pipe.

2.2.4.3 Compressor Stations

Compressor stations, also referred to as booster stations, are facilities used in all gas transportation processes, including extraction, re-injection, gathering, transmission and distribution activities, as well as volume reduction processes for shipment or storage [93]. They are usually divided into multiple compressor units, operating in series or in parallel.

A compressor station primarily serves the two full purposes: 1) it provides the pressure and energy necessary to keep the gas moving through the transportation line; and 2) it reduces the size of the NG molecules by many times, thus increasing the amount of NG that can be transported in a given size pipeline.

Various types of compressor units can be found in the gas industry, including *positive displacement devices* (see Figs. 2.10–2.11), such as reciprocating and rotary screw compressors, and *dynamic devices* (see Fig. 2.12), such as centrifugal and axial compressors.

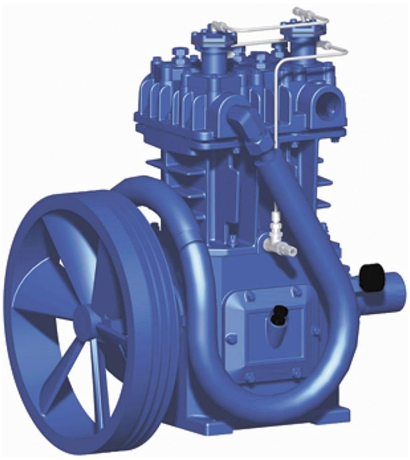


FIGURE 2.10: Reciprocating compressor for natural gas – *A positive displacement machine*

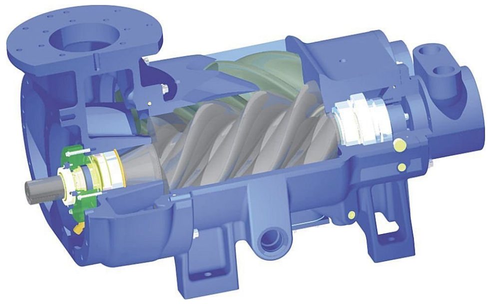


FIGURE 2.11: A rotary screw natural gas compressor– *A positive displacement machine*

Figs. 2.10–2.11 show a heavy-duty cast iron reciprocating compressor and a rotary screw compressor, respectively, which are engineered exclusively for natural gas applications.

In transportation pipeline systems, *reciprocating* and *centrifugal compressors* are the most widely used. The former is a positive displacement machine that increases the gas pressure by confinement within a closed space; here, the compressing element is a piston having a reciprocating motion in a cylinder. The latter is a compressor that uses a rotating disk or impeller (see Fig. 2.13) to force the gas to the inlet of the impeller in order to increase its velocity, and a diffuser to convert this velocity into pressure energy. (A better survey can be found in, e.g. [86] and [93]).

Two clear examples where compressor stations are required can briefly be described as follows. During the transportation process, energy and upstream

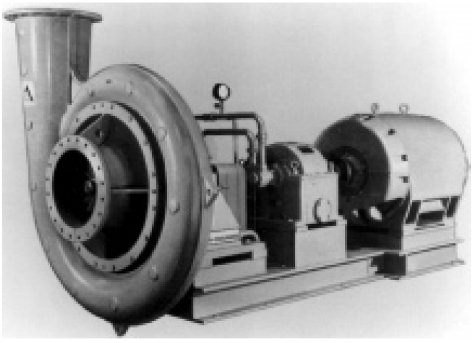


FIGURE 2.12: A single-stage centrifugal compressor driven by electric motor – *Photo: Dept. of Energy, U.S. Government (public domain)*

FIGURE 2.13: Centrifugal compressor impeller inside a wedge diffuser – *Photo: NASA, U.S. Government (public domain)*

pressure are dramatically reduced due to a number of factors such as the inherent resistance encountered when the gas is transported through confined spaces, such as pipelines, as well as heat transfer between the pipe and its environment. Thus, compression of the NG molecules is periodically required in order to keep the gas in motion over long distances; this is accomplished by means of mechanical work produced by compressor units installed along the transportation line. Nonetheless, the use of these compressors, which can be powered by either electric motors, steam turbines or gas fired engines, turns out in high associated costs that defy the gas transportation industry on a daily basis. Hence, the optimal operating setting of such devices is a challenging task that is addressed in Chapter 4 of the current work.

Moreover, gas producers typically strive to transport larger amounts of gas through a given pipeline system in order to satisfy client demands. However, gas operators have to face a number of unpredictable or scheduled events that may arise in the network, including demand uncertainty, load fluctuations, or anomalies/shortcomings. Hence, a strategy to overcome such difficulties, gas operators may use compressor stations to build up short-term line-packs in pipelines; this is accomplished by reducing the gas volume by operating at higher pressure. Chapter 6 of the current work presents a more detailed study of this subject.

2.2.4.4 Gate Settings

In addition to compressor stations, gate settings are installed periodically along the transmission pipeline system to help control the flow of the gas.

Gate settings are locations where the flow of gas can be stopped by closing a valve. Gate settings are generally installed about every 10 miles and function to help isolate a particular section of the pipeline. Generally gate sites are in fences or are located in locked boxes to prevent unauthorized operation of the valves. When multiple lines are located parallel to each other in the same rights-of-way, crossover piping that connects all of the pipelines together is built at the gate sites. This allows diversion of the flow of the natural gas between the adjacent pipelines. The flexibility to divert gas from one pipeline to an adjacent pipeline is often extremely useful and allows one pipeline to be isolated and worked on while gas continues to flow down the parallel pipeline system.

Since most parts of a natural gas transmission pipeline system are buried, it generally takes a special kind of agreement to install gate settings since they have above ground components. These agreements may require permission for permanent above ground gate setting facilities or may even require the specific purchase of real property.

2.2.4.5 Rights-of-Way Corridors

The corridors where natural gas transmission pipelines are installed and buried are commonly known as *rights-of-ways*. This terminology has to do with the fact that the pipeline owner has to obtain permissions from affected landowners that give the operator the right to construct, operate, and maintain the pipeline for its entire length. The specific agreements are not always all-inclusive and the individual pipeline companies often have to work with the current landowners when extensive maintenance activities are required.

Since pipelines have a long life span, generally many decades long, the original owner or owners of the land who gave permission to build the pipeline in its current location may not be the present owners of the land. In the life of a pipeline, the landowners may change numerous times. Keeping up with who the current owners are is a major task for the pipeline operator.

2.2.4.6 Valves & Regulators

Valves and regulators are typical components in a pipeline system. They are installed for operational and security reasons. For example, by closing or opening a valve, gas operators may restrict or direct the gas flow from one point to another in order to optimally satisfy clients' demand or to prevent loss of fluid when, e.g., a mal-function of the pipe is observed. The shutting down of flow through pipe sections is also required when conducting scheduled routine maintenance. In addition, natural gas pressure reduction is typically achieved using pressure reduction throttling valves.

In real-practice, regulatory requirements [125] oblige to pipeline industry to install mainline block valves at certain fixed spacing on large pipelines transporting natural gas.

Valves are constructed of steel while following both the specifications given by standards of *American Petroleum Institute* (API), *American National Standards Institute* (ANSI) and *American Gas Association* (AGA) [125], and the purchaser's requirements regarding the type of material and operating conditions required.

Chapter 3

Optimization – *The Science of Decision Making*

For several decades, both mathematical models and optimization algorithms have played a key role in tackling a considerable number of optimization problems in the day-to-day activities. Continuous and discrete models, as well as exact and heuristic methods have been developed with the promise of substantially improving the performance of a company or institution (be it profit/lucrative or not).

The natural gas industry is no exception. Due to issues such as ecological aspects, security and financial reasons, a sizeable number of mathematical models have been developed and thoroughly studied in order to solve many of the processes involved in the extraction, purification, transportation, distribution, consumption, sale, etc. of natural gas. Thousands of millions of dollars per year might be saved by optimizing either the infrastructure of the gas industry and its transmission systems, or the plans, schedules and operations of its day-to-day procedures.

Given that in this thesis, mathematical formulations involving *non-linear programming* (NLP) and *mixed-integer non-linear programming* (MINLP) models are proposed to solve three important challenges involved in the natural gas transportation network systems, an overview of the research field, as well as a brief description of the major challenges of optimizing gas pipeline systems are presented in subsequent sections.

3.1 An overview of the field of study

Operations Research (OR), as specified by Gass and Assad in [51], is neither a natural science nor a social science, but a *decision making science*. More precisely, OR can be defined as the application of advanced scientific solution methods to enhance or strengthen the efficiency and performance of a company, including its operations, equipment, manpower or policies. This is accomplished by conducting a number of challenging tasks: 1) (clear/full) comprehension and definition of the problem, 2) gathering and analysis of data, 3) development and assessment of a mathematical model, 4) design of solution algorithms, 5) interpretation and validation of the outcome, and 6) implementation of improvement actions. Note that it may be required to repeat some of these steps several times before achieving the desired results, i.e., those results that are meaningful and could be analyzed and implemented.

Note that OR differs from Management Science (MS - also referred to as Operations Management) [48] at a conceptual level, given that MS focuses on managing of production resources to enhance the competitiveness of a company or organization in a managerially oriented way, whereas OR is mainly a mathematically oriented technique involving a problem and finding an optimal solution for it.

Optimization, also referred to as *mathematical programming*, formally encompasses several stages described by OR, namely (a) understanding and formulation of the problem; (b) design, implementation and evaluation of solution methodologies; and (c) validation and interpretation of the outcome. It can be considered as a scientific research field that creates a narrow link between *computer science* and *applied mathematics*, i.e. its major focus is placed on the study of the theory of mathematical modeling and algorithmic design.

Nowadays, due to its profitability when applied to a wide variety of real-life situations, optimization is seen as a usual practice in sciences and business; however, it took almost a century to reach this point. While the foundations of OR can be traced several centuries back (approx. c. XVII-XIX) with those valuable scientific contributions made by, e.g. Newton, Leibnitz, Lagrange, Bernoulli and Euler, the formal introduction of optimization took place after the World War II in the twentieth century with the works of Kantorovich[1] and Dantzig[2] in Russia and USA, respectively, on the theory of *linear programming* (LP).

[1] *Leonid V. Kantorovich* (1912-1986), a Russian mathematician and economist, shared the Nobel Prize for economics in 1975 with Dutchman T. Koopmans based on the development of the mathematical theory of LP in 1939.

[2] *George B. Dantzig* (1914-2005) is called *the father of linear programming* and received the National Medal of Science in 1975.

A few decades later, along with the revolution of the computer systems in the 70s, its application was widely spread throughout the world, causing a significant change in the practicality and availability of modeling languages, software, hardware, algorithms and mathematical models. Many decision-making problems were thereby modeled and solved using optimization methods [94]. (An excellent collection of major contributions and more important events along the development of OR from the 16th to the 21st centuries can be found in the book entitled *"An Annotated Timeline of Operations Research: An Informal History"* by Gass and Assad in [51]).

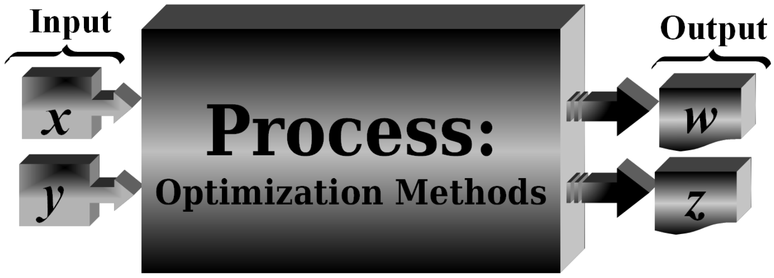


FIGURE 3.1: Abstract representation of a system – *Main parts: Input, process, output*

In agreement with OR, the aim of this scientific research area is to achieve the optimal behavior of a system by means of a thorough study and efficient exploitation of its abstract-mathematical representation. Note that evidently there exists a wide range of ideological definitions for a system based on the study area. In optimization, the notion of a system can be expressed as an entity composed of three main interacting parts capable to be analyzed both quantitatively and qualitatively, namely *input, process,* and *output*. Fig. 3.1 shows this system and the consequent relationship among its elements.

More formally, an optimization problem in its simplest form refers to the theory of minimizing or maximizing an objective function, i.e., a real-valued function of several variables. The domain of the objective may be subject to a wide range of (equality or inequality) constraints, which define the set of feasible choices. Thus, an optimization technique solves the problem by systematically selecting a solution that is within the feasible set and improves the objective. Improvement in this context refers to the assessment of productivity and quality, i.e., *being efficient when using the available resources and being effective when achieving the outcome.*

The theory and techniques required to address the wide variety of optimization models comprises a quite large area that it may fill several volumes (exceeding the expectations of this work). Nevertheless, in order to bound the research frame

Table 3.1: Main components of an optimization model.

Objective Function	Quantitative component of the problem that we want to maximize or minimize.
Decision Variables	Elements of the problem that can be manipulated (adjusted) by the optimizer to achieve a given objective.
Constraints	Equalities or inequalities relating variables and parameters of the problem that have to be satisfied while looking for the best performance in the objective. (Note that there may exist implicit and explicit constraints.)
Parameters	Input data that provide the values required in, e.g., the coefficients of the decision variables or the right hand side of the constraints.

while discussing the mathematical models and optimization techniques in subsequent chapters, some insights into the basics of the mathematical programming are provided next. (A detailed survey can be found in, e.g. [94]).

3.1.1 Mathematical Models

A mathematical model, also referred to as *optimization model*, is an abstract representation of a decision making problem that may arise in the real world and that we attempt to solve.

Readers familiar with the area of optimization may easily recall several classes of optimization models. For example, we may mention unconstrained or constrained optimization models, continuous or integer (discrete) programming models, deterministic or stochastic models, linear or non-linear models, convex or non-convex models, among others. Such classes, besides providing a clearer notion of the complexity of the model being faced, also indicate the more suitable research branch to follow.

A mathematical model can be placed into a particular class by considering two major aspects: (a) The characterization of its main components (see Table 3.1), and (b) the search space, also referred to as *the feasible region*, defined by, e.g., the set of (equality or inequality) constraints. For example, the simplest case among all optimization models is the *linear programming* (LP) model. The components of an LP model share a specific characteristic, i.e., both objective function and constraints are linear. The canonical form of an LP model can be put in the following way:

$$(LP) : \min f(x) = \{c^T x | Ax \geq b, x \geq 0\} \tag{3.1}$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is the real-valued objective function, i.e., the mathematical expression indicating what it needs to be improved (here in terms of minimizing its value); $\mathbb{R}$ denotes the set of real numbers; $x = (x_1, \cdots, x_n)^T$ is a nonnegative real vector representing the decision variables of the problem, i.e., the components that must be manipulated to achieve the objective f; c and b are real vectors of known coefficients; and A is a known matrix of coefficients.

Let $\Omega = \{x | Ax \geq b, x \geq 0\}$ be the feasible region (domain) for the LP formulation. Then, a vector x^* is an optimal solution for (3.1), if it has the smallest objective value among all vectors x in Ω, i.e.,

$$x^* \in argmin\{f(x) | x \in \Omega\}.$$

Several well-designed techniques for LP have been implemented since the development of the *simplex method* made by George B. Dantzig [27] in 1947. Although the simplex method shows a good practical performance, it still provides an exponential worst-case complexity as shown by Klee and Minty in [74]. Two decades later, Narendra Karmarkar proposed the well-known *interior point method* [72] to tackle, for both average and worst case complexity, LPs in polynomial time. Mehrotra, motivated by Karmarkar's method, in turn, proposed the *predictor-corrector method* in 1992 (the best in practice) for large-scale problems.

When certain applications require integrality of variables, or nonlinearities in relationships, or uncertainty in the data, LP formulations are not suitable and other more sophisticated models may be required. Nonetheless, these other models as well as their solution methodologies are heavily influenced by the structure, theory, and algorithms aligned with LP.

For example, a more difficult model is presented by a mixed-integer program (MIP), which can be seen as an extension of the LP model. *Branch-and-cut* algorithms have been successfully proposed to solve MIPs, which are based on LP relaxation. A MIP model can be put in the following form:

$$(MIP) : \min f(x, y) = c^T x + h^T y \tag{3.2}$$

s.t.:

$$Ax + Gy \leq b \tag{3.3}$$

$$x \in \mathbb{R}^n_+, y \in \mathbb{Z}^p \tag{3.4}$$

where $\mathbb{Z}$ denotes the set of all integers; $c \in \mathbb{R}^n$, $h \in \mathbb{R}^p$, and $b \in \mathbb{R}^m$ are given real vectors; and $A \in \mathbb{R}^{m \times n}$ and $G \in \mathbb{R}^{m \times p}$ are matrices of known coefficients.

This problem is called 'mixed' because it contains (non-negative) integer variables y as well as continuous variables x, although it still keeps the linearity. If there are no continuous variables, i.e. $n = 0$, then it is called a (pure) *Integer Program* (IP). A special case of an IP is the so-called *binary program*, where the variables are restricted to be either 0 or 1. If $p = 0$, i.e., there are no integer variables, IP becomes identical to the LP formulation introduced above. An excellent reference on the subject is, e.g., the book of *Integer Programming* [140] by Wolsey.

When either the objective function, the constraints or both contain a non-linear term, then we refer to the model as a *non-linear program* (NLP). The book of *Numerical Optimization* [100] by Nocedal and Wright offers an extensive and well-detailed basis in the state-of-the-art of *non-linear optimization*.

Regarding the MIP introduced above, a pair $(x, y) \in \mathbb{R}^n \times \mathbb{Z}^p$ is said to be a *feasible solution*, if it satisfies all constraints. Further on, (x^*, y^*) is called optimal, if $c^T x^* + h^T y^* \leq c^T x + h^T y$ for all pairs of feasible solutions (x, y).

We can also find optimization problems where several objective functions may be in conflict subject to certain constraints. For instance, in the well-known *traveling salesman problem* (TSP), an NP-hard problem in combinatorial optimization, we may need to minimize the time spent when visiting all cities exactly once, while it is simultaneously required to minimize the total fuel consumed. Here, a trade off between time and fuel would not be suitable, i.e., 10 minutes vs. 100 liters, or 8 minutes vs. 113 liters do not indicate anything since both reflect the same importance. Another example could be the situation in which a company may be interested in maximizing its profits from a production process and at the same time minimize the periods involved with the end-products. These kinds of problems encompass the so-called *multi-objective optimization*, also known as *multi-criteria* or *multi-attribute* optimization, where many solutions are possible. The *Pareto-front* constitutes all no-dominated solutions, i.e., they comprise the set of optimal solutions. The theoretical basics of multi-objective optimization can be found in, e.g. [39].

The models described above share a specific property, namely their deterministic nature, i.e., all coefficients are known with certainty. When the values of the parameters are given within uncertainty ranges, an approach called *robust optimization* is then applied with the promise of accounting for uncertainty in the input data. When such coefficients can assume probabilistic values, i.e., input data are governed or estimated by probability distributions, we must refer to another classification known as *stochastic programming* to achieve some decision policy affected by uncertainty. (The reader is referred to [124] for a survey on theoretical foundations of stochastic programs.)

3.2 Solution methods for gas pipeline systems

The resolution methods for a gas pipeline system typically focus on finding an optimum operating plan, which include ideal pressure settings and mass flow rate values through the pipeline network. This is a very challenging task since a large number of complicating (inequality and equality) constraints may be imposed in a typical natural gas transportation problem. Hence, two major approaches are commonly used to deal with these problems, namely (a) *numerical simulation* and (b) *mathematical optimization*.

3.2.1 Numerical simulation

The aim of a numerical simulation is to determine the behavior of a system under given conditions; this is typically accomplished by conducting a large number of runs. However, this approach has severe limitations.

First of all, the problem must be discretized in order to be capable to apply this kind of approaches. Depending on how refined the discretization is, the number of alternatives may become astronomical. Hence, only a tiny number of alternatives can be covered by the simulations, and there is no way to know if an optimum solution was included in the trials, i.e. by no way optimality can be guaranteed. Furthermore, the solution requires the interpretation of an expert user who is familiar with the network to confirm its applicability; this means that different users may draw different conclusions on the solution for the same problem.

The *trial and error procedure* based on pipeline simulation is a typical method found in this category for reducing operating costs. This procedure basically consists of three main steps: a) collect data from a centralized control system, such as SCADA; b) load all current data into a steady-state model; and c) set different configurations to the model until a significant improvement can no longer be observable. Thus, since an exhaustive comparison of all possible operating conditions on all possible scenarios is not practically feasible, finding the best setting that satisfies the steady-state simulation model becomes a very challenging task, as well as time consuming. There is simply not enough time to try the astronomical number of possible variable settings. Additionally, there is a limited number of simulated cases where a trial and error approach can be applied.

Clearly a more intelligent rigorous method is required for solving complex pipeline optimization problems. Hence, more sophisticated approaches based on mathematical optimization are presented next.

3.2.2 Mathematical Optimization

As long as a mathematical model is well-defined and its feasible region is known, a mathematical optimization method may be applied to find the solutions satisfying the model.

An optimization approach differs from a numerical simulation in many ways. First, in contrast to a numerical simulation, an optimization method finds the best set of control variables within a single optimization run, possibly executed by less experienced users [145]. Second, the search procedure must be substituted by a more sophisticated algorithm. Moreover, an optimization method usually works with simplified models, but it may yield optimum results for certain targets, subject to a number of constraints, if they all are defined *a priori*.

In the broadest sense, these methods can be classified as local or global search methods. A global solution method seeks to find the global optimum by, e.g., shrinking the gap between the lower and upper bounds of the problem (see e.g. [42, 43] and [59]). These methods may guarantee global optimality or can provide a measurement of relative error that proves global convergence.

The aim of a local solution method, however, is to achieve a local optimum; this is accomplished by focusing on a neighborhood that has been generated by a starting point. Typically, these methods are aided by additional techniques to generate multiple starting points in order to extend the search, thus approaching to the global optimum; yet there is no guarantee of that. Among these methods, the most widely used procedures are those based on the gradient, also known as *hill-climbing* procedures. Gradient-based methods basically consist of a sequence of steps in the negative gradient direction given by the partial derivative of the objective function. The search procedure in these methods continues until an inflection point is achieved, i.e., a point where an improvement in the solution can no longer be possible.

3.2.2.1 Analytical and numerical solutions

An outstanding number of solution techniques has been dramatically exposed with notable success in many branches of mathematical programming, including combinatorial and non-linear optimization. According to Papadimitriou and Steiglitz in [111], these solution methods can be classified in two major groups:

(a) exact methods, also referred to as analytical methods, and (b) approximate methods or numerical solutions, such as *heuristics*.

An analytical method frequently recalls the theory of differential and integral calculus to yield the optimal solution, also denominated as the exact solution. Its application, however, is bounded for some specific problems, whereas it turns out to be very time consuming for large real-world problems.

In contrast, a numerical solution, provided by an approximate method, expresses the behaviour of a system in function of numbers rather than mathematical formulas [104]. Thus, by solving general equations for specific values of parameters and variables, the numerical methods look for high quality solutions within reasonable computation times.

3.2.3 Search space

In the field of mathematical programming, either *constrained* or *unconstrained optimization*, the complexity of a mathematical model representing a real-world situation, is determined by both the objective function and constraints (if any). In the case of constrained optimization, the difficulties are commonly tied to the properties of model as a result of the modeling constraints, e.g., its combinatorial nature. Consequently, the design and application of an optimization algorithm has to face an important challenge, namely the search space, also referred to as *the feasible region*.

The search space is nothing but the set of all possible alternatives that satisfies the model. Note that for some instances the search space can become quite large. Let us consider the 'simple' case where a gas transportation network has to be optimized by simply deciding which compressor units should be on and off to maximize the gas flow through the pipeline system. Hence, the number of choices turns out to be 2^s, where s is the number of compressor units installed in the network. For example, for just 50 compressor units, the possible number of alternatives is 2^{50}, i.e., more than 10^3 trillion combinations, whereas for 200 units the number of possibilities, 2^{200}, increases to an astronomical number of combinations, such that in the hypothetical case that a supercomputer capable to carry out 10^3 billion of calculations per second would have started doing the calculations 15 million years ago, it would still last more than another 15 million years to assess all possible alternatives.

Given that complete enumeration typically turns out to be impossible, the theoretical and practical results of *combinatorial optimization* [76] are required to propose efficient and effective solution methods in order to tackle problems like the one shown above by, e.g., exploiting their mathematical structure. The aim of the branch of combinatorial optimization is to provide an optimal setting of a finite or countable infinite number of discrete entities [111].

In the optimization problems associated with natural gas transportation by means of pipeline systems (as the three projects discussed in this thesis), the resulting feasible region is, by definition, non-convex. Moreover, the size and complexity of the search space while addressing a gas transportation problem rapidly increases when, e.g., upstream and downstream pressures have to be set up for each specific choice of on/off combinations of compressor units. Thus, finding optimal pressure settings that maximize the flow through the network requires the theory and application of *non-linear constraint optimization* as well.

Since no standard optimization solvers are available to effectively cope with problems of this complexity, heuristic (and metaheuristic) approaches have been extensively developed with a notable success during the last decades. These methods, which are conspicuously preferable in practical applications, are briefly discussed in next section.

3.2.4 Heuristic and metaheuristic approaches

In the field of mathematical programming, it is well-known that achieving the global optimal solution to a complex problem usually turns out to be a very hard task, becoming sometimes impossible to fulfill (unless time constraints were no longer applicable). However, in many real-world situations, including some natural gas transportation problems, not only finding the optimal is hard, even the simple specification of a feasible solution can be a very challenging process. Hence, alternative methods are proposed to handle these problems in acceptable running times, namely *heuristic optimization algorithms*.

According to [137], the foundations to problem solving in a heuristic way were first laid in the book: *"How to solve it"* by Pólya [115]. Here, Pólya establishes a theoretical framework for solving mathematical problems, to what he called *'heuristic'*. There, he emphasized that the common procedure to solve a problem was no longer based on a deductive reasoning. Opposite to traditional conventionalism, he showed that problem solving is inductive, i.e., in function of educated conjectures and greedy intuitions that might or might not lead to promising results. (A complete survey of *the mathematical heuristic* of Pólya can be found in, e.g., [118]).

Currently, *heuristic* is a common term used to define a class of intelligent non-traditional strategies that efficiently overcomes the difficulties associated to a real-world decision making problem. Its goal is to, some extent, tackle problems in which an exact solution is not allowed due to time constraints or due to the high level of complexity being faced. In more precise terms, it refers to a number of experience-based steps, or educated guesses to effectively provide a good solution to an optimization problem. 'Good' in this sense refers to a solution that is hoped to be as close as possible to the best alternative, i.e., the optimal solution. However, no guarantee of optimality can be provided by a heuristic method.

The literature on the theory and applications of heuristic methods (see for example, [8], [56], [76], [96], [112] and [117]) offers a great variety of definitions that basically fit with the introductory remarks provided above. For instance, Reeves in [117] provides the following definition:

DEFINITION 1. *A heuristic is a technique which seeks good (i.e., near-optimal) solutions at a reasonable computational cost without being able to guarantee either feasibility or optimality, or even in many cases to state how close to optimality a particular feasible solution is.*

A super class of these methods, often called *modern heuristics* [117], are nowadays the well-known *metaheuristics*. Blum and Roli in [8] describe this kind of methods as a class that tries to combine different (heuristic) methods in higher level frameworks to explore a search space in an efficient and effective way. As stated in [87], *"a metaheuristic is a set of concepts that can be used to define heuristic methods that can be applied to a wide set of different problems. In other words, a metaheuristic can be seen as a general algorithmic framework which can be applied to different optimization problems with relatively few modifications to make them adapted to a specific problem".* Osman and Laporte in [110] provide the next formal definition:

DEFINITION 2. *A meta-heuristic is an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search spaces using learning strategies to structure information in order to find efficiently near-optimal solutions.*

This special class of optimization techniques has been dramatically developed since its inception in the early 1980s. The *Handbook of Metaheuristics* [56] edited by Glover and Kochenberger provides an in-depth survey of the most relevant metaheuristics currently applied to a wide variety of optimization problems. Here, the main concepts, algorithmic ideas, and essential strategic elements for a successful application of metaheuristics are introduced by pioneering and leading researchers in optimization, particularly in the branch of combinatorial optimization [76].

Popular meta-heuristics for combinatorial optimization problems include *tabu search* [57], *genetic algorithms* [62], *simulated annealing* [73], *neural networks* [63], *ant colony optimization*[34], *variable neighborhood search* [90], and *GRASP* [40]. These meta-heuristics may be placed in different categories based on the algorithmic idea used when tackling an optimization problem. A gentle introduction to the most important classifications, including *nature-inspired vs. non-nature inspired, population-based vs. single point search, dynamic vs. static objective function, one vs. various neighborhood structures, and memory usage vs. memory-less methods*, is provided in [8].

Even though the heuristic methods, or the super class (meta-heuristics), may be based on different philosophies, they certainly provide efficient mechanisms to effectively explore a search space with the aim of solving hard optimization problems where traditional methods have failed, or are practically not applicable.

3.2.5 Modeling language systems and optimization tools

Modeling language systems (MLSs) have become usual tools in the development and application of the mathematical models. The aim of these MLSs is to allow the user to build maintainable mathematical models in a usually straightforward manner; this includes the definition of parameters and simple or multidimensional sets, the selection of the most suitable solver and the execution of the model.

Nowadays, a wide range of MLSs can be commercially acquired. However, the most widely used MLSs are AMPL [46] (*A Modeling Language for Mathematical Programming*), optimization platforms from MSF [89] (Microsoft Solver Foundation) and GAMS [49] (*General Algebraic Modeling System*). They are complex systems that require a certain learning curve before building large, complex models in a controlled fashion.

AMPL, for example, is a modeling language created by Fourer et al. [45] and developed at Bell Laboratories in the 1990s. In contrast to other MSLs, AMPL must be incorporated into a system that manages data, models and solutions; otherwise, it can only be employed to specify classes of mathematical programming models.

The modeling language products behind MSF [89] optimization platforms are also used to implement and solve mathematical programming models, including LP, MIP, constraint programming (CSP) and quadratic programming (QP) models. MSF consists of a number of modules that includes an equation based modeling language OML [70], application programming interfaces (API's), external solvers (Cplex, Xpress and Mosek) and an excel based framework.

GAMS, on the other hand, was developed as a research project at the World Bank in the early 1980s. GAMS is a high level modeling system that integrates a modeling parser system, a language compiler and high-performance solvers. It was specifically designed for large, complex problems, including LP, NLP and MINLP models. In addition, GAMS outperforms AMPL and MSF in terms of language syntax and support for sparse matrices when solving large NLP models. In contrast to MSF, which has a CSP solver built-in, GAMS offers very little support for constraint programming.

GAMS solver options comprise a considerable number of optimization tools, including local solvers, such as MINOS [97] (a NLP solver from *Stanford University*), CONOPT [37, 38] (a large scale NLP solver from *ARKI Consulting and Development*), SNOPT [52] (a large scale SQP based NLP solver from *Stanford Business Software, Inc.*); and global solvers, such as BARON [131] (Branch-And-Reduce Optimization Navigator from *The Optimization Firm*), LGO [114] (Lipschitz global optimizer from *Pintér Consulting Services, PCS*) and OQNLP [49] (OptQuest/NLP multi-start solver system by *OptTek Systems/Optimal Methods*). The solvers differ from each other in several ways. For example, in the specific size and characteristics of models they may handle, as well as the different methods they apply for solving models with a given proven optimality.

Similar systems, such as IBM ILOG Concert Technology (CPLEX optimizer), MPL, Xpress-MP, and many others have also contributed to the state of the art of optimization problems (see [98] for a better survey on global and local optimization tools).

Several facts must be taken into account when using MLSs. For instance, to increase the reliability of a *global optimization* (GO) tool, we can consider two major classes based on its strategic idea for solving a problem, namely *deterministic* and *stochastic*. Moreover, we must also be aware that none of the currently available optimization packages is fully reliable for solving all global optimization problems [99]. In contrast to LGO, when applied to deterministic GO models, BARON and OQNLP do require explicit functions to work properly.

Note also that because the solvers integrated in MLSs are mostly 'black-box' software, it requires a good understanding of them in order to set up their most appropriate parameter values, and thus achieving their best performance. As stated by Dolan and Moré in [33], the performance of a particular solver integrated in a MLS may improve significantly if non-default options are given.

On the other hand, we must also realize that some of these high-performance optimization tools may outperform other solvers on particular problems or specific test instances. Neumaier et al. in [99] present a complete survey on the performance of various global solvers, including BARON, LGO, LINGO, OQNLP and

GlobSol, when applied to a set of over 1000 constrained optimization problems. In their study, BARON turned out to be the fastest and most robust among the global solvers tested.

BARON is an implementation of a branch-and-bound (B&B) algorithm where a convex relaxation of the submitted problem is solved in each node of the search tree. Basically, a B&B technique refers to the partition of the feasible region in some fashion and the estimation of the objective value by using upper and lower bounds. Moreover, for the sake of optimality-based range reduction, BARON must depend on MINOS or SNOPT (NLP solvers) to solve the convex subproblems in the search tree.

The aim of BARON is indeed to tackle GO problems. GO relies on the theory of finding the best solution of (constrained) optimization problem which may have multiple local optima. Given a bounded set $\Omega \subset \mathbb{R}^n$, i.e., an explicit constraint set, and a continuous real-valued function $f : \Omega \to \mathbb{R}$, GO aims at finding:

$$\min_{x \in \Omega} f(x), \tag{3.5}$$

where x is a real n-vector representing the decision variables.

In this thesis, due to its excellent performance when applied to various models, BARON is widely used in the numerical experiments conducted on GAMS formulations of NLP and MINLP models. Its application fulfils two primary purposes: (1) For comparison purposes, i.e., as a means of assessment of the performance of the optimization techniques proposed here; and (2) for analysis purposes when evaluating the quality and effectiveness of the mathematical models designed here for tackling specific natural gas transportation system problems.

3.3 Some words on the skepticism of the application

A gentle revision of the literature on optimization problems can easily show that the number of contributions done in this area is quite impressive. This is the result of work done by leading researchers and enthusiastic scientists around the world who have carried out a high-quality research for many years. Nobody can complain about the great effort made into the substantial expansion of the optimization theory during the last decades. Neither can anyone complain about the great advances made in designing sophisticated algorithms in order to deal with optimization problems in a more effective way. However, more than a few might complain of the limited practical implications that such theoretical contributions have posed directly to the society.

In an interview [2] conducted to George Dantzig on August 27^{th}, 1999, Peter Horner (*OR/MS Today* editor) posed an interesting issue: *"In the real world, coming up with an optimal solution is often only half the problem. The other 'difficult part' is convincing some decision-makers to implement the solution"*. Agreeing with Dantzig's answer to such statement, the decision-makers have absolutely right to be skeptical about implementing optimization methods in the real-world. A number of factors have to be thoroughly considered by the decision-makers before conducting such practical task. For instance, even though a numerical method may have been designed to optimally solve the mathematical-abstract representation of a real-life situation, its numerical solution could be implemented effectively in the real world only if: 1) it is exact, and 2) the mathematical design has taken into consideration all possible aspects (constraints) of the optimization problem being faced.

These conditions rarely (if ever) happen in practice because the nature of a mathematical model is typically a simplified idealization of reality. The complexity of a model significantly increases when it is approached to the real world [104]. Hence, a compromise between skepticism and urgency of practical applications becomes fundamentally necessary.

Concerning pipeline optimization, the optimizer recommendations pose a similar effect. Gas operators usually rely on their knowledge and expertise to manually set up flow and pressure parameters that might get the best performance of the system. This procedure, however, reflects a long learning curve and is based on receiving feedback on the daily performance of the whole system from the control room.

Because of the transient nature of pipelines and the absence of real-time measurements, coupled with the implicit limitation to duplicate real system operating conditions, the transition from theory to practice is a very challenging task. Taking into account that the pipeline system can not run identically under different operating conditions, we must rely on the results of the mathematical model as long as the constraints of the system have been properly defined. Hence, despite industry skepticism in implementing optimization methods, simulation and mathematical models are typically used on a day-to-day basis to draw conclusions and make decisions on multi-million dollar pipeline systems, including line pack calculations, flow capacity estimations, and compressor stations throughput.

A final remark: A gas operator who has worked for several years with a transportation line must certainly set up the system in a specific way; thus, the optimal settings given by the optimizer might be at first glance counter-intuitive. Hence, for a successful practical application, it is essential that modelers have open communication with the gas operator.

Chapter 4

Operability on Compressor Stations

Natural gas transportation systems, regardless of the service they may offer (see Section 2.2.3), comply with an intuitive but complex function: to allow natural gas to continuously flow through them from various supply points (e.g., wellheads, processing plants, regasification and storage facilities) towards several discharge points, including the petrochemical and power plants, local distribution centers and short-term storage facilities.

Among all possible challenges associated with the gas transportation process, the one tackled in this chapter is that of the optimal operability on compressor stations such that the fuel consumption cost associated with the compressor units is minimized. The optimization problem is described in detail next.

4.1 The fuel cost minimization problem

The *fuel cost minimization problem* (FCMP) can be stated as follows. In a typical gas transportation process conducted by means of pipeline systems, the initial energy and upstream pressure dramatically decrease due to several physical reasons. For example, this phenomenon occurs due to the friction exerted between the gas and pipes' inner walls, as well as heat exchange between the gas and the surroundings in order to reach a thermal equilibrium, as dictated by *the second law of thermodynamics*[1]. To overcome this loss, the gas operator must turn on the compressor stations, which consist of several identical centrifugal compressor units connected in parallel placed at strategic points along the transmission

[1] *The second law of thermodynamics* basically expresses the universal principle of entropy. This principle states that a system will tend to find a thermal equilibrium with its surroundings, i.e., everything that is close to it, by approaching to a maximum temperature value while an energy dissipation is taking place simultaneously over time.

line. Compressor stations or booster stations, as they are also called, provide the propelled force necessary to maintain the gas in motion through the system, and thus allow the gas to reach its final scheduled destination.

As a result of this inevitable procedure, a significant proportion of the transported gas (estimates vary between 3% and 5%) is consumed by these compressor stations before it arrives at the reception units. This represents high associated costs because the amount of gas transported on a daily basis is huge. Thus, keeping this consumption at a minimum is a task that not only represents large economic value to the industry, but also has an important environmental dimension.

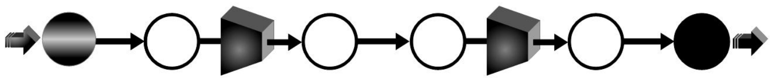


FIGURE 4.1: Linear or gun-barrel network

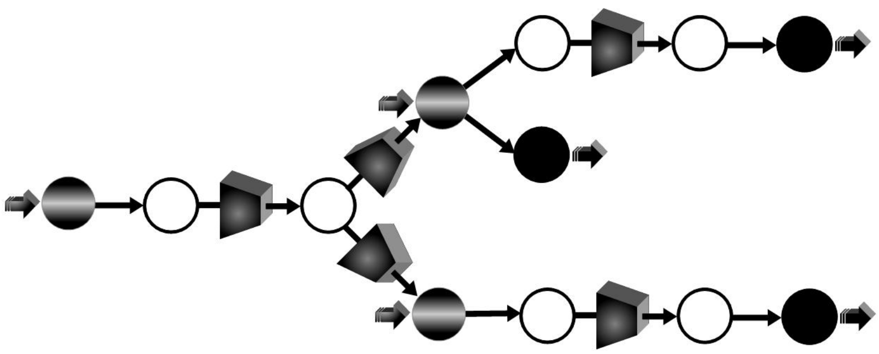


FIGURE 4.2: Tree-shaped network

The state of the art on steady-state flow models reveals two fundamental types of network topologies, namely *non-cyclic* (gun-barrel and tree-shaped, see Figs. 4.1–4.2) and *cyclic* (see Fig. 4.3). This classification is based on the associated compressor network (CN, also referred to as *reduced network*) explained in detail in Section 4.5.2.

In Figs. 4.1–4.3, a grey-gradient node (shown with an incoming arrow) represents a supply node, a black node (shown with an outgoing arrow) represents a discharge node, and a white node is just a transshipment node. A directed arc linking some pair of nodes corresponds to a compressor station if it contains a 3D-trapezoid, otherwise it corresponds to a pipeline.

An extensive literature on the FCMP described above has been published during the last decades. Most of the suggested solution methods have been practically limited to pipeline networks with acyclic structures. In such instances, the

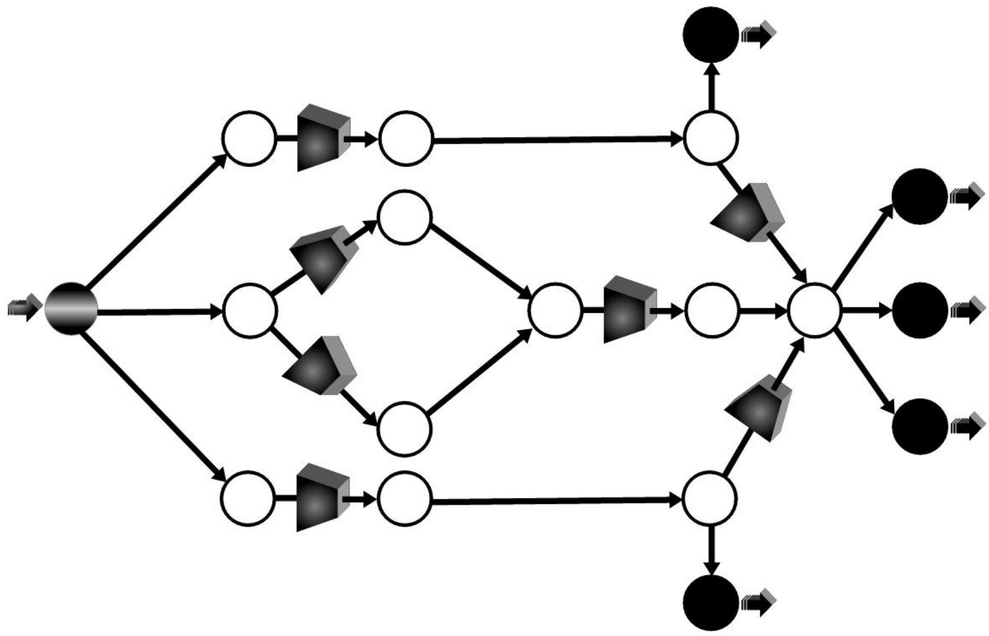


FIGURE 4.3: Cyclic topology

Table 4.1: Fundamental types of network instances of the FCMP.

Topology	**Description based on the *compressor network* (CN)**	**Fig.**
Linear or gun-barrel (Non-cyclic)	This corresponds to a linear arrangement of the compressor arcs, i.e., when *CN* is a single path.	4.1
Tree-shaped (Non-cyclic)	This occurs when the network system contains branches, i.e., when *CN* is a tree.	4.2
Cyclic	This happens when for some pair of nodes in the network there exist more than one path between them, which contain at least one compressor station each, i.e., when *CN* is cyclic.	4.3

suggested methods have shown a considerable success. In some of the more recent works, applications such as numerical simulations, dynamic programming, gradient techniques, among others, have been developed for cyclic networks. However, since these contributions require a certain sparse network structure, their applicability is somewhat restricted. The following section gives a more detailed overview of the most relevant literature.

4.2 Literature review

In this section, a detailed survey of the most relevant works for solving the FCMP or related problems is presented.

4.2.1 Methods based on dynamic programming

Dynamic Programming (DP) is an effective class of algorithmic strategies for solving optimization problems where the decisions can be characterized using Bellman's optimality principle. The technique has been applied to a wide range of optimization problems since its inception in 1957 [7]. Many optimization problems that are NP-complete may be solved efficiently by DP if they exhibit properties such as overlapping subproblems and optimal substructure in order to be able to restate the problem in a recursive form.

In pipeline optimization, DP has proved to be capable to easily handle the non-linearity found in flow optimization models. Quite a number of different applications can be found in the literature using DP, including the liquid pipes optimization on linear systems done in the late 1960s by Jefferson [66].

Concerning the FCMP, since it can be formulated as a combinatorial problem by discretizing the range of the pressure variables, it can also be approached by DP. The disadvantages are, however, that its application is practically limited to linear or tree-shaped networks, and the computation time increases exponentially with the dimension of the problem, commonly referred to as *the curse of dimensionality*.

In 1968, Wong and Larson [141] made the first publication to optimize natural gas transportation systems by means of DP. They first applied it to a gun-barrel (linear) network using a recursion formulation for the FCMP, and then in [142] they proceed to solve tree-structured topologies. A disadvantage of these methods was that the length and diameter of the pipeline segment were assumed to be constant because of the limitations of DP. Martch and McCall [85] modified the problem by adding branches to the pipeline segments and allowing that the length and diameter of the pipeline segments vary. However, since their problem formulation did not allow unbranched networks, more complicated network systems could not be handled.

In 1975, Zimmer [148] presented a technical report to calculate the optimum pipeline operations on tree-shaped networks. Several years later, a similar approach was described by Lall and Percell in [79], who allowed one diverging branch in their systems and included an integer decision variable into the model that represented the number of compressors operated into the compressor stations.

In 1989, Luongo, Gilmour, and Schroeder [82] published a hierarchical approach to solve tree-shaped and cyclic networks of arbitrary complexity for a pre-specified set of feasible flows. This was a great advance since some real-world configurations could already be considered. However, their application was not purely DP. Basically, DP was used to arrange the pipes in the network system in a sequential manner in order to reduce the problem to a much smaller combinatorial problem, but without any possibility of a recursive DP solution. Thus, whenever the reduced system was sufficiently small, it was solved via complete enumeration, otherwise it was solved by applying a simulated annealing technique [73]. This hierarchical approach worked very well for some complex pipelines, but for others it was very time consuming.

In 1998, Carter [15] developed an algorithm referred to as *non-sequential DP*. The principal idea of the method is to reduce the network by three basic reductions techniques until it consists of a single node. The method can handle a wide range of instances with cyclic networks, but fails if the networks are not sufficiently sparse. Borraz-Sánchez and Ríos-Mercado, motivated by Carter's work, presented in [11] a two-phase algorithm in which a non-sequential DP technique plays a key role when tackling natural gas cyclic network topologies. In the present work, Carter's ideas are incorporated within a tabu search scheme [55] for iteratively adjusting the set of flows. This will be further discussed in Section 4.6. The restriction that the network instance must be sparse is however a shortcoming that the hybrid method inherits from the original paper. Hence, Sections 4.7 and 4.8 formally discuss and provide algorithmic ideas to overcome the weakness presented by this hybrid method.

4.2.2 Methods based on gradient techniques

In 1987, Percell and Ryan [113] applied a *generalized reduced gradient* (GRG) method for solving FCMP. In comparison with DP, an advantage of GRG is that the rapid growth in instance size caused by many discretization points is avoided. Also, GRG is applicable to cyclic networks. Nonetheless, only a local optimum can be provided, of which instances of FCMP can have many, and the solution to be output depends on the choice of starting point. Jeníček [67] used a gradient technique, where the states of compressors and valves were fixed.

More recently, Flores-Villarreal and Ríos-Mercado in [41] extended with relative success the previous study by means of an extensive computational evaluation of the GRG method on cyclic structures. Villalobos-Morales and Ríos-Mercado in [134] and [135] proposed approximate objective functions to tackle the FCMP and evaluated preprocessing techniques for the GRG method, such as scaling, variable bounding, and choice of starting points, that presented better results for both cyclic and non-cyclic structures.

4.2.3 Other techniques and related problems

Wu et al. in [144] addressed the non-convex nature of FCMP, and suggested mathematical models that provided strong relaxations, and hence tight lower bounds on the minimum cost. Based on this model and the PhD Thesis of Wu [143], they demonstrated the existence of a unique solution to a nonlinear system of algebraic equations over a set of flow variables. This theoretical result lead to a technique for reducing the size of the original network without altering its mathematical structure (see [120]).

Discrete decisions such as the number of compressor units operating within a compressor station, and the binary decision of whether to open or to close a valve, are incorporated into *mixed-integer non-linear programming* models (MINLPs). In pipeline optimization, contributions on this research area are due to, for example, Pratt and Wilson [116], Osiadacz and Górecki [108], Carter et al. [16], Cobos-Zaleta and Ríos-Mercado [24], Villalobos-Morales et al.[133], and Chebouba et al. [22], to mention a few.

For example, [24], [116] and [133] presented satisfactory results given that their proposed methods were capable to solve many FCMP instances tested, although these solutions were only local optima. Cobos-Zaleta in [23] continued the study and solved the resulting MINLP model by means of an outer approximation with an equality relaxation algorithm. In [108], a modest success was obtained by addressing a pipeline network design problem. Several years later, a related work with promising results [16] was presented to propose some algorithms based on implicit filtering for a class of noisy optimization problems while considering discrete decision variables. More recently, in [22], an ant colony optimization algorithm was proposed to the FCMP while applied to a MINLP model. The technique was promising but the computational experiments were limited to a simple linear network instance.

Other works such as those presented by, for instance, O'Neill et al. in [102], Wilson et al. in [139], and more recently, De Wolf and Smeers in [30, 31] also dealt with MINLP models to describe the state of the compressor stations. However, they integrated transportation functions with gas sale planning functions. *Successive Linear Programming*[1] (SLP) was the solution method in the two former papers, whereas piecewise linear approximations solved by an extension of the Simplex algorithm [44] were proposed in the two latter papers in order to avoid both the selection of a trust region and the choice of a rule for updating the size of the trust region. In 2008, Bakhouya and De Wolf [5] separated the previous integrated problem while focusing only on minimizing the power used in the compressors. Here, they presented a study based on a two-phase method to solve Belgian and French gas transmission networks.

Optimization techniques have also been applied for transient (time dependent) models. For instance, Larson and Wismer [80] proposed a hierarchical control approach for a transient operation of a gunbarrel pipeline system. In 1986, Osiadacz and Bell [107] suggested a simplified algorithm for the optimization of the transient gas transmission network, which was based on a hierarchical control approach. The hierarchical control approach for transient models can be found in, e.g. [4], [106], and [109]. A certain degree of success has been reported by these approaches while optimizing the compressor stations. However, since these approaches do not guarantee global optima, their use is somehow limited. More recently, Mahlke et al. [83] considered the problem of time-dependent gas network optimization (also referred to as *transient technical optimization, TTO*) where the nonlinearities were approximated by piece wise linear functions. They developed a branch-and-cut algorithm to solve the linearized problem within a given accuracy. In addition, a simulated annealing (SA) algorithm was used as a primal heuristic, and a separation algorithm for the switching processes of the compressors was also developed. Even though they could not solve all test instances to optimality, the results obtained were acceptable given the difficulty that the transient gas network optimization problem presents.

It is important to mention that the optimization approaches developed to date work well under some general assumptions; however, as the problems become

[1]The *SLP* is an iterative optimization procedure based on both the construction of a linear approximation at some starting point and the definition of a trust region. The trust region is defined by a fixed parameter, δ, that restricts the maximum distance between the initial values and the optimal values achieved by solving the linearized problem. The optimal solution would in turn compose the basis for the linearization to be constructed at next iteration. Note that some authors may make δ dynamic to decrease or increase the trust region based on the predictions given by the approximate model at each iteration.

more complex, the need arises for further research and effective development of more advanced algorithms.

The mathematical model, which formulates the FCMP, is described in detail in the next section.

4.3 Mathematical formulation

4.3.1 Modeling assumptions

The natural gas transport operation on pipeline systems is characterized as a very complex process, hence, making several assumptions on the problem is essential to delimit the scope of the study. First, we study natural gas transmission pipeline systems. This refers exclusively to networks with large diameter pipelines that operate at high pressures.

Second, although in practice, gas transport operations are defined by inherently transient processes, we assume that the problem is in steady-state. That is, the mathematical model provides solutions for pipeline systems that have been operating for a relative large amount of time.

Since we assume the system to be in steady-state, we let the flow variables be independent of time. This allows the use of algebraic equations to describe the behavior of natural gas through the pipeline network. A transient analysis would require the use of partial differential equations to describe the continuity, energy, and momentum equations that relate the decision variables, such as gas flow, velocity, density, pressure, and temperature, as a function of time. Due to the challenge imposed by the transient case, while increasing the number of variables, as well as the complexity of the problem, works on this area are still in a developing phase. (The PhD thesis of Moritz [95] is a good reference on this topic.)

Another assumption is the principle of mass conservation at each node in the network. That is, the network is balanced, which implies that the net flow in each node, except sources and terminals, of the network is equal to zero.

Moreover, we confine our study to irreversible flows in steady-state, i.e, the gas can flow through a pipeline in only one direction. In other words, it is assumed that valves are present to restrict the direction of flow. For some other models, such as those connecting storage facilities to the network, the flow in either direction could be allowed. Additionally, the gas flow is considered isothermal at an inlet average effective temperature. That is, we assume that a heat transfer with the surroundings in the pipeline system causes the temperature to remain constant.

We also assume that the transmission lines are composed of horizontal pipelines. In practice, these systems have frequent changes in their elevation. Nonetheless, the necessary correction factors to compensate the changes in elevation would require special attention beyond the scope of this thesis.

Each parameter is assumed to be known in advance, i.e., we work with a deterministic model. Finally, given that pipe segments are long enough, the changes in the kinetic-energy term can be neglected.

4.3.2 Network representation

Let $G = (V, A)$ be a directed graph representing a gas transmission network, where V and A are the node and arc sets, respectively. Let V_v^+ and V_v^- denote the sets of out- and in-neighbors, respectively, of node $v \in V$. Let $V_s \subseteq V$ be the set of supply nodes, $V_d \subseteq V$ the set of demand nodes, and let $A = A_c \cup A_p$ be partitioned into a set of compressor arcs A_c and a set of pipeline arcs A_p. That is, if $(u, v) \in A_c$ then $u, v \in V$ are the network nodes representing the input and the output units, respectively, of some compressor (u, v). An analogous interpretation is made for pipeline arcs $(u, v) \in A_p$.

Two types of decision variables are defined: Let x_{uv} denote the mass flow rate at arc $(u, v) \in A$, and let p_v denote the gas pressure at node $v \in V$. For each $v \in V$, we define the parameters net mass flow rate B_v and pressure bounds P_v^L and P_v^U (lower and upper, respectively). By convention, $B_v > 0$ if $v \in V_s$, $B_v < 0$ if $v \in V_d$, and $B_v = 0$ otherwise. By the assumption that flow is conserved at the nodes, the decision variables are subject to the constraints $\sum_{u \in V_v^+} x_{vu} - \sum_{u \in V_v^-} x_{uv} = B_v$ for all $v \in V$. Constraints linking the pressure and flow variables are given for the arc sets A_c and A_p, and these are discussed next.

4.3.3 Compressor arc constraints

The variables that are manipulated in a compressor $(u, v) \in A_c$ in order to have the desired values of x_{uv}, p_u, and p_v are according to Wu et al. in [144] compressor speed S_{uv}, volumetric inlet flow rate Q_{uv}, adiabatic head H_{uv} and adiabatic efficiency η_{uv}. As explained more detailed in e.g. [144], these relate to (x_{uv}, p_u, p_v)

according to

$$H_{uv} = \frac{\alpha}{\kappa}\left[\left(\frac{p_v}{p_u}\right)^{\kappa} - 1\right], \quad \forall (u,v) \in A_c \quad (4.1)$$

$$Q_{uv} = \alpha \frac{x_{uv}}{p_u}, \quad \forall (u,v) \in A_c \quad (4.2)$$

$$\frac{H_{uv}}{S_{uv}^2} = A_H + B_H\left(\frac{Q_{uv}}{S_{uv}}\right) + C_H\left(\frac{Q_{uv}}{S_{uv}}\right)^2 + D_H\left(\frac{Q_{uv}}{S_{uv}}\right)^3, \quad \forall (u,v) \in A_c \quad (4.3)$$

$$\eta_{uv} = A_E + B_E\left(\frac{Q_{uv}}{S_{uv}}\right) + C_E\left(\frac{Q_{uv}}{S_{uv}}\right)^2 + D_E\left(\frac{Q_{uv}}{S_{uv}}\right)^3, \quad \forall (u,v) \in A_c \quad (4.4)$$

where $\kappa \in (0,1)$ represents the gas specific heat value, and $\alpha = ZRT$ is a global constant defined by the compressibility factor (Z), the gas constant (R), and the average temperature (T). The coefficients (A_H, B_H, C_H, D_H) and (A_E, B_E, C_E, D_E) of the polynomial functions (4.3) and (4.4), respectively, are assessed by applying least squares analysis to a set of selected data points.

For each $(u,v) \in A_c$, Q_{uv} is subject to lower and upper bounds Q_{uv}^L and Q_{uv}^U, and we adopt a similar notation for bounds on the variables S_{uv}, H_{uv} and η_{uv}.

The fuel consumption cost [144] is given by:

$$g_{uv}(x_{uv}, p_u, p_v) = \frac{c x_{uv}\left[\left(\frac{p_v}{p_u}\right)^{\kappa} - 1\right]}{\eta_{uv}} \quad \forall (u,v) \in A_c,$$

where $c > 0$ is a monetary constant.

The feasible operating domain of the compressor station $(u,v) \in A_c$ is defined by the set $D_{uv} \subset \Re^3$ of value assignments to (x_{uv}, p_u, p_v) for which there exist values of $(Q_{uv}, S_{uv}, H_{uv}, \eta_{uv})$ satisfying (4.1)-(4.4) and the bounds $Q_{uv}^L \leq Q_{uv} \leq Q_{uv}^U$, $S_{uv}^L \leq S_{uv} \leq S_{uv}^U$, $H_{uv}^L \leq H_{uv} \leq H_{uv}^U$, and $\eta_{uv}^L \leq \eta_{uv} \leq \eta_{uv}^U$.

We assume that for all $(x_{uv}, p_u, p_v) \in D_{uv}, \forall (u,v) \in A_c$, there is a *unique* feasible $(Q_{uv}, S_{uv}, H_{uv}, \eta_{uv})$. This correspondence defines the desired transformation from feasible flow and pressure variable values (x_{uv}, p_u, p_v) to an estimate $g_{uv}(x_{uv}, p_u, p_v)$ of the fuel cost.

4.3.4 Pipeline arc constraints

In this study, the Weymouth Equation (WE) proposed in [105] plays a key role due to both its simplicity in gas transmission problems, and its accuracy when dealing with gas flows at high pressure. WE basically expresses the relation between pipeline flow, x_{uv}, through each pipeline $(u,v) \in A_p$ and the drop between its inlet and outlet pressures p_u and p_v, respectively. We also refer to WE throughout this study as the pipe resistance equation in steady-state networks, which takes the following form for flows in horizontal pipes:

$$x_{uv}^2 = W_{uv}\left(p_u^2 - p_v^2\right), \forall (u,v) \in A_p, \tag{4.5}$$

where W_{uv} is typically considered as constant, which depends on both gas properties and pipe physical attributes, given by:

$$W_{uv} = \frac{d^5}{KzgTfL},$$

where z is the gas compressibility factor, g is the gas specific gravity, T is the average temperature, f represents the (Darcy-Weisbach) friction factor, L corresponds to the length of pipe, d is the inside diameter of pipe, and K is a global constant with value defined by the units used.

4.3.5 A non-convex NLP model

For each node $v \in V$, we impose lower and upper pressure bounds P_v^L, and P_v^U, respectively. We confine our study to irreversible flow, and impose $x_{uv} \geq 0$ for all $(u,v) \in A$. Summarizing the two last sections, the non-convex NLP model, originally introduced by Wu et al. in [144], can then be formulated as follows:

$$\min \quad \sum_{(u,v)\in A_c} g_{uv}(x_{uv}, p_u, p_v) \tag{4.6}$$

s.t.:

$$\sum_{u\in V_v^+} x_{vu} - \sum_{u\in V_v^-} x_{uv} = B_v, \quad \forall v \in V \tag{4.7}$$

$$x_{uv} \leq x_{uv}^U, \quad \forall (u,v) \in A_p \tag{4.8}$$

$$(x_{uv}, p_u, p_v) \in D_{uv}, \quad \forall (u,v) \in A_c \tag{4.9}$$

$$x_{uv}^2 = W_{uv}\left(p_u^2 - p_v^2\right), \quad \forall (u,v) \in A_p \tag{4.10}$$

$$P_u^L \leq p_u \leq P_u^U, \quad \forall u \in V \tag{4.11}$$

$$x_{uv} \geq 0, \quad \forall (u,v) \in A. \tag{4.12}$$

Eq. (4.6) represents the objective function to be minimized. It is given by the total amount of fuel consumption in the system. Constraints (4.7)–(4.8) are the typical network flow constraints representing node mass balance and pipeline capacity, respectively. Constraint (4.10) represents the flow-pressure relation in each pipeline of the network under steady-state assumption. Constraint (4.11) reflect the pressure limits in each node. Constraint (4.9) represents the feasible operating domain D_{uv} for compressor station (u,v). Fig. 4.4 shows a two-dimensional shape of the feasible operating domain D_{uv} for p_u fixed. As observed, D_{uv} is defined by a non-convex feasible region. This is a typical property of centrifugal compressor units. Finally, the mathematical model is bounded by nonnegative decision variables, given by constraint (4.12). The details on the nature of g_{uv} and D_{uv} can be found in [144].

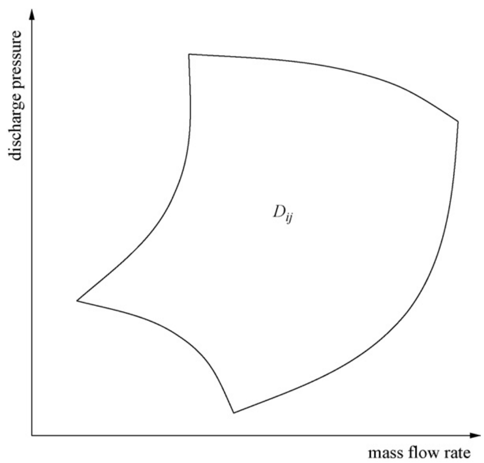


FIGURE 4.4: 2D compressor station feasible operating domain – *Suction pressure fixed*

4.4 Solution approaches for the FCMP

In order to propose solution methods for the FCMP, two major issues have to be taken into consideration. First, the approach must be capable of handling effectively the non-linearity and non-convexity observed in the mathematical model introduced above. Second, its applicability must not be restricted to networks with specific structures, i.e., it must overcome the weakness presented by previous approaches when applied to dense cyclic networks.

In this project, three solution methodologies are introduced with the promise of tackling effectively the FCMP.

The first methodology discussed in this chapter focuses on optimizing flows in the pipeline arcs and pressures at each node of a gas transmission system. It comprises a tabu search mechanism for optimizing the flows, which includes the non-sequential dynamic programming technique proposed by Carter [15] to find optimal pressures for a given set of feasible flows.

The second methodology is exclusively devoted to finding a set of optimal pressures for a given set of feasible flows. The need for a second approach relies on the weakness presented by the first methodology when optimizing pressure variables, which requires the network to be sufficiently sparse. To overcome this weakness, a two-stage based methodology is proposed to handle cyclic networks, including large network cases where the previous approach may have failed. In the first stage, a tree decomposition of the original network is found. Then, in the second stage, a dynamic programming technique is applied to the tree decomposition to find a set of optimal pressures.

Since the effectiveness of the second methodology lies on the width of the tree decomposition obtained in the first stage, as well as on the discretization of the pressure ranges used in the second stage, the application of dynamic programming is compromised, which is known as *the curse of the dimensionality.* This leads to the proposal of a third methodology to ensure somehow that the discretization factor does not diminish the applicability of dynamic programming.

A third methodology is thus proposed based on a key fact: The more refined the discretization is, the more efficient the solution. This methodology can be seen as an extension of the second methodology, in which an adaptive discretization method based on dynamic programming is applied to get an optimal set of pressures variables in a more effective and efficient way.

The three approaches described above are discussed in detail in the subsequent sections. First, two preprocessing techniques that play a key role in the effectiveness of the proposed solution methodologies are introduced.

4.5 Preprocessing techniques

Two preprocessing techniques, namely *bounding and reduction techniques* compose *Phase 0* in the solution methodologies proposed in papers I, II and III (see Table 1.1 in Section 1.2). The aim of this phase is to make the proposed algorithms more effective when applied to large FCMP instances with cyclic structures.

4.5.1 Bounding technique – *Shrinking the search region for DP*

Here, a bounding technique is proposed to speed up the convergence of the solution methods proposed next, particularly concerning the application of DP.

The idea behind this bounding technique is to shrink all pressure bounds in G based on the maximum and minimum potential pressure values given by the

physical properties of each compressor arc $(u, v) \in A_c$. This technique can be seen as elementary operations that may lead to faster algorithms for the FCMP, defined as follows.
Let $\phi\left(\frac{Q_{uv}}{S_{uv}}\right) = A_H + B_H\left(\frac{Q_{uv}}{S_{uv}}\right) + C_H\left(\frac{Q_{uv}}{S_{uv}}\right)^2 + D_H\left(\frac{Q_{uv}}{S_{uv}}\right)^3, \forall(u, v) \in A_c$, and let ρ^L_{uv} and ρ^U_{uv} be the lower and upper limits of the ratio $Q_{uv}/S_{uv}, \forall(u, v) \in A_c$, respectively. Based on (4.3), the bounds on the adiabatic head $H_{uv}, \forall(u, v) \in A_c$ are given by

$$H^L_{uv} = (S^L_{uv})^2\phi(\rho^U_{uv}), \qquad H^U_{uv} = (S^U_{uv})^2\phi(\rho^L_{uv}). \tag{4.13}$$

The lower and upper bounds on the pressure values at node $u \in V$ can then be computed by using (4.1), (4.11) and (4.13), as

$$\Pi^L_u = \max_{u \in V^+_u} P^L_v \left(\frac{\kappa H^U_{uv}}{\alpha} + 1\right)^{-1/\kappa} \tag{4.14}$$

and

$$\Pi^U_u = \min_{u \in V^+_u} P^U_v \left(\frac{\kappa H^L_{uv}}{\alpha} + 1\right)^{-1/\kappa} \tag{4.15}$$

with κ as the isotropic factor.

The new refined pressure bounds can be expressed using (4.14) and (4.15) as:

$$lb_u(P^L, P^U) = \max\{P^L_u,\ \Pi^L_u\} \leq\ p_u\ \leq ub_u(P^L, P^U) = \min\{P^U_u,\ \Pi^U_u\}. \tag{4.16}$$

4.5.2 Compressor network – *Reducing the size of the gas system*

Here, a preprocessing technique to reduce the size of a gas system without altering its mathematical properties is introduced. The outcome of this technique is a *compressor network* (referred to as the *reduced network* by Ríos-Mercado et al. in [120]), which can be obtained in three steps as depicted in Fig. 4.5. In Step 1, all compressor arcs in a given network are temporarily removed. In Step 2, every resulting connected subgraph is merged into a single component, i.e., a big supernode. In Step 3, all compressor arcs are put back into their place to finally get the compressor network.

Mathematically, this can be expressed as follows. Let $V' \subseteq V$ consist of exactly one node from each of the connected components in the directed graph (V, A_p), and let $G^v = (V^v, A^v)$ denote the component (subgraph, *Step 2*) to which $v \in V'$ belongs. Define the *compressor network* as the directed graph

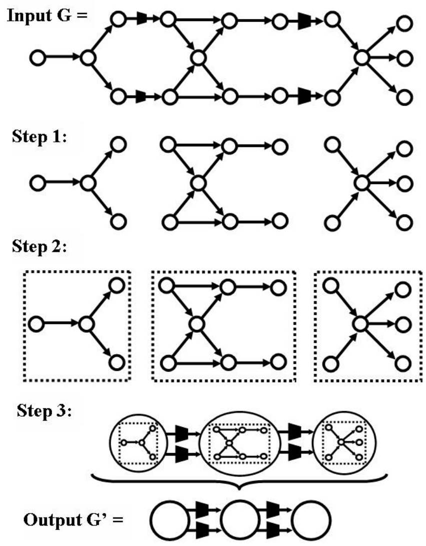


FIGURE 4.5: Transition to a compressor network by applying a reduction technique

$G' = (V', A'_c)$, where $(u, v) \in A'_c$ if and only if $u, v \in V'$ and there exists some arc in A_c from V^u to V^v. As in [120], we assume that G' does not contain loops, which means that no compressor arc has both its start node and its end node in the same connected component of (V, A_p). Equivalently, the node set of G' can be associated with the subgraphs G^v $(v \in V')$, as shown in the illustration of the transition from G to G' (Fig. 4.5 – *Step 3*).

THEOREM 1. *If $A_c = \emptyset$ then for any $B \in \Re^V$ satisfying $\sum_{v \in V} B_v = 0$, any real number $p^{ref} \geq 0$, and any $v \in V$, there exist unique $x \in \Re^A$ and $p \in \Re^V_+$ satisfying $p_v = p^{ref}$, (4.7) and (4.10).*

Proof. See Ríos-Mercado et al. [120]. □

The essence of Theorem 1 is that in any network consisting exclusively of pipeline arcs, the flow and pressure values are all given uniquely once the pressure at any reference node $v \in V$ is set to any value p^{ref}. If (x, p) also satisfies (4.11)-(4.12), the assignment $p_v = p^{\text{ref}}$ is feasible.

The observation that Theorem 1 applies to G^v for all $v \in V'$ suggests the following approach: Identify the connected components in (V, A_p), and nominate one reference node in each. Since x is fixed, all other pressure values are found by utilizing (4.10), and feasibility is checked by verifying whether (4.11) holds. As pointed out by Ríos-Mercado et al. in [119], and exploited in the algorithm given in the same reference, it follows that the original model is reduced to the problem of solving instances of (4.6)–(4.12) where $A_p = \emptyset$ and x is fixed.

Theorem 1 shows that if x_{uv} and p_v are fixed for all compressor arcs $(u, v) \in A_c$ and all reference nodes $v \in V'$, the remaining variable values are computed by solving the system of equations consisting of (4.7) and (4.10). We thus keep p_v fixed for all $v \in V'$, and optimize over $\{x_{uv} : (u, v) \in A_c\}$. To respect the flow balance constraints (4.7), flow updates must be made by sending flow along cycles in G', and by identifying cycles with negative net cost, a reduction in the objective function value is achieved. To check the cost of sending flow along a cycle, we have to take into account the change in x_{uv} for all compressor arcs (u, v) along the cycle, but also the change in p_v for all $v \in V \setminus V'$ in connected components of (V, A_p) intersected by the cycle. For more details, the reader is referred to [120].

4.6 Tabu Search and DP techniques for FCMP (Paper I)

The first approach for solving the FCMP is based on the application of Tabu Search and DP techniques. The method provides a feasible assignment of flows (x^*) and optimal pressures (p^*) to a given network instance $G = (V, A)$ by means of the steps depicted in Algorithm 1. Basically, Steps 2–3 of Algorithm 1 provide an initial feasible point (x, p), and Step 4 gives an improved solution. A brief summary of Algorithm 1 can be given as follows.

Algorithm 1 NDPTS()

Require: An instance of the FCMP
  Step 1: $Preprocessing()$;
  Step 2: $x \leftarrow FindInitialFlow()$;
  Step 3: $p \leftarrow NDP(x)$;
  Step 4: $(x^*, p^*) \leftarrow TS(x, p)$;
Output: A feasible assignment (x^*, p^*)

In Step 1 of Algorithm 1, the preprocessing techniques introduced in the previous section are applied in order to reduce the feasible operating domain of each compressor arc $(u, v) \in A_c$, and to get the associated compressor network G' of G. In Step 2, a set of initial feasible flows (x) is obtained. This can be accomplished by the approach suggested in, e.g., [12], in which a technique based on path selection and the residual network of G, is applied.

In Step 3, optimal pressure (p), for the given set of flows, is obtained by means of a non-sequential DP [15] (NDP) algorithm. Finally, an iterative procedure based on tabu search [55] (TS) is applied in Step 4 to properly optimize the flow values. At each iteration of this procedure, a call to NDP is made to obtain the corresponding optimal pressures (based on the new set of flow values). Details on the techniques applied in Steps 3 and 4 are provided next.

4.6.1 Discretized pressure and dynamic programming formulation

Carter [15] suggested to solve FCMP by discretizing $[P^L, P^U]$ and then apply a network reduction technique referred to as *non-sequential dynamic programming* (NDP).

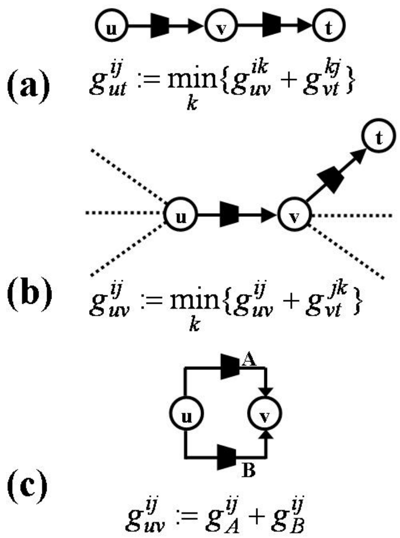


FIGURE 4.6: Network reduction types considered by NDP: serial, dangling and parallel

The NDP algorithm can be described as follows. Assume that there are τ discretization points $p_v^1, \ldots, p_v^\tau$ of the pressure at each $v \in V$ such that $P_v^L \leq p_v^1 < \cdots < p_v^\tau \leq P_v^U$, and for all $i, j = 1, \ldots, \tau$, let $g_{uv}^{ij} = g_{uv}(x_{uv}, p_u^i, p_v^j)$ if $(x_{uv}, p_u^i, p_v^j) \in D_{uv}$ and $g_{uv}^{ij} = \infty$, otherwise. Then NDP consists of a sequence of reductions of G until the resulting graph is a single node. The reduction types (see Fig. 4.6) considered by NDP are described as follows:

(a) Serial: If $v \in V$ has exactly two incident arcs (u, v) and (v, t) in G, then v, (u, v) and (v, t) are replaced by a new arc (u, t), and

$g^{ij}_{ut} = \min_k \left\{ g^{ik}_{uv} + g^{kj}_{vt} : k = 1, \ldots, \tau \right\}$. The same principle applies if both arcs incident to v enter (leave) v.

(b) Dangling: If $v \in V$ has only one incident arc (v, t), then t and (v, t) are removed, and, for all in-neighbors u of v in G, g^{ij}_{uv} is updated to

$g^{ij}_{uv} + \min_k \left\{ g^{jk}_{vt} : k = 1, \ldots, \tau \right\}$. Similar updates apply to the out-neighbors of v, and the principle applies also if the sole neighbor of t is an out-neighbor.

(c) Parallel: If $k > 1$ arcs $a_1, \ldots, a_k$ in G connect nodes u and v, then these are replaced by a single arc (u, v). The associated cost parameters are defined as $g^{ij}_{uv} = \sum_{\ell=1}^{k} g^{ij}_{a_\ell}\ \forall i, j = 1, \ldots, \tau$.

4.6.2 Heuristic approach: *Tabu Search*

Tabu search [55] (TS) is a meta-heuristic (see Section 3.2.4) that uses an adaptive memory structure, as well as strategies of exploration and intensification, to systematically guide a search in an intelligent way.

The core of TS is embedded in its *short-term memory process* [54], whereas its long-term memory basically defines the criteria for intensification and diversification trade-offs. That is, by using a short-term memory, also referred to as a *tabu list*, TS forbids revisiting solutions in order to escape from poor local optima. On the other hand, TS may intensify and diversify the search by using a long-term memory, i.e., by means of a learning process that collects information during the search procedure. These two memory-based strategies rely on four principles that guide the overall searching procedure: *recency* (to indicate how recent a visited solution is), *frequency* (to indicate how often the solution has been visited), quality (to indicate those solutions that have posed good fitness values in order to intensify the search in promising feasible areas) and *influence* (to identify the more promising updates carried out on the current solution structure).

The procedure starts with a given feasible solution (x, p). The nature of a feasible solution is defined by three basic components which are directly related with a cyclic network topology: (a) static component, a mass flow rate value not belonging to any cycle, (b) variable component, a flow value belonging to a cycle, and (c) search component, all pressure variables in the network. These components are depicted in Fig. 4.7.

The search space employed by TS is defined uniquely by the flow variables x_{uv}. This follows because once the flows are fixed, the optimal pressures are obtained by applying NDP. Furthermore, we do not need to handle the entire set of flow variables, but only one per cycle. This is because once a flow value is fixed in a

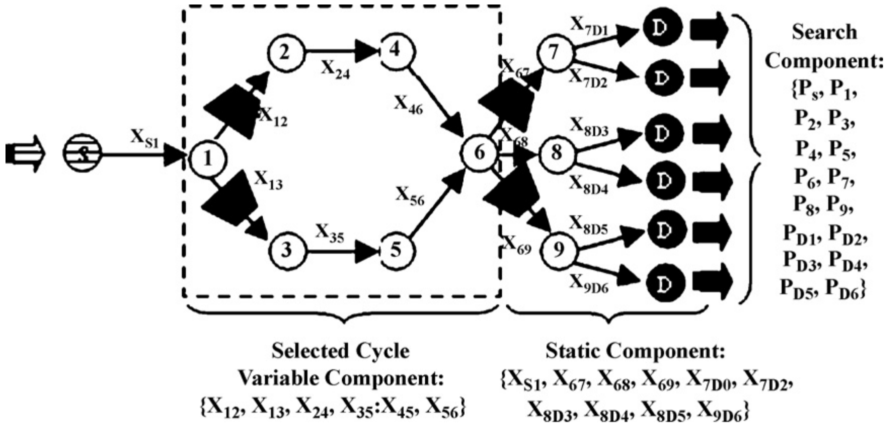


FIGURE 4.7: Basic components of a feasible solution of TS – *A cyclic topology*

cycle, the rest of the flows can be uniquely determined. In consequence, a vector of flows $\overline{x} = (x_1, \cdots, x_m)$ is required, which contains exclusively reference arcs of selected cycles from G, i.e., $x_w \in \overline{x}$ is a reference arc representing a selected cycle w. Note that $\overline{x}$ is arbitrarily chosen, and that converting a flow from x to and from $\overline{x}$ is straightforward, so in the description x and $\overline{x}$ are used interchangeably.

A neighborhood $V(\overline{x})$ of a given solution $\overline{x}$ is defined as the set of solutions reachable from $\overline{x}$ via a slight modification of Δ_x units in each of its components. This is given by

$$V(\overline{x}) = \{x' \in R^m | x'_w = x_w \pm j \cdot \Delta_x, \forall j = 1, 2, \cdots, N_{\text{size}}/2, w = 1, \cdots, m\} \quad (4.17)$$

where N_{size} is the predefined neighborhood size.

The best $x' \in V(\overline{x})$ which is non-tabu is then chosen and the corresponding subsets are updated accordingly. The size of the tabu list (*tabu tenure*) controls the number of iterations that a flow value $x' \in V(\overline{x})$ is kept in the list. The search terminates after *iter_max* iterations.

4.6.3 Overview of the numerical experiments

The TS based procedure, NDPTS, was developed in C++ and run on a Sun Ultra 10 workstation under Solaris 7 at the facilities of the University of Nuevo Leon, Mexico, during my Master studies. There, a lower bound procedure for the FCMP was also implemented in order to assess the quality of NDPTS-solutions.

Concerning the experiments conducted on NDPTS, from preliminary fine-tuning computations, the values for the discretization size in $V(x)$, the discretization size for pressure variables, the tabu tenure, the neighborhood size, and the iteration limit, were set to $\Delta_x = 5$, $\tau = 20$, $T_{\text{tenure}} = 8$, $N_{\text{size}} = 20$, and $iter\_max = 100$, respectively.

Deriving lower bounds for a non-convex minimization problem can be a very difficult task. Obtaining convex envelopes can be as difficult as solving the original problem. However, two important facts that lead to an approximate lower bound can be noted for this problem. First, by relaxing constraint (4.10) in the model introduced in Section 4.3.5, the problem becomes separable in each compressor station, i.e., the relaxed problem consists of optimizing each compressor station individually. Although the resulting relaxed problem is still non-convex, the objective becomes a function of only three variables in each compressor. Hence, we built a three-dimensional grid on these variables and performed an exhaustive evaluation for finding the global optimum of the relaxed problem (for a specified discretization).

Moreover, the experimental phase conducted during my PhD studies is based on the GRG implementation found in [41] for the FCMP. Here, we extended the GRG implementation within a multistart strategy on a GAMS formulation. The procedure generates randomly a set of, not necessarily feasible, solutions to the FCMP model, which are passed to CONOPT [37, 38] to perform local optimization. As stopping criterion, we apply an upper bound on the CPU-time that can be afforded. The experiments were run on the platform described above via remote connection.

All procedures were applied to 11 test instances with compressor-related data provided by a consulting firm in the pipeline industry (see [133] for details). Details on the numerical results are shown in Tables 1–2 from Paper I (see PART II of the thesis). A summary of the results are presented next.

4.6.3.1 Results

First, NDPTS could provide solutions to all instances tested, whereas the multistart local search failed to do so in 4 out of 11 cases.

The deviations of NDPTS-solutions and the multistart local search solutions from the lower bound obtained for the FCMP, are depicted in Fig. 4.8. The observed values from the comparisons of NDPTS-solutions are $\frac{g^{\text{NDPTS}} - g^{\text{LB}}}{g^{\text{LB}}} 100\%$, where g^{NDPTS} and g^{LB} denote the best solution produced by NDPTS and the lower bound obtained, respectively. Similarly, the observed deviations for the multistart local search are defined analogously to the lower bound for the FCMP.

As can be seen from Fig. 4.8, the results indicate that NDPTS outperformed the multistart local search in terms of solution quality. In instances where both procedures found feasible solutions, NDPTS obtained solutions of significantly

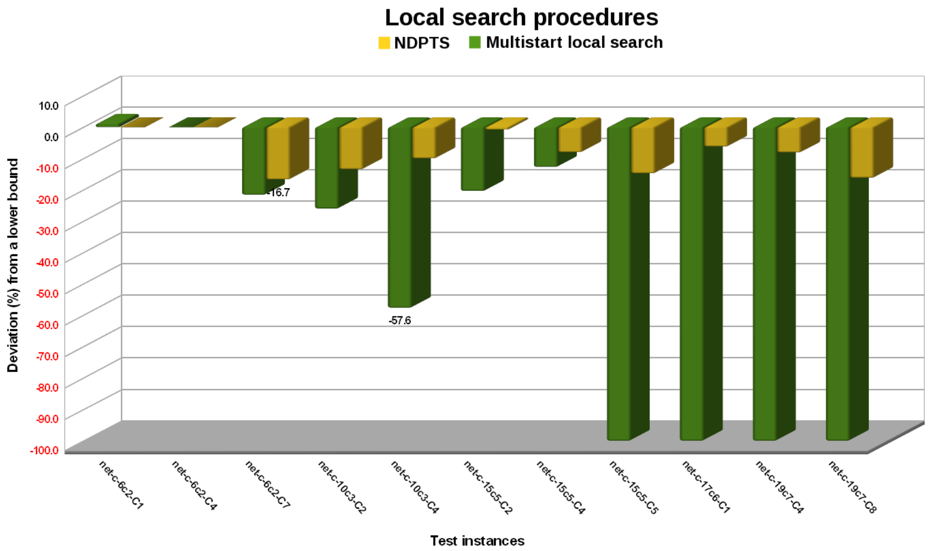


FIGURE 4.8: Deviation (%) of NDPTS and the multistart local search from a lower bound

better quality than those obtained by the multistart local search. This shows the effectiveness of the heuristic approach.

4.6.3.2 Conclusions

It is evident that the heuristic approach dominates the multistart local search. This has been verified empirically by comparing their solutions to lower bounds obtained for the FCMP-instances. Here, NDPTS reported an optimality gap less than 17% in all instances tested, whereas the multistart local search based on a GRG implementation reported an optimality gap up to 57%. Nonetheless, since Carter's technique is applied inside NDPTS to optimize pressure variables, NDPTS requires a sparse matrix when applied to networks with cyclic structures, and even small instances can easily be found where this method fails.

To overcome this limitation, based on the arguments given in Section 4.5.2, a resulting problem FCMP' is achieved by applying a reduction technique and solving (4.6)–(4.12), with the additional conditions that $A_p = \emptyset$ and x is fixed. Note that FCMP' refers to the problem solved in Step 3 of Algorithm 1 (Section 4.6), in which we consider x to be given, and optimize p. With the purpose of developing efficient computational methods for optimizing pressure variables, the rest of the chapter is devoted to problem FCMP'.

4.7 Tackling dense FCMP-instances (Paper II)

FCMP' can be defined as

$$\min_{p \in \Re^{V'}} \left\{ \sum_{(u,v) \in A'_c} g'_{uv}(p_u, p_v) : (p_u, p_v) \in D'_{uv} \forall (u,v) \in A'_c \right\}, \tag{4.18}$$

where $g'_{uv}(p_u, p_v)$ is the cost incurred on all arcs in A_c between V^u and V^v given that u and v are assigned pressure values p_u and p_v, respectively. Further, D'_{uv} is the feasible domain of (p_u, p_v), taking (4.9) into account for all arcs from V^u to V^v.

As a part of the method proposed in Section 4.6, Carter's technique [15] was applied to solve (4.18) by discretizing $[p^L, p^U]$. This technique basically consisted of three reduction techniques shown in Fig. 4.6 (see Section 4.6.1). However, when neither of the reductions (a)–(c) can be carried out, NDP fails. Fig. 4.9 shows a simple example where this occurs. To overcome this weakness, we now go on to demonstrate how such instances of FCMP' can be solved.

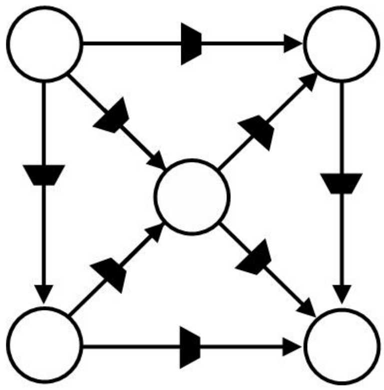


FIGURE 4.9: An instance of G where NDP fails

4.7.1 A tree decomposition approach to optimizing pressures

FCMP' has the mathematical structure of the *frequency assignment problem* [77], and can also be solved by the procedure suggested in the cited reference. This is based on the following concept introduced by Robertson and Seymour in [121]:

Algorithm 2 $\mathbf{DP}(\mathcal{J}, i, X, \pi)$

if i is a leaf in T **then**

return

$$\min_{p \in \mathcal{D}_{X_i \cup X}} \left\{ \sum_{\substack{(u,v) \in A \\ u,v \in X_i \cup X}} g_{uv}(x_{uv}, p_u, p_v) : p_v = \pi_v \forall v \in X \right\}$$

else

return

$$\min_{p \in \mathcal{D}_{X_i \cup X}} \left\{ \sum_{\substack{(u,v) \in A \\ u,v \in X_i \cup X}} g_{uv}(x_{uv}, p_u, p_v) + \sum_{j \in K_i} \mathbf{DP}(\mathcal{J}, j, X_i \cup X, p) : p_v = \pi_v \forall v \in X \right\}$$

Definition 3. *A tree decomposition of G is a pair $\mathcal{J} = (\{X_i : i \in I\}, T)$, where each X_i is a subset of V, called a bag, and T is a tree with node set I. The following properties must be satisfied:*

- $\bigcup_{i \in I} X_i = V$;
- *for all $(u, v) \in A$, there is an $i \in I$ such that $\{u, v\} \subseteq X_i$;*
- *$\forall i, j, k \in I$, if j lies on the path between i and k in T, then $X_i \cap X_k \subseteq X_j$.*

The width of a tree decomposition $\mathcal{J}$ is $\max_{i \in I} |X_i| - 1$.

For any $X \subseteq V$, define p_X as the vector with components p_v ($v \in X$) in any consistent order. Define $\mathcal{D}_v = \{p_v^1, \ldots, p_v^\tau\}$ for all $v \in V$, and let $\mathcal{D}_X = \{p_X : p_v \in \mathcal{D}_v, \forall v \in X\}$. For any $i \in I$, let K_i denote the set of child nodes of i in T.

Algorithm 2 applies a dynamic programming (DP) technique to a tree decomposition $\mathcal{J}$ of G. When bag X_i is processed, the union X of all ancestor bags of X_i are input along with a pressure vector $\pi \in \mathcal{D}_X$. The algorithm optimizes the value of p_v for all $v \in X_i$ by complete enumeration of $\mathcal{D}_v$, and by taking into account optimal pressure assignments to all nodes in all child bags of X_i. This is expressed in terms of a recursive call in Algorithm 2. Since $X_i \cap X$ may be nonempty, we must ensure that nodes contained in this set are not assigned new pressure values when processing X_i, and we impose the constraint that $p_v = \pi_v$ for all $v \in X$.

The running time of Algorithm 2 is $\mathcal{O}\left(|I|\tau^d\right)$, where d is the width of $\mathcal{J}$. This means that finding a tree decomposition of small width can be crucial for the running time of the algorithm. It is however well-known [121] that finding one with minimum width is an *NP-hard* problem, and it is therefore unlikely that a tree decomposition minimizing the running time of Algorithm 2 can be found in polynomial time. We will rely on a heuristic approach to constructing $\mathcal{J}$ with small width.

To solve (4.18), we thus apply a two-phase procedure, TreeDDP, where the computation of some tree decomposition $\mathcal{J}$ is the first phase, and where Algorithm 2 constitutes the second. The input to this procedure is a network which is reduced as much as possible by the techniques described in Section 4.6.1.

To compute $\mathcal{J}$, we apply the technique given in [130].

4.7.2 Overview of the numerical experiments

In the computational evaluation, we make use of the compressor-related data as mentioned in Section 4.6.3, and all compressors are identical within any given instance.

The first experiment is a feasibility study where we examine the performance of TreeDDP while varying the granularity of the discretization. We let $\tau \in \{50, 100, 1000\}$, and let all pressure values be uniformly distributed between their lower and upper bounds.

For a comparison of TreeDDP with a generic global optimization tool, we submit in the second set of experiments (4.18) to *BARON* [131]. In the third set of experiments, we applied MINOS to compute local optima to (4.18) for 1000 different starting-points. (See Section 3.2.5 for details on the optimization tools used.)

The TreeDDP procedure was coded in C++ and run on an Optiplex 745 architecture with Dual Core Technology XD under Fedora 10 based on Linux Red Hat. Experiments with BARON and MINOS were conducted by formulating the model in GAMS [49].

All experiments were carried out on a set of test instances composed of 16 network cases. Details on the test instances and the numerical results can be found in Table 1 and Tables 2–4 , respectively, in Paper II (see PART II of the thesis). Note that in the paper, we refer to the discretization parameter τ as the parameter m. All test instances can be downloaded in GAMS-format at http://www.ii.uib.no/∼conrado/TreeDDP/instances/index.html

4.7.2.1 Results

First, the proposed procedure found feasible solutions in all instances with the smallest granularity ($\tau = 100$ and $\tau = 1000$), whereas it failed to do so in 1 out of 16 cases (instance G) with $\tau = 50$. We also observe that as τ increases, better solutions are found (minimum cost decreases) in most of the instances. Nevertheless, a finer discretization also implies, as expected, that the computational requirements increase.

As for the solvers used, BARON failed to find a feasible solution in 8 out of 16 test instances (G, I, K, L, M, N, O, and P), whereas MINOS did so in 5 cases (G, I, K, N, and P). In most cases, in particular for the largest networks, providing good bounds on all nonlinear variables turned out to be crucial for BARON.

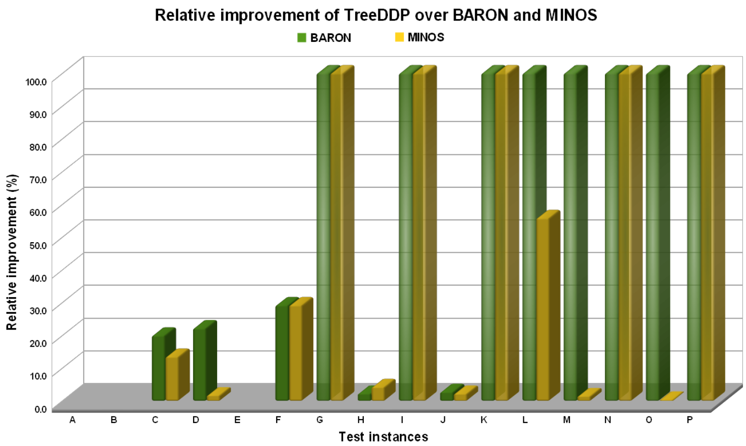


FIGURE 4.10: Relative improvement (%) of treeDDP over BARON and MINOS

The relative improvements (%) reached when comparing TreeDDP-solutions with BARON and MINOS solutions, are depicted in Fig. 4.10. The shown values are $\frac{g^{\text{BARON}} - g^{\text{TreeDDP}}}{g^{\text{BARON}}} 100\%$, where g^{TreeDDP} and g^{BARON} denote the best solution found by TreeDDP with $\tau = 1000$ and the solution found by BARON, respectively. The relative improvement of TreeDDP over MINOS is defined analogously.

As seen in Fig. 4.10, the results indicate that the model appears to be hard to solve to optimality when submitted to BARON within a given time limit of 1 CPU-hour. When the computation times are comparable, TreeDDP outperformed a straightforward multistart local optimization procedure in terms of solution quality. In instances where all methods found feasible solutions, TreeDDP obtained solutions of significantly better quality than those obtained by BARON and MINOS. This shows the effectiveness of the proposed procedure.

4.7.2.2 Conclusions

Still in agreement with previously suggested methods, the non-convex subproblem of optimizing pressure is approximated by a combinatorial one.

The experiments indicate that a method guaranteeing the global optimum in reasonable time seems unrealistic even for small instances. Further, discretizing the pressure variables and applying dynamic programming to a tree decomposition gives better results than multistart local optimization of the continuous version.

Unlike methods based on successive network reductions, our method does not rely on assumptions concerning the sparsity of the network. By constructing a tree decomposition of the network, and by applying DP to it, we are able to solve the discrete version of the pressure optimization problem without enumerating the whole solution space. The contribution of this research is a method for solving the discrete version of FCMP in instances where previously suggested methods fail.

4.8 An adaptive discretization method applied to FCMP (Paper III)

Several works have demonstrated that, at least in acyclic and sparse cyclic instances of FCMP, it is a promising approach to discretize the pressure variables, and apply DP [7] to the resulting combinatorial problem. The purpose of the method to be given in this section is twofold: First, we demonstrate how such approaches can be applied to networks of arbitrary structure. Second, in order to keep the running time down in dense instances, we propose a new scheme for discretizing the pressure variables. This scheme is adaptive in the sense that it avoids fine discretization of variables in area unlikely to contain good solutions, and intensifies discretization in more promising regions.

4.8.1 A heuristic approach

An important parameter of the approach suggested in Section 4.6.1, is the number of discretization points, τ, by which we represent each pressure variable. Assessing this parameter may be difficult. On the one hand, a large value of τ increases the possibility of finding a feasible solution of good quality. On the other hand, the previous section showed that the asymptotic increase in the running time is proportional to τ^d. With a large width d of the tree decomposition, choosing a large value of τ may lead to a slow method.

In this section, we therefore develop a method where the number of discretization points is initially small, and upgraded by a fixed factor until at least one feasible point is found by dynamic programming. Next, for each solution in a selection of the feasible ones hence found, we define an enclosing rectangular subset of the solution set, henceforth referred to as a *focus area*. The same procedure is then applied to each focus area.

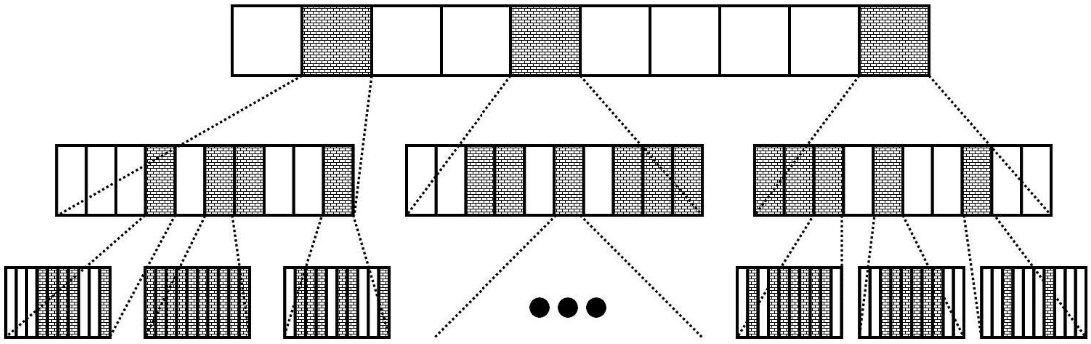


FIGURE 4.11: Search tree based on adaptive discretization – *Solution-neighborhood scheme*

By this approach, we focus the search in the neighborhood of some of the feasible solutions found, and repeat the idea recursively until the discretization distance within the focus area drops below a given threshold. The idea can be depicted by a search tree (see Fig. 4.11), where each node corresponds to a unique focus area and the branches correspond to the set of feasible solutions found by DP and selected for further exploration. To limit the size of the search tree, only a fixed proportion of the feasible solutions are selected to be explored. If Ω is the set of feasible solutions found, we select the $\lceil \sigma|\Omega| \rceil$ solutions in Ω with the smallest cost, where $\sigma \in (0, 1]$ is an input parameter.

The dynamic programming algorithm (Algorithm 2 in Section 4.7.1) can easily be generalized such that it produces a set of solutions rather than only the best solution found. For all possible value assignments to the variables corresponding to the root bag of $\mathcal{J}$, we make optimal value assignments to all the remaining variables. Hence, $|\Omega| \leq \tau^{|X_0|}$, where X_0 is the root bag of $\mathcal{J}$. Only a trivial modification of Algorithm 2, where the root of $\mathcal{J}$ is treated differently from the other bags, is needed, and for reasons of brevity we omit the details. The resulting algorithm, denoted by DP', returns Ω and takes as input the same arguments as does Algorithm 2.

If DP' returns the empty set when τ is set equal to an initial number τ_0 of discretization points, we update τ by a fixed factor γ and call DP' again. The process is repeated until $\Omega \neq \emptyset$ or $\Delta_v = \frac{P_v^U - P_v^L}{\tau - 1} < \epsilon$ for all $v \in V$, where the threshold value ϵ is an input parameter.

The focus area around any selected solution $p \in \Omega$ is defined as the Cartesian product of the intervals $[lb_v(p - \Delta), ub_v(p + \Delta)]$, where $\Delta \in \Re^V$ is the vector with

Algorithm 3 adaptiveDiscretization($\mathcal{J}$, G, p^L, p^U, ϵ, σ,τ_0, γ)

$(\Omega, \tau) \leftarrow$ findFeasibleSolutions($\mathcal{J}$, G, p^L, p^U, ϵ, σ,τ_0, γ)
$z \leftarrow \infty$
if $\Omega \neq \emptyset$ **then**
  $z \leftarrow \min\left\{\sum_{(u,v)\in A} g_{uv}(x_{uv}, p_u, p_v) : p \in \Omega\right\}$
  Let $\Omega' \subseteq \Omega$ consist of the $\lceil\sigma|\Omega|\rceil$ solutions in Ω with smallest cost
  $\Delta \leftarrow \frac{1}{\tau-1}\left(P^U - P^L\right)$
  for all $p \in \Omega'$ **do**
    $z \leftarrow \min\left\{z, \text{adaptiveDiscretization}(\mathcal{J}, G, lb\left(p^L - \Delta\right), ub\left(p^U + \Delta\right), \epsilon, \sigma, \tau_0, \gamma)\right\}$

Algorithm 4 findFeasibleSolutions($\mathcal{J}$, G, p^L, p^U, ϵ, σ,τ_0, γ)

$\tau \leftarrow \tau_0$
repeat
  $\Omega \leftarrow$ DP'(p^L, p^U, τ)
  $\tau \leftarrow \lceil\gamma\tau\rceil$
until $\Omega \neq \emptyset$ or $\frac{p_v^U - p_v^L}{\tau-1} < \epsilon \; \forall v \in V$
return (Ω, τ)

components Δ_v $(v \in V)$. Hence, the range of a variable in the child node covers at most two consecutive intervals between discretization points in the parent node. However, the improved bounds defined by (4.16) (see Section 4.5.1) are likely to narrow down the range.

A summary of the approach is given in Algorithms 3–4.

4.8.2 Overview of the numerical experiments

In the first experiment, we examine the performance of the dynamic programming approach when the number of discretization points is kept fixed. We let $\tau \in \{50, 100, 1000\}$, and let the pressure values be uniformly distributed between their lower and upper bounds. The purpose of the experiment is to study the impact of τ on the quality of the solution and the running time.

In the second experiment, we analyze the performance of the adaptive discretization approach, and compare it to fixed discretization. The idea behind the experiment is to investigate whether adaptive discretization produces solutions comparable to those of fixed discretization in less computer time.

The third experiment is a similar comparison between the dynamic programming approaches and a multistart local search procedure based on MINOS [97].

We run MINOS for 500 and 1000 randomly generated starting points. The points are drawn from the uniform distribution on $[p^L, p^U]$.

The fourth experiment is a comparison between the solutions produced by our methods to (a lower bound on) the true minimum cost. We submit FCMP' to BARON [131]. To solve the convex subproblems, BARON is set to call MINOS. We impose a time limit of 3600 CPU-seconds on each application of BARON, and let the relative optimality tolerance be 0.01. That is, any feasible solution is considered to be optimal if the gap between the objective function value and its lower bound is below one percent of the objective function value. In instances where BARON fails to compute the global optimum, it may still provide a lower bound on the minimum cost, and this bound may also give some indications on the quality of the output from our methods.

The reader is referred to Section 4.7.2 for details on the platform used in these experiments, and for the method used to obtain the tree decomposition of the compressor network G' defined in Section 4.5.2. Details on the optimization tools are given in Section 3.2.5.

All experiments were carried out on 22 test instances. Details on the test instances and the numerical results can be found in Table 1 and Tables 2–7 , respectively, from Paper III (see PART II of the thesis). All test instances can be downloaded in GAMS-format at
http://www.ii.uib.no/∼conrado/caie/instances/index.html

4.8.2.1 Results

First, the DP procedure based on a fixed discretization failed to find a feasible solution in only one case (instance K) for $\tau = 50$. We also observed that by increasing τ, better solutions were found in all instances, except in two cases (instances O and P) where the objective value increased when τ was increased from 50 to 100. Still in agreement with the solution method introduced in Section 4.7.1, an increase in the computational resources was also observed while refining the discretization size. Here, with $\tau = 1000$, the running time slightly exceeded one CPU-hour in one case (instance S).

For the procedure based on dynamic programming and adaptive discretization, we used the parameter values (see Section 4.8.1) $\tau_0 = 3$, $\gamma = 1.5$, $\sigma = 0.05$ and $\epsilon = 0.001$. The procedure found feasible solutions in all instances. Furthermore, we observe that the running time was no larger than 60 CPU-seconds in the most time-consuming case (instance S). Hence, adaptive discretization is much faster than fixed discretization, although the latter requires only one call to the DP-algorithm. The larger number of calls to DP' reported in Table 3 in Paper III

, seems to be more than compensated for by the smaller number of discretization points.

Regarding the multistart local optimization, we can make several observations. First, by increasing the number of starting points for MINOS from 500 to 1000, a relative improvement was observed in only 5 out of 22 cases, where it was no larger than 1% in 3 cases (instances B, C and E), 1.5% in one case (instance F), and up to 2.6% in the remaining case (instance J). For the experiments with 1000 starting points, we observe that MINOS failed to find a feasible solution in 9 out of 22 test instances, whereas for the remaining cases (except A and B), it did so for at least 50% of the starting points. On the other hand, the solver is fast, and a relatively large number of starting points can be affordable.

As for global optimization, BARON was able to find a feasible solution in 15 out of 22 instances, and in 2 instances (A and B), it was able to prove optimality within the given tolerance. In the remaining instances, no feasible solution was found before the time limit expired.

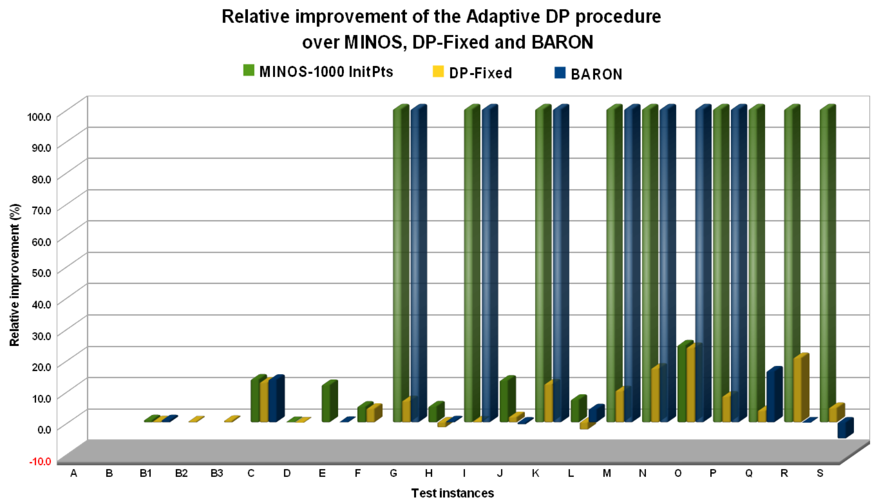


FIGURE 4.12: Relative improvement (%) of the adaptive DP procedure over the fixed discretization, MINOS, and BARON

The relative improvement of adaptive discretization over fixed discretization, is depicted in Fig. 4.12. The shown values are $\frac{g^{\text{DP-Fixed}} - g^{\text{DP-Adaptive}}}{g^{\text{DP-Fixed}}} 100\%$, where $g^{\text{DP-Fixed}}$ and $g^{\text{DP-Adaptive}}$ denote the best solution found by fixed discretization with $\tau = 1000$ and the solution found by adaptive discretization, respectively. The relative improvements of adaptive discretization over MINOS and BARON are defined analogously, and also given in the figure.

In all instances but D, H and L, where fixed discretization gives up to 2.3% lower cost, the adaptive discretization gives solutions of equal or better quality. Also when compared to MINOS, the adaptive discretization approach turns out to be superior. Applying MINOS with a single random starting point is certainly faster, but this involves a considerable risk of failing to find a feasible solution. When the number of random starting points is increased such that the total running time of MINOS exceeds the one of adaptive discretization, the total cost of the best MINOS-solution is in general higher than the cost of the solution produced by its competitor. The maximum relative improvements of adaptive discretization when compared to MINOS, fixed discretization and BARON, respectively, are 24.7%, 23.9% and 16.3%.

Fig. 4.12 also shows that BARON is able to find a better solution than does the adaptive discretization in instances J, R and S. However, extensive computations were needed to find these solutions.

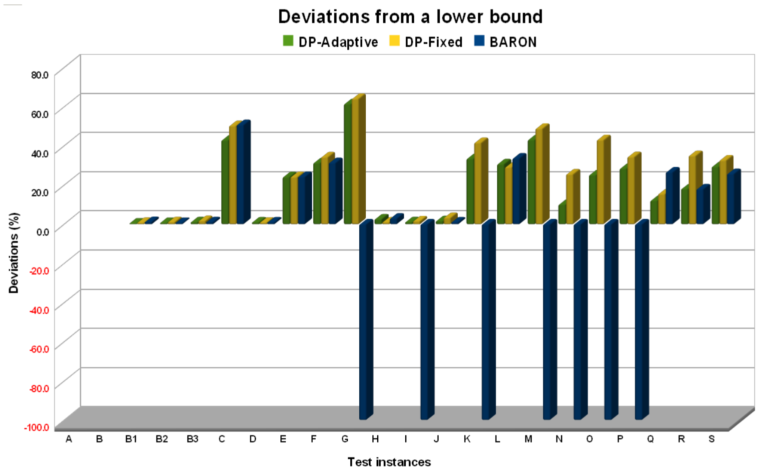


FIGURE 4.13: Relative distance of the adaptive discretization procedure and BARON from the minimum cost

The relative distances of adaptive discretization, fixed discretization and BARON solutions from the lower bound provided by BARON, are depicted in Fig. 4.13. The shown values of adaptive discretization are $\frac{g^{\text{DP-Adaptive}} - g^{\text{LB}}}{g^{\text{LB}}} 100\%$, where g^{LB} denotes the lower bound provided by BARON. The relative distances of fixed discretization ($\tau = 1000$) and BARON from the lower bound are defined analogously.

In 9 out of 22 test instances, adaptive discretization is less than 3% from the relative optimality gap (relative distance from minimum cost), but in some instances, the optimality gap is large (as large as 61.3% in instance G). The relative optimality gap of fixed discretization may be as large as 64.0% (instance G). In the instances where BARON found a feasible solution, the largest gap is 42.9% (instance C). It is however unknown whether this is due to weak lower bounds or shortcomings of our algorithm.

4.8.2.2 Conclusions

Following the general algorithmic idea of the approach proposed in the previous section, the contribution of this research is a heuristic method for solving the discrete version of the problem in instances where previously suggested methods fail.

We have tested our solution methods on a set of imaginary instances, and compared the results to those obtained by applying both a global and a local optimizer to the continuous version of the problem. By an adaptive discretization scheme, we obtain significant speed-up of the dynamic programming approach in comparison with fixed discretization.

CHAPTER 5

Variability of Gas Specific Gravity and Compressibility in Pipeline Systems

A typical approach in steady-state flow models is to consider volumetric parameters like the specific gravity and compressibility as universal constants. However, since these thermodynamic parameters are related to flow, pressure and temperature values, neglecting their variation may lead to significant misleading results. An example where this can occur is in the computation of pipeline resistance.

The aim of the project is to extend previously suggested models in the literature in order to bring them closer to physical reality. This is accomplished by formulating a model where not only the pressure, but also specific gravity and compressibility are defined as state variables during the transmission process. More precisely, the mathematical model proposed in this study maximizes the flow in transmission pipeline systems, and computes the resistance of the pipelines as functions of specific gravity, compressibility and pressure. The gas system is considered to be in steady-state, which implies that the partial derivative with respect to time for any property of the gas or system is zero. In other words, we assume that a certain time has elapsed after the system is initiated before reaching the current state. The gas flow is still considered isothermal, i.e., the temperature is assumed to be constant. Implicitly, all state variables become functions of the flow variables. The non-convex nature of these functions implies that also the feasible region of the model is non-convex.

The literature on optimization models for pipeline gas transportation does not seem to be very rich on models with variable specific gravity or compressibility, and most works focus on models for transient flow. Interested readers are referred to the works presented by Abbaspour and Chapman [1], Chaczykowski [20], Modisette and Modisette [92], and the references therein. The optimization problem is briefly described next.

5.1 Description of the problem

The resistance of a pipeline in steady-state flow problems has been modeled as a function of state variables ever since the inception of gas pipeline optimization in the last century. The work of Wong and Larson [141] when minimizing the total fuel cost incurred by compressor stations is a good example. They suggested to apply the well known Weymouth equation [105] to compute the pipeline capacity. The same principle is followed by more sophisticated works, as those presented by, e.g., Carter [15], Ríos-Mercado et al. [120], De Wolf and Smeers [30, 31], Borraz-Sánchez and Ríos-Mercado [11, 13], Bakhouya and De Wolf [6], Kalvelagen [50, 69], and Borraz-Sánchez and Haugland [10].

Nevertheless, one downside of all the cited works is that they neglect the fact that the parameter in the Weymouth equation depends not only on pipeline characteristics, but also on thermodynamic and physical gas properties. This includes temperature, specific gravity (relative density, g) and compressibility (z-factor), which are assumed as universal constants in these works.

Examples where the assumption is unrealistic exist. The pipeline network connecting wells on the Norwegian continental shelf with the European continent is supplied by gas from sources of relatively lean gas, situated in the North Sea, and sources located in e.g. the Haltenbank area. Since the latter area generally has richer gas, in the sense that it consists of components of higher specific gravity, the assumption of constant properties may be unrealistic. Also, gas compressibility depends on current temperature and pressure conditions, which also vary along the transmission line.

As a result, larger amounts of natural gas may be estimated if optimistic global values are considered. On the contrary, pessimistic input data would lead to reduced estimates in revenues for the transporter.

In this project, the effect caused by the variability of g and z is thus studied. We follow the same principle as the previous works and apply the Weymouth equation to compute the resistance of the pipeline. In the equation, g and z become decisions to reflect their dependence on the upstream flow and the pipeline pressure. Hence, the variability of g must be estimated at the interconnection points of the system as a function of specific gravities of entering flows. Gas compressibility is in turn computed in each pipeline arc by the California Natural Gas Association method [28], which depends on gas specific gravity and pressure values.

5.2 Goals of the project

The objectives of the study are listed as follows. We first evaluate the improvement obtained in terms of the maximum flow by considering the properties of the gas as decisions. This is achieved by a comparison with traditional models in which g and z are considered universal constants. Certainly, the fact of adding decision variables entails an additional computational effort, and this cost is also assessed.

Second, considering that providing global optimal solutions to instances of considerable size can become time consuming, the question of how to find approximate solutions to the original model without losing its realism must also be addressed. Here, the goal is to propose an effective heuristic method that provides sufficiently accurate solutions to the original model.

The heuristic procedure is based on an iterative scheme in which, for given g-values in each network element, a simpler model is solved in each iteration. By the end of each iteration, g is assigned a value consistent with flow values observed in the current iteration. The procedure is repeated until all variables are feasible in the original model. Numerical experiments are conducted to assess the performance of the heuristic, including a comparison with a global optimizer.

In the subsequent sections, the key aspects of the optimization model are briefly discussed.

5.3 The optimization model

To support the modeling of the pipeline resistance in such a way that the variability of g and z is taken into account, some fundamentals on these thermodynamic parameters are given in this section. First, the required notation and the assumptions of the model are presented next.

5.3.1 Notation

Let $G = (V, A)$ be a directed acyclic graph representing a gas transmission network, where V and A are the node (junction point) and arc (pipeline) sets, respectively. Let $V_s \subseteq V$ be the set of supply nodes (sources), and let $V_d \subseteq V$ be the set of demand nodes (terminals).

For each node $i \in V$, we define the net mass flow rate as the variable b_i. By convention, $b_i \geq 0$ if $i \in V_s$, $b_i \leq 0$ if $i \in V_d$, and $b_i = 0$ otherwise. For any node $i \in V$, let $V_i^- = \{j \in V : (j, i) \in A\}$ be the set of start nodes of incoming arcs, and let $V_i^+ = \{j \in V : (i, j) \in A\}$ be the set of end nodes of outgoing arcs. Let

p_i^L and p_i^U, respectively, be a lower and an upper bound on the pressure at node $i \in V$.

5.3.2 Assumptions

A number of assumptions become essential to delimit the mathematical model. First, since the study focuses on optimizing natural gas transmission systems with large diameter pipelines, the proposed model is intended to deal with gas systems operating at high pressures (200 psig and beyond). Second, the principle of mass conservation is applied at each node in the network, i.e., flow variables $x_{ij}, \forall (i,j) \in A$ are subject to the constraints:

$$\sum_{j \in V_i^+} x_{ij} - \sum_{j \in V_i^-} x_{ji} - b_i = 0, \forall i \in V. \tag{5.1}$$

For simplicity, we confine our study to irreversible flow in steady-state, i.e, the gas can flow through a pipeline in only one direction. Extension of the model to a bidirectional flow model, which may become relevant for example when connecting storage facilities to the network, is however straightforward.

The gas flow is considered isothermal at an average effective temperature. We also assume all pipelines to be horizontal pipelines, although in practice, transmission lines have frequent changes in their elevation. However, the need for correction factors to compensate these changes in elevation would require special attention out of the scope of the current work.

We assume $V_i^- = \emptyset \; \forall i \in V_s$, i.e., there are no incoming flow streams (pipelines) to a node which has been identified as source node. Instances violating this assumption can be converted by adding a dummy node pointing toward the original source node. The dummy node inherits all properties of the source node that it replaces, which in its turn becomes a simple transshipment node.

5.3.3 Modeling the resistance of the pipelines

As stated in [125], both the physical properties of the pipelines and the composition of the gas have an influence on the resistance of the pipelines. Several equations have been proposed for modeling the flow capacities in pipelines. We can find the Weymouth equation developed in 1912 (see [105]), the Panhandle A equation developed in 1940, and the Panhandle B equation developed in 1956, among others. Details on these equations can be found in e.g., [26] and [91].

Due to its simplicity and its accuracy when applied to gas flows at high pressures, the Weymouth equation is chosen for this study. Let p_i and p_j be the upstream and downstream pressure, respectively, in pipeline $(i,j) \in A$. The equation can be put in the following form:

$$x_{ij}^2 = W_{ij}\left(p_i^2 - p_j^2\right), \forall (i,j) \in A, \tag{5.2}$$

where W_{ij}, referred to as the Weymouth factor, is a parameter that depends on gas and pipeline properties as given by

$$W_{ij} = \frac{d_{ij}^5}{K z_{ij} g_i T f_{ij} L_{ij}},$$

where z_{ij} is the compressibility of the flow in pipeline (i,j), g_i the specific gravity of the flow arriving at node i, T the gas temperature, f_{ij} is the (Darcy-Weisbach) friction factor in pipeline (i,j), L_{ij} is the length of pipeline (i,j), d_{ij} the inside diameter of pipeline (i,j), and K is a global constant with value defined by the units used.

Eq. (5.2) basically defines the relationship between the mass flow rate x_{ij} through a horizontal pipeline $(i,j) \in A$ and the corresponding difference between the squares of the inlet and outlet pressures p_i and p_j, respectively.

By defining $w_{ij} = z_{ij} g_i W_{ij}$, (5.2) can be written

$$z_{ij} g_i x_{ij}^2 = w_{ij}\left(p_i^2 - p_j^2\right), \forall (i,j) \in A. \tag{5.3}$$

As observed, the pipeline parameter w_{ij} is independent of the flow properties z_{ij} and g_i.

5.3.4 Ideal gas law

"An ideal gas is a theoretical gas composed of a set of randomly-moving, non-interacting point particles" (Wikipedia).

From the kinetic theory, it can be deducted the so-called *ideal gas Law*, which can be written

$$z = \frac{PV}{NkT}, \tag{5.4}$$

where z is the gas compressibility, P is the absolute pressure, V is the volume, N is the number of molecules, k is the *Boltzmann constant* $(1.38066 \times 10^{-23} J/K)$, and T is the absolute temperature.

Eq. (5.4) defines the relationship between the state variables that characterize an ideal gas: P, V, and T. Contrary to real gases, an essential assumption of an ideal gas is that z is equal to 1 for any pressure value at a given temperature. However, real gases exhibit a clear dependence on current pressure and temperature conditions, thus changing the compressibility. This is studied next.

5.3.5 Gas compressibility

As observed in (5.4), the gas compressibility, also referred to as the z-factor, can be considered as the deviation from ideal gas. More formally, it is defined as the relative change in gas volume in response to a change in pressure and temperature. The importance of accurate estimates of this parameter is obvious from (5.2) and the definition of W_{ij}.

The literature on gas metering reveals a number of diverse methods for approximating the z-factor, including experimental measurements, equations of state methods [35], empirical correlations [71] and regression analysis methods [36, 58].

For instance, Katz et al. [71] presented a graphical correlation for the z-factor as a function of pseudo-reduced temperature and pressure based on experimental data. As a result, the Standing-Katz z-factor chart has been used to obtain natural gas compressibility factors for more than 40 years. Dranchuk and Abou-Kassem [35] used the equation of the state to fit the Standing-Katz data and extrapolated to higher reduced pressure. This was accomplished by a simple mathematical description of the Standing-Katz z-factor chart. The California Natural Gas Association (CNGA) developed a method [28] to compute the z-factor based on the gas specific gravity, temperature and pressure values. All these methods have a domain where they are reasonably accurate, and may break down outside.

In the current work, we make use of the CNGA method, which is briefly described next. For the sake of comparison in terms of accuracy of the adopted method, the AGA-NX19 and DPR methods are also introduced. A survey of the methods can be found in [125].

5.3.5.1 The California Natural Gas Association method

The CNGA method has been in use since the middle of the last century. One of its first applications is reported by Davisson [28], who makes use of the method in a computer program for precise flow calculations.

In our work, there are two key reasons for choosing this method. First, the method is valid for z-factor computations at high pressures, which agrees with the assumption made in Section 5.3.2. Second, in comparison with other procedures, the CNGA method presents a simple and effective way to compute compressibility as a function of specific gravity, temperature and pressure.

The CNGA equation can be stated as follows:

$$z_{ij} = \frac{1}{1 + \dfrac{\overline{p}_{ij} \alpha 10^{\beta g_i}}{T^{\delta}}}, \tag{5.5}$$

where $\overline{p}_{ij}$ is the average pipeline pressure, T is the gas temperature and g is the gas specific gravity, and α, β and δ are universal constants.

According to Shashi Menon [125], by using (5.5), the estimation of z_{ij} typically becomes more accurate for transportation pipelines working at pressures beyond 100 psig. Since we assume gas streams under isothermal conditions, i.e., T is assumed fixed, (5.5) can be written

$$z_{ij}\left(1+\omega\overline{p}_{ij}\times 10^{\beta g_i}\right)=1, \tag{5.6}$$

where $\omega=\frac{\alpha}{T^\delta}$ is an instance specific constant.

5.3.5.2 The AGA-NX19 method

The AGA-NX19 method is used to compute z-factor values based on the gas specific gravity and the average temperature and pressure conditions. The method was developed in a research project supported by the American Gas Association (AGA) between 1956 and 1962 [123]. Let P_m be the average pressure in the pipeline. The procedure can be stated as follows:

$$F_p=\frac{156.47}{160.8-7.22g},\ F_t=\frac{226.29}{99.15+211.9g},\ T_c=\frac{T_mF_t}{500}$$

$$P_c=\frac{P_mF_p+14.73}{1000},\ M=\frac{0.0330378}{T_c^2}-\frac{0.0221323}{T_c^3}+\frac{0.0161353}{T_c^5}$$

$$N=\left(\frac{\frac{0.265827}{T_c^2}+\frac{0.0457697}{T_c^4}-\frac{0.133185}{T_c}}{M}\right),\ B=\frac{3-MN^2}{9MP_c^2}$$

$$E=1-0.00075P_c^{2.3}(2-e^{-20(1.09)-T_c})-1.317(1.09-T_c)^4P_c(1.69-P_c)$$

$$A=\frac{9N-2MN^3}{54MP_c^3}-\frac{E}{2MP_c^2},\ D=\left(A+\sqrt{A^2+B^3}\right)^{1/3}$$

$$F_{pv}=\frac{\sqrt{\frac{B}{D}-D+\frac{N}{3P_c}}}{1+\frac{0.00132}{T_c^{3.25}}}$$

$$z=\frac{1}{F_{pv}^2}. \tag{5.7}$$

5.3.5.3 The Dranchuk, Purvis, and Robinson method

The Dranchuk, Purvis, and Robinson (DPR) method uses *the Benedict-Webb-Rubin equation of state* [91] to correlate the Standing-Katz chart in order to approximate z as a function of (g, T, P) [36].

The DPR method uses several coefficients in a polynomial function of the reduced density, ρ_r, as follows:

$$z = 1 + \left(A_1 + \frac{A_2}{T_{pr}} + \frac{A_3}{T_{pr}^3}\right)\rho_r + \left(A_4 + \frac{A_5}{T_{pr}}\right)\rho_r^2 + \frac{A_5 A_6 \rho_r^5}{T_{pr}} + \cdots$$

$$\frac{A_7 \rho_r^3}{T_{pr}^3 (1 + A_8 \rho_r^2) e^{-A_8 \rho_r^2}} \tag{5.8}$$

where the corresponding values to the coefficients are:

$$\begin{aligned} A_1 &= 0.31506237; & A_2 &= -1.04670990; \\ A_3 &= -0.57832729; & A_4 &= 0.53530771; \\ A_5 &= -0.61232032; & A_6 &= -0.10488813; \\ A_7 &= 0.68157001; & A_8 &= 0.68446549; \end{aligned}$$

In addition, ρ_r is given by:

$$\rho_r = \frac{0.270 P_{pr}}{T_{pr}}, \tag{5.9}$$

with P_{pr} and T_{pr} as the pseudo-reduced pressure and temperature values, respectively. The reduced temperature is defined as the ratio of the temperature of the gas to its critical temperature. Similarly, the reduced pressure is the ratio of gas pressure to its critical pressure.

According to [125], if the percentages of the various components in the natural gas mixture are not available, the pseudo-critical temperature and pressure of the gas mixture, T_{pc} and P_{pc}, respectively, can be estimated as a function of the gas specific gravity (g) as follows:

$$T_{pc} = 170.491 + 307.344g, \tag{5.10}$$

$$P_{pc} = 709.604 - 58.718g. \tag{5.11}$$

P_{pr} and T_{pr} can then be estimated by using (5.10) and (5.11) as follows:

$$P_{pr} = \frac{P}{P_{pc}}, T_{pr} = \frac{T}{T_{pc}}. \tag{5.12}$$

5.3.5.4 Comparative study

For the comparative study of the methods described above, we use a Matlab program for encoding the methods as specified by (5.6)–(5.8). We thus applied the methods to compute z on a discretized pressure range for constant temperature and specific gravity values.

Fig. 5.1 shows the gas compressibility curves on various constant temperature lines provided by the AGA-NX19, DPR and CNGA methods.

As observed in Fig. 5.1, the higher the temperature, the smaller the deviation among the compressibility curves provided by the methods. This supports the choice of the CNGA method in this study given that its estimates tally with those provided by AGA-NX19 and DPR for temperatures higher than $0^o F$ and up to about 3000 psia, thus meeting the requirements of this study.

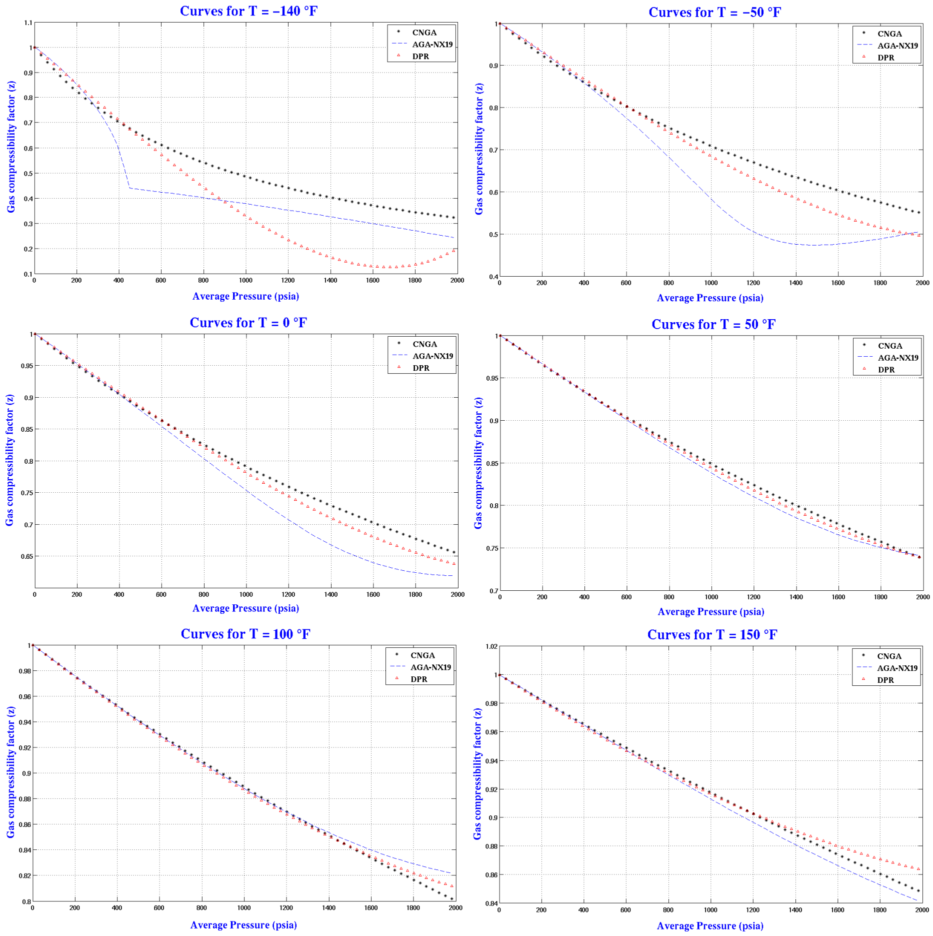


FIGURE 5.1: Gas compressibility curves on constant temperature lines – *Estimates provided by the AGA-NX19, DPR and CNGA methods.*

Table 5.1: Molecular weights of typical compounds in a natural gas mixture.

Compound	Molecular weight
N_2	28.02
O_2	32.00
C_2H_4	16.04
C_2H_6	30.07
C_3H_8	44.09

5.3.6 The gas specific gravity

The specific gravity, also known as the relative density, is a dimensionless unit defined as the ratio between the density (mass per unit volume) of the actual gas and the density of air at the same temperature. It can also be defined in terms of molecular weights (M_w) as the ratio of the apparent molecular weight of the gas mixture to the molecular weight of air, given by:

$$g = \frac{\sum_{c \in NG} \Gamma_c M_w^c}{\sigma}, \tag{5.13}$$

where Γ_c is the relative content of compound c in the natural gas mixture, M_w^c is the corresponding molecular weight of compound c, and σ is the molecular weight of air.

For the sake of illustration, let us assume an air mixture composed of 76% of nitrogen and 24% of oxygen. Based on the list of molecular weights given in Table 5.1, the molecular weight of air can then be calculated as

$$\sigma = 0.76 M_w^{N_2} + 0.24 M_w^{O_2} = 28.96,$$

with $M_w^{N_2}$ and M^{O_2} as the corresponding molecular weights of nitrogen and oxygen. Then, the specific gravity of a natural gas mixture that consists of 88% of methane, 7% of ethane, 3% of propane, and 2% of nitrogen, can be computed using (5.13) as

$$g = \frac{0.88 M_w^{C_2H_4} + 0.07 M_w^{C_2H_6} + 0.03 M_w^{C_3H_8} + 0.02 M_w^{N_2}}{\sigma} = 0.6251,$$

with $M_w^{C_2H_4}$, $M_w^{C_2H_6}$, $M_w^{C_3H_8}$, and $M_w^{N_2}$ as the corresponding molecular weights of methane, ethane, propane, and nitrogen.

A complete list of molecular weights and other properties of various hydrocarbon gases is provided by Shashi Menon [125]. Published values of the specific gravity of natural gas range from 0.554 to 0.870.

In this study, the specific gravity values used at the sources, i.e., $g_i, \forall i \in V_s$, are calculated in advance as specified by (5.13) based on some natural gas mixtures found in the literature, and for nodes $j \in V \setminus V_s$, we assume that

$$g_j = \frac{\sum_{i \in V_j^-} g_i x_{ij}}{\sum_{i \in V_j^-} x_{ij}} \tag{5.14}$$

That is, we let the specific gravity of a blend of different gases be the weighted average of specific gravities of entering flows. The flow values constitute the respective weights.

The equation of specific gravity balance is obtained by multiplying (5.14) by the total flow:

$$g_j \sum_{i \in V_j^-} x_{ij} - \sum_{i \in V_j^-} g_i x_{ij} = 0, \forall j \in V. \tag{5.15}$$

5.3.7 Computing average pressure in a pipeline

The pressure is decreasing along the pipeline. According to Shashi Menon [125], the formula

$$\overline{p}_{ij} = \frac{2}{3}\left(p_i + p_j - \frac{p_i p_j}{p_i + p_j}\right) \tag{5.16}$$

gives a better approximation to the average pressure $\overline{p}_{ij}$ in pipeline $(i, j) \in A$, than does the arithmetic mean of p_i and p_j. The suggested formula in its quadratic form $3\overline{p}_{ij}(p_i + p_j) = 2(p_i^2 + p_j^2 + p_i p_j)$ is adopted in the current work.

5.3.8 The proposed NLP model

Summarizing the above sections, we formulate a mathematical model ($\mathcal{M}_1$) with flow, pressure, compressibility and specific gravity as decisions. For convenience, a complete list of decision variables of $\mathcal{M}_1$ is given in Table 6.1.

The constraints include conservation of flow (see Section 5.3.2), the Weymouth equation as given in Section 5.3.3, the compressibility formula suggested in Section 5.3.5.1, the equation of specific gravity balance (see Section 5.3.6), the quadratic form of the average pressure formula suggested in the previous section, as well as pressure and flow bounds. The objective function, describing a flow maximization problem, takes the form $\max \sum_{i \in V_s} b_i$.

$\mathcal{M}_1$ can then be formulated as follows:

Table 5.2: Decision variables

$Y =$	Objective function value
$x_{ij} =$	Flow through arc $(i,j) \in A$,
$b_i =$	Net flow (Supply/demand) at node $i \in V$,
$p_i =$	Gas pressure at node $i \in V$,
$\overline{p}_{ij} =$	Average gas pressure in arc$(i,j) \in A$,
$z_{ij} =$	Compressibility of gas in arc $(i,j) \in A$
$g_i =$	Specific gravity of gas at node $i \in V \backslash V_s$

$$(\mathcal{M}_1) \quad Y_1 = \max \textstyle\sum_{i \in V_s} b_i \tag{5.17}$$

s.t.:

$$\sum_{j \in V_i^+} x_{ij} - \sum_{j \in V_i^-} x_{ji} - b_i = 0, \qquad \forall i \in V \tag{5.18}$$

$$g_j \sum_{i \in V_j^-} x_{ij} - \sum_{i \in V_j^-} g_i x_{ij} = 0, \qquad \forall j \in V \backslash V_s \tag{5.19}$$

$$z_{ij}(1 + \omega \overline{p}_{ij} \times 10^{\beta g_i}) = 1, \qquad \forall (i,j) \in A \tag{5.20}$$

$$3\overline{p}_{ij}(p_i + p_j) = 2(p_i^2 + p_j^2 + p_i p_j), \quad \forall (i,j) \in A \tag{5.21}$$

$$g_i z_{ij} x_{ij}^2 = w_{ij}\left(p_i^2 - p_j^2\right), \qquad \forall (i,j) \in A \tag{5.22}$$

$$p_i^L \leq p_i \leq p_i^U, \qquad \forall i \in V \tag{5.23}$$

$$b_i = 0, \qquad \forall i \in V \backslash V_s \backslash V_d \tag{5.24}$$

$$b_i \geq 0, \qquad \forall i \in V_s \tag{5.25}$$

$$b_i \leq 0, \qquad \forall i \in V_d \tag{5.26}$$

$$x_{ij} \geq 0, \qquad \forall (i,j) \in A. \tag{5.27}$$

Since constraints (5.19)-(5.22) are non-convex, computing a global optimal solution to $\mathcal{M}_1$ is expected to be time consuming. In instances of realistic size, fast and possibly inexact solution methods may be required, and suggestions to this are discussed next.

5.4 A heuristic method

In this section, we propose a heuristic method to tackle the model $\mathcal{M}_1$ presented in Section 5.3.8. To diminish the difficulty introduced by the non-convex constraint (5.19), we propose to deal with the variability of g outside the model, and develop an iterative procedure where g is kept fixed in each iteration. This leads to a simpler mathematical formulation ($\mathcal{M}_2$) that is solved once in each iteration. By declaring the specific gravity at each node $i \in V$ as an input parameter g_i, $\mathcal{M}_2$ is put in the following form:

$$(\mathcal{M}_2) \quad Y_2 = \max \textstyle\sum_{i \in V_s} b_i \tag{5.28}$$

s.t.:

$$(5.18), (5.20) - (5.27). \tag{5.29}$$

By virtue of the reduced number of decision variables and the absence of (5.19), $\mathcal{M}_2$ is easier to solve than $\mathcal{M}_1$. In addition, constraints (5.20) and (5.22) have become simpler since g is considered to be constant.

Algorithm 5 Heuristic(ε, $g^{(0)}_{i|i \in V_s}$)

$t \leftarrow 0$
compute $g_j^{(t)}$ such that $g_j^{(t)} = \frac{1}{|V_j^-|} \sum_{i \in V_j^-} g_i^{(t)}$ for all $j \in V \backslash V_s$.
repeat
  $t \leftarrow t + 1$
  fix g to $g^{(t)}$, and solve $\mathcal{M}_2$ for $x^{(t)}, z^{(t)}, p^{(t)}$ and $\overline{p}^{(t)}$.
  for $i \in V_s$ **do**
    $g_i^{(t+1)} \leftarrow g_i^{(t)}$
  for $j \in V \setminus V_s$ **do**
    if $\sum_{i \in V_j^-} x_{ij}^{(t)} > 0$ **then**

$$g_j^{(t+1)} \leftarrow \sum_{i \in V_j^-} g_i^{(t)} x_{ij}^{(t)} \Bigg/ \sum_{i \in V_j^-} x_{ij}^{(t)}$$

    else
      $g_j^{(t+1)} \leftarrow g_j^{(t)}$
until $\left(x^{(t)}, z^{(t)}, p^{(t)}, \overline{p}^{(t)}, g^{(t+1)}\right)$ is feasible in (5.20) and (5.22) within a tolerance ε.

Two main steps inside a loop constitute the main parts of the proposed heuristic method. Keeping g fixed to some value $g^{(t)}$ in iteration t, we first solve $\mathcal{M}_2$ to get a feasible point $\left(x^{(t)}, z^{(t)}, p^{(t)}, \overline{p}^{(t)}\right)$. Next we correct g in order to make it consistent with the feasible point found, as specified by (5.15). Note that at any source node i, g_i is given as input data, and remains unchanged throughout the heuristic. For a node $j \in V \setminus V_s$, the initial value $g_j^{(0)}$ is based on the specific gravities given for source nodes i on paths leading to j. More precisely, we let

$g_j^{(0)} = \frac{1}{|V_j^-|}\sum_{i\in V_j^-} g_i^{(0)}$ such that each node initially has a specific gravity equal to the mean specific gravity of upstream neighbor nodes. Since the network has no directed cycles, it is straightforward to compute the vector $g^{(0)}$.

The point $\left(x^{(t)}, z^{(t)}, p^{(t)}, \overline{p}^{(t)}, g^{(t+1)}\right)$ is declared feasible if

$$\left| z_{ij}^{(t)}(1+\omega\overline{p}_{ij}^{(t)} \times 10^{\beta g_i^{(t+1)}}) - 1\right| \leq \varepsilon \cdot \max_{(i,j)\in A} \max\{z_{ij}^{(t)}(1+\omega\overline{p}_{ij}^{(t)} \times 10^{\beta g_i^{(t+1)}}), 1\}$$

and

$$\left| g_i^{(t+1)} z_{ij}^{(t)} (x_{ij}^{(t)})^2 - w_{ij}\left((p_i^{(t)})^2 - (p_j^{(t)})^2\right)\right| \leq$$
$$\varepsilon \cdot \max_{(i,j)\in A} \max\{g_i^{(t+1)} z_{ij}^{(t)} (x_{ij}^{(t)})^2, w_{ij}\left((p_i^{(t)})^2 - (p_j^{(t)})^2\right)\}.$$

In that case, the algorithm stops. A summary of the approach is given in Algorithm 5.

5.5 A traditional approach

As mentioned in Section 5.1, optimizing the transport of natural gas via pipelines while neglecting the variability of specific gravity and compressibility, is a typical approach of steady-state flow models. Let $\bar{g}$ and $\bar{z}$, respectively, denote the universal constants for the specific gravity and compressibility. A traditional approach, $\mathcal{M}_3$, can then be achieved by simplifying $\mathcal{M}_1$ as follows:

$$(\mathcal{M}_3) \quad Y_3 = \max \textstyle\sum_{i\in V_s} b_i \tag{5.30}$$

s.t.:

$$\bar{g}\bar{z}x_{ij}^2 = w_{ij}\left(p_i^2 - p_j^2\right), \qquad \forall (i,j)\in A \tag{5.31}$$

$$(5.18), (5.23) - (5.27). \tag{5.32}$$

Note that to adapt the input data of $\mathcal{M}_1$ to $\mathcal{M}_3$ in the numerical experiments presented next, we let

$$\bar{g} = \frac{g^{\min} + g^{\max}}{2}$$

and

$$\bar{z} = \frac{1}{1+\omega\frac{p^{\min}+p^{\max}}{2}10^{\beta\bar{g}}},$$

where $g^{\min} = \min_{i\in V_s} g_i$, $g^{\max} = \max_{i\in V_s} g_i$, $p^{\min} = \min_{i\in V} p_i^L$, $p^{\max} = \max_{i\in V} p_i^U$. In $\mathcal{M}_2$, we replace all occurrences of g_i by $\bar{g}$, but no replacement is made for z_{ij}.

5.6 Overview of the numerical experiments

The purpose of the computational analysis is twofold. First, we assess the need for the proposed model $\mathcal{M}_1$ by making comparisons with the traditional model $\mathcal{M}_3$ and the model $\mathcal{M}_2$ used in Algorithm 5. Second, we evaluate the efficiency and effectiveness of Algorithm 5 by comparing it to the performance of commercially available solvers applied to $\mathcal{M}_1$.

In the first set of experiments, all three models were formulated in the GAMS modeling language [49] and submitted to a global optimizer, BARON [131]. BARON is set to call MINOS [97] to solve the relaxations. We imposed a time limit of 3600 CPU-seconds to each application of BARON, where two different relative optimality tolerances ($\varepsilon \in \{10^{-1}, 10^{-3}\}$) were applied. That is, BARON stopped once it is proved that a feasible solution with objective function value above $(1-\varepsilon)$ times the maximum flow is found. The reader is referred to Sections 3.2.5 and 4.7.2, respectively, for details on the optimization tools and the platform used in these experiments.

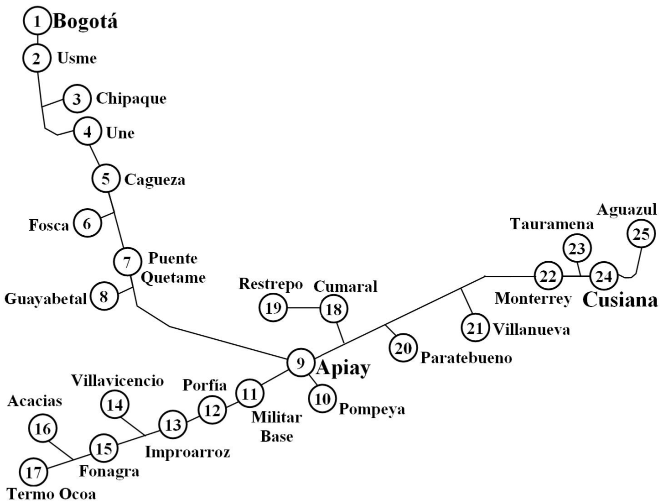


FIGURE 5.2: The Bogota-Apiay-Cusiana, Colombia transmission system

In the second set of of experiments, we compare both Algorithm 5-solutions to the global optimal solutions to $\mathcal{M}_1$ found by BARON, as well as the computational effort of the algorithm to the effort of computing Y_1 exactly.

Recall that Algorithm 5 requires access to solutions to instances of $\mathcal{M}_2$. This was accomplished by letting the algorithm call a local optimizer (MINOS) once in each iteration. The relative stopping tolerance, ϵ, was set to 10^{-3} in all runs.

Multistart local optimization is a straightforward heuristic approach, which by virtue of its simplicity deserves to be compared to Algorithm 5. We implemented a procedure that generates randomly a set of, not necessarily feasible, solutions to $\mathcal{M}_1$. Each solution is passed to MINOS, which is asked to perform local optimization starting from the submitted initial point. The procedure is assigned an upper bound on the CPU-time, and the number of initial points generated is determined as the maximum that can be afforded given that the multistart procedure terminates within this bound.

All experiments were carried out on a set of test instances composed of 31 imaginary cases and 1 real network. Details on the test instances are shown in Table 3 from Paper IV (see PART II of the thesis). All test instances can be downloaded in GAMS-format at
http://www.ii.uib.no/~conrado/Project2/instances/index.html

Fig. 5.2 shows the real gas transmission network from Bogota, Colombia used in the experiments (instance AD). The other instances are generated manually, and Fig. 5.3 shows a typical test network (instance T), where a striped node with an incoming arrow represents a source node, a gray node with an outgoing arrow represents a demand node, and a white node is a transshipment node.

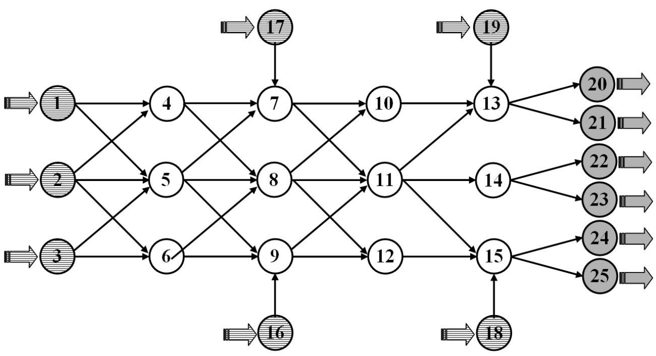


FIGURE 5.3: Typical test network

Details on the results from the numerical experiments can be found in Tables 4–6 from Paper IV (see Part II of the thesis). A brief summary of the major results achieved by the computational analysis is provided next.

5.6.1 Results

The deviations of $\mathcal{M}_2$-solutions and $\mathcal{M}_3$-solutions from $\mathcal{M}_1$-solutions when the models were submitted to BARON with a relative optimality tolerance of $\varepsilon = 10^{-3}$, are depicted in Fig. 5.4. The observed values from the comparisons of $\mathcal{M}_2$ and $\mathcal{M}_3$ are $\frac{Y_2-Y_1}{Y_1}100\%$ and $\frac{Y_3-Y_1}{Y_1}100\%$, respectively.

Fig. 5.4 shows to what extent the traditional approach ($\mathcal{M}_3$) results in wrong estimates of the total flow capacity of a network, and similar information is provided for an approach ($\mathcal{M}_2$) where only the compressibility is allowed to vary with the network components. In the figure, the deviations from $\mathcal{M}_1$ can be observed from -4.7 to 21.0% for $\mathcal{M}_2$, and from -31.6 to 7.6% for $\mathcal{M}_3$.

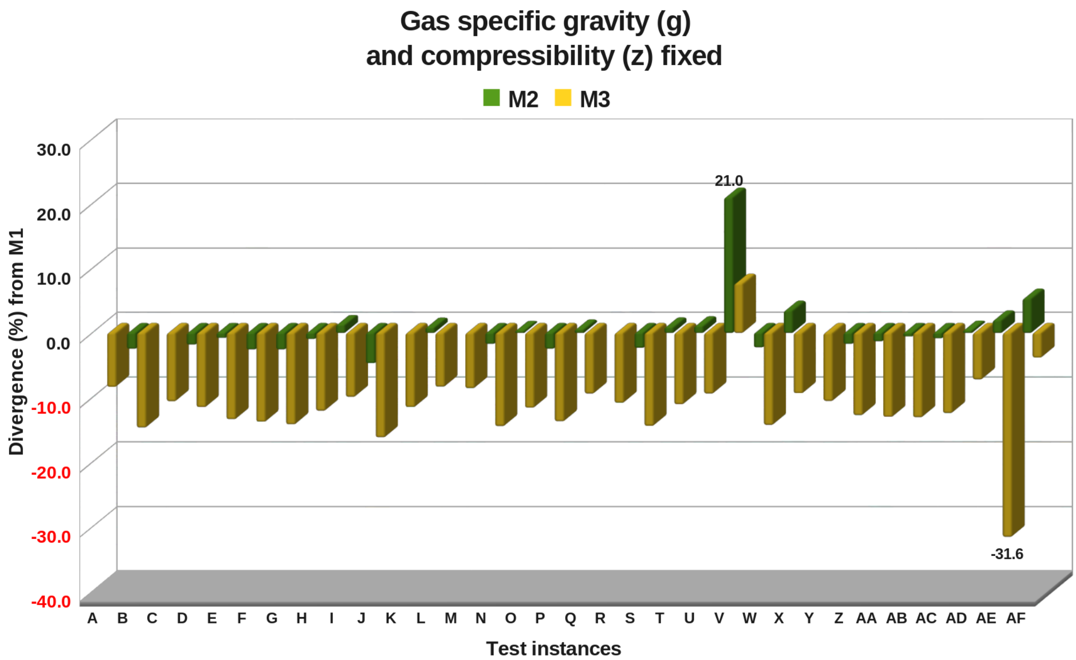


FIGURE 5.4: Divergence (%) of $\mathcal{M}_2$ and $\mathcal{M}_3$ from $\mathcal{M}_1$ for $\varepsilon = 10^{-3}$ in terms of maximum flow values

As seen in Fig. 5.4, the maximum flow values achieved by $\mathcal{M}_3$ overestimate considerably the resistance of the pipelines in a more pessimistic way in comparison with $\mathcal{M}_2$ where only g is fixed.

The deviations of Algorithm 5-solutions and the multistart local search solutions from $\mathcal{M}_1$-solutions ($\varepsilon = 10^{-3}$), when $\mathcal{M}_1$ was submitted to BARON with a relative optimality tolerance of $\varepsilon = 10^{-3}$, are depicted in Fig. 5.5. The shown values for Algorithm 5 are $\frac{Y_1^{\text{alg1}} - Y_1^{\text{BARON}}}{Y_1^{\text{BARON}}} 100\%$, where Y_1^{alg1} and Y_1^{BARON} denote the best solutions produced by Algorithm 5 and BARON, respectively (see Tables 4 and 6 in Paper IV).

The results for multistart local search were achieved by imposing a time limit of 20 CPU-seconds, where the largest objective function value observed was recorded. The time limit was chosen since Algorithm 5 in no instance needed more time to converge. Fairness to its competitor in terms of time allocation is hence assured. The deviations for multistart local search are defined analogously to the deviations for Algorithm 5.

As observed in Fig. 5.5, Algorithm 5 shows a significant deviation in two instances (AE and AF), whereas the multistart local search do so in 5 cases (G, W, V, AE and AF). More precisely, we observe that Algorithm 5 in all instances but AB, AE and AF produces solutions within 5% deviation from the best known solution. A comparison with multistart local search shows that Algorithm 5 looses with a difference less than 1% in 5 instances (F, K, O, P and S), with a difference no larger than 4% in 5 other instances (H, J, Z, AA and AC), and up to 8.2% in instance AB. In 5 instances (G, R, V, W and X), the relative difference between the two heuristics are observed up to 75.5% (instance G) in favor of Algorithm 5. In addition, multistart local search was unable to produce anything better than the zero solution in two instances (AE and AF), whereas for Algorithm 5 this occurred only in instance AE.

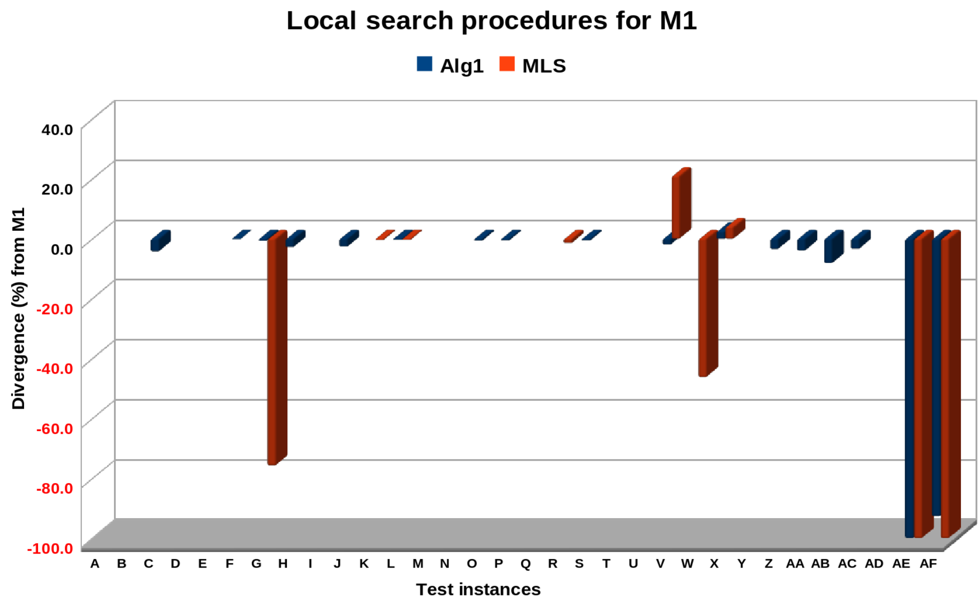


FIGURE 5.5: Deviations (%) of the proposed heuristic and multistart local search from $\mathcal{M}_1$ ($\varepsilon = 10^{-3}$)

We conclude that one promising approach to solving model $\mathcal{M}_1$, is a sequential procedure where some state variables are kept fixed in each iteration. A local optimum to a simplified version of the model (model $\mathcal{M}_2$ in the case of Algorithm 5) can be found quickly, and new and better estimates of the state variables are computed. The likelihood of converging to a solution far from optimum seems smaller than by multistart local search. Besides, the computer time needed for doing so is not close to the time needed to prove optimality.

5.7 Concluding remarks

A mathematical model has been developed for maximizing the flow of natural gas in pipeline transmission networks. Unlike previously suggested models, the model admits variations in gas specific gravity and compressibility.

The resulting model has a non-convex feasible domain, and represents a considerable computational challenge for global optimization algorithms. Simpler, but still non-convex models, where the variation in gas specific properties is neglected, can already be found in the literature. Through experiments, we have demonstrated that applying these to instances where the variation is high, tends to give misleading results.

The simpler models can however be useful building blocks in inexact, but fast, heuristic methods for our model. We have developed an iterative procedure, which in each iteration keeps the specific gravity constant while optimizing other decisions. As a preparation for the next round, the specific gravity is updated consistently with the flow values observed, and the procedure is repeated until convergence. Computational experiments demonstrate that this procedure yields optimal or near-optimal solutions in most of the instances.

Chapter 6

Line-Pack Management Optimization

Reasons for success in different arenas of the private sector are due to both efficient management of resources and equipment, as well as the effective implementation of appropriate strategies. The natural gas transport industry is no exception. Because of the substantial increase in both natural gas demand and its reserves in recent decades, coupled with the expected promising growth in its production and distribution in the years ahead, the gas industry has become more aware of the need for a sustainable infrastructure that may lead to increases in revenue.

In this project, a special attention is paid to the effective application of the transport and short-term storage of natural gas. More precisely, as a strategic idea to meet market demand, and motivated by previously suggested models, we focus on optimizing the gas contained (line-pack) in the pipelines over time. We thus propose a mixed-integer non-linear model for maximizing flow in transmission pipeline systems for a given planning horizon.

In the subsequent sections, the key issues addressed in designing the optimization model are discussed. The description of the problem is provided next.

6.1 The line-packing problem

Gas pipelines have proven to be the most suitable transportation means for the gas industry since the advent of metallurgical improvements and welding techniques after World War II. Since then, dependable and economic pipeline systems have become essential in preserving the continuous business growth of the gas transport industry in national and international arenas. Nevertheless, a common denominator in the transportation process is that a number of unpredictable or scheduled events do occur on a daily basis. Among these events we can find, e.g., the break down of flow capacities elsewhere in the system due to malfunctions, routine maintenance or inspection; failures in upstream process capacity; shortfall in downstream capacity; demand uncertainty; and high fluctuation in demand due to seasons (in the winter the demand is usually higher than in the summer). However, gas producers must be able to supply gas to their customers despite such difficulties.

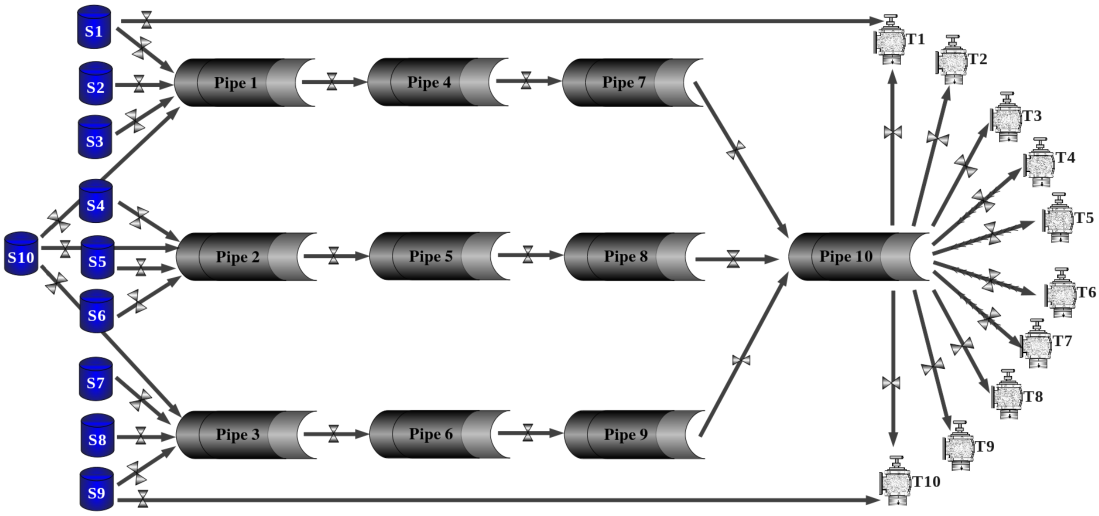


Figure 6.1: Test instance U – *(see Table 1 in Paper V)*

When such events can be predictable, as in the case of routine maintenance and scheduled inspections, the fundamentals of deterministic optimization are applied in this thesis. Hence, we propose an optimization model that aims at satisfying market demand for a given planning horizon based on given values in upstream capacity.

To alleviate the consequences caused by unforeseen events, it requires a stochastic model, which is out of the scope of this thesis. Interested readers on stochastic analyses for transport and storage of natural gas are referred to the works of Carter and Rachford [17] and Contesse et al. [25].

The model to be proposed in this project takes into account one key fact: Gas pipelines do not only serve as transportation links between producer and consumer, but they also represent potential storage units for safety stocks. That is, due to the compressible nature of dry gas, large reserves can be stored on a short-term basis inside the pipeline, referred to as the line-pack, for subsequent extraction when flow capacities elsewhere in the system break down. Hence, the problem of keeping a sufficient level of line-pack during a given planning horizon becomes critical to the transporter.

Fig. 6.1 shows one of the typical test cases used in this study. As observed, the network system is composed of 10 sources, 10 terminals and 10 pipeline segments that can serve as potential storage units to meet market demand over time.

To conceptualize this problem, let us see the simplest example. Let us suppose that there is a unique transmission line between one producer and one costumer, and let us assume that the amount of gas required by the client during several consecutive periods can easily be satisfied with only 60% of the maximum capacity. An obvious solution is simply to send the required amount for the mentioned periods. However, let us assume that the demand increases up to 120% of the maximum capacity for some subsequent period. Here, the producer can not meet such requirement, thus leading to considerable economic losses. Hence, the strategic idea would be to send for instance 80% of the maximum capacity, then consuming just the required demand in each period, and storing the remaining gas to satisfy future extraordinary requirements.

Summing up, the line-packing problem in gas pipeline systems basically refers to optimizing the refill of gas in pipelines in periods of sufficient capacity, and optimizing the withdrawals in periods of shortfall. This is accomplished by, e.g., closing (or throttling) a downstream valve while upstream compressors continue sending gas into the pipeline for future use.

Some attempts, although few, have been made in the direction of mathematical planning models for this problem (see [17] and [47]).

In this project, a *mixed-integer non-linear programming* (MINLP) model is proposed to tackle the line-packing problem in natural gas transmission pipeline systems. The model maximizes gas deliveries on a multi-source, multi-demand transmission system for a given multi-period planning horizon. To do so, we base the study on building up and consuming line-packs in the pipelines over time. This is discussed in detail in the subsequent sections.

6.2 Design of the optimization model

The design of the mathematical model to be given in this section obeys two fundamental assumptions. First, we assume the network to be in steady-state, which implies that the partial derivative with respect to time for any property of the gas or network is zero. In other words, we assume that a certain time has elapsed after the system is initiated before reaching the current state. Second, the gas flow is considered isothermal, i.e., we assume the temperature to be constant.

In the subsequent sections, the main ideas behind how we impose the constraints related to building up and consuming the line-pack in the pipelines during a given multi-period horizon, are discussed. The discussion also includes the way in which the gas composition is tracked during the transmission process. As a result, we present a single-objective, multi-source, multi-demand, multi-quality, and multi-period problem that is formulated as a MINLP model.

6.2.1 Heterogeneous batches

Contracts between suppliers and clients often specify a certain quality of the gas to be delivered. The quality will in this section be illustrated by the relative CO_2 content, but the model to be developed accepts an arbitrary set of quality parameters. The sources do not necessarily all supply gas of equal quality. Fig. 6.2 shows a case with 3 sources which all are different in this respect, and one terminal where the CO_2 content cannot exceed 0.8%. It follows that a certain proportion of the delivery must originate from source $S2$. If the flow from the sources to the pipelines changes over time, so does the quality of the gas stored in the pipelines.

In agreement with [47], we assume that no blending process takes place inside the pipelines, which results in batches of possible unequal quality. Hence, the notion of *heterogeneous batches*.

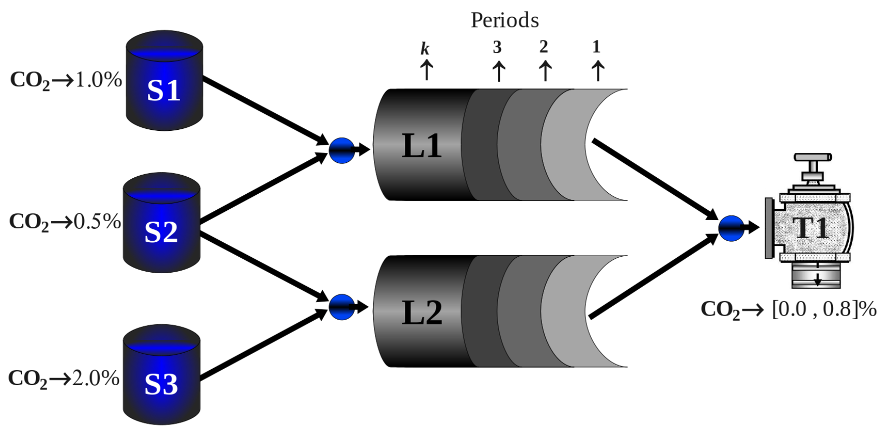


FIGURE 6.2: Heterogeneous gas – *The relative CO_2 content varies at the sources*

6.2.2 Notation

Let $G = (N, A)$ be an acyclic directed network representing a gas transmission pipeline system (see Fig. 6.1), where $N = S \cup L \cup T$ is the set of nodes partitioned into sources (S), pipelines (L) and terminals (T). The arc set A represents the set of links joining pairs of nodes, where each link is assumed to have a valve.

Let $K = \{1, \cdots, \kappa\}$ be the set of periods of length δ representing the planning horizon. Consider any two periods, k and l, where $l \leq k$. Some batch entered pipeline i in period l, and is henceforth referred to as batch (i,l). Due to extraction, the size of the batch may in period k be reduced, possibly to zero. Let β_i^{kl} be the size of batch (i,l) in period k. Clearly, we have $\beta_i^{ll} \geq \beta_i^{l+1,l} \geq \cdots \geq \beta_i^{\kappa l} \geq 0$.

Let y_i^{kl} be a binary variable, where $y_i^{kl} = 1$ means that batch (i,l) in pipeline i is extracted (fully or partly) in period k, and $y_i^{kl} = 0$ otherwise. Let r_i^{kl} be the proportion of batch (i,l) extracted in period k. Let α_i^{kl} be the amount of gas extracted from batch (i,l) in period k, and let x_{ij}^k be the flow through link $(i,j) \in A$ in period k.

For each source node $i \in S$, the total supply in period k is given by b_i^k. For each sink node $i \in T$, the total gas demand in period k is given by d_i^k. For each pipeline node $i \in L$, the minimum and maximum line-pack levels are given by Δ_i^{min} and Δ_i^{max}, respectively, with $0 < \Delta_i^{min} \leq \Delta_i^{max}$. Furthermore, the initial inventory is determined by β_i^{00}.

For any node $i \in V$, let $V_i^- = \{j \in V : (j,i) \in A\}$ be the set of start nodes of incoming arcs, and let $V_i^+ = \{j \in V : (i,j) \in A\}$ be the set of end nodes of outgoing arcs.

6.2.3 Building up batches in the pipelines

The constraints that we add to the model in order to build up heterogeneous gas batches in the pipelines during a given planning horizon are

$$\sum_{j \in V_i^-} x_{ji}^k - \sum_{j \in V_i^+} x_{ij}^k \leq f_i \cdot \left(\Delta_i^{max} - \sum_{l=0}^{k} \beta_i^{kl} \right), \quad \forall i \in L, k \in K \qquad (6.1)$$

$$\beta_j^{kk} = \sum_{j \in V_i^-} x_{ji}^k, \quad \forall j \in L, k \in K \qquad (6.2)$$

$$\Delta_i^{min} \leq \sum_{l=0}^{k} \beta_i^{kl} \leq \Delta_i^{max}, \quad \forall i \in L, k \in K. \qquad (6.3)$$

Constraint (6.1) says that the net input pipeline flow in any period has an upper bound proportional to the slack $\Delta_i^{max} - \sum_{l=0}^{k} \beta_i^{kl}$ in line-pack capacity. The proportionality factor is constant over time and denoted f_i.

Constraint (6.2) defines the new batch (j,k) of initial size $\sum_{i \in V_j^+} x_{ij}^k$. Finally, (6.3) imposes that the total line-pack in pipeline i ($\sum_{l=0}^{k} \beta_i^{kl}$) in any period must be kept within given specified limits. These bounds are physical or contractual limitations imposed by, e.g., the transporter.

Fig. 6.3 shows the relation between the initial size of batch (j,k) and the corresponding flow streams of which it is composed. Once the batches have

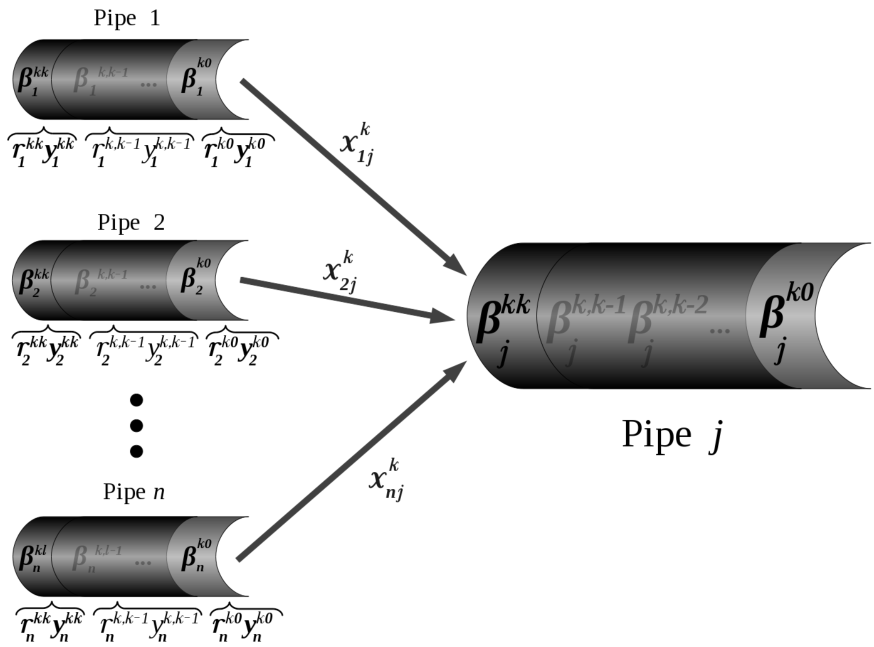


FIGURE 6.3: Relation between incoming flows in pipeline j and the batch built up in period k

entered the pipeline, they can be consumed. The consumption is however subject to physical and logical restrictions, which are studied next.

6.2.4 Consumption of batches

The constraints that we add to the model in order to systematically consume the batches stored in the pipelines are given by

$$\sum_{j\in V_i^+} x_{ij}^k - \sum_{j\in V_i^-} x_{ji}^k \le w_i \cdot \left(\sum_{l=0}^{k} \beta_i^{kl} - \Delta_i^{min}\right), \quad \forall i \in L, k \in K \tag{6.4}$$

$$r_i^{kl} \le y_i^{kl}, \quad \forall i \in L, 0 \le l \le k \le \kappa, \tag{6.5}$$

$$\alpha_i^{kl} = \beta_i^{ll} \cdot r_i^{kl}, \quad \forall i \in L, 0 \le l \le k \le \kappa \tag{6.6}$$

$$\sum_{l=0}^{k} \alpha_i^{kl} = \sum_{j\in V_i^+} x_{ij}^k, \quad \forall i \in L, k \in K \tag{6.7}$$

$$\beta_i^{k+1,l} = \beta_i^{kl} - \alpha_i^{kl}, \quad \forall i \in L, 0 \le l \le k \le \kappa. \tag{6.8}$$

$$y_i^{kl} \le \sum_{t=l-1}^{k} r_i^{t,l-1}, \quad \forall i \in L, 0 \le l \le k \le \kappa \tag{6.9}$$

$$\sum_{k\in K} r_i^{kl} \le 1, \quad \forall i \in L, 0 \le l \le \kappa \tag{6.10}$$

Constraint (6.4) imposes the input flow capacity in pipeline i to be proportional to the present line-pack $(\sum_{l=0}^{k} \beta_i^{kl} - \Delta_i^{min})$ after gas extraction in period k, where w_i denotes the constant proportionality factor over time.

Constraint (6.5) states that batch (i,l) can be extracted in period k only if $y_i^{kl} = 1$. (6.6) represents the extraction α_i^{kl} of batch (i,l) in period k. Consequently, (6.7) states that the total flow leaving pipeline i towards downstream pipelines in period k is given by the total gas extracted from the batches in the pipeline.

Since a batch may be consumed in fractions over time, (6.8) forces the model to update the contents of the batches in pipeline i from one time period to the next.

Constraint (6.9) basically applies *the FIFO principle of a queue*, i.e., the batch (i,l) can be consumed in period k (totally or partially), only if its predecessor is fully consumed during periods $l-1,\cdots,k$. (6.10) is a logical constraint which imposes that the sum of all proportions of gas $(\sum_{k\in K} r_i^{kl})$ extracted from batch (i,l) since it entered the pipeline up to the final period (κ) cannot exceed 1.

6.2.5 Gas quality estimation

Unlike source nodes, the gas quality at any other component of the network must be estimated accordingly to the gas streams that enter the network component in each period.

Keeping track of gas composition entails a modeling process that can briefly be described as follows.

Let us consider the network instance shown in Fig. 6.2. In the example, we assume only the relative CO_2 content in the natural gas mixture is of interest to the client. Let Λ_i^k be the relative CO_2 content in period k at node i. It is given as a weighted average of the CO_2 contents at its upstream neighbor nodes as

$$\Lambda_i^k = \frac{\sum_{j\in V_i^-} \Lambda_j^k x_{ji}^k}{\sum_{j\in V_i^-} x_{ji}^k}, \forall i \in N\backslash S, k \in K. \tag{6.11}$$

Eq. (6.11) defines the relative CO_2 content in node j as the weighted average of the relative CO_2 contents of all flow streams entering the node, where the flow values constitute the weights.

The flow leaving pipeline i in period k may be composed of several batches. We assume that these are blended upon extraction from i, and by following the principle of equation (6.11), the relative CO_2 content of the flow becomes $\Lambda_i^k = \sum_{l=0}^{k} \lambda_i^l \alpha_i^{kl} \Big/ \sum_{l=0}^{k} \alpha_i^{kl}$, where λ_i^l is the relative CO_2 content of batch (i,l).

By extending the arguments presented so far, let Γ be the set of gas quality parameters subject to constraints at any terminal. Let the variable λ_{ia}^{l} represent the quality parameter $a \in \Gamma$ of batch (i, l) in the current period. Similarly, let the variable Λ_{ia}^{k} represent the quality parameter $a \in \Gamma$ of the total flow leaving pipeline i, and the total flow entering sink node i in period k. The model constraints related to the quality requirements are then given by

$$\lambda_{ja}^{k} \cdot \beta_{j}^{kk} = \sum_{i \in V_j^+} \Lambda_{ia}^{k} \cdot x_{ij}^{k}, \quad \forall j \in L, k \in K, a \in \Gamma \tag{6.12}$$

$$\Lambda_{ia}^{k} \cdot \sum_{j \in V_i^+} x_{ij}^{k} = \sum_{l=0}^{k} \lambda_{ia}^{l} \cdot \alpha_{i}^{kl}, \quad \forall i \in L, k \in K, a \in \Gamma \tag{6.13}$$

$$\underline{\Lambda}_{ja}^{k} \leq \Lambda_{ja}^{k} \leq \overline{\Lambda}_{ja}^{k}, \quad \forall j \in T, k \in K, a \in \Gamma. \tag{6.14}$$

Constraint (6.12) imposes the quality of batch (i, k) as a result of the corresponding flow streams of which it is composed. (6.13) follows the same principle of equation (6.11) to impose the quality of flow streams leaving pipeline i based on the quality of the batches of which it composed. Let $[\underline{\Lambda}_{ja}^{k}, \overline{\Lambda}_{ja}^{k}]$ be a given interval of the quality required at terminal $j \in T$ for each quality parameter $a \in \Gamma$. Then, the quality at sink nodes is finally imposed by (6.14).

6.2.6 Flow capacities

Assuming that in any period k there is an upper bound b_i^k on the supply at source i, we arrive at the constraint

$$\sum_{j \in V_i^+} x_{ij}^{k} \leq b_i^k, \forall i \in S, k \in K. \tag{6.15}$$

Constraint (6.15) simply restricts that the sum of all gas streams leaving source node i in period k can be no larger than b_i^k.

At the terminals, it is assumed that no more than what is specified by contract regulations can be delivered. Thus, the following constraint is added to the model.

$$\sum_{i \in V_j^+} x_{ij}^{k} \leq d_j^k, \forall j \in T, k \in K, \tag{6.16}$$

where d_j^k is the maximum amount of gas required at terminal j in period k.

To reflect the physical resistance of the pipelines in each period, the following equation is included in the model:

$$\left(\sum_{j\in V_i^+} x_{ij}^k - \sum_{j\in V_i^-} x_{ji}^k\right)^2 = W_i\left(p_i^k - q_i^k\right), \forall i \in L, k \in K. \tag{6.17}$$

where W_i is the resistance factor of pipeline i, and p_i^k and q_i^k are the squared inlet and outlet pressure variables of the pipeline, respectively, in period k.

Eq. (6.17) refers to the Weymouth equation [105] introduced in Section 4.3.4 (see Chapter 4).

Every link $(i, j) \in A$ has a valve that is either open or closed. An open valve implies that the outlet pressure at pipeline i and the inlet pressure at pipeline j are equal. Since a closed valve obviously implies zero flow through the link, we have the constraint

$$x_{ij}^k(q_i^k - p_j^k) = 0, \forall (i, j) \in A, k \in K. \tag{6.18}$$

Eq. (6.18) essentially allows the model to open or close a valve between any pair of adjacent pipelines in the system. If $x_{ij}^k > 0$, valve (i, j) is open in period k, and (6.18) implies $p_j^k = q_i^k$. Correspondingly, if the pressures differ, the equation implies zero flow. By denoting the squared pressures in period k at source s and terminal t by q_s^k and p_t^k, respectively, (6.18) also holds for links (i, j) where either $i \in S$ or $j \in T$.

6.2.7 Final state conditions

The final state conditions of the pipelines are accomplished by imposing a minimum line-pack of adequate quality by the end of the planning horizon. Thus, the following constraints are also added to the model:

$$\sum_{l=0}^{\kappa} \beta_i^{\kappa+1,l} \geq \Delta_i^{final}, \qquad \forall i \in L, \tag{6.19}$$

$$\sum_{l=0}^{\kappa} \lambda_{ia}^{\kappa+1,l} \beta_i^{\kappa+1,l} = \mu_{ia} \cdot \sum_{l=0}^{\kappa} \beta_i^{\kappa+1,l}, \quad \forall i \in L, a \in \Gamma. \tag{6.20}$$

$$\underline{\mu}_{ia} \leq \mu_{ia} \leq \overline{\mu}_{ia}, \qquad \forall i \in L, a \in \Gamma, \tag{6.21}$$

where Δ_i^{final} is an input data representing the line-pack size required in pipeline i at the final state, μ_{ia} is a continuous decision variable representing the quality in pipeline i for each parameter $a \in \Gamma$ in the final period, and $\underline{\mu}_{ia}$ and $\overline{\mu}_{ia}$ are lower and upper quality bounds imposed in the final period.

Constraint (6.19) specifies that the gas remaining in pipeline i, which is given by the sum of batches created from period 0 to period κ must be at least equal to Δ_i^{final} at the end of the planning horizon (period $\kappa + 1$).

As Eq. (6.13) introduced in the previous section, (6.20) imposes the quality of the final inventory in pipeline i as the weighted average of all batches remaining in the pipeline in period $\kappa + 1$.

6.2.8 A MINLP Model

Summarizing all the above, we can now formulate a mixed-integer non-linear programming model as follows.

$$Obj = \max \sum_{k \in K} \sum_{j \in T} \sum_{i \in V_j^+} x_{ij}^k \tag{6.22}$$

$$\begin{aligned}
s.t.\quad & (x, p, q, \beta, r, y, \alpha, \lambda, \Lambda, \mu) \in \Omega, && && (6.23)\\
& p_i^L \le p_i^k \le p_i^U, && \forall i \in N, k \in K, && (6.24)\\
& q_i^L \le q_i^k \le q_i^U, && \forall i \in N, k \in K, && (6.25)\\
& x_{ij}^k \ge 0, && \forall (i,j) \in A, k \in K, && (6.26)\\
& p_i^k, q_i^k \ge 0, && \forall i \in N, k \in K, && (6.27)\\
& \beta_i^{kl}, r_i^{kl}, \alpha_i^{kl}, \lambda_i^{kl} \le 0, && \forall i \in L, 0 \le l \le k \le \kappa, && (6.28)\\
& y_i^{kl} \in \{0, 1\}, && \forall i \in L, 0 \le l \le k \le \kappa, && (6.29)\\
& \Lambda_{ia}^k \ge 0, && \forall i \in N \backslash S, a \in \Gamma, k \in K, && (6.30)\\
& \mu_{ia} \ge 0, && \forall i \in L, a \in \Gamma. && (6.31)
\end{aligned}$$

where $\Omega = \{x, p, q, \beta, r, y, \alpha, \lambda, \Lambda, \mu |$ *Eqs.* (6.1)–(6.21) *are satisfied.*$\}$ is the set of feasible solutions, and p_i^L, p_i^U and q_i^L, q_i^U are the lower and upper bounds, respectively, on the squared of the inlet and outlet pressures at node $i \in N$.

Table 6.1 shows a complete list of decision variables for the MINLP model, where all but variable y_i^{kl} are continuous decision variables.

A summary of the numerical results is provided next.

Table 6.1: Decision variables for the MINLP model proposed in Project 3.

Obj	=	*Objective function value*
		Variables defined in period $k \in K$:
x^k_{ij}	=	Total flow through link $(i, j) \in A$
p^k_i	=	Squared inlet pressure of pipeline $i \in L$
q^k_i	=	Squared outlet pressure of pipeline $i \in L$
β^{kl}_i	=	Amount of flow stored in batch l *in pipeline* $i \in L$
r^{kl}_i	=	Ratio of gas extracted from batch l *in pipeline* $i \in L$
y^{kl}_i	=	Activation of batch l *in pipeline* $i \in L$ *for gas extraction*
α^{kl}_i	=	Proportion of gas extracted from batch l in pipeline $i \in L$
λ^{kl}_{ia}	=	Gas quality of batch l in pipeline $i \in L$
Λ^k_{ia}	=	Weighted average quality of total flow stream leaving pipeline $i \in L$
μ_{ia}	=	Weighted average quality of the final inventory in pipeline $i \in L$

6.3 Overview of the numerical experiments

The aim of the numerical experiments is to examine the computability of the model proposed in the previous section, and thus to analyze what features make it more difficult to solve.

This is accomplished by means of a GAMS formulation and a global optimizer, BARON [131]. We submit 10 network instances to BARON and let $\kappa \in \{1, 3, 6\}$ for each case. Note that BARON is set to call MINOS [97] to solve the convex subproblems in each node of the search tree. Furthermore, a time limit of 3600 CPU-seconds is imposed on each application of BARON with a relative optimality tolerance $\varepsilon = 10^{-2}$. This implies that any feasible solution is considered to be optimal if the gap between the objective function value and its upper bound is below 1% of the objective function value. In instances where BARON fails to compute the global optimum, it may still provide an upper bound on the maximum flow to give some indications on the quality of the output.

The range of the size of the tested network instances is wide. The number of sources, pipelines, terminals, links and quality parameters range respectively from 2 to 18, from 2 to 11, from 3 to 14, from 10 to 176, and from 1 to 13.

6.3.1 Summing up the numerical results

For a planning horizon with 1 time period, all test cases were solved to optimality in less than 1 minute. When the number of periods was increased to 3, nine out of 10 cases were solved to optimality, whereas the remaining case showed an optimality gap of 15% once the time limit was reached. Concerning the CPU-time, six cases required less than 5 minutes, no more than 20 minutes in 2 cases, and up to 28 minutes in the remaining case.

For a planning horizon with 6 time periods, 6 out of 10 cases were solved to optimality, whereas the optimality gap found in the remaining cases was no larger than 20% after reaching the time limit imposed on BARON. Three cases required less than 4 minutes, no more than 30 minutes in 2 cases, and up to 54 minutes in the remaining case.

The number of variables, constraints and non-linear terms increases linearly with respect to κ. For $\kappa = 6$, the largest instance comprised more than 12000 constraints, 10000 variables, and up to 48000 non-linear terms.

6.4 Conclusions

In order to hedge against scheduled events that may affect the transport of natural gas via pipelines, the implementation of a strategy known as line-packing is required. The idea behind this strategy is to store gas temporarily in the pipeline system itself in order to consume it whenever it is required. This entails that a greater amount of natural gas can be supplied to delivery points during a given period of high demand than what it is currently injected at sources.

The optimization model is formulated as a mixed-integer non-linear program, which tackles the line-packing problem by building up and consuming heterogeneous batches in the pipelines over time. More precisely, the model maximizes the flow of natural gas in a transmission pipeline system, and keeps track of energy content and quality to meet market demand in a given multi-period horizon, thus maintaining the reliability of producers and transporters.

Several numerical experiments were conducted by means of a GAMS formulation and the application of a global optimizer, BARON. The aim was to assess the computability of the model. The numerical results showed that test cases of small and moderate size can be handled effectively by means of global optimization, even for the largest given planning horizon. Nevertheless, BARON showed to be sensitive to the number of time periods, and was time consuming when applied to the largest networks.

Chapter 7

Concluding remarks

Within three research projects on the optimization of natural gas transport in transmission pipeline systems, a number of various mathematical models, algorithms, and numerical experiments have been presented and discussed in this thesis. The proposed optimization methods are composed of NLP and MINLP models, as well as of exact and heuristic methods. In addition, the experimental analyses conducted on each project were devoted to gain insight into three major issues: 1) the assessment of the computability of the mathematical models, 2) the performance of the proposed optimization techniques, and 3) comparison of the proposed techniques with existing optimization algorithms and tools.

Project 1 focused on minimizing the total fuel consumption incurred by compressor stations installed in a gas pipeline system. The project was mainly devoted to tackle large natural gas pipeline systems with cyclic structures. After conducting a painstaking study on the NLP model introduced in Section 4.3, three different methodologies were proposed to effectively overcome both the difficulties encountered in the steady-state flow model, namely the non-linearity and non-convexity, as well as the weaknesses found in previously suggested optimization approaches.

As discussed in Chapter 4, the key to success in this project was to apply the strategic idea of discretizing the feasible operating domain of compressor stations, which in turn allowed the implementation of hybrid solution methods based on powerful optimization techniques such as DP, tabu search, and tree decomposition. Note that the idea of working within a discretized space has been successfully applied since the liquid pipeline optimization conducted in the late 1960s by Jefferson [66], until the non-traditional optimization technique suggested by Carter [15] in 1998.

The computational experiments conducted on each proposed optimization method, coupled with comparisons with typical approaches found in the literature, indicated that a continual improvement was achieved at each implementation. This is supported by the fact that the proposed algorithms were capable of effectively solving a wider variety of test instances, including difficult cases where previous methods had failed.

Project 2 focused on maximizing the gas flow through transmission network systems. Unlike previously suggested models, the proposed model admitted variations in gas specific gravity and compressibility. More precisely, we implemented the network model suggested in the literature where all arcs correspond to pipelines, and considered the pipeline flow and the gas pressure at each network node as the main decision variables. However, to support these decisions, the model also had to assess the gas specific gravity at each node, and the gas compressibility at each pipeline. Thus, in addition to flow conservation constraints, the model includes constraints that relate flow to pressure while considering the specific gravity and compressibility variations. Constraints defining the gas properties as functions of flow and pressure variables are also included.

Since global optimization is challenged by the non-convex feasible domain of the model, thus becoming time consuming, a heuristic approach was also proposed.

Several numerical experiments were conducted to support the need for a more accurate model and to show the effectiveness of the proposed heuristic approach. Through experiments, it was proved that the application of traditional models in which the variation of specific gravity or compressibility or both are neglected, tends to yield misleading results. The numerical experiments also showed that the heuristic is capable of providing optimal or near-optimal solutions to the proposed model. It also outperforms a multistart local optimization procedure.

Project 3 focused on optimizing the line-pack management in pipeline systems. Here, the network model included three different sets of nodes to represent sources, pipelines and terminals, and a set of arcs to indicate the links between the nodes.

To overcome the consequences caused by unforeseen events, it requires a stochastic model, which is out of the scope of this thesis. However, whenever such events can be predictable, the fundamentals of the deterministic optimization can be applied. Hence, a MINLP model was proposed for maximizing the flow through a pipeline system in order meet market demand over a given planning horizon. The key of the model was to build up line-packs in pipelines over time, and to consume them whenever so is required in order to comply with contract regulations.

Unlike previously suggested models, the proposed model allowed the line-pack in the pipelines to be composed of batches having different gas composition. This was a consequence of the fact that the market imposes quality constraints and gas sources may supply gas of different quality. Hence, the model is enabled to keep track of gas composition at all network components.

The numerical results conducted on the model showed that test cases of small and moderate size can be handled effectively by means of global optimization. Nevertheless, the global optimizer (BARON) showed to be sensitive to the number of time periods when applied to the largest networks, thus resulting to be time consuming.

In general, the systematic design of various sets of test instances produced in each project becomes an inherent contribution in this thesis. The aim of this design was to assess the proposed optimization methods in terms of efficiency and effectiveness when applied to large test instances.

A final remark: Regardless of the promising results shown and discussed in detail in the previous chapters, the findings may still be far from a successful practical implementation. Unfortunately, it seems to be a common agreement that the inherent transition between theory and practice requires an extra effort that takes the theoretical fundamentals beyond a traditional developing phase. Conspicuous breakthroughs, such as the simplex or interior point methods during the last century, the extensively developed metaheuristics, among others, have already shown that this transition is possible and worth the effort no matter the number of external considerations that have to be done.

References

[1] Abbaspour, M. & Chapman, K.S. Nonisothermal transient flow in natural gas pipeline. Journal of Applied Mechanics, ASME Transactions, Vol. 75 (3), pp. 181-188, doi:10.1115/1.2840046 (2008) 8, 85

[2] Albers, D.J., Reid, C. & Dantzig, G.B. An Interview with George B. Dantzig: The Father of Linear Programming. The College Mathematics Journal, Vol. 17 (4), pp. 293-314, Sep. (1986) 51

[3] Allison, I.S. & Palmer, D.F. *Geology, the science of a changing Earth.* McGraw-Hill, New York, USA. (1980) 19

[4] Anglard, P. & David, P. Hierarchical steady state optimization of very large gas pipelines. In: Proceedings of the 20th PSIG annual meeting, Toronto. (1988) 59

[5] Bakhouya, B. & De Wolf, D. Solving gas transmission problems by taking compressors into account. HEC Ecole de Gestion de l'Université de Liége (ULG), Working Paper 2008 27/10. (2008) 59

[6] Bakhouya,B. & De Wolf, D. The gas transmission problem when the merchant and the transport functions are disconnected. HEC Ecole de Gestion de l'Université de Liége (ULG), Working Paper 2007 01/01. (2007) 8, 86

[7] Bellman, R. *Dynamic Programming*. Princeton University Press, Princeton, USA. (1957) 7, 56, 78

[8] Blum, C. & Roli, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, Vol. 35 (3), pp. 268-308. (2003) 47, 48

[9] Bonnefoy, O., Gruy, F. & Herri, J.M. Van der Waals interactions in systems involving gas hydrates. Journal of Fluid Phase Equilibria, Vol. 231 (2), pp. 176-187. (2005) 20

[10] Borraz-Sánchez C., Haugland D. A tree decomposition algorithm for minimizing fuel cost in gas transmission networks. In: Proceedings of the 39th International Conference: Computers & Industrial Engineering (cie2009), pp 244-249, ISBN: 978-1-4244-4135-8 (2009) 8, 86

[11] Borraz-Sánchez, C. & Ríos-Mercado, R.Z. A Non-sequential Dynamic Programming Approach for Natural Gas Network Optimization, WSEAS Transactions on Systems, Vol. 3 (4), pp. 1384-1389. (2004) 8, 57, 86

[12] Borraz-Sánchez, C. & Ríos-Mercado, R.Z. A Procedure for Finding Initial Feasible Solutions on Cyclic Natural Gas Networks. In Proceedings of the 2004 NSF Design, Service and Manufacturing Grantees and Research Conference, Dallas, USA. (2004) 68

[13] Borraz-Sánchez, C. & Ríos-Mercado, R.Z. Improving the operation of pipeline systems on cyclic structures by tabu search, Computers & Chemical Engineering, Vol. 33 (1), pp. 58-64. (2009) 8, 86

[14] Broeker, R.J. A New Process for the Transportation of Natural Gas. In: Proceedings of the 1st International Conference on Liquified Natural Gas. (1974) 18, 27

[15] Carter, R.G. Pipeline optimization: Dynamic programming after 30 years, In: Proceedings of the PSIG Meeting, pp. 1-19, Denver, CO, USA. (1998) 6, 8, 57, 64, 69, 74, 86, 117

[16] Carter, R.G., Gablonsky, J.M., Patrick, A., Kelley, C.T. & Eslinger, O.J. Algorithms for noisy problems in gas transmission pipeline optimisation. Optimization and Engineering, Vol. 2 (2), pp. 139157. (2002) 58

[17] Carter, R.G. & Rachford, H.H. Jr. Optimizing line-pack management to hedge against future load uncertainty. PSIG 0306. (2003) 9, 106, 107

[18] Cedigaz. Cedigaz News Report. Vol. 47, No. 3, January 18. (2008) 4, 5

[19] Cedigaz. Natural gas in the World. November 2009 Edition, France. (2009) 4

[20] Chaczykowski, M. Transient flow in natural gas pipeline - The effect of pipeline thermal model. Applied Mathematical Modelling, Vol. 34, pp. 1051-1067. (2010) 8, 85

[21] Chapple C., Paula. Terminal de regasificación - MARINERO. HitoTecnológico, Magazine BIT, Vol. 60, pp. 32-38, May. (2008)
http://www.gnlquintero.com 5

[22] Chebouba, A., Yalaoui, F., Amodeo, L., Smati, A. & Tairi, A. A new method to minimize fuel consumption of gas pipeline using ant colony optimization algorithms. In: Proceedings of the 2006 International Conference on Service Systems and Service Management (ICSSSM'06), Vol.2, Troyes, France. (2006) 58

[23] Cobos-Zaleta, D. Modelos de Optimización Entera Mixta No Lineal en Sistemas de Transporte de Gas Natural. Master Thesis, Graduated Program in System Engineering, UANL, Mexico. In Spanish. (2003) 58

[24] Cobos-Zaleta, D. & Ríos-Mercado, R.Z. A MINLP model for minimizing fuel consumption on natural gas pipeline networks. In: Proceedings of the XI-Latin-Ibero-American Conference on Operations Research, pp. 1-9, Concepcion, Chile. (2002) 58

[25] Contesse, L., Ferrer, J.C. & Maturana, S. A Mixed-Integer Programming Model for Gas Purchase and Transportation. Annals of Operation Research, Springer Science, Vol. 139 (1), pp. 39-63. (2005) 30, 106

[26] Crane, Co. Flow of fluids through valves, fittings and pipe. Technical Paper 410. Metric Edition, USA. (1982) 88

[27] Dantzig, G.B. Programming of Interdependent Activities: II Mathematical Model. *Econometrica*, Vol. 17 (3-4), 200-211. (1949) 41

[28] Davisson, E.G. A computer program for flow calculations. Tech Report ORNL-TM-1093, OAK Ridge National Laboratory, U.S. Atomic Energy Commision, USA. (1965) 8, 86, 90

[29] Davy, H. On a Combination of Oxymuriatic Gas and Oxygene Gas. Philosophical Transactions of the Royal Society, Vol. 101, pp. 155-162. (1811) 20

[30] De Wolf, D. & Smeers, Y. Mathematical properties of formulations of the gas transmission problem. SMG Preprint 94/12. Technical Report, Université Libre De Bruxelles. (1994) 8, 59, 86

[31] De Wolf, D. & Smeers, Y. The gas transmission problem solved by an extension of the simplex algorithm. Management Science, Vol. 46 (11), pp. 1454-1465. (2000) 8, 59, 86

[32] Dempsey, Judy. Gazprom plans to re-route controversial European pipeline. The New York Times, International Herald Tribune, August 23. (2007)
http://www.iht.com/articles/2007/08/23/news/pipeline.php?page=1. 27

[33] Dolan, E.D. & Moré, J.J. Benchmarking Optimization Software with Performance Profiles. Tech. Report ANL/MCS-P861-1200, Argonne Nat. Lab., January 2001
http://www-unix.mcs.anl.gov/ more/cops 49

[34] Dorigo, M. Optimization, Learning and Natural Algorithms (Phd Thesis). Politecnico di Milano, Italy. (1992) 48

[35] Dranchuk, P.M. & Abou-Kassem, J.H. Calculations of Z factors for natural gases using equations of state. J Can Pet Tech, Vol. 14 (3), pp 34-36. (1975) 90

[36] Dranchuk, P.M., Purvis, R.A. & Robinson, D.B. Computer calculations of natural gas compressibility factors using the Standing and Katz correlations. Inst of Pet Tech Series, No. IP-74-008, pp 1-13. (1974) 90, 92

[37] Drud, A. CONOPT: A GRG Code for Large Sparse Dynamic Nonlinear Optimization Problems. Mathematical Programming, Vol. 31 (2), pp. 153-191. (1985) 49, 72

[38] Drud, A.S. CONOPT: A System for Large Scale Nonlinear Optimization. Tutorial for CONOPT Subroutine Library, 16p, ARKI Consulting and Development A/S, Bagsvaerd, Denmark. (1995) 49, 72

[39] Eichfelder, G. Adaptive Scalarization Methdos in Multiobjective Optimization. Vector Optimization. Springer-Verlag Berlin Heidelberg, Germany. (2008) 42

[40] Feo, T.A., & Resende, M.G.C. Greedy randomized adaptive search procedures. Journal of Global Optimization, Vol. 6, pp. 109-133. (1995) 48

[41] Flores-Villarreal, H.J. & Ríos-Mercado, R.Z. Efficient operation of natural gas pipeline networks: Computational finding of high quality solutions. In: Proceedings of the International Applied Business Research Conference, Paper 352, pp. 1-7. Acapulco, Mexico. (2003) 58, 72

[42] Floudas, C.A. *Deterministic Global Optimization: Theory, methods and applications.* The Netherlands: Kluwer Academic Publishers. (2000) 44

[43] Floudas, C.A. & Visweswatan, V. A global optimization algorithm (GOP) for certain classes of non convex NLPs I Theory. Computers and Chemical Engineering, Vol. 14, pp. 1397-1417. (1990) 44

[44] Fourer, R. A simplex algorithm for piecewise-linear programming I: Derivation and proof. Mathematical Programming, Vol. 33, pp. 204-233. (1985) 59

[45] Fourer, R., Gay, D.M & Kernighan, B.W. A Modeling Language for Mathematical Programming. Management Science, Vol. 36, pp. 519-554. (1990) 48

[46] Fourer, R., Gay, D.M. & Kernighan, B.W. AMPL: A Modeling Language for Mathematical Programming. Second edition, Duxbury Press, Brooks/Cole Publishing Company, Monterrey, C.A., USA. (2002) 48

[47] Frimannslund, L. & Haugland, D. Line pack management for improved regularity in pipeline gas transportation networks. In: Safety, Reliability and Risk Analysis. Theory, Methods and Applications. CRC Press 2009 ISBN 9780415485135. pp.2963-2969. (2008) 9, 107, 108

[48] Fuller, Jack A. & Martinec, C. Lee. Operations Research and Operations Management: From selective optimization to system optimization. Journal of Business & Economics Research, Vol. 3 (7), pp. 11-16. (2005) 38

[49] GAMS Development Corporation. GAMS: The solver manuals. Washington, DC, USA. (2008) 8, 48, 49, 76, 99

[50] GAMS Development Corporation. *The GAMS model library.* http://www.gams.com/modlib/modlib.htm 86

[51] Gass, S.I. & Assad, A. An Annotated Timeline of Operations Research: An Informal History. International Series in Operations Research and Management Science, *Kluwer Academic Publisher*, USA. (2005) 38, 39

[52] Gill, P.E., Murray, W. & Saunders, M.A. User's Guide for SNOPT Version 7: Software for Large-Scale Nonlinear Programming. (2006) http://www.sbsi-sol-optimize.com/index.htm 49

[53] Ginzburg, Y. ANG storage as a technological solution for the "chicken-and-egg" problem of NGV refueling infrastructure development. In Proceedings of 23^{rd} World Gas Conference, Amsterdam. (2006) 18

[54] Glover, F. Tabu Search: A Tutorial. *Interfaces*, Vol. 20 (4), pp. 74-94. (1990) 70

[55] Glover, F. & Laguna, M. *Tabu Search.* Kluwer, Boston, USA. (1997) 57, 69, 70

[56] Glover, F. & Kochenberger, G.A. (Editors) *Handbook of Metaheuristics.* International series in Operations Research & Management Science, Kluwer Academic Publishers. (2003) 47

[57] Glover, F. & Laguna, M. *Tabu Search.* Kluwer Academic Publishers. (1997) 48

[58] Gopal, V.N. Gas Z-factor equations developed for computers. Journal of Oil and Gas, Aug 8, pp. 58-60. (1977) 90

[59] Grossmann, I.E. *Global optimization in engineering design.* Dordretcht, The Netherlands: Kluwer Academic Puiblishers. (1996) 44

[60] Guo, Boyun & Ghalambor, Ali. *Natural Gas Engineering Handbook.* University of Louisiana at Lafayette. Gulf Publishing Company, Houston, USA. ISBN 0-9765113-3-9. (2005) 19

[61] Headquarters Department of the Army. Electrical design, cathodic protection. Technical Manual, Washington, D.C. (1985) 28

[62] Holland, J.H. Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, MI. (1975) 48

[63] Hopfield, J.J. & Tank, D.W. Neural Computation of Decisions in Optimization Problems. Biological Cybernetics, Vol. 52, pp. 141-152. (1985) 48

[64] Ingersoll, J.G. Natural Gas Vehicles. Fairmont Press Inc., Lilburn, GA, USA. (1996) 16, 17

[65] Instromet, Inc. Regulator Station Handbook. Gas measurement and control equipment. Publications by Instromet, TDC, UK. (2001) 31

[66] Jefferson, J.T. Dynamic programming. Journal of Oil and Gas, pp. 102-107. (1961) 56, 117

[67] Jeníček, T. Steady-state optimization of gas transport. In: Proceedings of the 2^{nd} International Workshop SIMONE on Innovative Approaches to Modelling and Optimal Control of Large Scale Pipeline Networks, September. (1993) 58

[68] Jensen, W.B. "The Origin of the Bunsen Burner". Journal of Chemical Education, Vol. 84 (4), p. 518. (2005) 16

[69] Kalvelagen, E. An improved formulation for a gas transportation problem. GAMS Development Corporation, USA. Scientific Literature Digital Library and Search Engine, Doi:10.1.1.13.4509. (2007) 8, 86

[70] Kalvelagen, E. *Modeling with Excel+OML, a practical guide.* A tutorial from Amsterdam Optimization Modeling Group, LLC. (2009)
www.amsterdamoptimization.com/models/msf/oml.pdf 48

[71] Katz, D.L., Cornell, D., Kobayashi, R., Poettmann, F.H., Vary, J.A., Elenbaas, J.R. & Weinaug, C.F. Properties of natural gases and volatile hydrocarbon liquids, Chap 4 in Handbook of Natural Gas Engineering, McGraw-Hill Book Co, New York, NY, USA, pp 94-188. (1959) 90

[72] Karmarkar, N. A new polynmial time algorithm for linear programming. Combinatorica, Vol. 4 (4), pp. 373-395. (1984) 41

[73] Kirkpatrick, S., Gelatt, C.D. & Vecchi, M.P. Optimization by Simulated Annealing. *Science*, Vol. 220 (4598), pp. 671-680. (1983) 48, 57

[74] Klee, V. & Minty, G.J. *How Good is the Simplex Algorithm?* In O. Shisha, editor, *Inequalities, III.* Academic Press, New York, NY, pp. 159-175. (1972) 41

[75] Koppl Pipeline Services, Inc. Hot tapping - Quality, value and service. Montebello, CA, USA. (2008)
http://www.koppl.com/ 30

[76] Korte, B. & Vygen, J. Combinatorial Optimization: Theory and Algorithms. 4th. Springer Publishing Company, Incorporated. (2007) 45, 47

[77] Koster, A.M.C.A., van Hoesel, S.P.M. & Kolen, A.W.J. Optimal solutions for frequency assignment problems via tree decomposition, Lecture Notes in Computer Science, 1665, pp. 338-350. (1999) 7, 74

[78] Kuenen, J.P. Measurements concerning the surface of Van Der Waals for mixtures of carbonic acid and methyl chloride. Amsterdam, pp. 422-424. Translation made by Dr. H. Kamerlingh Onnes in *Communications from the Laboratory of Physics*. University of Leiden. (1892) 26

[79] Lall, H.S. & Percell, P.B. A dynamic programming based gas pipeline optimizer. In A. Bensoussan, J. L. Lions (eds.): *Analysis and Optimization of Systems*, pp. 123-132, Springer, Berlin, Germany. (1990) 57

[80] Larson, R.E. & Wismer, D.A. Hierarchical control of transient flow in natural gas pipeline networks. In: Proceedings of the IFAC symposium on distributed parameter systems, Banff, Alberta, Canada. (1971) 59

[81] Lockermann, G. The centenary of the Bunsen burner. (Translated by Ralph E. Oesper) Journal of Chemical Education, Vol. 33 (1), pp. 20-22. (1956) 16

[82] Luongo, C.A., Gilmour, B.J. & Schroeder, D.W. Optimization in natural gas transmission networks: A tool to improve operational efficiency. *Technical Report*, Stoner Associates Inc., presented at the 3^{rd} SIAM conference on optimization, Boston, USA. (1989) 57

[83] Mahlke, D., Martin, A. & Moritz, S. A mixed integer approach for time-dependent gas network optimization. Optimization Methods & Software, Vol. 25 (4), pp. 625-644. (2009) 59

[84] Majorowicz, J.A., Hannigan, P.K. & Osadetz, K.G. Study of the Natural Gas Hydrate "Trap Zone" and the Methane Hydrate Potential in the Sverdrup Basin, Canada. Natural Resources Research, Vol. 11 (2), pp. 79-97. (2002) 20

[85] Martch, H.B. & McCall, N.J. Optimization of the Design and Operation of Natural Gas Pipeline Systems. Paper No. SPE-4006, Society of Petroleum Engineers of AIME. (1972) 56

[86] McMillan, G.K. Centrifugal and axial compressor control. IRP, Instrument Society of America, USA. (1983) 33

[87] Metaheuristics Network Website 2000. http://www.metaheuristics.net/. Visited in October 2009. 47

[88] Michot Foss, Michelle. Introduction to LNG - An overview on liquefied natural gas (LNG), its properties, organization of the LNG industry and safety considerations. Published by the Center for Energy Economics (CEE), Bureau of Economic Geology, Houston, USA. (2007) 18, 19

[89] Microsoft Solver Foundation. *Amsterdam Optimization and Microsoft Solver Foundation.* http://code.msdn.microsoft.com/solverfoundation http://www.amsterdamoptimization.com/solverfoundation.html 48

[90] Mladenović, N., & Hansen, P. Variable neighborhood search. Computers & Operations Research, Vol. 24 (11), pp. 1097-1100. (1997) 48

[91] Modisette, J.L. Equation of State Tutorial, In: Proceedings of the 32nd Annual Meeting of Pipeline Simulation Interest Group (PSIG), Savannah, GA, USA (2000) 88, 92

[92] Modisette, J.L. & Modisette, J.P. Transient and Succession-of-steady-states Pipeline Flow Models, Publication, Energy Solutions International Inc. (1998) http://www.energy-solutions.com/pdf/case_study_Modisette_Transient.pdf 86

[93] Mokhatab, S., Poe, W.A. & Speight, J.G. *Handbook of Natural Gas Transmission and Processing.* Gulf Professional Publishing, MA, USA. ISBN-13: 978-0-7506-7776-9 (2006) 16, 19, 33

[94] Mordecai, Avriel & Boaz, Golany. *Mathematical Programming for Industrial Engineers.* Marcel Dekker, Inc., New York, NY, USA. (1996) 39, 40

[95] Moritz, S. A mixed integer approach for the transient case of gas network optimization. Phd Thesis. Technische Universität Darmstadt, Germany. (2007) 60

[96] Müller-Merback, H. Heuristics and their design: a survey. *European Journal of Operational Research*, Vol. 8, pp. 1-23. (1981) 47

[97] Murtaugh, B.A. & Saunders, M.A. MINOS 5.1 User's guide. Tech Report SOL-83-20R, Systems Optimization Laboratory, Stanford University, Stanford, CA, USA. (1983) 8, 49, 80, 99, 115

[98] Neumaier, A. *Global Optimization – Software, Techniques and Applications.* Computational Mathematics group at the University of Vienna, Austria.
http://www.mat.univie.ac.at/ neum/glopt.html
http://www.mat.univie.ac.at/ neum/glopt/software_g.html 49

[99] Neumaier, A., Shcherbina, O., Huyer, W. & Vinkó, T. A comparison of complete global optimization solvers. Mathematical Programming, Vol. 103 (2), pp. 335-356. (2005) 49

[100] Nocedal, J. & Wright, Stephen. Numerical Optimization. Springer Series in Operations Research. Second Edition. New York, NY, USA. (2006) 42

[101] Nord Stream. The new gas supply route for Europe. (2010)
http://www.nord-stream.com/en/ 27

[102] O'Neill, R.P., Williard, M., Wilkins, B. & Pike, R.A Mathematical Programming Model for Allocation Of Natural Gas. Operations Research Vol. 27 (5), pp. 857-873. (1979) 59

[103] OGJ editors. "Norway approves Gudrun field development". Journal of Oil and Gas. *International Petroleum News and Technology*, June 18. (2010)
http://www.ogj.com/index/article-display/1489929928/articles
/oil-gas-journal/exploration-development-2/2010
/06/norway-approves_gudrun.html 4

[104] Oñate E. Límites de los Métodos Numéricos. International Center of Numerical Metods in Engineering. Publication CIMNE No. 191. (2000) 45, 51

[105] Osiadacz, A.J. Simulation and analysis of gas networks. Gulf Publishing Company, Houston, USA. (1987) 8, 62, 86, 88, 113

[106] Osiadacz, A.J. Dynamic optimization of high pressure gas networks using hierarchical systems theory. In: Proceedings of the 26th PSIG annual meeting, San Diego, USA. (1994) 59

[107] Osiadacz, A.J. & Bell, D.J. A simplified algorithm for optimization of largescale gas networks. Optimal Control Applications & Methods, Vol. 7, pp. 95-104. (1986) 59

[108] Osiadacz, A.J. & Górecki, M. Optimization of pipe sizes for distribution gas network design. In Proceedings of the 27th PSIG Annual Meeting, Albuquerque. (1995) 58

[109] Osiadacz, A.J. & Swierczewski, S. Optimal control of gas transportation systems. In: Proceedings of the 3^{rd} IEEE conference on control applications, pp. 795-796. (1994) 59

[110] Osman, I.H. & Laporte, G. Metaheuristics: A bibliography. Annals of Operations Research, Vol. 63, pp. 513-623. (1996) 47

[111] Papadimitriou, C.H. & Steiglitz, K. *Combinatorial Optimization: Algorithms and Complexity.* Dover Publications. (1999) 44, 46

[112] Pearl, P. *Heuristics: Intelligent Search Strategies for Computer Problem Solving.* Addison-Wesley, Reading. (1984) 47

[113] Percell, P.B. & Ryan, M.J. Steady-state optimization of gas pipeline network operation. In: Proceedings of the 19th PSIG Annual Meeting, Tulsa, USA, October. (1987) 57

[114] Pintér, J.D. Global Optimization with GAMS/LGO - Introduction, Usage, and Applications. PCS Inc, Halifax, NS. GAMS Global Optimization Workshop, Washington, DC, September. (2003) 49

[115] Pólya, G. *How to solve it.* Princeton University Press, Princeton, 1945. 46

[116] Pratt, K.F. & Wilson, J.G. Optimisation of the operation of gas transmission systems. Transactions of the Institute of Measurement and Control, Vol. 6 (5), pp. 261-269. (1984) 58

[117] Reeves, C.R. Modern Heuristic Techniques for Combinatorial Problems. Halstead Press (Wiley), New York, NY. (1993) 47

[118] Rhee, R.J. *The Socratic Method and the Mathematical Heuristic of George Pólya*, St. John's Law Review, pp. 881-898. (2007) 46

[119] Ríos-Mercado, R.Z., Kim, S. & Boyd, E.A. Efficient operation of natural gas transmission systems: A network-based heuristic for cyclic structures. Computers & Operations Research, Vol. 33 (8), pp. 2323-2351. (2006) 67

[120] Ríos-Mercado, R.Z., Wu, S., Scott, L.R. & Boyd, E.A. A reduction technique for natural gas transmission network optimization problems. Annals of Operations Research, Vol. 117 (1-4), pp. 217-234. (2002) 6, 8, 58, 66, 67, 68, 86

[121] Robertson, N. & Seymour, P.D. Graph minors II. Algorithmic aspects of treewidth, Journal of Algorithms, Vol. 7, pp. 309-322. (1986) 74, 75

[122] Rojey, A., Jaffret, C., Cornot-Gandolph, S., Durand, B., Jullin, S., and Valais, M. *Natural Gas Production, Processing, Transport.* Editions Technip, Paris, France. (1997) 16

[123] Savidge, J.L. Compressibility of natural gas, ISHM Lawrence Reid Award Recipient 2000, pp 12-26. (2000) 91

[124] Shapiro, A., Dentcheva D., & Ruszczyński, A. Lectures on stochastic programming: modeling and Theory. Society for Industrial and Applied Mathematics (SIAM) and Mathematical Programming Society, Philadelphia, PA, USA. (2009) 43

[125] Shashi Menon, E. Gas pipeline hydraulics. CRC Press, Taylor & Francis Group, LLc. Boca Raton, FL. (2005) 36, 88, 90, 91, 92, 94, 95

[126] Shell Co. Meeting the Energy Challenge. Shell booklet. Special advertorial section. (www.popsci.com/files/shell-booklet.pdf?ref=psc)
http://www.shell.com/home/content/environment_society/nef/e_diversity/ 4

[127] Sheriff, R.E. & Lloyd P.G. Exploration Seismology. Cambridge University Press, Cambridge. (1995) 24

[128] Sloan, E.D. Jr. *Clathrate Hydrates of Natural Gases.* Second edition, Marcel Dekker Inc., New York, USA. (1998) 21

[129] Speight, J.G. *Gas Processing: Environmental Aspects and Methods.* Butterworth Heinemann, Oxforx, England. (1993) 16

[130] Subbarayan, S. An Empirical Comparison of CSP Decomposition Methods, In: B.Hnich and K.Stergiou (Eds.): *Proceedings of the CP 2007 Doctoral Programme*, pp. 163-168, Providence RI, USA, September 23-27. (2007) 76

[131] Tawarmalani, M. & Sahinidis, N.V. Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. Mathematical Programming, Vol. 99 (3), pp. 563-591. (2004) 8, 9, 49, 76, 81, 99, 115

[132] U.S. Energy Information Administration. Environment – Energy-related emissions data & environmental analyses. Independent Statistics and Analysis. http://www.eia.doe.gov/environment.html 15

[133] Villalobos-Morales, Y., Cobos-Zaleta, D., Flores-Villarreal, H.J., Borraz-Sánchez, C. & 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, Birmingham, USA. (2003) 58, 72

[134] Villalobos-Morales, Y. & Ríos-Mercado, R.Z. Approximating the fuel consumption function on natural gas centrifugal compressors. In Proceedings of the 2002 NSF Design, Service and Manufacturing Grantees Research Conference. San Juan, Puerto Rico. (2002) 58

[135] Villalobos-Morales, Y. & Ríos-Mercado, R.Z. Preprocesamiento efectivo de un problema de minimización de combustible en sistemas de transporte de gas natural. *Revista Ingeniera de Sistemas*, Vol. 19, pp. 79-103. In Spanish. (2005) 58

[136] Visweswaran, V. & Floudas, C.A. Global optimization in engineering design. Kluwer book series in nonconvex optimization and its applications. Kluwer, Netherlands. (1996) 30

[137] Voβ, S. *Meta-heuristics: The State of the Art.* Lectures Notes in Computer Science, Springer-Verlag, Vol. 2148, pp. 1-23. ISBN:3-540-42898-4 (2000) 46

[138] WHO Working Group. Health Aspects of Air Pollution with Particulate Matter, Ozone and Nitrogen Dioxide. *Reports on World Health Organization*, Bonn, Germany, 13-15 January. (2003) 15

[139] Wilson, J.G., Wallace, J. & Furey, B.P. Steady-state Optimization of Large gas transmission systems. In Simulation and optimization of large systems, A.J. Osiadacz (Ed), Clarendon Press, Oxford. (1988) 59

[140] Wolsey, L.A. Integer Programming. Wiley-Interscience Series in Discrete Mathematics and Optimization, USA. (1998) 42

[141] Wong, P.J. & Larson, R.E. Optimization of natural gas pipeline systems via dynamic programming. IEEE Transactions on Automatic Control, Vol. 13 (5), pp. 475-481. (1968a) 8, 56, 86

[142] Wong, P.J. & Larson, R.E. Optimization of tree-structured natural-gas transmission networks. Journal of Mathematical Analysis and Applications, Vol. 24 (3), pp. 613-626. (1968b) 56

[143] Wu, S. Steady-State Simulation and Fuel Cost Minimization of Gas Pipeline Networks. Ph.D. Thesis. Houston, TX: University of Houston. (1998) 58

[144] Wu, S., Ríos-Mercado, R.Z., Boyd, E.A. & Scott, L.R. Model relaxations for the fuel cost minimization of steady-state gas pipeline networks. Mathematical and Computer Modeling, Vol. 31 (2-3), pp. 197-220. (2000) 6, 58, 61, 62, 63, 64

[145] Wu, Y., Lai, K.K. & Liu, Y. Deterministic global optimization approach to steady-state distribution gas pipeline networks. Optim Eng, Vol. 8, pp. 259-275. DOI 10.1007/s11081-007-9018-y (2007) 30, 44

[146] Zaghloul, J., Adewumi M. & Ityokumbul M.T. Hydrodynamic Modeling of Three-Phase Flow in Production and Gathering Pipelines. J. Energy Resour. Technol. - Transactions of the ASME, Vol. 130 (4), 043004 doi:10.1115/1.3000135. (2008) 20

[147] Zhenwu, L., Liming, S., Shitai, D., Zhiwen, D. & Guangcheng, X. Current situation and trend of geophysical technology in CNPC. Petrol. Explor. Develop., Vol. 37 (1), pp. 1-10. (2010) 24

[148] Zimmer, H.I. Calculating optimum pipeline operations. Technical Report, Natural Gas Company El Paso, presented at the AGA Transmission Meeting, Texas, USA. (1975) 57

Part II

Scientific Contributions

PAPER I

Improving the operation of pipeline systems on cyclic structures by tabu search*

Conrado Borraz-Sánchez   Roger Z. Ríos-Mercado[a]

* Published in: *Journal of Computers & Chemical Engineering*, Vol. 33 (1), pp. 58–64. (2009)
[a] Roger Z. Ríos-Mercado is currently an Associate Professor of Operations Research and Systems Engineering at Department of Mechanical and Electrical Engineering, Universidad Autónoma de Nuevo León (UANL), Mexico.

Computers and Chemical Engineering 33 (2009) 58–64

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

Short note

Improving the operation of pipeline systems on cyclic structures by tabu search

Conrado Borraz-Sánchez [a], Roger Z. Ríos-Mercado [b,*]

[a] Institute of Informatics, University of Bergen, Postboks 7803, 5020 Bergen, Norway
[b] Graduate Program in Systems Engineering, Universidad Autónoma de Nuevo León, AP 111-F, Cd. Universitaria, San Nicolás de los Garza, NL 66450, Mexico

A R T I C L E I N F O

Article history:
Received 30 November 2005
Received in revised form 1 July 2008
Accepted 7 July 2008
Available online 24 July 2008

Keywords:
Steady state
Natural gas
Transmission networks
Cyclic systems
Nonconvex problem
Dynamic programming
Tabu search

A B S T R A C T

In this paper, the problem of how to efficiently operate a natural gas transmission network under steady-state assumptions is considered. The problem is modeled as a nonlinear network optimization problem where the decision variables are mass flow rate in each arc and gas pressure in each node. The objective function to be minimized is the total amount of fuel consumed in the system by the compressor stations. In the past, several techniques ranging from classical gradient-based procedures to dynamic programming, for solving this difficult nonconvex problem have been applied with limited success, particularly when applied to cyclic network topologies. A cyclic system is defined as a network containing at least one cycle involving two or more compressor stations. In this paper we propose a hybrid metaheuristic procedure that efficiently exploits the problem structure. This hybrid procedure combines very effectively a nonsequential dynamic programming algorithm for finding an optimal set of pressure variables for a fixed set of mass flow rate varables, and short-term memory tabu search procedure for guiding the search in the flow variable space. The proposed procedure represents an improvement to the best existing approach to the best of our knowledge. In addition, empirical evidence over a number of instances supports the effectiveness of the proposed procedure outperforming a multi-start GRG method both in terms of solution quality and feasibility. Furthermore, to assess the quality of the solutions obtained by the algorithm, a lower bound is derived. It is found that the solution quality obtained by the proposed procedure is relatively good.

1. Introduction

In this paper, we address the problem of minimizing the fuel consumption incurred by compressor stations in a natural gas pipeline transmission system. During this process, energy and pressure are lost due to both friction between the gas and the pipes' inner wall, and heat transfer between the gas and the environment. To keep the gas flowing through the system, it is necessary to periodically increase its pressure, so compressor stations are installed through the network. It is estimated that compressor stations typically consume about 3–5% of the transported gas. This transportation cost is significant because the amount of gas being transported in large-scale systems is huge. On the other hand, even a marginal improvement in gas operations can have a significant positive impact from the economic standpoint, so this provides the main motivation from the practical side of the proposed work.

The problem is represented by a network, where arcs correspond to pipelines and compressor stations, and nodes correspond to their physical interconnection points. We consider two types of continuous decision variables: mass flow rates through each arc, and gas pressure level at each node. So, from the optimization perspective, this problem is modeled as a nonlinear program (NLP), where the cost function is typically nonlinear and nonconvex, and the set of constraints is typically nonconvex as well. It is well known that nonconvex NLP is NP-hard (Horst, Pardalos, & Thoai, 1995). This motivates the choice of the proposed heuristic approach.

The state of the art on research on this problem reveals a few important facts. First, there are two fundamental types of network topologies: noncyclic and cyclic. We would like to emphasize that, the former is a type of topology that has received most of the attention during the past 30 years. Several methods of solution have been developed, most of them based on dynamic programming (DP), which were focused on noncyclic networks.

In particular, as far as handling cyclic topologies are concerned, gradient search and DP approaches have been applied with little or limited success. The main limitation of the former is its local optimality status. The drawback of the latter is that its application is limited to problems where the flow variables are fixed so the final solution is "optimal" with respect to a prespecified set of flow variables. This is because cyclic topologies are a lot harder to solve.

In this paper, we proposed a novel solution methodology for addressing the problem of how to optimally operate the compressor stations in a natural gas pipeline system, focusing in cyclic

* Corresponding author.
E-mail addresses: Borraz-Sanchez@ii.uib.no (C. Borraz-Sánchez), roger@mail.uanl.mx (R.Z. Ríos-Mercado).

0098-1354/$ – see front matter © 2008 Published by Elsevier Ltd.
doi:10.1016/j.compchemeng.2008.07.009

topologies. The proposed technique combines a nonsequential DP technique (originally proposed by Carter, 1998) within a tabu search (TS) framework. For the past few years, TS has established its position as an effective metaheuristic guiding the design and implementation of algorithms for the solution of combinatorial optimization problems in a number of different areas (Glover & Laguna, 1997). In this case, even though we are dealing with a continuous optimization problem, the high nonconvexity of the objective function and the versatility of TS to overcome local optimality make TS, with an appropriate discrete solution space, a very attractive choice.

Empirical evidence over several instances with data taken from industry shows the efficiency of the proposed approach. A comparison with a multi-start GRG-based method demonstrates the significant superiority of the proposed procedure. The method represents an improvement over existing state-of-the-art approaches. Furthermore, in order to assess the quality of the solutions delivered by the algorithm, a lower bound was derived. It is shown that the optimality gaps found by our technique are less than 16%, most of them less than 10%, which represents a significant progress to the current state of the art in this area. The scientific contribution of this work is providing the best technique known to date, to the best of our knowledge, for addressing this type of problem in cyclic topologies.

The rest of this paper is organized as follows. In Section 2, we formally introduce the fuel consumption minimization problem (FCMP), describing its main features, modeling assumptions, and important properties. Then, in Section 3, we present a review of earlier approaches for this problem, highlighting the most related to our work, and how we attempt to exploit some of them. The proposed methodology is outlined in Section 4. A computational evaluation of the heuristic, including comparison with a multi-start GRG method, is presented in Section 5. Finally, we wrap up this work with the conclusions and directions for future research in Section 6.

2. Problem description

Pipeline system models can be mainly classified into steady-state and transient systems. Like all those previous works (reviewed in Section 3), here we assume a steady-state model. That is, our model provides solutions for systems that have been operating for a relatively large amount of time, which is a common practice in industry. Transient analysis has been done basically by descriptive models, because transient models are a highly intractable from the optimzation perspective. Optimization for transient systems remains as one of the great research challenges in this area. We also assume we work with a deterministic model, that is, each parameter is known with certainty. In terms of the compressor stations, we assume we work with centrifugal compressor units, which are the most commonly found in industry. As far as the network model is concerned, we assumed the network is balanced, that is, no gas is lost, and that each arc in the network has a prespecified direction.

2.1. The model

This model was originally introduced by Wu, Ríos-Mercado, Boyd, and Scott (2000). $V_s \subset V$ and $V_d \subset V$ are the set of supply and demand nodes, respectively. The set of arcs A is partitioned into a set of pipeline arcs A_p and a set of compressor station (or simply compressor) arcs A_c, i.e. $A_p \cup A_c$ and $A_p \cap A_c$. Let U_{ij} and R_{ij} the capacity and resistance of pipeline $(i,j) \in A_p$, respectively. Let P_i^L, P_i^U be the pressure lower and upper limits at node $i \in V$. Let B_i the net mass flow rate at node $i \in V$, where $B_i > 0$ if $i \in V_s$, $B_i < 0$ if $i \in V_d$, and $B_i = 0$ otherwise. The decision variables are given by x_{ij}, the mass flow rate in arc $(i,j) \in A$, and p_i, the pressure at node $i \in V$.

2.1.1. Formulation FCMP

$$\text{Minimize} \quad \sum_{(i,j)\in A_c} g_{ij}(x_{ij}, p_i, p_j) \tag{1}$$

$$\text{subject to} \quad \sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = B_i, \quad i \in V \tag{2}$$

$$x_{ij} \le U_{ij}, \quad (i,j) \in A_p \tag{3}$$

$$p_i^2 - p_j^2 = R_{ij} x_{ij}^2, \quad (i,j) \in A_p \tag{4}$$

$$p_i \in [p_i^L, p_i^U], \quad i \in V \tag{5}$$

$$(x_{ij}, p_i, p_j) \in D_{ij}, \quad (i,j) \in A_c \tag{6}$$

$$x_{ij}, p_i \ge 0, \quad (i,j) \in A, \ i \in V \tag{7}$$

The objective function (1) represents the total amount of fuel consumption in the system. We use a function g_{ij} in the following form:

$$g(x_{ij}, p_i, p_j) = \alpha x_{ij} \left\{ \left(\frac{p_j}{p_i} \right)^m - 1 \right\}, \quad (x_{ij}, p_i, p_j) \in D_{ij},$$

where α and m are assumed constant (and known) parameters that depend on the gas physical properties. Constraints (2) and (3) are the typical network flow constraints representing node mass balance and arc capacity, respectively. Constraint (4) represents the gas flow dynamics in each pipeline under the steady-state assumption. Constraints (5) denote the pressure limits in each node. These limits are defined by the compressor physical properties. Constraint (6) represents the nonconvex feasible operating domain D_{ij} for compressor station (i,j). The algebraic representation of D_{ij} is the result of curve fitting methods based on empirical data taken from the compressors. The details on the nature of the compressor station domain and how it is derived can be found in Wu et al. (2000). Fig. 1 shows a two-dimensional shape of this domain for p_i fixed. The precise model formulation for each of instances tested is available for download from http://yalma.fime.uanl.mx/ roger/ftp/. Finally, the mathematical model is bounded by nonnegative decision variables (7).

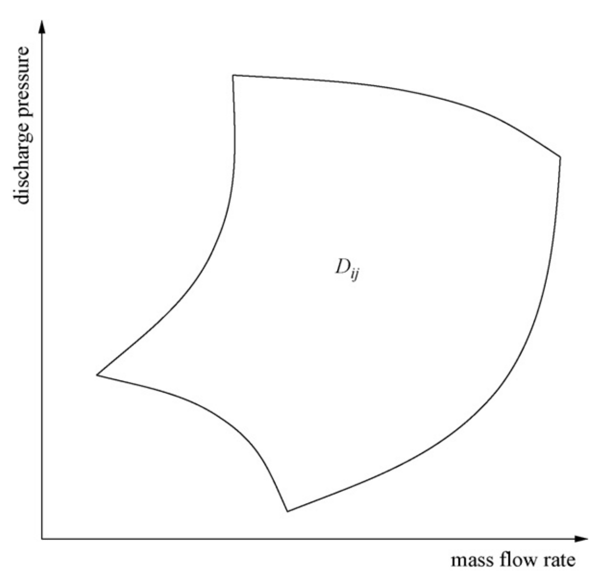


Fig. 1. 2D compressor station feasible domain for p_i (suction pressure) fixed.

3. Previous work

In this section, we review the most significant contributions over the last 30 years for solving the FCMP or related problems.

3.1. Methods based on dynamic programming

The key advantages of DP are that a global optimum is guaranteed to be found and that nonlinearity can be easily handled. In contrast, its application is practically limited to noncyclic networks, such as linear (also known as gun-barrel) or tree topologies, and that computation increases exponentially with the dimension of the problem, commonly referred as the curse of dimensionality.

DP for pipeline optimization was originally applied to gun-barrel systems in the late 1960s. It has been one of the most useful techniques due to both its computational behavior and its versatility for handling nonlinearity on sequential systems. DP was first applied to linear systems by Wong and Larson (1968a), and then applied to tree-structured topologies by Wong and Larson (1968b). A similar approach was described by Lall and Percell (1990), who allow one diverging branch in their system.

Luongo, Gilmour, and Schroeder (1989) published a hierarchical approach that allowed for both cycles and branches of arbitrary complexity. This represented significant progress in terms of finally addressing the issue of real world pipeline configurations. Their technique was no longer pure DP. Basically, DP was used to optimally describe the pieces of the pipeline that were arranged in a sequential manner. This typically reduced the system to a much smaller combinatorial problem, without any possibility of a recursive DP solution. A sufficiently small instance could be solved exactly via enumeration; otherwise it was solved inexactly using simulated annealing. This hierarchical approach worked very well for some complex pipelines, but for others the computational cost was very high.

The most significant work on cyclic networks known to date is due to Carter (1998) who developed a nonsequential DP algorithm, but limited to a fixed set of flows. This led to an interesting question of how to find the optimal setting of the flow variables and how to modify the current flow setting to obtain a better objective value. So with this in mind, recently Ríos-Mercado, Kim, and Boyd (2006) propose a network-based heuristic for modifying the flow values. The computational results showed an improvement with respect to Carter's NDP approach. A limitation of that work, though, is that the experimental phase was done over a small number of instances. Unfortunately, that code is no longer available for research purposes. In the present work, we use Carter's ideas and incorporate them within a tabu search scheme for iteratively adjusting the set of flows with great success. This will be further described in Section 4.

3.2. Methods based on gradient search

Percell and Ryan (1987) applied a different methodology based on a generalized reduced gradient (GRG) nonlinear optimization technique for noncyclic structures. One of the advantages of GRG, when compared with DP, is that they can handle the dimensionality issue relatively well, and thus, can be applied to cyclic structures. Nevertheless, being a method based on a gradient search, there is no guarantee for a global optimal solution, especially when there are discrete decision variables. Villalobos-Morales and Ríos-Mercado (2005) evaluated preprocessing techniques for GRG, such as scaling, variable bounding, and choice of starting solution, that resulted in better results for both cyclic and noncyclic structures. More recently, Flores-Villarreal and Ríos-Mercado (2003) performed an extensive computational evaluation of the GRG method over a large set of instances on cyclic structures with relative success. No comparison to DP was done in that work, so part of our contribution is to provide a comparison frame among Carter's NDP, GRG, and our method tested in the same set of instances.

3.3. Related models

Discrete decisions such as numbre of units operating within compressor stations are incoporated into a mixed-integer nonlinear programming model (MINLP). MINLP models in pipeline optimization have been studied by Pratt and Wilson (1984) and Cobos-Zaleta and Ríos-Mercado (2002). They present satisfactory results as they were able to find local optima for many instances tested.

Optimization of individual compressor stations has been studied by Osiadacz (1980), Percell and Reet (1989), and Wu, Boyd, and Scott (1996). Later, Wu et al. (2000) completed the analysis for the same problem, but considering several units within compressor stations. In a related work, some of the most important theoretical properties regarding pipeline networks are developed by Ríos-Mercado, Wu, Scott, and Boyd (2002).

Carter, Gablonsky, Patrick, Kelley, and Eslinger (2002) present some algorithms based on implicit filtering for a class of noisy optimization problems which also consider discrete decision variables with promising results. In a related work, Osiadacz and Górecki (1995) address a pipeline network design problem with modest success. More recently, Costa, de Medeiros, and Pessoa (2000) use linear programming for the optimal design of pressure relief header networks.

Optimization techniques have also been applied for transient (time dependent) models. For instance, Larson and Wismer (1971) propose a hierarchical control approach for a transient operation of a gunbarrel pipeline system. Osiadacz and Bell (1986) suggest a simplified algorithm for the optimization of the transient gas transmission network, which is based on a hierarchical control approach. The hierarchical control approach for transient models can be found in Anglard and David (1988), Osiadacz (1994), and Osiadacz and Swierczewski (1994). Some degree of success has been reported from these approaches as far as optimizing the compressor station subproblem. However, these approaches have limitations in globally optimizing the minimum cost.

See display Ríos-Mercado (2002, Chapter 18.8.3) for more references on optimization techniques applied to gas pipeline problems. It is important to mention that optimization approaches developed to date work well under some general assumptions; however, as the problems become more complex, the need arises for further research and effective development of algorithms from the optimization perspective.

4. Solution procedure

The proposed methodology (depicted in Fig. 2) proceeds as follows. In Step 1, a preprocessing phase is performed both to refine the feasible operating domain given by tightening decision variable bounds, and to reduce the size of the network by a reduction technique (motivated by the work of Ríos-Mercado et al., 2002). Then, in Step 2, a set of initial feasible flows (x) is found by two different methods: a classic assignment technique and a reduced graph algorithm.

In Step 3, a set of optimal pressures (p), for teh specified flow obtained before is found by applying a nonsequential DP (NDP) algorithm. At this point, we have an initial feasible solution (x, p) which enters a TS local search procedure.

```
function NDPTS()
Input: An instance of the FCMP.
Output: A feasible assigment (x, p).

1  Preprocessing();
2  x ← FindInitialFlow();
3  p ← NDP(x);
4  (x, p) ← TS((x, p));
5  return (x, p);
end NDPTS
```

Fig. 2. Pseudocode of NDPTS.

Within the TS, there are two main components for neighbor generation: a flow modification component and a pressure computation component. In the former, an attempt is made to find a different set of flows, and in the latter, a corresponding set of optimal pressure values is found by NDP. The TS is performed until a stopping criteria (in the case, a number of iterations) is met. As we know from theoretical properties of pipeline networks (Ríos-Mercado et al., 2002), the flow modification step is unnecessary for noncyclic topologies because there exists a unique set of optimal flow values which can be determined in advance at preprocessing. So, here we focus on cyclic topologies. For finding the optimal set of pressures, we implemented a NDP technique motivated by the work of Carter (1998). The overall procedure is called NDPTS. The methods employed in Steps 1 and 2 have been fairly well documented in our previous work (Borraz-Sánchez & Ríos-Mercado, 2004b), so, in the reminder of this section, we assume we have an initial feasible flow and provide a description of the NDP and the TS components, which is the core of the proposed work.

4.1. Nonsequential dynamic programming

We include in this section a brief description of the essence of the NDP algorithm. The details can be found in Borraz-Sánchez and Ríos-Mercado (2004a). Starting with a feasible set of flow variables, the NDP algorithm searches for the optimal set of nodal pressure values associated to that prespecified flow. Rather than attempting to formulate DP as a recursive algorithm, at a given iteration, the NDP procedure grabs two connected compressors and replace them by a "virtual" composite element that represents the optimal operation of both compressors. These two elements can be chosen from anywhere in the system, so the idea of "recursion" in classical DP does not quite apply here. After performing this step, the system has been replaced by an equivalent system with one less compressor station. The procedure continues until only one virtual element, which fully characterizes the optimal behavior of the entire pipeline system, is left. Aftwerwards, the optimal set of pressure variables can be obtained by a straight-forward backtracking process. The computational complexity of this NDP technique is $O(|A_c|N_p^2)$, where N_p is the maximum number of elements in a pressure range discretization.

4.2. Tabu search

We start the procedure with a given feasible solution (x, p). We define the nature of a feasible solution based on three basic components which are directly related with a cyclic network topology: (a) *static component*, a mass flow rate value not belonging to any cycle, (b) *variable component*, a mass flow rate value belonging to a cycle, and (c) *search component*, all pressure variables in the network. These components are depicted in Fig. 3. The search space employed by TS is defined by the flow variables x_{ij} only because once the flow rates are fixed, the pressure variables are optimally found by NDP. Furthermore, we do not need to handle the entire set of flow variables, but only one per cycle. This is so because once you fix a flow rate in a cycle, the rest of the flows can be uniquely determined. Thus, a given state is represented by a vector $\bar{x} = (x_{\alpha_1}, \ldots, x_{\alpha_m})$, where α_w is an arc that belongs to a selected cycle w. Note that this set of arcs is arbitrarily chosen, and that converting a flow from x to and from $\bar{x}$ is straightforward, so in the description x and $\bar{x}$ are used interchangeably.

Then a neighborhood $V(\bar{x})$ of a given solution $\bar{x}$ is defined as the set of solutions reachable from $\bar{x}$ via a slight modification of Δ_x units in each of its components. This is given by

$$V(\bar{x}) = \{x' \in R^m | x'_w = \bar{x}_w \pm j\Delta_x, j = 1, 2, \ldots, N_{size}/2, \ w = 1, \ldots, m\} \tag{8}$$

where N_{size} is the predefined neighborhood size and Δ_x accounts for the mesh size. Note that, for a given solution, we do not store the entire solution but only the flow in the selected arc to be modified. Note also that once this value is set, the rest of the flow variables in the cycle are easily determined, so in this sense, it is precisely this mass flow rate which becomes the attribute. Then the best $x' \in V(\bar{x})$ which is non-tabu is chosen and the corresponding subsets are updated accordingly. A tabu list (TL) stores recently used attributes, in our case, values of the x variables. The size of the TL (*tabu tenure*) controls the number of iterations a particular attribute is kept in the list. The search terminates after iter_max iterations.

5. Computational evaluation

The purpose of designing and setting up a database with problem instances is twofold. First, it is necessary for testing our proposed algorithms. Second, it aims at providing a common framework for benchmarking different algorithms. As far as we know, there is no such database for this type of problems. So this becomes an important contribution of this work.

There are three different kinds of network topologies: (a) linear or gun-barrel, (b) tree or branched, and (c) cyclic. Technically, the procedure for making this classification is as follows. In a given network, the compressor arcs are temporarily removed. Then each of the remaining connected components are merged into a big supernode. Finally, the compressor arcs are put back into their place. This new network is called the associated reduced network.

Linear topology: This corresponds to a linear arrangement of the compressor station arcs, that is, when the reduced network is a single path.
Tree topology: This occurs when the compressors are arranged in branches through the system, that is, when the reduced network is a tree.
Cyclic topology: This happens when compressors are arranged forming cycles with other compressor stations. That is, it refers to a cyclic reduced network.

As stated before, linear and tree topologies can be solved by dynamic programming since it has been shown that the flow variables in every arc can be uniquely determined. So in this work, our focus is on addressing cyclic topologies. Fig. 4 shows examples of cyclic topologies. A stripped node (shown with an ingoing arrow next to it) represents a supply node, a black node (shown with an outgoing arrow next to it) represents a demand noid, and

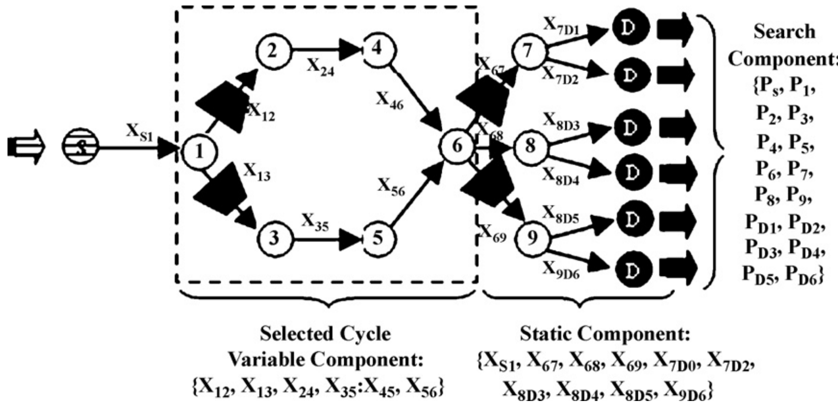


Fig. 3. Basic components of a feasible solution on a cyclic topology.

a white node is a transshipment node. A single directed arc joining two nodes represents a pipeline, and a directed arc with a black trapezoid joining two nodes represents a compressor arc.

So in the instance database tested below, a name net-x-m cn represents an instance of type x, $x \in \{a, b, c\}$, with m nodes and n compressor arcs. In addition, a suffix -Cy present means the instance uses compressors type y, where y is one of nine different type of compressors used in industry. This database is available at: http://yalma.fime.uanl.mx/ roger/ftp/, or directly from the authors upon request. Each instance is given as a GAMS file.

The proposed TS was developed in C++ and run on a Sun Ultra 10 workstation under Solaris 7. All of the compressor-related data, described in Villalobos-Morales, Cobos-Zaleta, Flores-Villarreal, Borraz-Sánchez, and Ríos-Mercado (2003), was provided by a consulting firm in the pipeline industry. For the tabu list size and the neighborhood size, several preliminar experiments were done using values of {5, 8, 10} and {20, 30, 40}, respectively. For the experiments we use the following values for the algorithmic parameters which were found to produce the best results in preliminar fine-tuning computations: Iteration limit (iter_max = 100), discretization size in $V(x)(\Delta_x = 5)$, discretization size for pressure variables ($\Delta_p = 20$), tabu tenure $T_{tenure} = 8$, and eighborhood size $N_{size} = 20$. In order to assess the effectiveness of the proposed procedures, we apply the algorithms to solving several instances under different cyclic network topologies on the same platform.

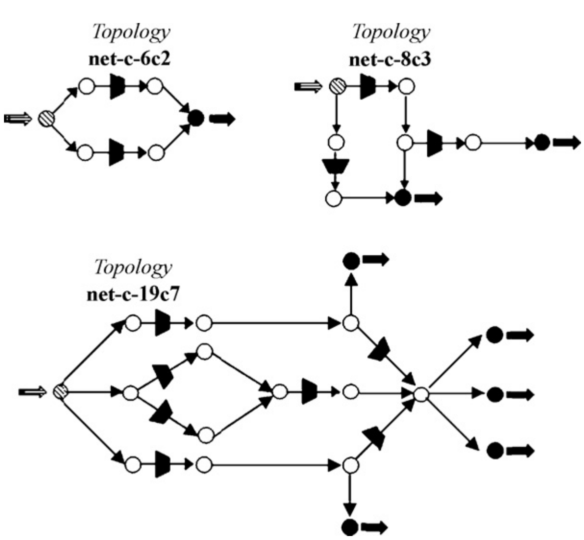


Fig. 4. Cyclic topology instances.

It is evident that the proposed NDPTS approach dominates NDP. This has been verified empirically, where the NDTPS has reported improvements in solution quality of up to 27% with respect to NDP. This comparison between NDPTs and NDP is presnted in (Borraz-Sánchez & Ríos-Mercado, 2005). In this paper, we present a comparison between the proposed NDPTS and the best GRG-based implementation known to date. In a second experiment, we provide evidence of the quality of the solution reported by NDPTS by comparing to a lower bound.

Table 1 shows a comparison between the GRG and NDPTS on cyclic networks. For the GRG we used the implementation by Flores-Villarreal and Ríos-Mercado (2003) within a multi-start strategy. That is, given that GRG is basically a local search method, the idea is to apply GRG from multiple different starting solutions for an amount of time equal to the time used by the NDPTS in each instance. In preliminary work, it was also observed that the multi-start GRG produced better results than the single application of the GRG as expected.

The first column shows the instances tested. The second column shows the total number of iterations employed by the multi-start GRG method. The third and fifth column show the GRG and NDPTS solution, respectively. The fourth column shows the running time of both methods. The last column shows the relative improvement of NDPTS over GRG given by

$$RI = \frac{g_{GRG} - g_{NDPTS}}{g_{NDPTS}} \times 100\%,$$

where g_Z denotes the objective function value found by method Z and Z being any of GRG or NDPTS.

First, the NDPTS obtained solutions to all instances tested, whereas GRG failed for four of these, that is, for the four harder instances the GRG could not find feasible solutions. The results indicate that NDPTS procedure outperforms GRG in terms of solu-

Table 1
Comparison between GRG and NDPTS

Instance	Iterations	GRG	Time	NDPTS	RI (%)
net-c-6c2-C1	8712	2,312,548.24	271.72	2,288,252.53	1.06
net-c-6c2-C4	8535	1,393,061.12	270.02	1,393,001.99	0.004
net-c-6c2-C7	9637	1,210,687.22	272.33	1,140,097.39	6.19
net-c-10c3-C2	7581	5,811,713.60	288.92	4,969,352.82	16.95
net-c-10c3-C4	7633	4,751,940.94	283.61	2,237,507.93	112.37
net-c-15c5-C2	5040	6,219,045.57	228.32	4,991,453.59	24.59
net-c-15c5-C4	5377	3,554,598.11	317.26	3,371,985.41	5.41
net-c-15c5-C5	10040	Not found	334.01	7,962,687.43	N/A
net-c-17c6-C1	9654	Not found	368.12	8,659,890.72	N/A
net-c-19c7-C4	8906	Not found	393.45	8,693,003.78	N/A
net-c-19c7-C8	18574	Not found	398.72	7,030,280.45	N/A

Table 2
Solution quality

Instance	LB	NDPTS	Gap (%)
net-c-6c2-C1	2,287,470.58	2,288,252.53	0.03
net-c-6c2-C4	1,392,354.29	1,393,001.99	0.05
net-c-6c2-C7	949,909.48	1,140,097.39	16.68
net-c-10c3-C2	4,303,483.50	4,969,352.82	13.40
net-c-10c3-C4	2,015,665.98	2,237,507.93	9.91
net-c-15c5-C2	4,955,752.90	4,991,453.59	0.72
net-c-15c5-C4	3,103,697.48	3,371,985.41	7.96
net-c-15c5-C5	6,792,248.08	7,962,687.43	14.69
net-c-17c6-C1	8,129,730.11	8,659,890.72	6.12
net-c-19c7-C4	7,991,897.18	8,693,003.78	8.06
net-c-19c7-C8	5,897,768.92	7,030,280.45	16.10

tion quality. In the instances where both procedures found feasible solutions, NDPTS obtains solution of significantly better quality than those obtained by GRG as can be observed from the relative improvement of NDPTS over GRG. In terms of computational effort, both procedures employed the same amount of time in a range of 270–400 s.

To assess the quality of the solutions delivered by the algorithm it is necessary to derive a lower bound. Now, deriving lower bounds for a nonconvex problem can become a very difficult task. Obtaining convex envelopes can be as difficult as solving the original problem. However, for this problem we note two important facts that lead us to an approximate lower bound. First, by relaxing constraint (4) in model FCMP the problem becomes separable in each compressor station. That is, the relaxed problem consists of optimizing each compressor station individually. Now, this is still a nonconvex problem, however, we exploit the fact that in each compressor, the objective is a function of three variables only, so we build a three-dimensional grid on these three variables and perform an exhaustive evaluation for finding the global optimum of the relaxed problem (for a specified discretization).

Table 2 shows these results. The first column displays the instances tested, the second and third columns show the lower bound and the best value found by the heuristic, respectively, and the last column shows the relative optimality gap obtained by NDPTS. As can be seen from the table, all of the tested instances have a relative optimality gap of less than 17%, 7 out of 11 instances tested have a relative gap of less than 10%, and three of these observed an optimality gap of less than 1%. This shows the effectiveness of the proposed approach. Finally, although our NDPTS algorithm finds better solutions than the GRG method or the simple NDP, it is more computationally expensive. In general, any additional time leading to even small improvements can be easily justified since the costs involved in natural gas transportation are relatively huge.

Fig. 5 shows the convergence of the NDPTS algorithm on instance net-c-10c3-C1. It can be seen how, at some iterations, the solution deteriorates but then it improves to a better solution, which illustrates how getting stuck at a local optimum is overcome by the TS mechanism. This figure also shows that the algorithm often did not improve beyond 50 iterations. In fact, we have observed similar behavior in all other tested instances.

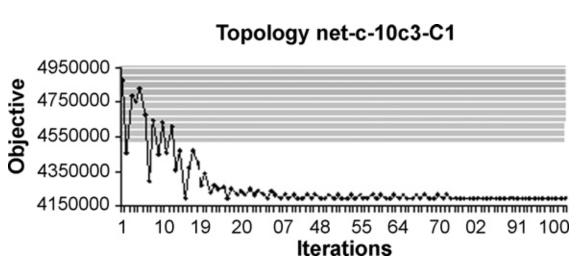


Fig. 5. Convergence on instance net-c-10c3-C1.

6. Conclusions

In this work we have proposed a hybrid heuristic based on NDP and TS for a very difficult problem arising in the natural gas pipeline industry. The NDPTS implementation, based primarily in a short-term memory strategy, proved very successful in the experimental work as it was able to obtained solutions of much better quality than those delivered by earlier GRG-based approaches when tested on a number of instances with data taken from industry. In addition, the way the method operates clearly produces better solutions that those found by Carter's NDP method. This represents, to the best of our knowledge, a significant contribution to the state of the art in this area of work. Other contributions include the evaluation of a simple lower bounding scheme and a data set collection which can be used for benchmarking.

There are still many areas for forthcoming research. The proposed procedure is a basic short-term memory tabu search. It would be interesting to incorporate advanced TS strategies such as intensification and diversification. In addition, one of the great challenges in the industry is to address time-dependent systems from the optimization perspective.

Acknowledgments

This research was supported by the Mexican National Council for Science and Technology (CONACYT grant J33187-A) and Universidad Autónoma de Nuevo León under its Scientific and Technological Research Support Program (UANL-PAICYT grant CA820-04). We are grateful to the reviewers whose comments helped us to improve the presentation of this work.

References

Anglard, P., & David, P. (1988). Hierarchical steady state optimization of very large gas pipelines. In *Proceedings of the 20th PSIG annual meeting*, Toronto.

Borraz-Sánchez, C., & Ríos-Mercado, R. Z. (2004). A non-sequential dynamic programming approach for natural gas network optimisation. *WSEAS Transactions on Systems*, *3*(4), 1384–1389.

Borraz-Sánchez, C., & Ríos-Mercado, R. Z. (2004b). A procedure for finding initial feasible solutions on cyclic natural gas networks. In *Proceedings of the 2004 NSF design, service and manufacturing grantees and research conference*, Dallas

Borraz-Sánchez, C. & Ríos-Mercado, R. Z. (2005). *Improving the operation of pipeline systems on cyclic structures by tabu seach*. Technical Report PISIS-2005–04, Graduate Program in Systems Engineering, Universidad Autónoma de Nuevo León, San Nicolás de los Garza, Mexico.

Carter, R. G. (1998). Pipeline optimization: Dynamic programming after 30 years. In *Proceedings of the 30th PSIG annual meeting*, Denver.

Carter, R. G., Gablonsky, J. M., Patrick, A., Kelley, C. T., & Eslinger, O. J. (2002). Algorithms for noisy problems in gas transmission pipeline optimisation. *Optimization and Engineering*, *2*(2), 139–157.

Cobos-Zaleta, D., & Ríos-Mercado, R. Z. (2002). A MINLP model for minimizing fuel consumption on natural gas pipeline networks. In *Proceedings of the XI Latin-Ibero-American conference on operations research*, Concepción, Chile.

Costa, A. L. H., de Medeiros, J. L., & Pessoa, F. L. P. (2000). Optimization of pressure relief header networks: A linear programming formulation. *Computers and Chemical Engineering*, *24*(1), 153–156.

Flores-Villarreal, H. J., & Ríos-Mercado, R. Z. (2003). Computational experience with a GRG method for minimizing fuel consumption on cyclic natural gas networks. In N. E. Mastorakis, I. A. Stathopulos, C. Manikopoulos, G. E. Antoniou, V. M. Mladenov, & I. F. Gonos (Eds.), *Computational methods in circuits and systems applications* (pp. 90–94). Athens, Greece: WSEAS Press.

Glover, F., & Laguna, M. (1997). *Tabu search*. Boston: Kluwer.

Horst, R., Pardalos, P. M., & Thoai, N. V. (1995). *Introduction to global optimization*. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Lall, H. S., & Percell, P. B. (1990). A dynamic programming based gas pipeline optimizer. In A. Bensoussan & J. L. Lions (Eds.), *Analysis and optimization of systems, volume 144 of lecture notes in control and information sciences* (pp. 123–132). Berlin: Springer-Verlag.

Larson, R. E. & D. A. Wismer (1971). Hierarchical control of transient flow in natural gas pipeline networks. In *Proceedings of the IFAC symposium on distributed parameter systems*, Banff, Alberta, Canada.

Luongo, C. A., Gilmour, B. J., & Schroeder, D. W. (1989). *Optimization in natural gas transmission networks: A tool to improve operational efficiency*. Technical report, Stoner Associates, Inc., Houston.

Osiadacz, A. J. (1980). Nonlinear programming applied to the optimum control of a gas compressor station. *International Journal for Numerical Methods in Engineering*, *15*(9), 1287–1301.

Osiadacz, A. J. (1994). Dynamic optimization of high pressure gas networks using hierarchical systems theory. In *Proceedings of the 26th PSIG annual meeting*, San Diego.

Osiadacz, A. J., & Bell, D. J. (1986). A simplified algorithm for optimization of large-scale gas networks. *Optimal Control Applications & Methods*, *7*, 95–104.

Osiadacz, A. J., & Górecki, M. (1995). Optimization of pipe sizes for distribution gas network design. In *Proceedings of the 27th PSIG Annual Meeting*, Albuquerque.

Osiadacz, A. J. & Swierczewski, S. (1994). Optimal control of gas transportation systems. In *Proceedings of the 3rd IEEE conference on control applications* (pp. 795–796).

Percell, P. B. & Reet, J. D. V. (1989). A compressor station optimizer for planning gas pipeline operation. In *Proceedings of the 21st PSIG annual meeting*, El Paso.

Percell, P. B., & Ryan, M. J. (1987). Steady-state optimization of gas pipeline network operation. In *Proceedings of the 19th PSIG annual meeting*, Tulsa.

Pratt, K. F., & Wilson, J. G. (1984). Optimisation of the operation of gas transmission systems. *Transactions of the Institute of Measurement and Control*, *6*(5), 261–269.

Ríos-Mercado, R. Z. (2002). Natural gas pipeline optimization. In P. M. Pardalos & M. G. C. Resende (Eds.), *Handbook of applied optimization* (pp. 813–825). New York: Oxford University Press.

Ríos-Mercado, R. Z., Kim, S., & Boyd, E. A. (2006). Efficient operation of natural gas transmission systems: A network-based heuristic for cyclic structures. *Computers & Operations Research*, *33*(8), 2323–2351.

Ríos-Mercado, R. Z., Wu, S., Scott, L. R., & Boyd, E. A. (2002). A reduction technique for natural gas transmission network optimization problems. *Annals of Operations Research*, *117*(1–4), 217–234.

Villalobos-Morales, Y., D. Cobos-Zaleta, H. J. Flores-Villarreal, C. Borraz-Sánchez, and R. Z. Ríos-Mercado (2003). 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*, Birmingham.

Villalobos-Morales, Y., & Ríos-Mercado, R. Z. (2005). Preprocesamiento efectivo de un problema de minimización de combustible en sistemas de transporte de gas natural. *Revista Ingeniería de Sistemas*, *19*, 79–103. In Spanish.

Wong, P. J., & Larson, R. E. (1968a). Optimization of natural-gas pipeline systems via dynamic programming. *IEEE Transactions on Automatic Control*, *AC-13*(5), 475–481.

Wong, P. J., & Larson, R. E. (1968b). Optimization of tree-structured natural-gas transmission networks. *Journal of Mathematical Analysis and Applications*, *24*(3), 613–626.

Wu, S., Boyd, E. A., & Scott, L. R. (1996). Minimizing fuel consumption at gas compressor stations. In J.J.-W. Chen & A. Mital (Eds.), *Advances in industrial engineering applications and practice I* (pp. 972–977), Cincinnati, OH. International Journal of Industrial Engineering.

Wu, S., Ríos-Mercado, R. Z., Boyd, E. A., & Scott, L. R. (2000). Model relaxations for the fuel cost minimization of steady-state gas pipeline networks. *Mathematical and Computer Modelling*, *31*(2–3), 197–220.

PAPER II

A tree decomposition algorithm for minimizing fuel cost in gas transmission networks*

Conrado Borraz-Sánchez Dag Haugland[a]

* Published in: *IEEE conference proceedings 2009*. The cie39 International Conference on Computers & Industrial Engineering. ISBN 978-1-4244-4136-5, pp. 244–249. (2009)

[a] Dag Haugland is currently an Associate Professor in Optimization at Department of Informatics, University of Bergen, Norway. (Main supervisor of the PhD thesis)

A Tree Decomposition Algorithm for Minimizing Fuel Cost in Gas Transmission Networks

Conrado Borraz-Sánchez[1], **Dag Haugland**[1]

[1]Department of Informatics, University of Bergen, Norway
{Conrado.Borraz-Sanchez, Dag.Haugland}@ii.uib.no

ABSTRACT

In this paper, we address the problem of computing optimal transportation plans of natural gas by means of compressor stations in pipeline networks. This non-linear (non-convex) problem takes into account two types of continuous decision variables: mass flow rate through each arc, and gas pressure level at each node. Compressors consume fuel at rates depending on flow and pressure, and the problem is to assign values to these variables such that the total fuel cost is minimized.
We propose a dynamic programming algorithm based on tree decomposition, which applies to a broader class of instances than currently available techniques can solve. Through computational experiments, we demonstrate that our algorithm is capable to solve several instances where previously suggested methods and commercially avialable solvers for non-linear optimization fail.

Keywords: Gas Transmission Network, Fuel Cost, Dynamic Programming, Tree Decomposition

1. INTRODUCTION

Natural gas has become one of the most important energy resources worldwide. Consequently, the volumes of gas flowing from the fields through transmission networks to the market have been increasing steeply during the past decades, and in parallel, a growing interest in reducing costs associated with pipeline gas transportation has been observed.

In this paper, the fuel cost minimization problem (FCMP) to transport natural gas in a general class of transmission networks is addressed. The FCMP involves two types of continuous decision variables: mass flow rate through each arc, and gas pressure level at each node. The problem can be described as follows: We need to move natural gas over large distances from several possible sources to different distribution centers through various devices including pipes and compressor stations. During the transmission, energy and pressure are lost, and the compressor stations installed in the pipeline system are crucial for keeping the gas moving. Consequently, fuel consumption associated costs are incurred at these stations. The problem is to determine a transportation plan on an existing network minimizing the total fuel cost, while meeting specified demand at the distribution centers.

1.1. Related work

An extensive literature on the FCMP has been published over the past 30 years. This includes applications of numerical simulations (see [10]), Dynamic Programming (DP) (see [6], [8] and [17]), gradient techniques (see [4]). and others. Most of these contributions are practically limited to pipelines networks with non-cyclic structures or to sparse cyclic networks, and have obtained a considerable success on such instances.

Several works based on successive reductions of the network (see [1], [2] and [3]), and graph theory and functional analysis (see [12]) have been developed with the promise to handle cyclic topologies. However, since these optimization approaches require a certain sparse network structure, their application is still in a development phase. The purpose of the current work is to present a solution approach that admits a more general network structure, and hence overcome the limitations of network reduction techniques.

The remainder of this paper is organized as follows. In Section 2, we define the problem in mathematical terms. In Section 3, we present a contemporary solution method, and point out a serious point of weakness. In Section 4, the tree decomposition based algorithm to solve the FCMP via DP is described. Our numerical results based on different computational experiments are shown in Section 5, where we compare our results to those obtained by alternative methods when applied to several network configurations. Finally, concluding remarks are given in Section 6.

2. PROBLEM DEFINITION

Let $G = (V, A)$ be a directed graph representing a gas transmission network, where V and A are the node and arc sets, respectively. Let V_v^+ and V_v^- denote the sets of out- and in-neighbors, respectively, of node $v \in V$, let $V_s \subseteq V$ be the set of supply nodes, $V_d \subseteq V$ the set of demand nodes, and let $A = A_c \cup A_p$ be partitioned into a set of compressor arcs A_c and a set of pipeline arcs A_p. That is, if $(u, v) \in A_c$ then $u, v \in V$ are the network node representing the input and the output units, respectively, of some compressor (u, v). An analogous interpretation is made for pipeline arcs $(u, v) \in A_p$.

Two types of decision variables are defined: Let x_{uv} denote the mass flow rate at arc $(u, v) \in A$, and let p_v denote the gas pressure at node $v \in V$. For each $v \in V$, we define the parameters net mass flow rate B_v and (lower and upper, respectively) pressure bounds P_v^L and P_v^U. By convention, $B_v > 0$ if $v \in V_s$, $B_v < 0$ if $v \in V_d$, and $B_v = 0$ otherwise. By the assumption that flow is conserved at the

978-1-4244-4136-5/09/$25.00

nodes, the decision variables are subject to the constraints $\sum_{u\in V_v^+} x_{vu} - \sum_{u\in V_v^-} x_{uv} = B_v$ for all $v \in V$. Constraints linking the pressure and flow variables are given for the arc sets A_c and A_p, and these are discussed next.

2.1. Compressor arc constraints

The variables that are manipulated in a compressor $(u, v) \in A_c$ in order to have the desired values of x_{uv}, p_u, and p_v are according to Wu et al. [18] compressor speed S_{uv}, volumetric inlet flow rate Q_{uv}, adiabatic head H_{uv} and adiabatic efficiency η_{uv}. As explained more detailed in e.g. [18], these relate to (x_{uv}, p_u, p_v) according to

$$H_{uv} = \alpha \left[\left(\frac{p_v}{p_u} \right)^m - 1 \right] \quad \forall (u,v) \in A_c \tag{1}$$

$$Q_{uv} = \alpha m \frac{x_{uv}}{p_u} \quad \forall (u,v) \in A_c \tag{2}$$

$$\frac{H_{uv}}{S_{uv}^2} = \phi^1 \left(\frac{Q_{uv}}{S_{uv}} \right) \quad \forall (u,v) \in A_c \tag{3}$$

$$\eta_{uv} = \phi^2 \left(\frac{Q_{uv}}{S_{uv}} \right) \quad \forall (u,v) \in A_c \tag{4}$$

where $m \in (0, 1)$ and $\alpha > 0$ are gas specific constants, and ϕ^1 and ϕ^2 are polynomial functions (typically of degree 3). The coefficients of ϕ^1 and ϕ^2 are assessed by applying least squares analysis to a set of selected data points. For each $(u, v) \in A_c$, Q_{uv} is subject to lower and upper bounds Q_{uv}^L and Q_{uv}^U, and we adopt a similar notation for bounds on the variables S_{uv}, H_{uv} and η_{uv}.

The fuel consumption cost is given by (see [18])

$$g_{uv}(x, p) = \frac{c x_{uv} \left[\left(\frac{p_v}{p_u} \right)^m - 1 \right]}{\eta_{uv}} \quad \forall (u,v) \in A_c,$$

where $c > 0$ is a monetary constant.

The *operating domain* of compressor $(u, v) \in A_c$ is the set $D_{uv} \subset \Re^3$ of value assignments to (x_{uv}, p_u, p_v) for which there exist values of $(Q_{uv}, S_{uv}, H_{uv}, \eta_{uv})$ satisfying (1)-(4) and the bounds $Q_{uv}^L \le Q_{uv} \le Q_{uv}^U$, $S_{uv}^L \le S_{uv} \le S_{uv}^U$, $H_{uv}^L \le H_{uv} \le H_{uv}^U$, and $\eta_{uv}^L \le \eta_{uv} \le \eta_{uv}^U$.

We assume that for all $(x_{uv}, p_u, p_v) \in D_{uv}$, there is a *unique* feasible $(Q_{uv}, S_{uv}, H_{uv}, \eta_{uv})$. This correspondence defines the desired transformation from feasible flow and pressure variable values (x_{uv}, p_u, p_v) to an estimate $g_{uv}(x, p)$ of the fuel cost.

2.2. Pipeline arc constraints

Following [18], the relation between pipeline flow and (sufficiently high) pressure in steady state networks can be written as $x_{uv}^2 = W_{uv} \left(p_u^2 - p_v^2 \right)$, where $W_{uv} > 0$ is some constant depending on characteristics of the gas and the pipeline $(u, v) \in A_p$.

2.3. Mathematical model

For each node $v \in V$, we impose lower and upper pressure bounds P_v^L, and P_v^U, respectively. We confine our study to irreversible flow, and impose $x_{uv} \ge 0$ for all $(u, v) \in A$.

Summarizing the two last sections, the FCMP can then be formulated as follows:

$$\min \quad \sum_{(u,v)\in A_c} g_{uv}(x, p) \tag{5}$$

s.t.:

$$\sum_{u\in V_v^+} x_{vu} - \sum_{u\in V_v^-} x_{uv} = B_v \quad \forall v \in V \tag{6}$$

$$(x_{uv}, p_u, p_v) \in D_{uv} \quad \forall (u,v) \in A_c \tag{7}$$

$$x_{uv}^2 = W_{uv} \left(p_u^2 - p_v^2 \right) \quad \forall (u,v) \in A_p \tag{8}$$

$$P_v^L \le p_v \le P_v^U \quad \forall v \in V \tag{9}$$

$$x_{uv} \ge 0 \quad \forall (u,v) \in A \tag{10}$$

3. SOLUTION METHODS

Several solution methods have been suggested for FCMP, including those by Ríos-Mercado et al. [11] and Borraz-Sánchez and Ríos-Mercado [1], which all follow the idea of Algorithm 1.

Algorithm 1 SolveFCMP

Step 1: Choose initial (feasible) flow
repeat
  Step 2: Optimize pressure while keeping the flow fixed
  Step 3: Optimize flow while keeping the pressure fixed
until flow does not change

With the risk of missing the global optimum, flow and pressure are determined separately in Steps 2 and 3, respectively. As we show next, this can be accomplished by focusing on only a subset of the variables.

3.1. Compressor network

In [12], it was shown that if $A_c = \emptyset$ then for any $B \in \Re^V$ there exists a unique solution to the set of equations defined by (6) and (8). That is, the flow assignment to A_p is unique (and infeasible if it violates (10)).

Let $V' \subseteq V$ consist of exactly one node from each of the connected components in the directed graph (V, A_p), and let $G^v = (V^v, A^v)$ denote the component (subgraph) to which $v \in V'$ belongs. By applying the result in [12] to G^v for any $v \in V'$, we get that the pipeline flow is uniquely determined once the compressor flow is given. If also p_v is given, we can by repeated application of (8) also find p_u for all other nodes $u \in V^v$. Hence FCMP is reduced to finding the flow on all arcs in A_c (Step 2 in Algorithm 1) and p_v for all $v \in V'$ (Step 3).

Step 2 can be approached by identifying cycles in G with negative net cost, as suggested in e.g. [12], and will not be discussed further here. Step 3 can be viewed as follows: Define the *compressor network* (in [12] referred to as the *reduced* network) as the directed graph $G' = (V', A_c')$, where $(u, v) \in A_c'$ if and only if $u, v \in V'$ and there exists some arc in A_c from V^u to V^v. As in [12], we assume that G' does not contain loops, which means that no compressor arc has both it start node and its end node in the same connected component of (V, A_p). The node set of G' can alternatively be associated with the subgraphs G^v, as shown in the illustration of the transition from G to G' (Fig. 1). Optimizing the pressure is now equivalent to solving

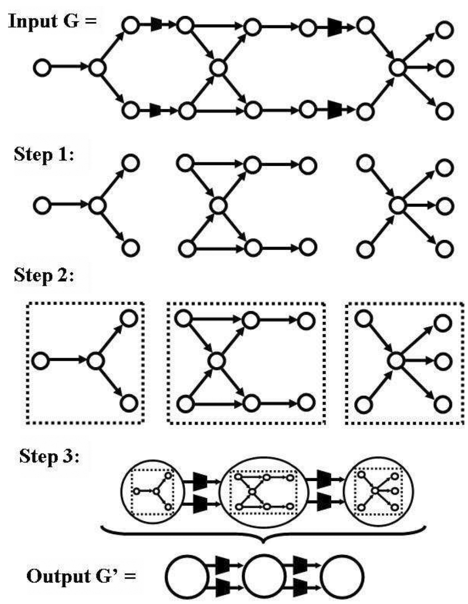


Fig. 1: Transition to compressor network

$$\min_{p \in \Re^{V'}} \left\{ \sum_{(u,v) \in A'_c} g'_{uv}(p_u, p_v) : (p_u, p_v) \in D'_{uv} \forall (u,v) \in A'_c \right\}, \quad (11)$$

where $g'_{uv}(p_u, p_v)$ is the cost incurred on all arcs in A_c between V^u and V^v given that u and v are assigned pressure values p_u and p_v, respectively. Further, D'_{uv} is the feasible domain of (p_u, p_v), taking (7) into account for all arcs from V^u to V^v.

Carter [3] suggested to solve (11) by discretizing $[P^L, P^U]$ and then apply a network reduction technique referred to as *Non-sequential* Dynamic Programming (NDP). Assume that there are m discretization points denoted $p_v^1, \ldots, p_v^m$ for each $v \in V'$, and let $g_{uv}^{ij} = g'_{uv}\left(p_u^i, p_v^j\right)$ if $(p_u, p_v) \in D'_{uv}$ and $g_{uv}^{ij} = \infty$, otherwise. Then NDP consists of a sequence of reductions of G' until the resulting graph is a single node. Three reduction types (see Fig. 2) are considered:

(a) Serial: If $v \in V'$ has exactly two incident arcs (u, v) and (v, t) in G', then v, (u, v) and (v, t) are replaced by a new arc (u, t), and $g_{ut}^{ij} = \min_k \left\{ g_{uv}^{ik} + g_{vt}^{kj} : k = 1, \ldots, m \right\}$. The same principle applies if both arcs incident to v enter (leave) v.

(b) Dangling: If $v \in V'$ has only one incident arc (v, t), then t and (v, t) are removed, and, for all in-neighbors u of v in G', g_{uv}^{ij} is updated to $g_{uv}^{ij} + \min_k \left\{ g_{vt}^{jk} : k = 1, \ldots, m \right\}$. Similar updates apply to the out-neighbors of v, and the principle applies also if the sole neighbor of t is an out-neighbor.

(c) Parallel: If $k > 1$ arcs $a_1, \ldots, a_k$ in G' connect nodes u and v, then these are replaced by a single arc (u, v). The associated cost parameters are defined as $g_{uv}^{ij} = \sum_{\ell=1}^{k} g_{a_\ell}^{ij}\ \forall i, j = 1, \ldots, m$.

The serial and parallel reductions constitute the preprocessing procedure suggested by Koster et al. [7].

When neither of the reductions (a)-(c) can be carried out, NDP fails. Fig. 3 shows a simple example where this occurs. To overcome this weakness, we now go on to demonstrate how such instances of (11) can be solved.

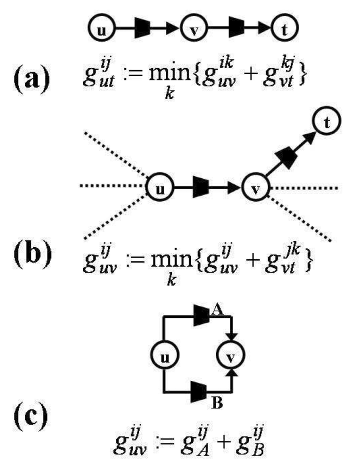


Fig. 2: Network reduction types

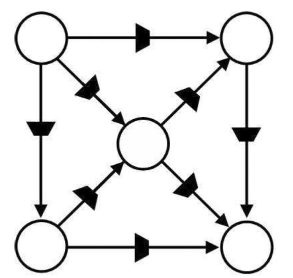


Fig. 3: An instance of G' where NDP fails

4. A TREE DECOMPOSITION APPROACH TO OPTIMIZING THE PRESSURE VALUES

Problem (11) has the mathematical structure of the *frequency assignment problem* [7], and can also be solved by the procedure suggested in [7]. This is based on the following concept introduced by Robertson and Seymour [13]:

Definition 1 *A* tree decomposition *of G' is a pair $\mathcal{J} = (\{X_i : i \in I\}, T)$, where each X_i is a subset of V', called a bag, and T is a tree with node set I. The following properties must be satisfied:*

Algorithm 2 $\mathbf{DP}(\mathcal{J}, i, X, \pi)$

if i is a leaf in T **then**

$$\textbf{return} \min_{p \in \mathcal{D}_{X_i \cup X}} \left\{ \sum_{\substack{(u,v) \in A'_c \\ u,v \in X_i \cup X}} g'_{uv}(p_u, p_v) : p_v = \pi_v \forall v \in X \right\}$$

else

$$\textbf{return} \min_{p \in \mathcal{D}_{X_i \cup X}} \left\{ \sum_{\substack{(u,v) \in A'_c \\ u,v \in X_i \cup X}} g'_{uv}(p_u, p_v) + \sum_{j \in K_i} \mathbf{DP}(\mathcal{J}, j, X_i \cup X, p) : p_v = \pi_v \forall v \in X \right\}$$

- $\bigcup_{i \in I} X_i = V'$;
- *for all* $(u, v) \in A'_c$, *there is an* $i \in I$ *such that* $\{u, v\} \subseteq X_i$;
- $\forall i, j, k \in I$, *if* j *lies on the path between* i *and* k *in* T, *then* $X_i \cap X_k \subseteq X_j$.

The width *of a* tree decomposition $\mathcal{J}$ *is* $\max_{i \in I} |X_i| - 1$.

For any $X \subseteq V'$, define p_X as the vector with components p_v $(v \in X)$ in any consistent order. Define $\mathcal{D}_v = \{p_v^1, \ldots, p_v^m\}$ for all $v \in V'$, and let $\mathcal{D}_X = \{p_X : p_v \in \mathcal{D}_v \forall v \in X\}$. For any $i \in I$, let K_i denote the set of child nodes of i in T.

Algorithm 2 applies dynamic programming to a tree decomposition $\mathcal{J}$ of G'. When bag X_i is to be processed, the union X of all ancestor bags of X_i are input along with a pressure vector $\pi \in \mathcal{D}_X$. The algorithm optimizes the value of p_v for all $v \in X_i$ by complete enumeration of $\mathcal{D}_v$, and by taking into account optimal pressure assignments to all nodes in all child bags of X_i. This is expressed in terms of a recursive call in Algorithm 2. Since $X_i \cap X$ may be nonempty, we must ensure that nodes contained in this set are not assigned new pressure values when processing X_i, and we impose the constraint that $p_v = \pi_v$ for all $v \in X$.

The running time of Algorithm 2 is $\mathcal{O}\left(|I| m^d\right)$, where d is the width of $\mathcal{J}$. This means that finding a tree decomposition of small width can be crucial for the running time of the algorithm. It is however well known [13] that finding one with minimum width is an NP-hard problem, and it is therefore unlikely that a tree decomposition minimizing the running time of Algorithm 2 can be found in polynomial time. We will rely on a heuristic approach to constructing $\mathcal{J}$ with small width.

5. NUMERICAL EXPERIMENTS

To solve (11), we thus apply a two-phase procedure, *TreeDDP*, where the computation of some tree decomposition $\mathcal{J}$ is the first phase, and where Algorithm 2 constitutes the second. The input to this procedure is a network, which is reduced as much as possible by the techniques described in Fig. 2. To compute $\mathcal{J}$, we apply the technique given in [14] based on *Maximum Cardinality Search* [15].

5.1. Test instances

All experiments reported in this work were carried out on the set of test instances shown in Tab. 1. Each row gives an identifier of an instance, the size in terms of nodes and arcs in G' after reduction, and the type of compressor used. We consider 9 different compressor types, and all compressors are identical within any given instance. Furthermore, the width and the number of bags in the tree decomposition are given in the two last columns of Tab. 1.

Tab. 1: Test instances

	Size			$\mathcal{J}$	
Ref	$\|V'\|$	$\|A'_c\|$	type	width	$\|I\|$
A	3	3	4	3	1
B	3	3	5	3	1
C	4	6	1	3	3
D	4	6	2	3	4
E	4	6	3	3	5
F	4	6	4	3	4
G	4	6	5	3	4
H	4	6	6	3	3
I	4	6	7	3	4
J	5	8	4	3	6
K	5	8	8	3	4
L	9	20	4	4	9
M	9	20	5	4	8
N	18	25	2	3	25
O	18	25	4	3	18
P	18	25	9	3	22

5.2. Experiments

The experiments can be briefly described as follows. The first experiment is a feasibility study where we examine the performance of TreeDDP while varying the granularity of the discretization. We let $m \in \{50, 100, 1000\}$, and let the pressure values be uniformly distributed between their lower and upper bounds.

For a comparison of TreeDDP to a generic global optimization tool, we submit in the second set of experiments (11) to *BARON* [16]. The algorithm of BARON is a variant of branch-and-bound where a convex program is solved in each node of the search tree. We use version 8.1.5 of BARON with version 5.51 of *MINOS* [9] to solve the convex subproblems.

In the third set of experiments, we applied MINOS to compute local optima to problem (11) for 1000 randomly generated starting points.

The TreeDDP procedure was coded in C++ under Linux Red Hat, and all experiments were run on a 2.4 GHz In-

Tab. 2: Performance of TreeDDP

	$m = 50$		$m = 100$		$m = 1000$	
Ref	CPU(s)	Obj	CPU(s)	Obj	CPU(s)	Obj
A	0.0	1.12	0.0	0.77	0.8	0.75
B	0.0	2.63	0.0	2.62	2.2	2.62
C	1.0	10.29	15.9	9.34	245.8	8.79
D	0.1	7.45	11.3	7.34	421.8	7.34
E	1.4	9.66	21.9	6.36	836.1	5.29
F	1.8	6.87	29.5	5.69	1845.2	4.12
G	0.6	9.43	9.5	6.30	1322.5	6.30
H	0.7	6.34	12.7	5.93	712.8	5.09
I	0.6	2.83	9.5	2.82	412.2	2.77
J	0.8	6.07	13.4	5.59	2201.3	5.27
K	0.6	—	9.4	35.67	1052.7	35.67
L	3.1	68.89	49.9	61.83	3424.1	61.73
M	2.5	89.68	39.5	74.80	3092.4	60.74
N	2.1	60.71	34.4	52.46	3554.1	46.00
O	2.6	127.31	41.5	44.51	3623.1	32.62
P	1.4	35.25	23.1	37.67	3417.2	26.54

tel(R) processor with 2 GByte RAM. Experiments with BARON and MINOS were conducted by formulating the model in GAMS [5].

5.3. Results

Table 2 shows the results achieved by TreeDDP while varying m. Instance references are given in the first column, and computation times (CPU-seconds) and objective function values for the respective values of m are given in columns 2-7. The only case where TreeDDP failed to find a feasible solution was for $m = 50$ in instance K. We observe that as m increases, better solutions are found (minimum cost decreases) in all instances, except from a cost increase from $m = 50$ to $m = 100$ in instance P. Nevertheless, a finer discretization also implies, as expected, that the computational requirements increase, and the running time slightly exceeds one CPU-hour in one instance (O).

Table 3 shows the performance of BARON when applied to the test instances. A time limit of 3600 CPU-seconds is imposed, and the relative optimality tolerance is set to 0.01. That is, any feasible solution is considered to be optimal if the gap between the objective function value and its lower bound is below one percent of the objective function value. Columns 2-5 contain the number of iterations in BARON, the maximum number of open nodes the search tree ever had, the objective function value of the best feasible solution found (if any), and the lower bound on the minimum cost.

In 9 out of 16 instances, BARON was able to find a feasible solution, and in 4 instances (A, B, D and J) it was able to prove optimality within the given tolerance. In instances C, E, F, H and L, the relative optimality gap ranged from 2.9% (H) to 50.1% (C), whereas in the remaining instances, no feasible solution was found before the time limit expired. By comparing the last column in Tab. 2 to the lower bounds in Tab. 3, we also observe that the relative optimality gap of TreeDDP in one instance (G) is as large as 64.0%. In the instances where BARON found a feasible solution, the largest gap is 49.4% (instance C).

Tab. 3: Performance of BARON

Ref	Its	#nodes	Obj	LB
A	551	131	0.75	0.75
B	1148	342	2.62	2.62
C	21521	7462	9.02	4.45
D	445	38	7.35	7.28
E	17059	7023	5.30	4.02
F	26765	7480	3.94	2.71
G	5231	1283	—	2.27
H	2109	204	5.19	5.04
I	3267	324	—	2.73
J	27832	2299	5.15	5.10
K	14968	3344	—	20.86
L	740	451	65.94	43.81
M	2438	765	—	31.12
N	1830	839	—	34.28
O	1124	168	—	15.74
P	978	655	—	17.43

Tab. 4: TreeDDP vs. other optimizers

	Minimum cost			TreeDDP vs	
Ref	BARON	MINOS	TreeDDP	BARON	MINOS
A	0.75	0.75	0.75	0.0	0.0
B	2.62	2.62	2.62	0.0	0.0
C	9.02	10.97	8.79	2.5	19.9
D	7.35	7.34	7.34	0.1	0.0
E	5.30	5.63	5.29	0.2	6.0
F	3.94	4.74	4.12	-4.6	13.1
G	—	—	6.30	—	—
H	5.19	5.31	5.09	1.9	4.1
I	—	—	2.77	—	—
J	5.15	5.69	5.27	-2.3	7.4
K	—	—	35.67	—	—
L	65.94	69.16	61.73	6.4	10.7
M	—	61.58	60.74	—	1.4
N	—	—	46.00	—	—
O	—	32.71	32.62	—	0.3
P	—	—	26.54	—	—

In Tab. 4, we compare our results (when $m = 1000$) to the best results obtained by MINOS applied to 1000 randomly generated initial solutions. For an overview, we also include the results from BARON. Columns 2-4 contain the best objective function values obtained by each solver. Whenever applicable, we give in the two last columns the relative cost reduction in percentages when TreeDDP is applied in place of BARON and MINOS, respectively. The numerical values in Column 5 show that neither BARON nor TreeDDP outperforms the other when both are able to compute feasible solutions. We observe that also MINOS failed to find a feasible solution in 5 of the instances, and that it in most instances produced solutions that are of poor quality compared to the TreeDDP and BARON solutions.

6. CONCLUDING REMARKS

In this paper, we have studied a model (FCMP) for minimizing compressor fuel cost in transmission networks for natural gas. An arc in the network model corresponds to either a pipe or a compressor, and the decision variables are arc flow and node pressure. In addition to flow conservation constraints, the model contains non-linear constraints relating pipeline flow to inlet and outlet pressure, as well as non-convex constraints defining the operation domain of the compressors.

Following a general algorithmic idea, which has been suggested and supported experimentally in several recent works, we consider a procedure where each iteration consists of a flow improvement step and a pressure optimization step. Alternating between flow and pressure, one set of decision variables is kept fixed in each step. Still in agreement with previously suggested methods, the non-convex subproblem of optimizing pressure is approximated by a combinatorial one. This is accomplished by discretization of the pressure variables. The contribution of this paper is a method for solving the discrete version of the problem in instances where previously suggested methods fail.

Unlike methods based on successive network reductions, our method does not make any assumptions concerning the sparsity of the network. By constructing a tree decomposition of the network, and apply dynamic programming to it, we are able to solve the discrete version of the pressure optimization problem without enumerating the whole solution space.

We have tested our solution method on a set of imaginary instances, and compared the results to those obtained by applying both a global and a local optimizer to the continuous version of the problem. The experiments indicate that a method guaranteeing the global optimum in reasonable time seems unrealistic even for small instances. Further, discretizing the pressure variables and applying dynamic programming to a tree decomposition gives better results than local optimization, with multiple initial solutions, of the continuous version.

REFERENCES

[1] C. Borraz-Sánchez, R.Z. Ríos-Mercado, "Improving the operation of pipeline systems on cyclic structures by tabu search", *Computers & Chemical Engineering*, Vol. 33, No. 1, pp. 58-64, 2009.

[2] C. Borraz-Sánchez, R.Z. Ríos-Mercado, "A Non-sequential Dynamic Programming Approach for Natural Gas Network Optimization", *WSEAS Transactions on Systems*, Vol. 3, No. 4, pp. 1384-1389, 2004.

[3] R.G. Carter, "Pipeline optimization: Dynamic programming after 30 years", In *Proceedings of the PSIG Meeting*, pp. 1-19, Denver, USA, October 1998.

[4] H.J. Flores-Villarreal, R. Z. Ríos-Mercado, "Efficient operation of natural gas pipeline networks: Computational finding of high quality solutions", In *Proceedings of the International Applied Business Research Conference*, Paper 352, pp. 1-7. Acapulco, Mexico, March 2003.

[5] GAMS Development Corporation, "GAMS: The Solver Manuals. Washington, DC, USA, 2008.

[6] B.J. Gilmour, C.A. Luongo, D.W. Schroeder, "Optimization in natural gas transmission networks: A tool to improve operational efficiency", Technical Report, Stoner Associates Inc., presented at *The Third Meeting on SIAM Optimization*, Boston, USA, April 1989.

[7] A.M.C.A. Koster, S.P.M. van Hoesel, A.W.J. Kolen, "Optimal solutions for frequency assignment problems via tree decomposition", *Lecture Notes in Computer Science*, Vol. 1665, pp. 338-350, 1999.

[8] H.S. Lall, P.B. Percell, "A dynamic programming based gas pipeline optimizer", In A. Bensoussan, J. L. Lions (eds.): *Analysis and Optimization of Systems*, pp. 123-132, Springer, Berlin, Germany, 1990.

[9] B.A. Murtaugh, M.A. Saunders, "MINOS 5.1 User's Guide", Technical Report SOL-83-20R, Systems Optimization Laboratory, Stanford University, Stanford, California, 1983.

[10] A.J. Osiadacz, "Simulation and Analysis of Gas Networks", *Gulf Publishing Company, Houston, USA*, 1987.

[11] R.Z. Ríos-Mercado, S. Kim, E.A. Boyd, "Efficient operation of natural gas transmission systems: A network-based heuristic for cyclic structures", *Computers & Operations Research*, Vol. 33, No. 8, pp. 2323-2351, 2006.

[12] R.Z. Ríos-Mercado, S. Wu, L.R. Scott, E.A. Boyd, "A reduction technique for natural gas transmission network optimization problems", *Annals of Operations Research*, Vol. 117, No. 1-4, pp. 217-234, 2002.

[13] N. Robertson, P.D. Seymour, "Graph minors II. Algorithmic aspects of treewidth", *Journal of Algorithms*, Vol. 7, pp. 309-322, 1986.

[14] S. Subbarayan, "An Empirical Comparison of CSP Decomposition Methods", In: B.Hnich, K.Stergiou (Eds.): *Proceedings of the CP 2007 Doctoral Programme*, pp. 163-168, Providence RI, USA, September 23-27, 2007.

[15] R.E. Tarjan, M. Yannakakis, "Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs", *SIAM Journal of Computing*, Vol. 13, pp. 566-579, 1984.

[16] M. Tawarmalani, N. V. Sahinidis, "Global optimization of mixed-integer nonlinear programs: A theoretical and computational study", *Mathematical Programming*, Vol. 99, No. 3, pp. 563-591, 2004.

[17] P.J. Wong, R.E. Larson, "Optimization of natural gas pipeline systems via dynamic programming", *IEEE Transactions on Automatic Control*, Vol. 13, No. 5, pp. 475-481, 1968.

[18] S. Wu, R.Z. Ríos-Mercado, E.A. Boyd, L.R. Scott, "Model relaxations for the fuel cost minimization of steady-state gas pipeline networks", *Mathematical and Computer Modeling*, Vol. 31, No. 2-3, pp. 197-220, 2000.

PAPER III

Minimizing fuel cost in gas transmission networks by dynamic programming and adaptive discretization*

Conrado Borraz-Sánchez   Dag Haugland

* Article in press: *Journal of Computers & Industrial Engineering* (2010), doi:10.1016/j.cie.2010.07.012

Computers & Industrial Engineering xxx (2010) xxx–xxx

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Minimizing fuel cost in gas transmission networks by dynamic programming and adaptive discretization

Conrado Borraz-Sánchez, Dag Haugland *

Department of Informatics, University of Bergen, P.B. 7803, N-5020 Bergen, Norway

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Gas transmission network
Compressor
Fuel cost
Discretization
Dynamic programming
Tree decomposition

ABSTRACT

In this paper, the problem of computing optimal transportation plans for natural gas by means of compressor stations in pipeline networks is addressed. The non-linear (non-convex) mathematical model considers two types of continuous decision variables: mass flow rate along each arc, and gas pressure level at each node. The problem arises due to the presence of costs incurred when running compressors in order to keep the gas flowing through the system. Hence, the assignment of optimal values to flow and pressure variables such that the total fuel cost is minimized turns out to be essential to the gas industry. The first contribution from the paper is a solution method based on dynamic programming applied to a discretized version of the problem. By utilizing the concept of a tree decomposition, our approach can handle transmission networks of arbitrary structure, which makes it distinguished from previously suggested methods. The second contribution is a discretization scheme that keeps the computational effort low, even in instances where the running time is sensitive to the size of the mesh. Several computational experiments demonstrate that our methods are superior to a commercially available local optimizer.

1. Introduction

Natural gas has become one of the most important energy resources worldwide. Consequently, the volumes of gas flowing from the fields through transmission networks to the market have been increasing steeply during the past decades, and in parallel, a growing interest in reducing costs associated with pipeline gas transportation has been observed.

A gas transmission network is a system consisting of sources, pipelines, compressors and distribution centers. At the sources, a supply of gas received from external fields is refined, and transported via pipelines and compressors to the distribution centers. The distribution centers are the end points of the transmission network, and the gas finally received here is input to local distribution networks supporting households and other clients.

The flow capacity of any pipeline increases with the inlet pressure and decreases with the outlet pressure of the pipeline. If no compressors are installed along a flow path, the pressure will be continuously decreasing. Since the pressure at the distribution centers typically is fixed, the flow capacity may therefore eventually become prohibitively small. To increase the pressure, and thereby the flow capacity, compressors are hence installed at the entry points of selected pipelines. Operation of the compressors incurs a cost depending on the flow and their inlet and outlet pressures.

* Corresponding author. Tel.: +47 55 58 40 33; fax: +47 55 58 41 99.
E-mail addresses: Conrado.Borraz-Sanchez@ii.uib.no (C. Borraz-Sánchez), dag@ii.uib.no (D. Haugland).

In this paper, the fuel cost minimization problem (FCMP) to transport natural gas in a general class of transmission networks is addressed. The FCMP involves two types of continuous decision variables: mass flow rate through each arc, and gas pressure level at each node. The problem is to determine a transportation plan minimizing the total fuel cost, while meeting a specified demand at the distribution centers.

An extensive literature on the FCMP has been published over the past 30 years. Most of the suggested solution methods are limited to pipelines networks with acyclic structures, and in such instances, the suggested methods have shown a strong potential. In some of the more recent works, methods for cyclic networks have been developed. However, since these optimization approaches require a certain sparse network structure, their applicability is somewhat restricted. The following sections give a more detailed overview of the most relevant methods. A common assumption is that the system is in steady-state, which means that rapid changes in parameter values do not occur.

1.1. Methods based on dynamic programming

By discretizing the range of the pressure variables, FCMP has in several works been formulated as a combinatorial problem that can be approached by dynamic programming (DP). Wong and Larson (1968) published the first work on optimization of pipeline transportation of natural gas by DP. They applied it to a gun-barrel (linear) network, that is a problem instance where the underlying

0360-8352/$ - see front matter © 2010 Published by Elsevier Ltd.
doi:10.1016/j.cie.2010.07.012

network is a path, using a recursive formulation. A disadvantage was that the length and diameter of the pipeline segment were assumed to be constant because of limitations of DP. Martch and McCall (1972) modified the problem by adding branches to the pipeline segments and letting the length and diameter of the pipeline segments vary. However, since their problem formulation did not allow unbranched network, more complicated network systems could not be handled.

The first attempt to solve instances with tree-shaped networks by DP was done by Zimmer (1975). A similar approach was described by Lall and Percell (1990). They allowed a divergent branch in their systems and included an integer decision variable into the model that represented the number of operating compressors in the stations.

Carter (1998) developed an algorithm referred to as non-sequential DP. The principal idea of the method is to reduce the network by three basic reductions techniques until it consists of a single node. The method can handle a wide range of instances with cyclic networks, but fails if the networks are not sufficiently sparse. Based on this approach, Borraz-Sánchez and Ríos-Mercado (2004, 2009) developed a hybrid meta-heuristic combining tabu search and non-sequential DP. The restriction that the networks must be sparse is however a shortcoming that the hybrid method inherits from the original paper.

1.2. Methods based on gradient techniques

Percell and Ryan (1987) applied a generalized reduced gradient (GRG) method for solving FCMP. In comparison with DP, an advantage of GRG is that the rapid growth in instance size caused by many discretization points is avoided. Also, GRG is applicable to cyclic networks. Nonetheless, only a local optimum can be provided, of which instances of FCMP can have many, and the solution to be output depends on the choice of starting point. Flores-Villarreal and Ríos-Mercado (2003) extended the previous study by means of an extensive computational evaluation of the GRG method.

1.3. Other techniques and related problems

Wu, Ríos-Mercado, Boyd, and Scott (2000) address the non-convex nature of FCMP, and suggest mathematical models that provide strong relaxations, and hence tight lower bounds on the minimum cost. Based on this model and the PhD thesis of Wu (1998), they demonstrated the existence of a unique solution to a non-linear algebraic equations system over a set of flow variables. This theoretical result lead to a technique for reducing the size of the original network without altering its mathematical structure.

Villalobos-Morales, Cobos-Zaleta, Flores-Villarreal, Borraz-Sánchez, and Ríos-Mercado (2003) formulated a non-linear optimization model that also contains integer variables representing the number of compressor units inside a compressor station. Cobos-Zaleta and Ríos-Mercado (2002) extended this model, and suggested a solution technique based on outer approximation.

1.4. Contributions from the current work

Several works have demonstrated that, at least in acyclic and sparse cyclic instances of FCMP, it is a promising approach to discretize the pressure variables, and apply DP to the resulting combinatorial problem. The purpose of this research is twofold: First, we demonstrate how such approaches can be applied to networks of arbitrary structure. Second, in order to keep the running time down in dense and cyclic instances, we propose a new scheme for discretizing the pressure variables. This scheme is adaptive in the sense that it avoids fine discretization of variables in area unlikely to contain good solutions, and intensifies discretization in more promising regions.

The remainder of the paper is organized as follows: In the next section, we define the problem in mathematical terms. In Section 3, we present a contemporary solution method, and point out a simple instance where it fails. In Section 4, we show how the weakness of the method discussed in Section 3 can be overcome by our alternative method. Our adaptive discretization method is given in Section 5. Results from computational experiments are reported in Section 6, and concluding remarks are given in Section 7.

2. Problem definition

Let $G = (V,A)$ be a directed graph representing a gas transmission network, where V and A are the node and arc sets, respectively. Let V_v^+ and V_v^- denote the sets of out- and in-neighbors, respectively, of node $v \in V$. Let $V_s \subseteq V$ be the set of supply nodes representing the sources, $V_d \subseteq V$ the set of demand nodes representing the distribution centers, and let $A = A_c \cup A_p$ be partitioned into a set of compressor arcs A_c and a set of pipeline arcs A_p. That is, if $(u,v) \in A_c$ then $u, v \in V$ are the network nodes representing the input and the output units, respectively, of some compressor (u,v). An analogous interpretation is made for pipeline arcs $(u,v) \in A_p$.

Two types of decision variables are defined: Let x_{uv} denote the mass flow rate at arc $(u,v) \in A$, and let p_v denote the gas pressure at node $v \in V$. For each $v \in V$, we define the parameters net mass flow rate B_v and pressure bounds P_v^L and P_v^U (lower and upper, respectively). By convention, $B_v > 0$ if $v \in V_s$, $B_v < 0$ if $v \in V_d$, and $B_v = 0$ otherwise. By the assumption that flow is conserved at the nodes, the decision variables are subject to the constraints $\sum_{u \in V_v^+} x_{vu} - \sum_{u \in V_v^-} x_{uv} = B_v$ for all $v \in V$. Constraints linking the pressure and flow variables are given for the arc sets A_c and A_p, and these are discussed next.

2.1. Compressor arc constraints

The variables that are manipulated in a compressor $(u,v) \in A_c$ in order to have the desired values of x_{uv}, p_u, and p_v are according to Wu et al. (2000) compressor speed S_{uv}, volumetric inlet flow rate Q_{uv}, adiabatic head H_{uv} and adiabatic efficiency η_{uv} of the compressor. These can briefly be explained as follows (more details can be found in the cited work):

- The variable S_{uv} is the speed at which each molecule flows through compressor (u,v), and should not be confused with the flow x_{uv} itself.
- While x_{uv} is the mass flow per time unit, the volumetric flow Q_{uv} is simply x_{uv} divided by the gas density at the inlet point of the compressor. Due to pressure variations, the density is not constant throughout the network.
- The adiabatic head H_{uv} says how much energy is required to compress one mass unit of gas from one pressure level to another without altering the gas temperature.
- The adiabatic efficiency η_{uv} is the ratio between the energy effective in compressing the gas and the total energy spent.

As explained more detailed by, e.g. Wu et al. (2000), the above magnitudes relate to (x_{uv}, p_u, p_v) according to

$$H_{uv} = \alpha\left[\left(\frac{p_v}{p_u}\right)^{\kappa} - 1\right] \quad \forall (u,v) \in A_c \tag{1}$$

$$Q_{uv} = \alpha\kappa \frac{x_{uv}}{p_u} \quad \forall (u,v) \in A_c \tag{2}$$

$$\frac{H_{uv}}{S_{uv}^2} = \phi^1\left(\frac{Q_{uv}}{S_{uv}}\right) \quad \forall (u,v) \in A_c \tag{3}$$

$$\eta_{uv} = \phi^2\left(\frac{Q_{uv}}{S_{uv}}\right) \quad \forall (u,v) \in A_c \tag{4}$$

Please cite this article in press as: Borraz-Sánchez, C., & Haugland, D. Minimizing fuel cost in gas transmission networks by dynamic programming and adaptive discretization. *Computers & Industrial Engineering* (2010), doi:10.1016/j.cie.2010.07.012

where $\kappa \in (0,1)$ and $\alpha > 0$ are gas specific constants, and ϕ^1 and ϕ^2 are polynomial functions (typically of degree 3). The coefficients of ϕ^1 and ϕ^2 are assessed by applying least squares analysis to a set of selected data points. For each $(u,v) \in A_c$, Q_{uv} is subject to lower and upper bounds Q^L_{uv} and Q^U_{uv}, and we adopt a similar notation for bounds on the variables S_{uv},H_{uv} and η_{uv}.

The fuel consumption cost is given by Wu et al. (2000):

$$g_{uv}(x_{uv}, p_u, p_v) = \frac{c x_{uv}\left[\left(\frac{p_v}{p_u}\right)^{\kappa} - 1\right]}{\eta_{uv}} \quad \forall (u,v) \in A_c$$

where $c > 0$ is a monetary constant.

The feasible *operating domain* of compressor station $(u,v) \in A_c$ is the set $D_{uv} \subset \Re^3$ of value assignments to (x_{uv},p_u,p_v) for which there exist values of $(Q_{uv},S_{uv},H_{uv},\eta_{uv})$ satisfying (1)–(4) and the bounds $Q^L_{uv} \leqslant Q_{uv} \leqslant Q^U_{uv}$, $S^L_{uv} \leqslant S_{uv} \leqslant S^U_{uv}$, $H^L_{uv} \leqslant H_{uv} \leqslant H^U_{uv}$, and $\eta^L_{uv} \leqslant \eta_{uv} \leqslant \eta^U_{uv}$.

We assume that for all $(x_{uv},p_u,p_v) \in D_{uv}$, $\forall (u,v) \in A_c$, there is a *unique* feasible $(Q_{uv},S_{uv},H_{uv},\eta_{uv})$. This correspondence defines the desired transformation from feasible flow and pressure variable values (x_{uv},p_u,p_v) to an estimate $g_{uv}(x_{uv},p_u,p_v)$ of the fuel cost.

2.2. Pipeline arc constraints

Following Wu et al. (2000), the relation between pipeline flow and (sufficiently high) pressure in steady state networks can be written as $x^2_{uv} = W_{uv}(p^2_u - p^2_v)$, where $W_{uv} > 0$ is some constant depending on characteristics of the gas and the pipeline $(u,v) \in A_p$.

2.3. Mathematical model

For each node $v \in V$, we impose lower and upper pressure bounds P^L_v, and P^U_v, respectively. We confine our study to irreversible flow, and impose $x_{uv} \geqslant 0$ for all $(u,v) \in A$. Summarizing the two last sections, the FCMP can then be formulated as follows:

$$\min \quad \sum_{(u,v)\in A_c} g_{uv}(x_{uv}, p_u, p_v) \tag{5}$$

$$\text{s.t.:} \quad \sum_{u\in V_v^+} x_{vu} - \sum_{u\in V_v^-} x_{uv} = B_v \quad \forall v \in V \tag{6}$$

$$(x_{uv}, p_u, p_v) \in D_{uv} \quad \forall (u,v) \in A_c \tag{7}$$

$$x^2_{uv} = W_{uv}(p^2_u - p^2_v) \quad \forall (u,v) \in A_p \tag{8}$$

$$P^L_v \leqslant p_v \leqslant P^U_v \quad \forall v \in V \tag{9}$$

$$x_{uv} \geqslant 0 \quad \forall (u,v) \in A \tag{10}$$

Wu et al. (2000) give simple illustrations of the domains D_{uv}, pointing out the fact that these typically are non-convex sets. It is therefore unlikely that simple local optimization methods are sufficient to solve the above model, and the remainder of this article is devoted to methods aimed for non-convex problem instances.

3. Solution methods

Several solution methods have been suggested for FCMP, including those by Ríos-Mercado, Kim, and Boyd (2006) and Borraz-Sánchez and Ríos-Mercado (2009), which all follow the idea of Algorithm 1.

Algorithm 1. SolveFCMP

Step 1: Choose initial (feasible) flow
repeat
  Step 2: Optimize pressure while keeping the flow fixed
  Step 3: Optimize flow while keeping the pressure fixed
until flow does not change

With the risk of missing the global optimum, flow and pressure are determined separately in Steps 2 and 3, respectively. As we show next, this can be accomplished by focusing on only a subset of the variables.

3.1. Compressor network

The focus in this paper is to accomplish Step 2 of the above algorithm, and we now demonstrate how this can be done by optimizing over only a subset of the pressure variables.

Let $V' \subseteq V$ consist of exactly one node from each of the connected components in the directed graph (V,A_p), and let $G^v = (V^v,A^v)$ denote the component (subgraph) to which $v \in V'$ belongs. Define the *compressor network* (by Ríos-Mercado, Wu, Scott, & Boyd (2002) referred to as the *reduced* network) as the directed graph $G' = (V', A'_c)$, where $(u,v) \in A'_c$ if and only if $u,v \in V'$ and there exists some arc in A_c from V^u to V^v. As in (Ríos-Mercado et al., 2002), we assume that G' does not contain loops, which means that no compressor arc has both its start node and its end node in the same connected component of (V,A_p). Equivalently, the node set of G' can be associated with the subgraphs G^v $(v \in V')$, as shown in the illustration of the transition from G to G' (Fig. 1).

Theorem 1. *If $A_c = \emptyset$ then for any $B \in \Re^V$ satisfying $\sum_{v\in V} B_v = 0$, any real number $p^{ref} \geqslant 0$, and any $v \in V$, there exist unique $x \in \Re^A$ and $p \in \Re^V_+$ satisfying $p_v = p^{ref}$, (6) and (8).*

Proof. See Ríos-Mercado et al. (2002). □

The essence of Theorem 1 is that in any network consisting exclusively of pipeline arcs, the flow and pressure values are all gi-

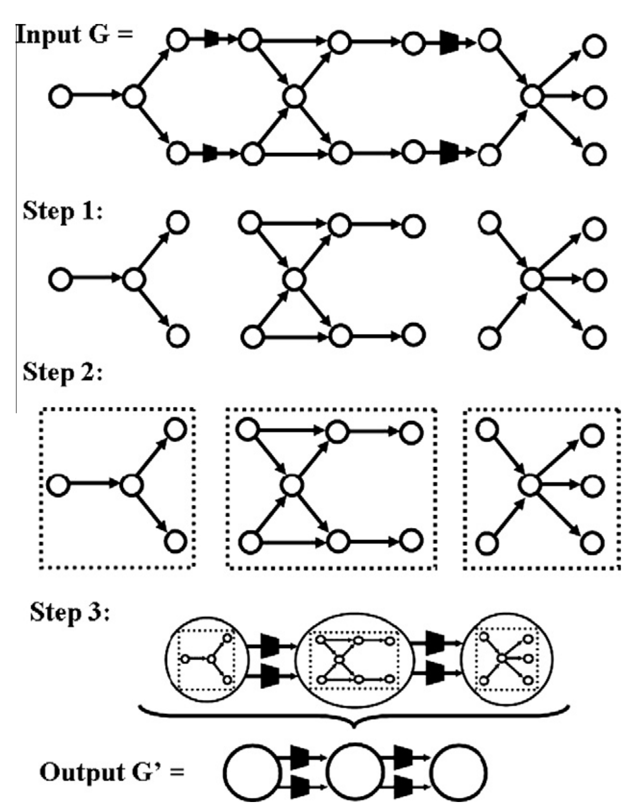


Fig. 1. Transition to compressor network.

Please cite this article in press as: Borraz-Sánchez, C., & Haugland, D. Minimizing fuel cost in gas transmission networks by dynamic programming and adaptive discretization. *Computers & Industrial Engineering* (2010), doi:10.1016/j.cie.2010.07.012

ven uniquely once the pressure at any reference node $v \in V$ is set to any value p^{ref}. If (x,p) also satisfies (9) and (10), the assignment $p_v = p^{\text{ref}}$ is feasible.

The observation that Theorem 1 applies to G^v for all $v \in V'$ suggests the following approach to Step 2 in Algorithm 1: Identify the connected components in (V,A_p), and nominate one reference node in each. Since x is fixed in this step, all other pressure values are found by utilizing (8), and feasibility is checked by verifying whether (9) holds. As pointed out by Ríos-Mercado et al. (2006), and exploited in the algorithm given in the same reference, it follows that Step 2 is reduced to the problem of solving instances of (5)–(10) where $A_p = \emptyset$ and x is fixed.

Theorem 1 shows that if x_{uv} and p_v are fixed for all compressor arcs $(u,v) \in A_c$ and all reference nodes $v \in V'$, the remaining variable values are computed by solving the system of equations consisting of (6) and (8). In Step 3, we thus keep p_v fixed for all $v \in V'$, and optimize over $\{x_{uv}:(u,v) \in A_c\}$. To respect the flow balance constraints (6), flow updates must be made by sending flow along cycles in G', and by identifying cycles with negative net cost a reduction in the objective function value is achieved. To check the cost of sending flow along a cycle, we have to take into account the change in x_{uv} for all compressor arcs (u,v) along the cycle, but also the change in p_v for all $v \in V \backslash V'$ in connected components of (V,A_p) intersected by the cycle. For more details, we refer the reader to Ríos-Mercado et al. (2002).

Step 3 will not be discussed further here. We define the problem FCMP' to be equivalent to (5)–(10), with the additional conditions that $A_p = \emptyset$ and x is fixed. With the purpose of developing efficient computational methods for Step 2, the rest of the paper is devoted to problem FCMP'.

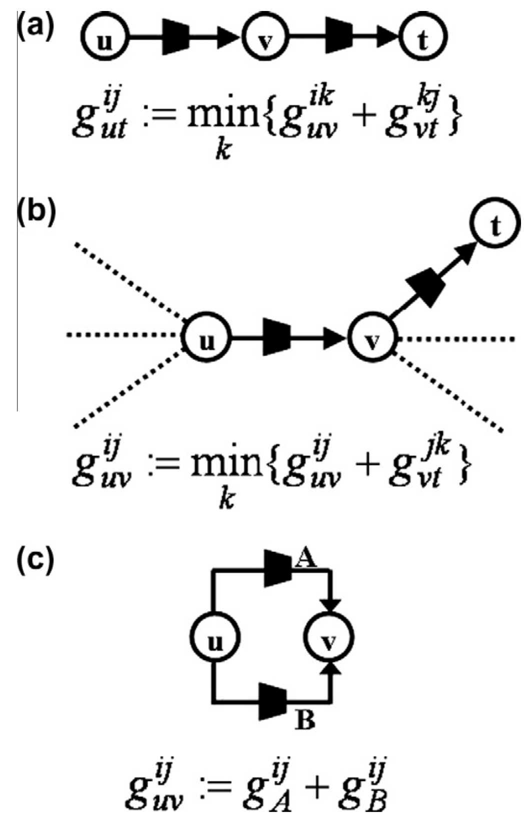


Fig. 2. Network reduction types.

3.2. Discretized pressure and dynamic programming formulation

Carter (1998) suggested to solve FCMP' by discretizing $[P^L,P^U]$ and then apply a network reduction technique referred to as non-sequential dynamic programming (NDP). Assume that there are τ discretization points $p_v^1,\ldots,p_v^\tau$ for each $v \in V$ such that $P_v^L \leqslant p_v^1 < \cdots < p_v^\tau \leqslant P_v^U$, and for all $i,j = 1,\ldots,\tau$, let $g_{uv}^{ij} = g_{uv}\left(x_{uv},p_u^i,p_v^j\right)$ if $\left(x_{uv},p_u^i,p_v^j\right) \in D_{uv}$ and $g_{uv}^{ij} = \infty$, otherwise. Then NDP consists of a sequence of reductions of G until the resulting graph is a single node. Three reduction types (see Fig. 2) are considered:

(a) *Serial*: If $v \in V$ has exactly two incident arcs (u,v) and (v,t) in G, then v, (u,v) and (v,t) are replaced by a new arc (u,t), and $g_{ut}^{ij} = \min_k\left\{g_{uv}^{ik} + g_{vt}^{kj} : \ k = 1,\ldots,\tau\right\}$. The same principle applies if both arcs incident to v enter (leave) v.

(b) *Dangling*: If $v \in V$ has only one incident arc (v,t), then t and (v,t) are removed, and, for all in-neighbors u of v in G, g_{uv}^{ij} is updated to $g_{uv}^{ij} + \min_k\left\{g_{vt}^{jk} : \ k = 1,\ldots,\tau\right\}$. Similar updates apply to the out-neighbors of v, and the principle applies also if the sole neighbor of t is an out-neighbor.

(c) *Parallel*: If $k > 1$ arcs $a_1,\ldots,a_k$ in G connect nodes u and v, then these are replaced by a single arc (u,v). The associated cost parameters are defined as $g_{uv}^{ij} = \sum_{\ell=1}^k g_{a_\ell}^{ij} \ \forall i,j = 1,\ldots,\tau$.

The serial and parallel reductions constitute the pre-processing procedure suggested by Koster, van Hoesel, and Kolen (1999).

When neither of the reductions (a)–(c) can be carried out, NDP fails. Fig. 3 shows a simple example where this occurs. To overcome this weakness, we now go on to demonstrate how such instances of FCMP' can be solved.

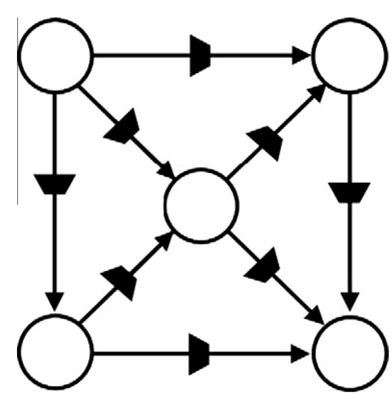


Fig. 3. An instance of G where NDP fails.

4. A tree decomposition approach to optimizing the pressure values

FCMP' has the mathematical structure of the *frequency assignment problem* (Koster et al., 1999), and can also be solved by the procedure suggested in the cited reference. This is based on the following concept introduced by Robertson and Seymour (1986):

Definition 1. A tree decomposition of G is a pair $\mathscr{I} = (\{X_i : i \in I\}, T)$, where each X_i is a subset of V, called a bag, and T is a tree with node set I. The following properties must be satisfied:

- $\bigcup_{i \in I} X_i = V$;
- for all $(u,v) \in A$, there is an $i \in I$ such that $\{u,v\} \subseteq X_i$;

- $\forall i,j,k \in I$, if j lies on the path between i and k in T, then $X_i \cap X_k \subseteq X_j$.

The width of a tree decomposition $\mathscr{J}$ is $\max_{i \in I}|X_i| - 1$.

For any $X \subseteq V$, define p_X as the vector with components p_v $(v \in X)$ in any consistent order. Define $\mathscr{D}_v = \{p_v^1, \ldots, p_v^\tau\}$ for all $v \in V$, and let $\mathscr{D}_X = \{p_X : p_v \in \mathscr{D}_v, \forall v \in X\}$. For any $i \in I$, let K_i denote the set of child nodes of i in T.

Algorithm 2. DP $(\mathscr{J}, i, X, \pi)$

if i is a leaf in T **then**
return

$$\min_{p \in \mathscr{D}_{X_i \cup X}} \left\{ \sum_{\substack{(u,v) \in A \\ u,v \in X_i \cup X}} g_{uv}(x_{uv}, p_u, p_v) : p_v = \pi_v \forall v \in X \right\}$$

else
return

$$\min_{p \in \mathscr{D}_{X_i \cup X}} \left\{ \sum_{\substack{(u,v) \in A \\ u,v \in X_i \cup X}} g_{uv}(x_{uv}, p_u, p_v) + \sum_{j \in K_i} \mathbf{DP}(\mathscr{J}, j, X_i \cup X, p) : p_v = \pi_v \forall v \in X \right\}$$

Algorithm 2 applies dynamic programming to a tree decomposition $\mathscr{J}$ of G. When bag X_i is processed, the union X of all ancestor bags of X_i are input along with a pressure vector $\pi \in \mathscr{D}_X$. The algorithm optimizes the value of p_v for all $v \in X_i$ by complete enumeration of $\mathscr{D}_v$, and by taking into account optimal pressure assignments to all nodes in all child bags of X_i. This is expressed in terms of a recursive call in Algorithm 2. Since $X_i \cap X$ may be non-empty, we must ensure that nodes contained in this set are not assigned new pressure values when processing X_i, and we impose the constraint that $p_v = \pi_v$ for all $v \in X$.

The running time of Algorithm 2 is $\mathscr{O}(|I|\tau^d)$, where d is the width of $\mathscr{J}$. This means that finding a tree decomposition of small width can be crucial for the running time of the algorithm. It is however well known (Robertson & Seymour, 1986) that finding one with minimum width is an NP-hard problem, and it is therefore unlikely that a tree decomposition minimizing the running time of Algorithm 2 can be found in polynomial time. We will rely on a heuristic approach to constructing $\mathscr{J}$ with small width.

4.1. Pre-processing variable bounds

In this section, we propose a bounding technique to be applied as a pre-processing technique in order to speed-up the convergence of our proposed methods. The aim of applying this pre-processing technique is to avoid as much as possible huge computational efforts when applying a finer discretization. We basically shrink all pressure bounds in G based on the maximum and minimum potential pressure values given by the physical properties in each compressor arc $(u, v) \in A$. We can define this bounding technique as elementary operations that may lead to better algorithmic properties before attempting to apply any of our proposed methods to G.

Given $\left[P_u^L, P_u^U\right], \forall u \in V$, the new refined pressure bounds can then be expressed as:

$$lb_u(P^L, P^U) = \max\left\{P_u^L, \Pi_u^L\right\} \leqslant p_u \leqslant ub_u(P^L, P^U) = \min\left\{P_u^U, \Pi_u^U\right\} \quad (11)$$

where (1) is used to obtain the pressure bounds

$$\Pi_u^L = \max_{v:(u,v) \in A} P_v^L \left(\frac{\kappa H_{uv}^U}{ZRT_s} + 1 \right)^{-(1/\kappa)} \quad (12)$$

and

$$\Pi_u^U = \min_{v:(u,v) \in A} P_v^U \left(\frac{\kappa H_{uv}^L}{ZRT_s} + 1 \right)^{-(1/\kappa)} \quad (13)$$

with κ as the isotropic factor defined as

$$\kappa = \frac{(1.287 - 1)}{1.287} \approx 0.223.$$

5. An adaptive discretization method

An important parameter of the approach suggested in the previous section, is the number of discretization points, τ, by which we represent each pressure variable. Assessing this parameter may be difficult. On the one hand, a large value of τ increases the possibility of finding a feasible solution of good quality. On the other hand, the previous section showed that the asymptotic increase in the running time is proportional to τ^d. With a large width d of the tree decomposition, choosing a large value of τ may lead to a slow method.

In this section, we therefore develop a method where the number of discretization points is initially small, and upgraded by a fixed factor until at least one feasible point is found by dynamic programming. Next, for each solution in a selection of the feasible ones hence found, we define an enclosing rectangular subset of the solution set, henceforth referred to as a *focus area*. The same procedure is then applied to each focus area.

By this approach, we focus the search in the neighborhood of some of the feasible solutions found, and repeat the idea recursively until the discretization distance within the focus area drops below a given threshold. The idea can be depicted by a search tree S (see Fig. 4), where each node corresponds to a unique focus area and the branches correspond to the set of feasible solutions found by DP and selected for further exploration. To limit the size of the search tree, only a fixed proportion of the feasible solutions are selected to be explored. If Ω is the set of feasible solutions found, we select the $\lceil \sigma|\Omega| \rceil$ solutions in Ω with the smallest cost, where $\sigma \in (0, 1]$ is an input parameter.

The dynamic programming algorithm (Algorithm 2) can easily be generalized such that it produces a set of solutions rather than only the best solution found. For all possible value assignments to the variables corresponding to the root bag of $\mathscr{J}$, we make optimal value assignments to all the remaining variables. Hence, $|\Omega| \leqslant \tau^{|X_0|}$, where X_0 is the root bag of $\mathscr{J}$. Only a trivial modification of Algorithm 2, where the root of $\mathscr{J}$ is treated differently from the other bags, is needed, and for reasons of brevity we omit the details. The resulting algorithm, denoted by DP', returns Ω and takes as input the same arguments as does Algorithm 2.

If DP' returns the empty set when τ is set equal to an initial number τ_0 of discretization points, we update τ by a fixed factor γ and call DP' again. The process is repeated until $\Omega \neq \emptyset$ or $\Delta_v = \frac{P_v^U - P_v^L}{\tau - 1} < \epsilon$ for all $v \in V$, where the threshold value ϵ is an input parameter.

The focus area around any selected solution $p \in \Omega$ is defined as the Cartesian product of the intervals $[lb_v(p - \Delta), ub_v(p + \Delta)]$, where $\Delta \in \Re^V$ is the vector with components $\Delta_v (v \in V)$. Hence, the range of a variable in the child node covers at most two consecutive intervals between discretization points in the parent node.

Please cite this article in press as: Borraz-Sánchez, C., & Haugland, D. Minimizing fuel cost in gas transmission networks by dynamic programming and adaptive discretization. *Computers & Industrial Engineering* (2010), doi:10.1016/j.cie.2010.07.012

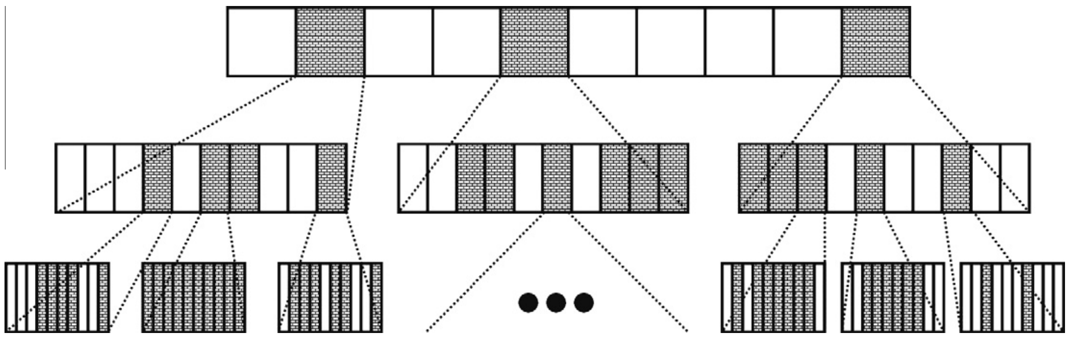


Fig. 4. Search tree based on adaptive discretization.

However, the improved bounds defined by (11) are likely to narrow down the range.

A summary of the approach is given in Algorithms 3 and 4.

Algorithm 3. adaptiveDiscretization $(\mathscr{J}, G, p^L, p^U, \epsilon, \sigma, \tau_0, \gamma)$

$(\Omega, \tau) \leftarrow$ findFeasibleSolutions$(\mathscr{J}, G, p^L, p^U, \epsilon, \sigma, \tau_0, \gamma)$
$z \leftarrow \infty$
if $\Omega \neq \emptyset$ **then**
  $z \leftarrow \min\left\{\sum_{(u,v)\in A} g_{uv}(x_{uv}, p_u, p_v) : p \in \Omega\right\}$
  Let $\Omega^{'} \subseteq \Omega$ consist of the $\lceil \sigma|\Omega| \rceil$ solutions in Ω with smallest cost
  $\Delta \leftarrow \frac{1}{\tau-1}(P^U - P^L)$
  for all $p \in \Omega'$ **do**
    $z = \min\{z,$ adaptiveDiscretization $(\mathscr{J}, G, lb(p^L - \Delta), ub(p^U + \Delta), \epsilon, \sigma, \tau_0, \gamma)\}$

Algorithm 4. findFeasibleSolutions$(\mathscr{J}, G, p^L, p^U, \epsilon, \sigma, \tau_0, \gamma)$

$\tau \leftarrow \tau_0$
repeat
  $\Omega \leftarrow \mathrm{DP}'(p^L, p^U, \tau)$
  $\tau \leftarrow \lceil \gamma\tau \rceil$
until $\Omega \neq \emptyset$ or $\frac{p_v^U - p_v^L}{\tau-1} < \epsilon \; \forall v \in V$
return (Ω, τ)

6. Numerical experiments

6.1. Overview of the experiments

In the first experiment, we examine the performance of the dynamic programming approach when the number of discretization points is kept fixed. We let $\tau \in \{50, 100, 1000\}$, and let the pressure values be uniformly distributed between their lower and upper bounds. The purpose of the experiment is to study the impact of τ on the quality of the solution and the running time.

In the second experiment, we analyze the performance of the adaptive discretization approach, and compare it to fixed discretization. The idea behind the experiment is to investigate whether adaptive discretization produces solutions comparable to those of fixed discretization in less computer time.

The third experiment is a similar comparison between the dynamic programming approaches and the commercially available local optimizer, *MINOS* (Murtaugh & Saunders, 1983). Since the local optimum output by MINOS, if any, turns out to be sensitive to the starting point, we run MINOS for 500 and 1000 randomly generated starting points. The points are drawn from the uniform distribution on $[p^L, p^U]$.

The fourth experiment is a comparison between the solutions produced by our methods to (a lower bound on) the true minimum cost. We submit FCMP' to the generic global optimization tool, *BARON* (Tawarmalani & Sahinidis, 2004), which is an implementation of a variant of branch-and-bound where a convex program is solved in each node of the search tree. To solve the convex subproblems, BARON is set to call MINOS. We impose a time limit of 3600 CPU-seconds on each application of BARON, and the relative optimality tolerance be 0.01. That is, any feasible solution is considered to be optimal if the gap between the objective function value and its lower bound is below one percent of the objective function value. In instances where BARON fails to compute the global optimum, it may still provide a lower bound on the minimum cost, and this bound may also give some indications on the quality of the output from our methods.

Our solution procedures were coded in C++ under Linux Red Hat, and all experiments were run on a 2.4 GHz Intel(R) processor with 2 GByte RAM. To compute the tree decomposition $\mathscr{J}$ to be input to the dynamic programming algorithms, we apply the technique given by Subbarayan (2007) based on *Maximum Cardinality Search* (Tarjan & Yannakakis, 1984). Experiments with BARON and MINOS were conducted by formulating the model in GAMS (GAMS Development Corporation, 2008), and we have used version 8.1.5 of BARON and version 5.51 of MINOS.

6.2. Test instances

All experiments reported in this work were carried out on the set of test instances shown in Table 1. Each row gives an identifier of an instance, the size in terms of nodes and arcs in G after reduction, and the type of compressor used. We consider nine different compressor types, and all compressors are identical within any given instance. Furthermore, the width and the number of bags in the tree decomposition are given in the two last columns of Table 1.

All test instances can be downloaded in GAMS-format at http://www.ii.uib.no/~conrado/caie/instances/index.html.

6.3. Results

Table 2 shows the results achieved by fixed discretization for three different values of τ. Instance references are given in the first column, and computation times (CPU-seconds) and objective function values for the respective values of τ are given in columns 2–7. The only case where the method failed to find a feasible solution was for $\tau = 50$ in instance K. We observe that as τ increases, better solutions are found (minimum cost decreases) in all instances, except from a cost increase from $\tau = 50$ to $\tau = 100$ in instances O and P. Nevertheless, a finer discretization also implies, as expected, that

Please cite this article in press as: Borraz-Sánchez, C., & Haugland, D. Minimizing fuel cost in gas transmission networks by dynamic programming and adaptive discretization. *Computers & Industrial Engineering* (2010), doi:10.1016/j.cie.2010.07.012

Table 1
Test instances.

Ref	Size		Type	$\mathcal{J}$	
	$\lvert V\rvert$	$\lvert A'_c\rvert$		Width	$\lvert I\rvert$
A	3	3	3	3	1
B	3	3	4	3	1
B1	3	3	5	3	1
B2	3	3	6	3	1
B3	3	3	8	3	1
C	4	6	1	3	3
D	4	6	2	3	4
E	4	6	3	3	5
F	4	6	4	3	4
G	4	6	5	3	4
H	4	6	6	3	3
I	4	6	7	3	4
J	5	8	4	3	6
K	5	8	8	3	4
L	9	20	4	4	9
M	9	20	5	4	9
N	18	25	2	4	15
O	18	25	4	4	15
P	18	25	9	3	18
Q	8	10	4	3	6
R	8	10	6	3	6
S	17	23	6	4	8

Table 2
Performance of dynamic programming with fixed discretization.

Ref	$\tau = 50$		$\tau = 100$		$\tau = 1000$	
	CPU (secs)	Obj ($\times 10^6$)	CPU (secs)	Obj ($\times 10^6$)	CPU (secs)	Obj ($\times 10^6$)
A	0	1.12	0	0.77	1	0.75
B	0	2.63	0	2.62	2	2.62
B1	0	2.83	0	2.63	138	2.61
B2	0	3.30	0	2.98	138	2.84
B3	0	2.17	0	2.08	132	1.83
C	1	10.29	16	9.34	1935	8.93
D	0	7.45	11	7.34	1047	7.34
E	1	9.66	22	6.36	1082	5.29
F	2	6.87	30	5.69	1845	4.12
G	1	9.43	10	6.30	817	6.30
H	1	6.34	13	5.93	692	5.09
I	1	2.83	10	2.82	529	2.77
J	1	6.07	13	5.59	842	5.27
K	1	–	9	35.67	772	35.67
L	3	68.89	50	61.83	2987	61.73
M	3	89.68	40	74.80	2715	60.74
N	2	60.71	34	52.46	2554	46.00
O	6	63.03	180	63.38	3422	38.80
P	1	35.25	23	37.67	2417	26.54
Q	3	23.31	80	21.32	3015	15.15
R	5	24.02	149	22.78	3310	20.10
S	15	72.01	482	69.59	3662	65.51

Table 3
Performance of dynamic programming with adaptive discretization.

Ref	CPU (secs)	Calls to DP'	Obj ($\times 10^6$)
A	1	395	0.75
B	1	117	2.62
B1	0	267	2.60
B2	0	12	2.83
B3	0	11	1.82
C	1	429	7.79
D	2	16	7.35
E	1	263	5.29
F	11	852	3.94
G	1	239	5.87
H	4	319	5.17
I	4	558	2.76
J	2	122	5.18
K	2	204	31.32
L	35	210	63.14
M	16	212	54.64
N	26	631	38.09
O	31	596	29.55
P	44	547	24.34
Q	13	438	14.58
R	17	1631	15.96
S	60	815	62.46

Table 4
Performance of MINOS.

Ref	500 Starting points			1000 Starting points			*RI* (%)
	CPU (secs)	Feas (%)	Obj ($\times 10^6$)	CPU (secs)	Feas (%)	Obj ($\times 10^6$)	
A	30	59.2	0.75	64	59.4	0.75	0.0
B	59	100.0	2.63	115	99.8	2.62	0.4
B1	21	37.4	2.62	55	37.7	2.62	0.0
B2	26	22.2	2.83	57	24.3	2.83	0.0
B3	18	20.4	1.82	52	22.3	1.82	0.0
C	29	7.4	9.09	66	7.8	9.03	0.7
D	31	19.6	7.36	88	20.4	7.36	0.0
E	36	31.2	6.02	68	29.0	6.01	0.2
F	53	23.8	4.21	117	23.2	4.15	1.4
G	20	0.0	–	61	0.0	–	–
H	54	39.2	5.45	129	40.7	5.45	0.0
I	52	0.0	–	124	0.0	–	–
J	49	50.0	6.14	110	49.3	5.98	2.6
K	65	0.0	–	120	0.0	–	–
L	125	2.8	68.09	241	2.5	68.09	0.0
M	19	0.0	–	36	0.0	–	–
N	58	0.0	–	124	0.0	–	–
O	52	1.0	39.22	139	0.7	39.22	0.0
P	37	0.0	–	75	0.0	–	–
Q	147	0.0	–	365	0.0	–	–
R	102	0.0	–	238	0.0	–	–
S	57	0.0	–	116	0.0	–	–

the computational requirements increase, and the running time slightly exceeds one CPU-hour in one instance (S).

Table 3 shows the results achieved by dynamic programming and adaptive discretization. In these runs, we have used the parameter values (see Section 5) $\tau_0 = 3$, $\gamma = 1.5$, $\sigma = 0.05$ and $\epsilon = 0.001$. Instance references are given in the first column, and computation time (CPU-seconds), number of calls to DP' in Algorithm 4, and objective function values for the corresponding test instance are given in columns 2–4. We observe that the running time slightly exceeds 30 CPU-seconds in the most time-consuming instance (S).

Table 4 shows corresponding results from MINOS. Columns 2–4 give respectively the CPU-time, percentage of the 500 starting points by which MINOS found a feasible solution, and the cost of the best feasible solution found. Columns 5–8 give corresponding results for 1000 starting points. We observe that MINOS fails to find a feasible solution in 9 instances, and in the other instances (except A and B), it does so for at least 50% of the starting points. On the other hand, the solver is fast, and a relatively large number of starting points is affordable.

Tables 5 and 6 give a comparison between MINOS (with 1000 starting points) and dynamic programming with the two discretization techniques. In Table 5, we summarize and compare running times, while costs are compared in Table 6.

First, we observe from Table 5 that adaptive discretization is much faster than fixed discretization, although the latter requires only one call to the DP-algorithm. The larger number of calls to DP' reported in Table 3, seems to be more than compensated for

Table 5
Dynamic programming vs. MINOS: CPU-time.

Ref	MINOS		Discretization	
	500 Iters	1000 Iters	Fixed[a]	Adaptive
A	30	64	1	1
B	59	115	2	1
B1	21	55	138	0
B2	26	57	138	0
B3	18	52	132	0
C	29	66	1935	1
D	31	88	1047	2
E	36	68	1082	1
F	53	117	1845	11
G	20	61	817	1
H	54	129	692	4
I	52	124	529	4
J	49	110	842	2
K	65	120	772	2
L	125	241	2987	35
M	19	36	2715	16
N	58	124	2554	26
O	52	139	3422	31
P	37	75	2417	44
Q	147	365	3015	13
R	102	238	3310	17
S	57	116	3662	60

[a] With $\tau = 1000$.

Table 6
Dynamic programming vs. MINOS: cost.

Ref	Cost ($\times 10^6$)			Adaptive vs. (*RI*%)	
	MINOS[a]	Fixed[b]	Adaptive	MINOS[a]	Fixed[b]
A	0.75	0.75	0.75	0.0	0.0
B	2.62	2.62	2.62	0.0	0.0
B1	2.62	2.61	2.60	0.8	0.4
B2	2.83	2.84	2.83	0.0	0.4
B3	1.82	1.83	1.82	0.0	0.5
C	9.03	8.93	7.79	13.7	12.8
D	7.36	7.34	7.35	0.1	-0.1
E	6.01	5.29	5.29	12.0	0.0
F	4.15	4.12	3.94	5.1	4.4
G	–	6.30	5.87	–	6.8
H	5.45	5.09	5.17	5.1	-1.6
I	–	2.77	2.76	–	0.4
J	5.98	5.27	5.18	13.4	1.7
K	–	35.67	31.32	–	12.2
L	68.09	61.73	63.14	7.3	-2.3
M	–	60.74	54.64	–	10.0
N	–	46.00	38.09	–	17.2
O	39.22	38.80	29.55	24.7	23.8
P	–	26.54	24.34	–	8.3
Q	–	15.15	14.48	–	4.4
R	–	20.10	15.96	–	20.6
S	–	65.51	62.46	–	4.7

[a] With 1000 starting points.
[b] With $\tau = 1000$.

by the smaller number of discretization points. Second, Table 6 shows that in all instances but D, H and L, where fixed discretization gives up to 2.3% lower cost, the faster approach gives solutions of equal or better quality.

Also when compared to MINOS, the adaptive discretization approach turns out to be superior. Applying MINOS with a single random starting point is certainly faster, but this involves a considerable risk of failing to find a feasible solution. When the number of random starting points is increased such that the total running time of MINOS exceeds the one of adaptive discretization, the total cost of the best MINOS solution is in general higher than the cost of the solution produced by its competitor. The relative improvement of adaptive discretization when compared to MINOS and fixed discretization, respectively, is given in columns 5–6 of Table 6.

Table 7
Adaptive discretization vs. a global optimizer.

Ref	Performance of BARON				Adaptive discretization		
	#Its	#Nodes	Obj	LB	Obj	RI(%)	GAP(%)
A	551	131	0.75	0.75	0.75	0.0	0.0
B	1148	342	2.62	2.62	2.62	0.0	0.0
B1	47	5	2.62	2.59	2.60	0.4	0.8
B2	125	11	2.83	2.81	2.83	0.0	0.0
B3	37	5	1.82	1.80	1.82	0.0	0.0
C	21521	7462	9.02	4.45	7.79	13.6	42.9
D	445	38	7.35	7.28	7.35	0.0	1.0
E	17059	7023	5.30	4.02	5.29	0.2	24.0
F	26765	7480	3.94	2.71	3.94	0.0	31.2
G	5231	1283	–	2.27	5.87	–	61.3
H	2109	204	5.19	5.04	5.17	0.4	2.5
I	3267	324	–	2.73	2.76	–	1.1
J	27832	2299	5.15	5.10	5.18	−0.6	1.5
K	14968	3344	–	20.86	31.32	–	33.4
L	740	451	65.94	43.81	63.14	4.2	30.6
M	2438	765	–	31.12	54.64	–	43.0
N	1830	839	–	34.28	38.09	–	10.0
O	234	59	–	22.13	29.55	–	25.1
P	978	655	–	17.43	24.34	–	28.4
Q	330	212	17.43	12.82	14.58	16.9	11.5
R	1182	468	15.94	13.09	15.96	−0.1	18.0
S	3123	632	59.39	44.17	62.46	−5.2	29.3

Table 7 shows the performance of BARON when applied to the test instances. In addition, we compare the adaptive discretization approach to the best results obtained by BARON in terms of the quality of the solution. Columns 2–5 contain the number of iterations in BARON (the number of convex subproblems solved), the maximum number of open nodes the search tree ever had, the objective function value of the best feasible solution found (if any), and the lower bound on the minimum cost. For more convenient comparison, we give the cost obtained by adaptive discretization in column 6 (identical to column 3 in Table 3), and in the 7th column, we give whenever applicable the relative improvement (in percentages) of these solutions over the best BARON solutions.

As seen in Table 7, in 15 out of 22 instances, BARON was able to find a feasible solution, and in two instances (A and B), it was able to prove optimality within the given tolerance. In the remaining instances, no feasible solution was found before the time limit expired. By comparing columns 5 and 6, we also observe that the relative optimality gap (relative distance from minimum cost) of adaptive discretization in one instance (G) may be as large as 61.3%. In the instances where BARON found a feasible solution, the largest gap is 42.9% (instance C).

Column 7 of Table 7 shows that BARON is able to find a better solution than does our method in instances J, R and S. However, extensive computations were needed to find these solutions. The last column of the table gives the relative distance from the lower bound on optimality provided by BARON. In nine of the instances, we are less then 3% from the minimum cost, but in some instances, the optimality gap is large (as large as 61.3% in instance G). It is however unknown whether this is due to weak lower bounds or shortcomings of our algorithm.

7. Concluding remarks

In this paper, we have studied a model (FCMP) for minimizing compressor fuel cost in transmission networks for natural gas. An arc in the network model corresponds to either a pipe or a compressor, and the decision variables are arc flow and node pressure.

In addition to flow conservation constraints, the model contains non-linear constraints relating pipeline flow to inlet and outlet pressure, as well as non-convex constraints defining the operation domain of the compressors.

Following a general algorithmic idea, which has been suggested and supported experimentally in several recent works, we consider a procedure where each iteration consists of a flow improvement step and a pressure optimization step. Alternating between flow and pressure, one set of decision variables is kept fixed in each step. Still in agreement with previously suggested methods, the non-convex subproblem of optimizing pressure is approximated by a combinatorial one. This is accomplished by discretization of the pressure variables.

The contribution of this paper is a method for solving the discrete version of the problem in instances where previously suggested methods fail. Unlike methods based on successive network reductions, our method does not make any assumptions concerning the sparsity of the network. By constructing a tree decomposition of the network, and apply dynamic programming to it, we are able to solve the discrete version of the pressure optimization problem without enumerating the whole solution space. By an adaptive discretization scheme, we obtain significant speed-up of the dynamic programming approach in comparison with fixed discretization.

We have tested our solution methods on a set of imaginary instances, and compared the results to those obtained by applying both a global and a local optimizer to the continuous version of the problem. The experiments indicate that a method guaranteeing the global optimum in reasonable time seems unrealistic even for small instances. Further, discretizing the pressure variables and applying dynamic programming to a tree decomposition gives better results than applying a commercially available local optimization package.

Non-convex continuous optimization problems can in general be approached by discretization of the variable space, and in many cases, the resulting discrete problem can be solved by dynamic programming. The challenge of finding the ideal balance between accuracy in the discrete model and speed of the DP-algorithm is however hardly avoidable by the approach. We therefore believe that the adaptive discretization scheme developed in this paper may have merit beyond the specific application in gas transmission networks studied here.

References

Borraz-Sánchez, C., & Ríos-Mercado, R. Z. (2009). Improving the operation of pipeline systems on cyclic structures by tabu search. *Computers & Chemical Engineering, 33*(1), 58–64.

Borraz-Sánchez, C., & Ríos-Mercado, R. Z. (2004). A non-sequential dynamic programming approach for natural gas network optimization. *WSEAS Transactions on Systems, 3*(4), 1384–1389.

Carter, R. G. (1998). Pipeline optimization: Dynamic programming after 30 years. In *Proceedings of the PSIG meeting, Denver, CO, USA, October 1998* (pp. 1–19).

Cobos-Zaleta, D., & Ríos-Mercado, R. Z. (2002). A MINLP model for minimizing fuel consumption on natural gas pipeline networks. In *Proceedings of the XI-Latin-Ibero-American conference on operations research, Concepcion, Chile* (pp. 1–9).

Flores-Villarreal, H. J., & Ríos-Mercado, R. Z. (2003). Efficient operation of natural gas pipeline networks: Computational finding of high quality solutions. In *Proceedings of the international applied business research conference, paper 352, Acapulco, Mexico, March 2003* (pp. 1–7).

GAMS Development Corporation (2008). GAMS: The solver manuals. Washington, DC, USA.

Koster, A. M. C. A., van Hoesel, S. P. M., & Kolen, A. W. J. (1999). Optimal solutions for frequency assignment problems via tree decomposition. *Lecture Notes in Computer Science, 1665*, 338–350.

Lall, H. S., & Percell, P. B. (1990). A dynamic programming based gas pipeline optimizer. In A. Bensoussan & J. L. Lions (Eds.), *Analysis and optimization of systems* (pp. 123–132). Berlin, Germany: Springer.

Martch, H. B., & McCall, N. J. (1972). Optimization of the design and operation of natural gas pipeline systems. Paper no. SPE 4006, Society of Petroleum Engineers of AIME.

Murtaugh, B. A., & Saunders, M. A. (1983). MINOS 5.1 user's guide, technical report SOL-83-20R. Stanford, CA, USA: Systems Optimization Laboratory, Stanford University.

Percell, P. B., & Ryan, M. J. (1987). Steady-state optimization of gas pipeline network operation. In *Proceedings of the 19th PSIG annual meeting, Tulsa, USA*.

Ríos-Mercado, R. Z., Kim, S., & Boyd, E. A. (2006). Efficient operation of natural gas transmission systems: A network-based heuristic for cyclic structures. *Computers & Operations Research, 33*(8), 2323–2351.

Ríos-Mercado, R. Z., Wu, S., Scott, L. R., & Boyd, E. A. (2002). A reduction technique for natural gas transmission network optimization problems. *Annals of Operations Research, 117*(1–4), 217–234.

Robertson, N., & Seymour, P. D. (1986). Graph minors II. Algorithmic aspects of treewidth. *Journal of Algorithms, 7*, 309–322.

Subbarayan, S. (2007). An empirical comparison of CSP decomposition methods. In Hnich, B., & Stergiou, K. (Eds.), *Proceedings of the CP 2007 doctoral programme, Providence, RI, USA, September 23–27* (pp. 163–168).

Tarjan, R. E., & Yannakakis, M. (1984). Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM Journal of Computing, 13*, 566–579.

Tawarmalani, M., & Sahinidis, N. V. (2004). Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Mathematical Programming, 99*(3), 563–591.

Wong, P. J., & Larson, R. E. (1968). Optimization of natural gas pipeline systems via dynamic programming. *IEEE Transactions on Automatic Control, 13*(5), 475–481.

Wu, S. (1998). Steady-state simulation and fuel cost minimization of gas pipeline networks. Ph.D. Thesis. Houston, TX: University of Houston.

Wu, S., Ríos-Mercado, R. Z., Boyd, E. A., & Scott, L. R. (2000). Model relaxations for the fuel cost minimization of steady-state gas pipeline networks. *Mathematical and Computer Modeling, 31*(2-3), 197–220.

Villalobos-Morales, Y., Cobos-Zaleta, D., Flores-Villarreal, H. J., Borraz-Sánchez, C., & Ríos-Mercado, R. Z. (2003). 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, Birmingham, USA*.

Zimmer, H. I. (1975). Calculating optimum pipeline operations. Technical report, Natural Gas Company El Paso, presented at the AGA transmission meeting, Texas, USA.

Please cite this article in press as: Borraz-Sánchez, C., & Haugland, D. Minimizing fuel cost in gas transmission networks by dynamic programming and adaptive discretization. *Computers & Industrial Engineering* (2010), doi:10.1016/j.cie.2010.07.012

PAPER IV

Optimization methods for pipeline transportation of natural gas with variable specific gravity and compressibility*

Conrado Borraz-Sánchez Dag Haugland

* Preprint submitted to *the International Journal of the Spanish Society of Statistics and Operations Research: TOP. (2010)*

TOP manuscript No.
TOPR-D-10-00014

Optimization methods for pipeline transportation of natural gas with variable specific gravity and compressibility

Conrado Borraz-Sánchez · Dag Haugland

Received: date / Accepted: date

Abstract In this paper, the problem of flow maximization in pipeline systems for transmission of natural gas is addressed. We extend previously suggested models by incorporating the variation in pipeline flow capacities with gas specific gravity and compressibility. Flow capacities are modeled as functions of pressure, compressibility and specific gravity by the commonly-used Weymouth equation, and the California Natural Gas Association method is used to model compressibility as a function of specific gravity and pressure. The sources feeding the transmission network do not necessarily supply gas with equal specific gravity. In our model, it is assumed that when different flow streams enter a junction point, the specific gravity of the resulting flow is a weighted average of the specific gravities of entering flows. We also assume the temperature to be constant, and the system to be in steady state.

Since the proposed model is non-convex, and global optimization hence can be time consuming, we also propose a heuristic method based on an iterative scheme in which a simpler NLP model is solved in each iteration. Computational experiments are carried out in order to assess the computability of the model by applying a global optimizer, and to evaluate the performance of the heuristic approach. When applied to a wide set of test instances, the heuristic method provides solutions with deviations less than 12% from optimality, and in many instances turns out to be exact. We also report several experiments demonstrating that letting the compressibility and the specific gravity be global constants can lead to significant errors in the estimates of the total network capacity.

Keywords Natural gas · Compressibility factor · Specific gravity · Weymouth equation · Transmission network · Nonlinear optimization · CNGA method

1 Introduction

Steady state models for optimizing pipeline transportation of natural gas are distinguished from classical network flow models by their non-constant arc capacities. This is implied by the dependence between the pipeline flow and the pressure drop along the pipeline, and the fact that the pressure can be considered as a state variable. The pressure value at any network node is in its turn determined by flow and pressure values found in network elements in the upstream direction of the node.

Borraz-Sánchez and Haugland
Department of Informatics, University of Bergen. P. B. 7803, N-5020 Bergen, Norway.
Tel.: +47 555-84150, Fax: +47 555-84199
E-mail: {Conrado.Borraz-Sanchez, Dag.Haugland}@ii.uib.no

The idea of modeling arc capacities as decision dependent functions is already well established in the optimization literature. In an early work on minimization of compressor fuel cost, Wong and Larson (1968) suggested to model the pipeline capacity by means of the popular Weymouth equation (Osiadacz, 1987). In more developed versions, the same principle has been applied by, e.g., Carter (1999), Ríos-Mercado et al. (2002), De Wolf and Smeers (1994, 2000), Borraz-Sánchez and Ríos-Mercado (2004, 2009), Bakhouya and De Wolf (2007), Kalvelagen (2007), and Borraz-Sánchez and Haugland (2009).

All the cited works neglect the fact that the parameter in the Weymouth equation depends not only on pipeline characteristics, but also on thermodynamic and physical gas properties. This includes temperature, specific gravity (relative density) and compressibility (z-factor). In instances where the network elements show no or only modest variation in these properties, it is sound modeling practice to represent them by global constants. This does however not seem to be the case in all real-life instances.

The pipeline network connecting wells on the Norwegian continental shelf with the European continent is supplied by gas from sources of relatively lean gas, situated in the North Sea, and sources located in e.g. the Haltenbank area. Since the latter area generally has richer gas, in the sense that it consists of components of higher specific gravity, the assumption of constant properties may be unrealistic. In this paper, we consequently extend previously suggested models such that variability in gas specific properties is reflected.

The first objective of the current work is to assess the improvement in estimates of total throughput gained by modeling variable gas properties. It is unlikely that such an improvement comes without additional computational cost, and this cost must be assessed. Second, we address the question of how the improved model can be solved approximately in instances where exact optimal solutions are too time consuming. The goal is to have a fast computational procedure, which also is sufficiently accurate to avoid that improved model realism gets lost in weak computational effectiveness.

To accomplish these objectives, we formulate a model where not only the pressure, but also compressibility and specific gravity are defined as state variables. The gas temperature is assumed to be constant also in this work. We assume that the network is in steady state, which essentially means that time variations can be neglected. Implicitly, all state variables become functions of the flow variables. The non-convex nature of these functions implies that also the feasible region of the model is non-convex.

Global optimization is expected to be time consuming in many instances of realistic size, and we therefore also propose a fast heuristic method. Our method is based on an iterative scheme in which, for given values of the gas specific gravity in each network element, a simpler model is solved in each iteration. By the end of each iteration, the specific gravity is assigned a value consistent with flow values observed in the current iteration. A computational evaluation to assess the performance of the proposed method, including a comparison with a global optimizer, is presented. The procedure is repeated until all variables converge.

The literature on optimization models for pipeline gas transportation does not seem to very rich on models with variable specific gravity or compressibility, and most works focus on models for transient flow. Interested readers are referred to the works by Abbaspour and Chapman (2008) and Chaczykowski (2010), and the references therein.

The remainder of this paper is organized as follows. In Section 2, we develop the mathematical model. Special emphasis is put on formulation of the relations between the various state variables, and how they relate to the flow. A heuristic method for the resulting non-convex model is described in Section 3. In Section 4, we make an experimental comparison between our model and a model based on the traditional approach of considering gas properties as global constants. The section also reports computational results from a comparison between our heuristic method and the results from a global optimizer applied to our model. Finally, concluding remarks are given in Section 5.

Table 1: Parameters and their typical values in English units

Parameters	Typical value
Pipe efficiency factor, e :	$[0.85, 1] \rightarrow 0.92$
Darcy-Weisbach friction factor, f :	8.5×10^{-4}
Pipe length, L :	$[15, 100](miles)$
Pipe inside diameter, d :	$[3, 5](ft)$
Pipe resistance factor, K :	$(1.3305 \times 10^5)T$
Net mass flow rate LB and UB, $[B^L, B^U]$:	$Known(lbf/min)$
Pressure LB and UB, $[p^L, p^U]$:	$[200, 1200](psig)$

2 Problem Definition

Let $G = (V, A)$ be a directed acyclic graph representing a gas transmission network, where V and A are the node (junction point) and arc (pipeline) sets, respectively. Let $V_s \subseteq V$ be the set of supply nodes (sources), and let $V_d \subseteq V$ be the set of demand nodes.

For each node $i \in V$, we define the net mass flow rate as the variable b_i. By convention, $b_i \geq 0$ if $i \in V_s$, $b_i \leq 0$ if $i \in V_d$, and $b_i = 0$ otherwise. For any node $i \in V$, let $V_i^- = \{j \in V : (j, i) \in A\}$ be the set of start nodes of incoming arcs, and let $V_i^+ = \{j \in V : (i, j) \in A\}$ be the set of end nodes of outgoing arcs. Let p_i^L and p_i^U, respectively, be a lower and an upper pressure bound on the pressure at node $i \in V$. A full list of parameters associated with the models presented in this paper, including their typical values, is shown in Table 1.

2.1 Assumptions

Several assumptions are necessary to delimit the scope of our proposed model. We work on natural gas transmission networks with large diameter pipelines that operate at high pressures (200 psig and beyond). The model to be presented is not intended for lower pressures, since we do not expect it to reflect reality in such instances. We assume that the mass is conserved at each node in the network, which means that the decision variables $x_{ij}, \forall(i, j) \in A$ representing arc flow are subject to the constraints:

$$\sum_{j \in V_i^+} x_{ij} - \sum_{j \in V_i^-} x_{ji} - b_i = 0, \forall i \in V. \tag{1}$$

For simplicity, we confine our study to irreversible flow in steady-state, i.e, the gas can flow through a pipeline in only one direction. Extension of our model to a bidirectional flow model, which may become relevant e.g. when connecting storage facilities to the network, is however straightforward.

The gas flow is considered isothermal at an average effective temperature. We also assume all pipelines to be horizontal pipelines, although in practice, transmission lines have frequent changes in their elevation. However, the need for correction factors to compensate these changes in elevation would require special attention out of the scope of the current work.

We assume $V_i^- = \emptyset \; \forall i \in V_s$, i.e., there are no incoming flow streams (pipes) to a node which has been identified as source node. Instances violating this assumption can be converted by adding a dummy node pointing toward the original source node. The dummy node inherits all properties of the source node that it replaces, which in its turn becomes a simple transshipment node.

2.2 Relating flow capacity to gas properties

Following Shashi Menon (2005), both the physical properties of the pipelines and the composition of the gas have an influence on the flow capacity. Several equations have been proposed for modeling the steady state flow in pipelines. We can find the Weymouth equation developed in 1912 (see Osiadacz, 1987), the Panhandle A equation developed in 1940, and the Panhandle B equation developed in 1956, among others. Details these equations can be found in e.g., Crane (1982) and Modisette (2000).

In this paper, we make use of the Weymouth equation due to its simplicity and its accuracy when applied to gas flows at high pressures. The Weymouth equation defines the relationship between the mass flow rate x_{ij} through a pipeline $(i, j) \in A$ and the corresponding difference between the squares of the inlet and outlet pressures p_i and p_j, respectively. We also refer to it as the pipeline resistance equation, which takes the following form for flow in a horizontal pipe:

$$x_{ij}^2 = W_{ij}\left(p_i^2 - p_j^2\right), \forall (i, j) \in A. \tag{2}$$

Parameter W_{ij}, referred to as the Weymouth factor, depends on gas and pipeline properties as given by

$$W_{ij} = \frac{d_{ij}^5}{K z_{ij} g_i T f_{ij} L_{ij}},$$

where z_{ij} is the compressibility of the flow in pipeline (i, j), g_i the specific gravity of the flow arriving at node i, T the gas temperature, f_{ij} is the (Darcy-Weisbach) friction factor in pipeline (i, j), L_{ij} is the length of pipeline (i, j), d_{ij} the inside diameter of pipeline (i, j), and K is a global constant with value defined by the units used.

By defining $w_{ij} = z_{i,j} g_i W_{ij}$, (2) can be written

$$z_{ij} g_i x_{ij}^2 = w_{ij}\left(p_i^2 - p_j^2\right), \forall (i, j) \in A. \tag{3}$$

We observe that w_{ij}, contrary to W_{ij}, is independent of the variable flow properties z_{ij} and g_i. For this reason, the Weymouth equation will henceforth be written on the form (3).

2.3 Gas compressibility

The gas compressibility, also referred to as the z-factor, can be considered as the deviation from ideal gas. More formally, it is defined as the relative change in gas volume in response to a change in pressure. The importance of good estimates of this parameter is obvious from (2) and the definition of W_{ij}.

The literature on gas metering reveals a number of diverse methods for approximating the z-factor, including experimental measurements, equations of state methods (Dranchuk and Abou-Kassem, 1975), empirical correlations (Katz et al., 1959) and regression analysis methods (Dranchuk et al., 1974; Gopal, 1977.

For instance, Katz et al. (1959) presented a graphical correlation for the z-factor as a function of pseudo-reduced temperature and pressure based on experimental data. As a result, the Standing-Katz z-factor chart has been used to obtain natural gas compressibility factors for more than 40 years. Dranchuk and Abou-Kassem (1975) used the equation of the state to fit the Standing-Katz data and extrapolated to higher reduced pressure. This was accomplished by a simple mathematical description of the Standing-Katz z-factor chart. The California Natural Gas Association (CNGA) developed a method (Davisson, 1965) to compute the z-factor based on the gas specific gravity, temperature and pressure values. All these methods have a domain where they are reasonably accurate, and may break down outside. A survey of the methods can be found in Shashi Menon (2005).

In this paper, we make use of the CNGA method, which is briefly described next.

2.3.1 The California Natural Gas Association method

The CNGA method has been in use since the middle of the last century. One of its first applications is reported by Davisson (1965), who makes use of the method in a computer program for precise flow calculations.

In our work, there are two key reasons for choosing this method. First, the method is valid for z-factor computations at high pressures, which agrees with the assumption made in Section 2.1. Second, in comparison with other procedures, the CNGA method presents a simple and effective way to compute compressibility as a function of specific gravity, temperature and pressure.

The CNGA equation can be stated as follows:

$$z_{ij} = \frac{1}{1 + \dfrac{\overline{p}_{ij}\alpha 10^{\beta g_i}}{T^{\delta}}}, \tag{4}$$

where $\overline{p}_{ij}$ is the average pipeline pressure, T is the gas temperature and g is the gas specific gravity, and α, β and δ are universal constants.

According to Shashi Menon (2005), by using (4), the estimation of z_{ij} typically becomes more accurate for transportation pipelines working at pressures beyond 100 psig. Since we assume gas streams under isothermal conditions, i.e., T is assumed fixed, (4) can be written

$$z_{ij}\left(1 + \omega\overline{p}_{ij} \times 10^{\beta g_i}\right) = 1, \tag{5}$$

where $\omega = \frac{\alpha}{T^{\delta}}$ is an instance specific constant.

2.4 Computing average pressure in a pipeline

The pressure is decreasing along the pipeline. According to Shashi Menon (2005), the formula

$$\overline{p}_{ij} = \frac{2}{3}\left(p_i + p_j - \frac{p_i p_j}{p_i + p_j}\right). \tag{6}$$

gives a better approximation to the average pressure $\overline{p}_{ij}$ in pipeline $(i, j) \in A$, than does the arithmetic mean of p_i and p_j. The suggested formula in its quadratic form $3\overline{p}_{ij}(p_i + p_j) = 2(p_i^2 + p_j^2 + p_i p_j)$ is adopted in the current work.

2.5 The gas specific gravity

The specific gravity is a dimensionless unit defined as the ratio between the density (mass per unit volume) of the actual gas and the density of air at the same temperature. A list of specific gravity and other properties of various hydrocarbon gases is provided by Shashi Menon (2005). Published values of the specific gravity of natural gas range from 0.554 to 0.870.

For a source node $i \in V_s$, we assume g_i to be known, and for nodes $j \in V \setminus V_s$, we assume that

$$g_j = \frac{\sum_{i \in V_j^-} g_i x_{ij}}{\sum_{i \in V_j^-} x_{ij}} \tag{7}$$

That is, we let the specific gravity of a blend of different gases be the weighted average of specific gravities of entering flows. The flow values constitute the respective weights.

Table 2: Decision variables

$Y =$	Objective function value
$x_{ij} =$	Flow through arc $(i,j) \in A$,
$b_i =$	Net flow (Supply/demand) at node $i \in V$,
$p_i =$	Gas pressure at node $i \in V$,
$\overline{p}_{ij} =$	Average gas pressure in arc$(i,j) \in A$,
$z_{ij} =$	Compressibility of gas in arc $(i,j) \in A$
$g_i =$	Specific gravity of gas at node $i \in V \backslash V_s$

The equation of specific gravity balance is obtained by multiplying (7) by the total flow:

$$g_j \sum_{i \in V_j^-} x_{ij} - \sum_{i \in V_j^-} g_i x_{ij} = 0, \forall j \in V. \tag{8}$$

2.6 Mathematical Formulation

Summarizing the above sections, we formulate a mathematical model ($\mathcal{M}_1$) with flow, pressure, compressibility and specific gravity as decisions. For convenience, a complete list of decision variables of $\mathcal{M}_1$ is given in Table 2.

The constraints include conservation of flow (see Section 2.1), the Weymouth equation as given in Section 2.2, the compressibility formula suggested in Section 2.3.1, the equation of specific gravity balance (see Section 2.5), the quadratic form of the average pressure formula suggested in Section 2.4, as well as pressure and flow bounds. The objective function, describing a flow maximization problem, takes the form $\max \sum_{i \in V_s} b_i$.

$\mathcal{M}_1$ can then be formulated as follows:

$$(\mathcal{M}_1) \quad Y_1 = \max \textstyle\sum_{i \in V_s} b_i \tag{9}$$

s.t.:

$$\sum_{j \in V_i^+} x_{ij} - \sum_{j \in V_i^-} x_{ji} - b_i = 0, \quad \forall i \in V \tag{10}$$

$$g_j \sum_{i \in V_j^-} x_{ij} - \sum_{i \in V_j^-} g_i x_{ij} = 0, \quad \forall j \in V \backslash V_s \tag{11}$$

$$z_{ij}(1 + \omega \overline{p}_{ij} \times 10^{\beta g_i}) = 1, \quad \forall (i,j) \in A \tag{12}$$

$$3\overline{p}_{ij}(p_i + p_j) = 2(p_i^2 + p_j^2 + p_i p_j), \quad \forall (i,j) \in A \tag{13}$$

$$g_i z_{ij} x_{ij}^2 = w_{ij}\left(p_i^2 - p_j^2\right), \quad \forall (i,j) \in A \tag{14}$$

$$p_i^L \le p_i \le p_i^U, \quad \forall i \in V \tag{15}$$

$$b_i = 0, \quad \forall i \in V \backslash V_s \backslash V_d \tag{16}$$

$$b_i \ge 0, \quad \forall i \in V_s \tag{17}$$

$$b_i \le 0, \quad \forall i \in V_d \tag{18}$$

$$x_{ij} \ge 0, \quad \forall (i,j) \in A. \tag{19}$$

Since constraints (11)–(14) are non-convex, computing a global optimal solution to $\mathcal{M}_1$ is expected to be time consuming. In instances of realistic size, fast and possibly inexact solution methods may be required, and suggestions to this are discussed next.

3 A Heuristic Method

In this section, we propose a heuristic method to tackle the model $\mathcal{M}_1$ presented in Section 2.6. To diminish the difficulty introduced by the non-convex constraint (11), we propose to deal with the variability of g outside the model, and develop an iterative procedure where g is kept fixed in each iteration. This leads to a simpler mathematical formulation ($\mathcal{M}_2$) that is solved once in each iteration. By declaring the specific gravity at each node $i \in V$ as an input parameter g_i, $\mathcal{M}_2$ is put in the following form:

$$(\mathcal{M}_2) \quad Y_2 = \max \textstyle\sum_{i \in V_s} b_i \tag{20}$$

s.t.:

$$(10), (12) - (19). \tag{21}$$

By virtue of the reduced number of decision variables and the absence of (11), $\mathcal{M}_2$ is easier to solve than $\mathcal{M}_1$. In addition, constraints (12) and (14) have become simpler since g is considered to be constant.

Algorithm 1 Heuristic(ε, $g^{(0)}_{i|i\in V_s}$)

$t \leftarrow 0$
compute $g_j^{(t)}$ such that $g_j^{(t)} = \frac{1}{|V_j^-|} \sum_{i \in V_j^-} g_i^{(t)}$ for all $j \in V \backslash V_s$.
repeat
  $t \leftarrow t + 1$
  fix g to $g^{(t)}$, and solve $\mathcal{M}_2$ for $x^{(t)}, z^{(t)}, p^{(t)}$ and $\overline{p}^{(t)}$.
  for $i \in V_s$ **do**
    $g_i^{(t+1)} \leftarrow g_i^{(t)}$
  for $j \in V \setminus V_s$ **do**
    if $\sum_{i \in V_j^-} x_{ij}^{(t)} > 0$ **then**
      $g_j^{(t+1)} \leftarrow \sum_{i \in V_j^-} g_i^{(t)} x_{ij}^{(t)} \Big/ \sum_{i \in V_j^-} x_{ij}^{(t)}$
    else
      $g_j^{(t+1)} \leftarrow g_j^{(t)}$
until $\left(x^{(t)}, z^{(t)}, p^{(t)}, \overline{p}^{(t)}, g^{(t+1)}\right)$ is feasible in (12) and (14) within a tolerance ε.

Two main steps inside a loop constitute the main parts of the proposed heuristic method. Keeping g fixed to some value $g^{(t)}$ in iteration t, we first solve $\mathcal{M}_2$ to get a feasible point $\left(x^{(t)}, z^{(t)}, p^{(t)}, \overline{p}^{(t)}\right)$. Next we correct g in order to make it consistent with the feasible point found, as specified by (8). Note that at any source node i, g_i is given as input data, and remains unchanged throughout the heuristic. For a node $j \in V \setminus V_s$, the initial value $g_j^{(0)}$ is based on the specific gravities given for source nodes i on paths leading to j. More precisely, we let $g_j^{(0)} = \frac{1}{|V_j^-|} \sum_{i \in V_j^-} g_i^{(0)}$ such that each node initially has a specific gravity equal to the mean specific gravity of upstream neighbor nodes. Since the network has no directed cycles, it is straightforward to compute the vector $g^{(0)}$.

The point $\left(x^{(t)}, z^{(t)}, p^{(t)}, \overline{p}^{(t)}, g^{(t+1)}\right)$ is declared feasible if

$$\left| z_{ij}^{(t)} (1 + \omega \overline{p}_{ij}^{(t)} \times 10^{\beta g_i^{(t+1)}}) - 1 \right| \leq \varepsilon \cdot \max_{(i,j) \in A} \max\{ z_{ij}^{(t)} (1 + \omega \overline{p}_{ij}^{(t)} \times 10^{\beta g_i^{(t+1)}}), 1\}$$

and

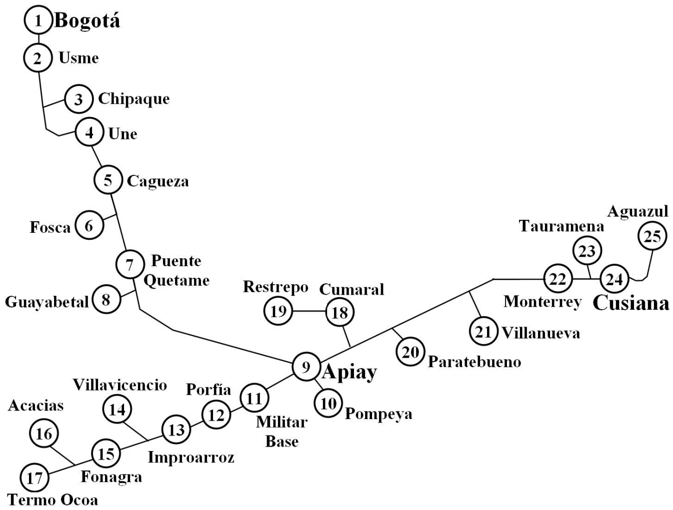


Fig. 1: Real network instance: The Bogota-Apiay-Cusiana, Colombia Transmission System

$$\left| g_i^{(t+1)} z_{ij}^{(t)} (x_{ij}^{(t)})^2 - w_{ij} \left((p_i^{(t)})^2 - (p_j^{(t)})^2 \right) \right| \leq \varepsilon \cdot \max_{(i,j)\in A} \max\{ g_i^{(t+1)} z_{ij}^{(t)} (x_{ij}^{(t)})^2, w_{ij} \left((p_i^{(t)})^2 - (p_j^{(t)})^2 \right) \}.$$

In that case, the algorithm stops. A summary of the approach is given in Algorithm 1.

4 Computational Analysis

In this section, we present results from a computational evaluation of the model and the solution method presented in earlier sections. The purpose of the experiments is twofold. First, we compare the model $\mathcal{M}_1$ to a model ($\mathcal{M}_3$) based on the traditional approach where specific gravity and compressibility are considered as global constants. Considering only the specific gravity as a global constant yields the model $\mathcal{M}_2$ used in Algorithm 1, and therefore, we also compare $\mathcal{M}_1$ to $\mathcal{M}_2$. Second, we evaluate the efficiency and effectiveness of Algorithm 1 by comparing it to the performance of commercially available solvers applied to $\mathcal{M}_1$.

4.1 Test instances

All experiments reported in this work were carried out on the set of test instances shown in Table 3, where an identifier for each test instance is given in the first column. Columns 2 and 3 show the size of the network in terms of nodes and arcs, respectively. The number of constraints, variables, and non-linear terms in $\mathcal{M}_1$ are given in columns 4–6 of Table 3, respectively, and the two last columns give respectively the smallest and largest specific gravity defined at any source. In all instances, we put $p_i^L = 400$ and $p_i^U = 1600$ for all $i \in V$.

Fig. 1 shows a real gas transmission network from Bogota, Colombia (instance AD). The other instances are generated manually, and Fig. 2 shows a typical test network (instance T), where a striped node with an incoming arrow represents a source node, a gray node with an outgoing arrow represents a demand node, and a white node is a transshipment node. To challenge the heuristic method, we also designed more complex test instances, such as instance AF shown in Fig. 3.

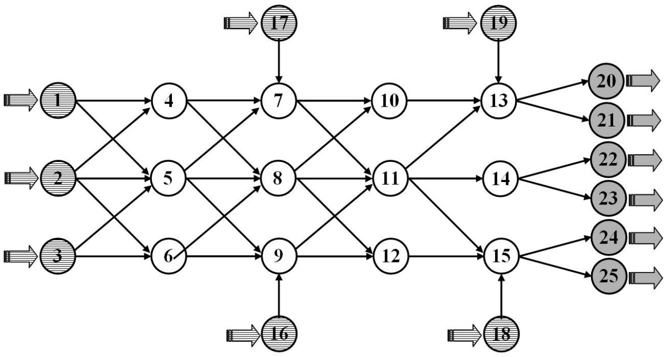


Fig. 2: Typical test network: Instance T (see Table 3)

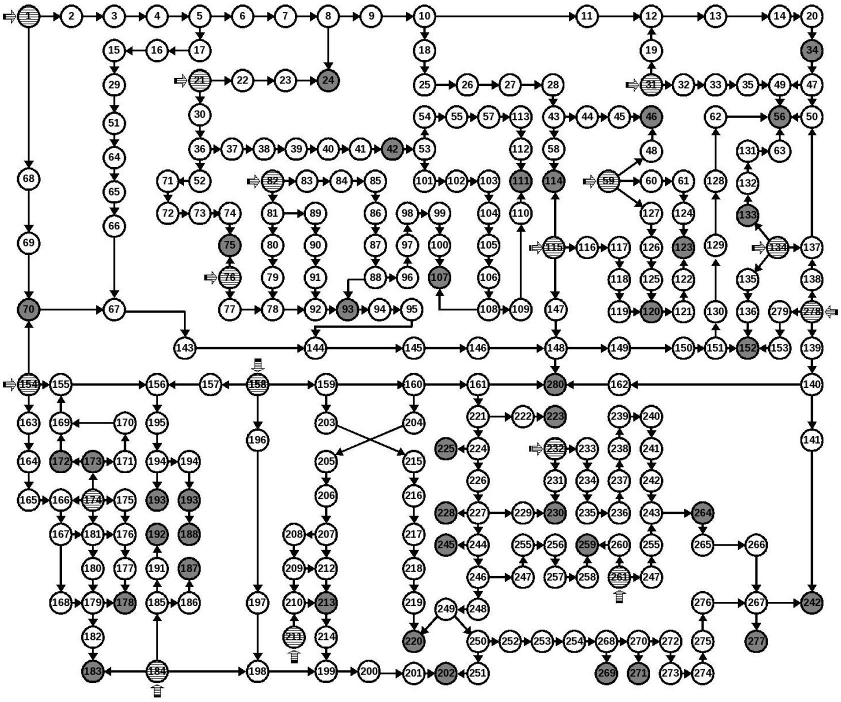


Fig. 3: A more challenging network: Instance AF (see Table 3)

4.2 Comparison to the traditional approach

In accordance with the traditional approach, model $\mathcal{M}_3$ admits specific gravity and compressibility only in terms of global constants. Letting $\bar{g}$ and $\bar{z}$, respectively, denote these input parameters, $\mathcal{M}_3$ is obtained

Table 3: Test instances

Ref	Size		Model statistics			Gravity range	
	$\|V\|$	$\|A\|$	# Consts	# Vars	# NLP terms	$g^{\min}$	$g^{\max}$
A	2	1	8	11	10	0.62	0.62
B	6	7	38	47	66	0.62	0.76
C	10	14	74	87	133	0.52	0.78
D	11	17	86	102	159	0.64	0.85
E	12	29	137	153	269	0.63	0.85
F	13	26	127	144	243	0.56	0.87
G	26	50	243	279	466	0.55	0.69
H	30	32	179	219	308	0.58	0.82
I	14	17	92	111	162	0.56	0.62
J	18	32	160	183	301	0.54	0.69
K	16	45	208	229	416	0.60	0.82
L	18	47	218	243	434	0.55	0.74
M	19	36	182	202	342	0.68	0.68
N	21	37	196	220	369	0.62	0.84
O	23	35	182	210	333	0.56	0.72
P	37	58	288	344	540	0.56	0.86
Q	23	38	192	218	353	0.62	0.68
R	24	53	254	285	494	0.56	0.82
S	25	28	156	188	270	0.55	0.69
T	25	36	182	210	333	0.55	0.87
U	28	63	302	337	588	0.55	0.82
V	26	55	264	299	512	0.55	0.82
W	29	50	236	276	441	0.57	0.89
X	30	65	310	347	599	0.62	0.84
Y	35	70	343	386	657	0.54	0.82
Z	40	75	372	421	706	0.54	0.88
AA	29	58	281	320	539	0.61	0.81
AB	37	66	314	364	591	0.58	0.87
AC	45	80	401	456	755	0.54	0.88
AD	26	25	150	179	248	0.56	0.63
AE	45	260	1097	1152	2321	0.60	0.60
AF	280	309	1767	2065	3016	0.60	0.60

by simplifying $\mathcal{M}_1$ as follows:

$$(\mathcal{M}_3) \quad Y_3 = \max \textstyle\sum_{i \in V_s} b_i \tag{22}$$

s.t.:

$$\bar{g}\bar{z}x_{ij}^2 = w_{ij}\left(p_i^2 - p_j^2\right), \quad \forall (i,j) \in A \tag{23}$$

$$(10), (15) - (19). \tag{24}$$

To adapt the input data of $\mathcal{M}_1$ to $\mathcal{M}_3$, we let

$$\bar{g} = \frac{g^{\min} + g^{\max}}{2}$$

and

$$\bar{z} = \frac{1}{1 + \omega \frac{p^{\min} + p^{\max}}{2} 10^{\beta \bar{g}}},$$

where $g^{\min} = \min_{i \in V_s} g_i$, $g^{\max} = \max_{i \in V_s} g_i$, $p^{\min} = \min_{i \in V} p_i^L$, $p^{\max} = \max_{i \in V} p_i^U$. In $\mathcal{M}_2$, we replace all occurrences of g_i by $\bar{g}$, but no replacement is made for z_{ij}.

All three models were formulated in the GAMS modeling language (GAMS, 2008) and submitted to version 8.1.5 of the global optimizer BARON (Tawarmalani and Sahinidis, 2004). BARON is an implementation of a branch-and-bound algorithm where a convex relaxation of the submitted problem is solved in each node of the search tree. We made BARON call version 5.51 of MINOS (Murtaugh and Saunders, 1983) to solve the relaxations.

We imposed a time limit of 3600 CPU-seconds to each application of BARON, where two different relative optimality tolerances ($\varepsilon \in \{10^{-1}, 10^{-3}\}$) were applied. That is, BARON stopped once it is proved that a feasible solution with objective function value above $(1-\varepsilon)$ times the maximum flow is found. In instances where BARON fails to compute the global optimum within the time limit, it may still provide an upper bound on the maximum flow to give some indications on the quality of the output.

All experiments reported in this work were run on a 2.4 GHz Intel(R) processor with 2 GByte RAM under Linux Red Hat operating system.

The results from these experiments can be found in Table 4. Column 1 shows the instance id, and the next four columns give the CPU-time in seconds and the best objective function value obtained for each value of ε when running $\mathcal{M}_1$. The same information for $\mathcal{M}_2$ and $\mathcal{M}_3$ is provided in columns 6–9 and columns 10–13, respectively. In instances where BARON did not converge within the time limit, the upper bound on the solution is provided in parenthesis.

Serving as a measure of the deviations of the $\mathcal{M}_2$-solutions from the $\mathcal{M}_1$-solutions, column 2 of Table 5 contains the observed values of $\frac{Y_2-Y_1}{Y_1}100\%$ when both $\mathcal{M}_1$ and $\mathcal{M}_2$ were submitted to BARON with a relative optimality tolerance of $\varepsilon = 10^{-1}$. Column 3 holds corresponding results for $\varepsilon = 10^{-3}$, and columns 4–5 give the deviations observed for $\mathcal{M}_3$. The percentages in the last two columns indicate to what extent the traditional approach ($\mathcal{M}_3$) results in wrong estimates of the total flow capacity of a network, and columns 2–3 provide similar information for an approach ($\mathcal{M}_2$) where only the compressibility is allowed to vary with the network components.

Model $\mathcal{M}_1$ appears to be hard to solve to optimality. Given the largest optimality tolerance and a time limit of 1 CPU-hour, BARON failed to do so in 9 (R, U, V, X, Y, Z, AC, AE and AF) out of 32 instances. With the tolerance reduced to 10^{-3}, it failed in 6 instance more (K, L, S, T, AA and AB). As expected, models $\mathcal{M}_2$ and $\mathcal{M}_3$ appear to be less difficult to handle by BARON. It failed to provide optimality in only 2 (AE and AF) and 1 (AE) out of 32 test cases, respectively, with $\epsilon = 10^{-1}$. With ϵ reduced to 10^{-3}, it failed in 1 instance more in both models (AB and AE, respectively).

We however observe that the traditional approach in several instances results in considerable deviations from the max flow. In cases where BARON could not close the optimality gap, large deviations may be implied by weak lower bounds on the optimal objective function value. However, even when BARON found the solutions within the given tolerances, the deviation is below -14% for $\mathcal{M}_3$ in 6 instances (B, G, J, N, W and AE) and above 7% in one instance (V). Also $\mathcal{M}_2$ results in deviations as large as 5.3% in one instance (AF), up to 21% in instance V, and below -4% in instance J.

4.3 Evaluation of the heuristic approach

The aim of the second set of experiments is to compare the output from Algorithm 1 to the true optimal solution to $\mathcal{M}_1$, and to compare the computational effort of the algorithm to the effort of computing Y_1 exactly.

Recall that Algorithm 1 requires access to solutions to instances of $\mathcal{M}_2$. This was accomplished by letting the algorithm call a local optimizer (MINOS) once in each iteration. The relative stopping tolerance, ϵ, was set to 10^{-3} in all runs.

Multistart local optimization is a straightforward heuristic approach, which by virtue of its simplicity deserves to be compared to Algorithm 1. We implemented a procedure that generates randomly a set of, not necessarily feasible, solutions to $\mathcal{M}_1$. Each solution is passed to MINOS, which is asked to perform

Table 4: Results from BARON applied to $\mathcal{M}_1$, $\mathcal{M}_2$, and $\mathcal{M}_3$

	$\mathcal{M}_1$				$\mathcal{M}_2$				$\mathcal{M}_3$			
	$\varepsilon = 10^{-1}$		$\varepsilon = 10^{-3}$		$\varepsilon = 10^{-1}$		$\varepsilon = 10^{-3}$		$\varepsilon = 10^{-1}$		$\varepsilon = 10^{-3}$	
Ref	CPU	Y_1	CPU	Y_1	CPU	Y_2	CPU	Y_2	CPU	Y_3	CPU	Y_3
A	0	0.36	0	0.36	0	0.36	0	0.36	0	0.33	0	0.33
B	0	0.41	1	0.41	0	0.40	0	0.40	0	0.35	0	0.35
C	0	1.42	4	1.42	0	1.42	1	1.42	0	1.27	0	1.27
D	0	3.92	5	3.92	0	3.85	2	3.85	0	3.47	0	3.47
E	3	5.17	32	5.17	0	5.13	9	5.13	0	4.48	0	4.48
F	0	7.84	3	7.86	0	7.66	1	7.66	0	6.78	0	6.78
G	18	2.76	105	2.76	1	2.69	12	2.69	0	2.34	0	2.37
H	240	1.07	1624	1.08	6	1.07	45	1.07	0	0.95	0	0.95
I	4	0.81	40	0.81	0	0.82	3	0.82	0	0.73	0	0.73
J	4	1.92	1204	1.92	0	1.83	61	1.83	0	1.61	9	1.61
K	137	2.70	(2.96)	2.70	16	2.70	71	2.70	0	2.39	0	2.39
L	275	3.61	(4.44)	3.61	5	3.64	69	3.64	0	3.31	0	3.31
M	0	1.17	6	1.17	0	1.15	5	1.17	0	1.07	0	1.07
N	11	1.18	44	1.18	0	1.16	8	1.16	0	1.01	0	1.01
O	19	3.10	590	3.11	1	3.12	17	3.12	0	2.75	0	2.75
P	12	5.34	260	5.34	1	5.21	42	5.21	0	4.61	0	4.61
Q	25	1.92	282	1.92	9	1.93	117	1.93	1	1.74	1	1.74
R	(0.94)	0.74	(0.94)	0.74	4	0.74	131	0.74	0	0.66	0	0.66
S	9	1.73	(1.77)	1.74	1	1.70	13	1.70	0	1.49	0	1.49
T	3574	1.18	(1.58)	1.18	6	1.19	82	1.19	0	1.05	0	1.05
U	(1.42)	0.96	(1.42)	0.96	28	0.97	1184	0.97	0	0.87	0	0.87
V	(1.69)	1.05	(1.69)	1.05	4	1.27	132	1.27	73	1.13	122	1.13
W	19	7.17	162	7.17	0	7.01	7	7.01	0	6.15	0	6.15
X	(2.48)	2.04	(2.48)	2.04	52	2.11	658	2.11	0	1.85	84	1.85
Y	(1.39)	1.14	(1.39)	1.14	8	1.14	178	1.14	1	1.02	1	1.02
Z	(1.49)	1.18	(1.49)	1.18	10	1.16	224	1.16	1	1.03	1	1.03
AA	108	1.54	(1.79)	1.54	5	1.52	168	1.52	1	1.34	1	1.34
AB	2156	1.84	(2.32)	1.84	38	1.83	(2.02)	1.83	1	1.60	23	1.60
AC	(1.52)	1.21	(1.52)	1.21	11	1.20	275	1.20	1	1.06	1	1.06
AD	1	8.18	13	8.36	1	8.39	4	8.39	0	7.76	0	7.76
AE	(13.91)	10.30	(13.91)	10.30	(13.82)	10.50	(13.82)	10.50	(10.19)	7.05	(10.19)	7.05
AF	(4.77)	3.95	(4.77)	3.95	(4.63)	4.16	(4.63)	4.16	615	3.80	(3.88)	3.80

local optimization starting from the submitted initial point. The procedure is assigned an upper bound on the CPU-time, and the number of initial points generated is determined as the maximum that can be afforded given that the multistart procedure terminates within this bound.

All instances in Table 3 were submitted to Algorithm 1 and to the multistart local optimization procedure, where the latter was given a time bound of 20 CPU-seconds. This limit was chosen since Algorithm 1 in no instance needed more time to converge. Fairness to its competitor in terms of time allocation is hence assured.

Results from Algorithm 1 and multistart local search are reported in Table 6. Columns 2–5 concern Algorithm 1. Results given are respectively the CPU-time (seconds), the number of iterations, the best solution found, and the relative deviation from the best known solution. More precisely, column 5 shows $\frac{Y_1^{\text{alg1}} - Y_1^{\text{BARON}}}{Y_1^{\text{BARON}}} 100\%$, where Y_1^{alg1} and Y_1^{BARON} denote the best solutions produced by Algorithm 1 and BARON (see Table 4), respectively. The results for multistart local search (columns 6–9) are respectively the number of local searches accomplished within the time limit, after how many searches the best solution was found, the largest objective function value observed, and the relative deviation from the result of BARON. The deviation is defined analogously to the deviation for Algorithm 1.

Table 5: Deviations (%) from $\mathcal{M}_1$-results

	Y_2		Y_3	
Ref	$\varepsilon = 10^{-1}$	$\varepsilon = 10^{-3}$	$\varepsilon = 10^{-1}$	$\varepsilon = 10^{-3}$
A	0.0	0.0	-8.3	-8.3
B	-2.4	-2.4	-14.6	-14.6
C	0.0	0.0	-10.6	-10.6
D	-1.8	-1.8	-11.5	-11.5
E	-0.8	-0.8	-13.3	-13.3
F	-2.3	-2.5	-13.5	-13.7
G	-2.5	-2.5	-15.2	-14.1
H	0.0	-0.9	-11.2	-12.0
I	1.2	1.2	-9.9	-9.9
J	-4.7	-4.7	-16.1	-16.1
K	0.0	0.0	-11.5	-11.5
L	0.8	0.8	-8.3	-8.3
M	-1.7	0.0	-8.5	-8.5
N	-1.7	-1.7	-14.4	-14.4
O	0.6	0.3	-11.3	-11.6
P	-2.4	-2.4	-13.7	-13.7
Q	0.5	0.5	-9.4	-9.4
R	0.0	0.0	-10.8	-10.8
S	-1.7	-2.3	-13.9	-14.4
T	0.8	0.8	-11.0	-11.0
U	1.0	1.0	-9.4	-9.4
V	21.0	21.0	7.6	7.6
W	-2.2	-2.2	-14.2	-14.2
X	3.4	3.4	-9.3	-9.3
Y	0.0	0.0	-10.5	-10.5
Z	-1.7	-1.7	-12.7	-12.7
AA	-1.3	-1.3	-13.0	-13.0
AB	-0.5	-0.5	-13.0	-13.0
AC	-0.8	-0.8	-12.4	-12.4
AD	2.6	0.4	-5.1	-7.2
AE	1.9	1.9	-31.6	-31.6
AF	5.3	5.3	-3.8	-3.8

We observe that Algorithm 1 in all instances but AB, AE and AF produces solutions within 5% deviation from the best known solution. A comparison with multistart local search shows that Algorithm 1 looses with a difference less than 1% in 5 instances (F, K, O, P and S), with a difference no larger than 4% in 5 other instances (H, J, Z, AA and AC), and up to 8.2% in instance AB. In 5 instances (G, R, V, W and X), the relative difference between the two heuristics are observed up to 75.5% (instance G) in favor of Algorithm 1. In addition, multistart local search was unable to produce anything better than the zero solution in two instances (AE and AF), whereas for Algorithm 1 this occurred only in instance AE.

We conclude that one promising approach to solving model $\mathcal{M}_1$, is a sequential procedure where some state variables are kept fixed in each iteration. A local optimum to a simplified version of the model (model $\mathcal{M}_2$ in the case of Algorithm 1) can be found quickly, and new and better estimates of the state variables are computed. The likelihood of converging to a solution far from optimum seems smaller than by multistart local search. Besides, the computer time needed for doing so is not close to the time needed to prove optimality.

Table 6: Performance of heuristic methods

	Algorithm 1				Multistart local optimization			
Ref	CPU	Its	Y_1	Dev	Runs	Best	Y_1	Dev
A	0	1	0.36	0.0	385	86	0.36	0.0
B	0	2	0.41	0.0	334	79	0.41	0.0
C	0	4	1.36	-4.2	254	28	1.42	0.0
D	0	3	3.92	0.0	196	182	3.92	0.0
E	0	5	5.17	0.0	162	67	5.17	0.0
F	0	2	7.85	-0.1	164	71	7.86	0.0
G	1	7	2.74	-0.7	89	1	0.67	-75.7
H	1	9	1.05	-2.8	60	49	1.08	0.0
I	0	2	0.81	0.0	149	6	0.81	0.0
J	0	3	1.87	-2.6	99	15	1.92	0.0
K	1	3	2.70	0.0	120	85	2.69	-0.4
L	1	3	3.60	-0.3	105	64	3.60	-0.3
M	0	1	1.17	0.0	257	213	1.17	0.0
N	0	2	1.18	0.0	206	189	1.18	0.0
O	0	2	3.09	-0.6	129	24	3.11	0.0
P	0	3	5.31	-0.6	94	58	5.34	0.0
Q	1	4	1.92	0.0	182	68	1.92	0.0
R	1	5	0.74	0.0	25	16	0.73	-1.4
S	0	4	1.73	-0.6	95	32	1.74	0.0
T	1	5	1.18	0.0	126	82	1.18	0.0
U	2	4	0.96	0.0	77	2	0.96	0.0
V	2	5	1.03	-1.9	24	4	1.27	21.0
W	1	8	7.17	0.0	184	1	3.86	-46.2
X	1	5	2.09	2.5	85	74	2.12	3.9
Y	2	5	1.14	0.0	37	24	1.14	0.0
Z	1	8	1.14	-3.4	37	14	1.18	0.0
AA	2	5	1.48	-3.9	23	3	1.54	0.0
AB	2	8	1.69	-8.2	36	9	1.84	0.0
AC	2	5	1.17	-3.3	26	11	1.21	0.0
AD	2	2	8.36	0.0	55	1	8.36	0.0
AE	11	25	0.00	-100.0	40	40	0.00	-100.0
AF	18	1	0.29	-92.7	12	1	0.00	-100.0

5 Concluding remarks

In this paper, we have developed a mathematical model for maximizing the flow of natural gas in pipeline transmission networks. Unlike previously suggested models, our models admits variations in gas specific gravity and compressibility. All arcs in the network model correspond to pipelines, and the main decision variables are pipeline flow and gas pressure at each network node. To support these decisions, the model also has to assess the gas specific gravity at each node, and the gas compressibility at each pipeline. In addition to flow conservation constraints, the model contains constraints relating pipeline flow to pressure, as well as constraints defining the gas properties as functions of the flow.

The resulting model has a non-convex feasible domain, and represents a considerable computational challenge for global optimization algorithms. Simpler, but still non-convex models, where the variation in gas specific properties is neglected, can already be found in the literature. Through experiments, we have demonstrated that applying these to instances where the variation is high, tends to give misleading results.

The simpler models can however be useful building blocks in inexact, but fast, heuristic methods for our model. We have developed an iterative procedure, which in each iteration keeps the specific gravity constant while optimizing other decisions. As a preparation for the next round, the specific gravity is updated consistently with the flow values observed, and the procedure is repeated until convergence. Computational experiments demonstrate that this procedure yields optimal or near-optimal solutions in most of the instances.

Acknowledgements This work was financed by the Norwegian Research Council, Gassco and Statoil under contract 175967/S30.

References

Abbaspour M, Chapman KS (2008) Nonisothermal transient flow in natural gas pipeline. J Appl Mech Trans ASME, vol 75 (3), pp 181-188, doi:10.1115/1.2840046

Bakhouya B, De Wolf D (2007) The gas transmission problem when the merchant and the transport functions are disconnected. HEC Ecole de Gestion de l'Université de Liége (ULG), Working Paper 2007 01/01

Borraz-Sánchez C, Haugland D (2009) A tree decomposition algorithm for minimizing fuel cost in gas transmission networks . In: Proceedings of the 39th International Conference: Computers & Industrial Engineering (cie2009), pp 244-249, ISBN: 978-1-4244-4135-8

Borraz-Sánchez C, Ríos-Mercado RZ (2004) A non-sequential dynamic programming approach for natural gas network optimization. WSEAS Transactions on Systems, vol 3 (4), pp 1384-1389

Borraz-Sánchez C, Ríos-Mercado RZ (2009) Improving the operation of pipeline systems on cyclic structures by tabu search. Computers & Chemical Engineering, vol 33 (1), pp 58-64

Carter RG (1998), Pipeline optimization: Dynamic programming after 30 years, In: Proceedings of the 30th PSIG Annual Meeting, Denver, CO, USA.

Chaczykowski M, (2010) Transient flow in natural gas pipeline - The effect of pipeline thermal model. Appl Math Modelling, vol (34), pp 1051-1067

Crane Co (1982) Flow of fluids through valves, fittings and pipe. Technical Paper 410. Metric Edition, USA

Davisson, E.G. (1965) A computer program for flow calculations. Tech Report ORNL-TM-1093, OAK Ridge National Laboratory, U.S. Atomic Energy Commision, USA.

De Wolf D, Smeers Y (1994) Mathematical properties of formulations of the gas transmission problem. SMG Preprint 94/12. Tech Report, Université Libre De Bruxelles

De Wolf D, Smeers Y (2000) The gas transmission problem solved by an extension of the simplex algorithm. Management Sci, vol 46 (11), pp 1454-1465

Dranchuk PM, Abou-Kassem JH (1975) Calculations of Z factors for natural gases using equations of state. J Can Pet Tech, vol 14 (3), pp 34-36

Dranchuk PM, Purvis RA, Robinson DB (1974) Computer calculations of natural gas compressibility factors using the Standing and Katz correlations. Inst of Pet Tech Series, No. IP-74-008, pp 1-13

GAMS Development Corporation (2008) GAMS: The solver manuals. Washington, DC, USA

Gopal VN (1977) Gas Z-factor equations developed for computers. Oil & Gas J, Aug 8, pp 58-60

Kalvelagen E (2003) An improved formulation for a gas transportation problem. Scientific Literature Digital Library and Search Engine, Doi:10.1.1.13.4509

Katz DL, Cornell D, Kobayashi R, Poettmann FH, Vary JA, Elenbaas JR, Weinaug CF (1959) Properties of natural gases and volatile hydrocarbon liquids, Chap 4 in Handbook of Natural Gas Engineering, McGraw-Hill Book Co, New York, NY, USA, pp 94-188

Modisette JL (2000) Equation of State Tutorial, In: Proceedings of the 32nd Annual Meeting of Pipeline Simulation Interest Group (PSIG), Savannah, GA, USA

Murtaugh BA, Saunders MA (1983) MINOS 5.1 User's guide. Tech Report SOL-83-20R, Systems Optimization Laboratory, Stanford University, Stanford, CA, USA

Osiadacz AJ (1987) Simulation and analysis of gas networks. Gulf Publishing Company, Houston, USA

Ríos-Mercado RZ, Wu S, Scott LR, Boyd, EA (2003) A Reduction Technique for Natural Gas Transmission Network Optimization Problems. Annals of Operations Research, vol 117, pp 217234, 2002.

Shashi Menon E (2005) Gas pipeline hydraulics. CRC Press, Taylor & Francis Group, LLc. Boca Raton, FL

Tawarmalani M, Sahinidis NV (2004) Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. Math Programming, vol 99 (3), pp 563-591

Wong PJ, Larson RE (1968) Optimization of natural gas pipeline systems via dynamic programming. IEEE Transactions on Automatic Control, vol 13, No 5, pp 475-481.

PAPER V

Modeling line-pack management in natural gas transportation pipeline systems*

Conrado Borraz-Sánchez Dag Haugland

* Preprint submitted for potential inclusion in the volume of *the INFORMS Computing Society 2011 Conference.* (2010)

ICS 2011
12th INFORMS Computing Society Conference
Monterey, California, January 9–11, 2011
pp. 000–000

 | ISBN 000-00000-00000
DOI 10.1287/ics.2011.XXXX

Modeling Line-Pack Management in Natural Gas Transportation Pipeline Systems

Conrado Borraz-Sánchez and Dag Haugland
Department of Informatics, University of Bergen,
{Conrado.Borraz-Sanchez, dag.haugland}@ii.uib.no

Abstract The gas industry, in order to meet clients' demand, always strives to levelling the gas sending rates as much as possible. However, unpredictable or scheduled events that may occur in the network, including break down of flow capacities elsewhere in the system, shortfall in downstream capacity and demand uncertainty, play a critical role while establishing an optimal plan during a given period. As strategy to diminish the effects caused by such events, line-packing methods are applied to gas transmission pipeline systems in order to increase the safety stock levels, i.e., customer satisfaction. In this paper, the problem of determining an optimum line-pack to satisfy clients' requirements for a given multi-period horizon is addressed. In order to satisfy market requirements, the proposed MINLP model keeps track of energy content and quality. An extensive computational experimentation based on a global optimizer by means of a formulation on a general algebraic modelling system (GAMS) is presented.

Keywords Natural gas; line-packing; transmission network; safety stock; MINLP.

1. Introduction

Reliable and economic pipeline systems for transporting natural gas are essential to the gas industry. Undoubtedly, they play a significant role in preserving the continuous business growth around the world. Nevertheless, a common denominator in the transportation process is that a number of unpredictable or scheduled events do occur on a daily basis. Among these events we can find, e.g., the break down of flow capacities elsewhere in the system due to malfunctions, routine maintenance or inspection; failures in upstream process capacity; shortfall in downstream capacity; demand uncertainty; and high fluctuation in demand due to seasons (in the winter the demand is usually higher than in the summer). Yet, gas producers must be able to supply gas to their customers despite such difficulties.

The aim of this paper is to propose a strategy to some extent alleviate the consecuences of these events by taking into account one key fact: Gas pipelines do not only serve as transportation links between producer and consumer, but they also represent potential storage units for safety stocks. That is, due to the compressible nature of dry gas, large reserves can be stored on a short-term basis inside the pipeline for subsequent extraction when flow capacities elsewhere in the system break down. Hence, keeping a sufficient level of line-pack during a given planning horizon becomes critical to the transporter.

1.1. The line-packing problem

According to the *Council of European Energy Regulators* (CEER), line-pack refers to the "storage of gas by compression in gas transmission and distribution systems, but excluding facilities reserved for transmission system operators carrying out their functions (Article

2(15))" [2]. In this definition, we clearly observe which facilities or portion of storage and line-pack are excluded from Third Party Access (TPA) for transmission operations.

In [2], CEER also claims that the access to storage and line-pack services play a crucial role in the development of a competitive European gas market since they provide flexibility services, and conform one of the prerequisites for entering and operating in the gas market.

From a technical point of view, line-pack is an operating pipeline where a downstream valve is closed (or throttled) while upstream compressors continue sending gas into the pipeline for future use.

From an industrial perspective, line-pack is *the ability (quality) of a natural gas pipeline* to effectively "store" small quantities of gas on a short-term basis by increasing the operating pressure of the pipeline. The aim is to use it as a resource to handle load fluctuations in a pipeline system. The strategy is to build up line-pack during periods of decreased demand and drawing it down during periods of increased demand.

A simple example that may conceptualize this problem can be described as follows. Let us suppose that there is a unique transmission line between one producer and one costumer, and let us assume that only 60% of maximum capacity is required for several periods due to client demands. Here, the gas producer could simply send the required amount of gas during the mentioned periods. However, let us suppose that for some subsequent periods the demand increases up to 120% of maximum capacity, then the producer would not be able to satisfy the demand, thus leading to considerable economic losses. Hence, the strategic idea would be to send for instance 80% of maximum capacity, then consuming just the required demand in each period, and storing the remaining gas in order to satisfy future extraordinary requirements.

Summarizing all the above, the line-packing problem in a gas transportation network practically means optimizing the refill of gas in pipelines in periods of sufficient capacity, and optimizing the withdrawals in periods of shortfall. Some attempts, although few, have been made in the direction of mathematical planning models for this problem (see [1] and [3]).

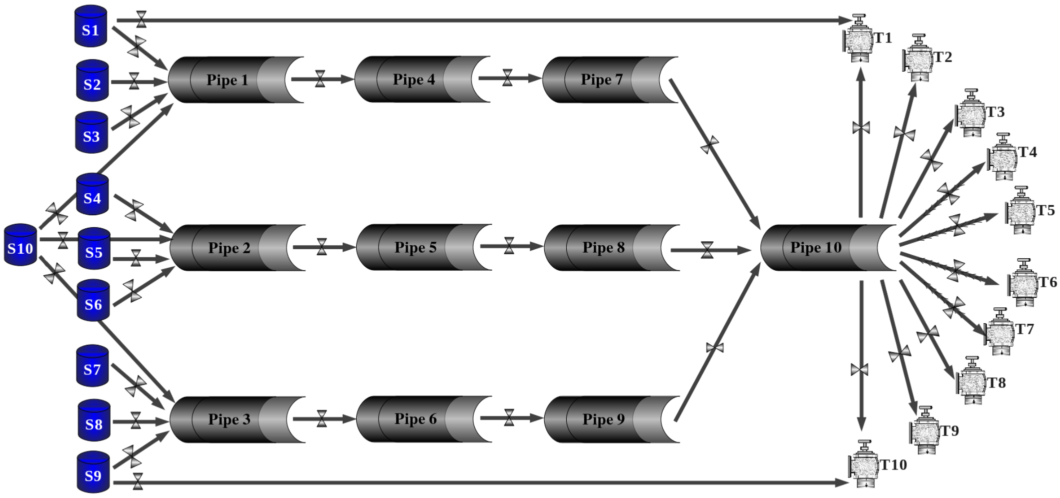
In this paper, we propose a *mixed-integer non-linear programming* (MINLP) model to tackle the line-packing problem by building up heterogeneous batches in gas transmission pipeline networks for a multiple-time period planning horizon. The model also includes the ability to keep track of energy content and quality at the nodes of the network to ensure that contract terms are met. An extensive computational experimentation is carried out by means of a formulation on a general algebraic modeling system (GAMS). The results achieved on a wide range of network topologies show that the application of a global optimizer tool, BARON [6], is effective and efficient when solving large networks for a planning horizon with a moderate number of time periods, whereas it turns out to be time consuming for a large number of time periods.

The remainder of this paper is organized as follows. In the next section, we introduced a MINLP mathematical formulation to tackle the line-packing problem in gas transmission networks. Section 3 presents the computational results of the application of a global optimizer when solving the MINLP model on a wide range of test instances. Finally, concluding remarks are given in Section 4.

2. Mathematical Formulation

The design of the mathematical model presented in this section is based on two fundamental steps. First, we build up (heterogeneous) batches of natural gas in the pipelines of the network system during a given multi-period horizon. Second, we consume the batches in a logical and schematical way such that customer contracts are met. In both steps of the design, specific quality requirements of the natural gas mixture that is supplied have to be considered as well. As a result, we present a single-objective, multi-source, multi-demand,

FIGURE 1. Network topology: *Serie 10* (see Table 2).



multi-quality, and multi-period problem that is formulated as a MINLP model. The objective function of this model is to maximize the flow through the pipeline system as a strategic idea to meet market demand during the planning horizon. All constraints related to this model are studied in detail in subsequent sections. First, a notion of what we refer to as heterogeneous batches is provided next.

2.1. Heterogeneous batches

Since the dispatcher strives to satisfy future deliveries (consignments) specified by costumer contracts, it may be required building up the line-packs (gas batches) over several time periods. This is accomplished by blending various flow streams coming from any available gas source in the system before entering the pipeline. Assuming all sources in the transportation system to have the same gas properties would lead to consider homogeneous batches, and thus the complexity of the problem would considerably be reduced. However, it is unlikely that such an assumption reflects the reality in practice, and therefore gas sources having different properties must be considered. As a consequence, the pipeline may end up having batches of different composition. Here, the notion of *heterogeneous batches*.

On the other hand, according to [3], a blending process between the batches inside the pipeline seems to be unrealistic unless a long lasting shortfall in downstream capacity takes place. Hence, considering no blending process among the batches inside the pipelines, a common practice in the gas industry, is an essential assumption that follows this research.

2.2. Notation

Let $G=(N,A)$ be an acyclic directed network representing a gas transmission pipeline system (see Fig. 1), where $N=S\cup L\cup T$ is the set of nodes partitioned into three classes: sources (S), pipelines (L) and terminals (T). The arc set A represents the set of links joining some pairs of nodes, where each link is assumed to have a valve.

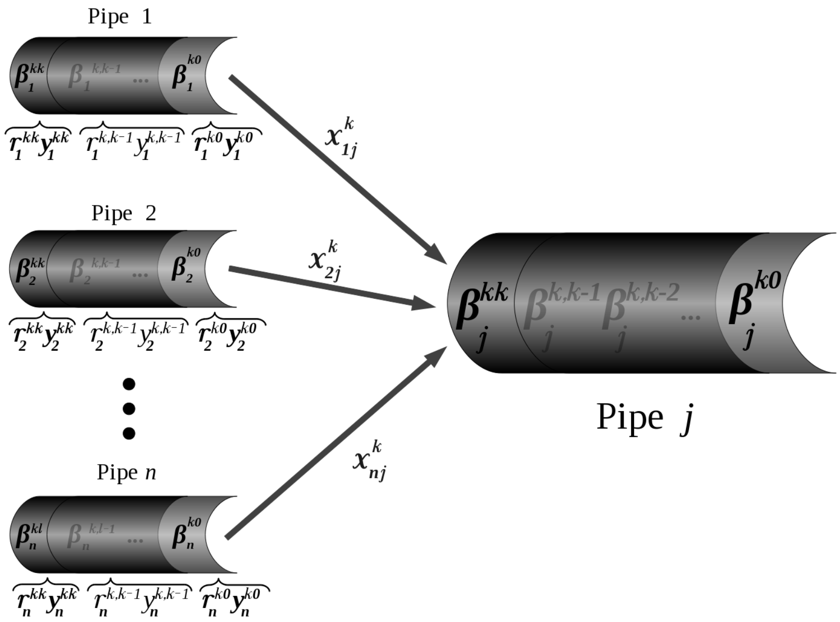
Let $K=\{1,\cdots,\kappa\}$ be the set of periods of length δ representing the planning horizon, where the superscript $k\in K$ denotes the current period.

For each source node $i\in S$, the total supply in period k is given by b_i^k.

For each sink node $i\in T$, the total gas demand in period k is given by d_i^k.

For each pipeline node $i\in L$, the minimum and maximum line-pack levels are given by Δ_i^{min} and Δ_i^{max}, respectively, with $0<\Delta_i^{min}\le\Delta_i^{max}$. Furthermore, the initial inventory is determined by β_i^{00}, the required final inventory by Δ_i^{final}, and the reduction and increase

FIGURE 2. Interaction between incoming flows in pipeline $j \in L$ and the new batch in period $k \in K$.



factors for the current line-pack are given by w_i and f_i, respectively. Basically, w_i and f_i indicate the minimum and maximum amount of gas permissible for extraction and filling in the pipeline under steady-state flow conditions, i.e., from a thermodynamic modelling perspective, they restrict the total amount of gas that can still be stored or extracted in any period of the planning horizon.

Let $V_i^+ = \{j \in N \mid (j,i) \in A\}$ be the set of start nodes of incoming arcs to node i. Similarly, let $V_i^- = \{j \in N \mid (i,j) \in A\}$ be the set of end nodes of outgoing arcs from node i.

Let $\beta_i^{kl} \in \mathbb{R}_+$ be a variable representing the size of batch l in pipeline $i \in L$ in period $k \in K$.

Let $y_i^{kl} \in \{0,1\}$ be a binary variable defined as follows:

$$y_i^{kl} = \begin{cases} 1 & \text{if batch } l \text{ in pipeline } i \in L \text{ is extracted} \\ & \text{(fully or partly) in period } k \in K \\ 0 & \text{otherwise} \end{cases}$$

Let $r_i^{kl} \leq 1$ be a variable defining the ratio of gas extracted in the current period from the batch that in period l entered pipeline i.

Let $x_{ij}^k \in \mathbb{R}_+$ be a continuous decision variable representing the total flow through link $(i,j) \in A$ in period $k \in K$.

2.3. Building up heterogeneous batches in the pipelines

The line-pack of the gas system is based on building up heterogeneous batches in every pipeline. We first bound the maximum available space to build up the batch in period $k \in K$ by considering the physical limitation of the pipeline, which is based on a given line-pack increase factor, f_i, and the current remaining capacity for the new batch. The maximum size of the new batch is then restricted by:

$$\sum_{j \in V_i^+} x_{ji}^k - \sum_{j \in V_i^-} x_{ij}^k \leq f_i \cdot \left(\Delta_i^{max} - \sum_{l=0}^{k} \beta_i^{kl} \right), \forall i \in L, k \in K \tag{1}$$

Eq. (1) basically limits the building up of a batch in pipeline $i \in L$ in period k based on the maximum available space determined by the difference between the maximum line-pack level (Δ_i^{max}) and the current total amount of gas in the pipeline. The model will then build up in pipeline j in period k a new batch as follows:

$$\beta_j^{kk} = \sum_{i \in V_j^+} x_{ij}^k, \forall j \in L, k \in K \tag{2}$$

As shown by Eq. (2), the batch β_j^{kk} to be introduced in pipeline j corresponds to the sum of all incoming flow streams in the pipeline in the current period.

Fig. 2 shows the interaction between the new batch and the corresponding flow streams of which it is composed. Note also that each gas stream (x_{ij}^k) entering the pipeline j is in turn determined by the total amount of gas extracted from all batches previously stored up to period k in pipeline i. This is explained in detail in subsequent sections.

Since $r_i^{kl} \in [0,1]$ is the ratio of gas extracted from batch l in pipeline $i \in L$ in period $k \in K$, the following constraint is required:

$$r_i^{kl} \leq y_i^{kl}, \forall i \in L, 0 \leq l \leq k \leq \kappa, \tag{3}$$

stating that it can be extracted gas from the batch that in period l entered pipeline i, only if its corresponding variable $y_i^{kl} = 1$, i.e., the batch is enabled to supply gas in the current period.

Concerning the limitations of the line-pack in pipeline $i \in L$, we finally impose that

$$\Delta_i^{min} \leq \sum_{l=0}^{k} \beta_i^{kl} \leq \Delta_i^{max}, \forall i \in L, k \in K. \tag{4}$$

Eq. (4) specifies the physical limitations imposed by the transporter on the total line-pack allowed in the pipeline. That is, the total amount of gas stored in the batches of the pipeline ($\sum_{l=0}^{k} \beta_i^{kl}$) for any period must always be keep in between specific given values.

2.4. Consumption of batches

Once the batches have entered the pipeline, they can be extracted in order to satisfy market demand. The consumption of batches is however subject to physical and logical restrictions, and the corresponding model constraints are studied next.

We first bound the maximum amount of gas that can be extracted from each pipeline in any period. This is based on the current line-pack and a given reduction factor w_i of the pipeline. The constraint can then be expressed as follows:

$$\sum_{j \in V_i^-} x_{ij}^k - \sum_{j \in V_i^+} x_{ji}^k \leq w_i \cdot \left(\sum_{l=0}^{k} \beta_i^{kl} - \Delta_i^{min} \right), \forall i \in L, k \in K. \tag{5}$$

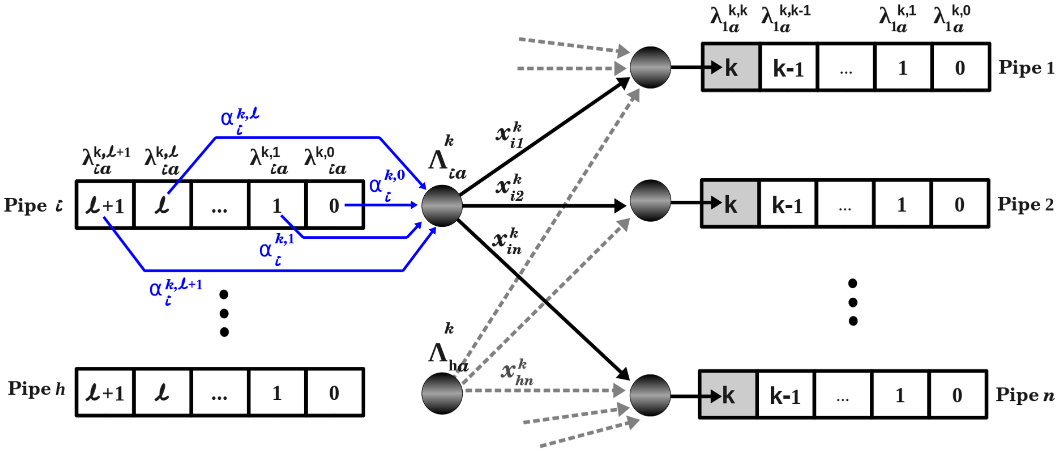
Eq. (5) expresses the relation between the net flow leaving the pipeline and the total amount of gas currently stored in the batches of the pipe.

Let α_i^{kl} be a variable representing the amount of gas extracted from batch l in pipeline $i \in L$ in period $k \in K$, i.e.,

$$\alpha_i^{kl} = \beta_i^{ll} \cdot r_i^{kl}, \quad \forall i \in L, 0 \leq l \leq k \leq \kappa. \tag{6}$$

As specified by the variable r_i^{kl}, the model is allowed to consume the batches as a whole entity or in fractions in each period. Hence, the model is forced to update the contents of the batches in pipeline $i \in L$ from one time period to the next by imposing the next constraint:

FIGURE 3. Interaction of the gas quality between the batches stored in a pipeline and the total flow stream leaving it in the current period.



$$\beta_i^{k+1,l} = \beta_i^{kl} - \alpha_i^{kl}, \forall i \in L, 0 \leq l \leq k \leq \kappa. \tag{7}$$

Eq. (7) specifies the amount of gas remaining in batch l in period $k+1$ by simply subtracting the amount of gas extracted (α_i^{kl}) in period k from its current value.

On the other hand, note that more than one of the batches present in a pipeline might be required to supply gas in the same period. We thus impose in each pipeline i that the sum of all corresponding variables $r_i^{t,l-1}$ of batch $l-1$ from the period when it was created ($t = l-1$) up to the current period (k), must complete the unit (i.e., the batch $l-1$ must be consumed completely) in order to be able to extract gas from its predecessor batch l in period k. This constraint can be put in the following form:

$$y_i^{kl} \leq \sum_{t=l-1}^{k} r_i^{t,l-1}, \forall i \in L, 0 \leq l \leq k \leq \kappa. \tag{8}$$

Eq. (8) basically implies *the FIFO principle of a queue*, i.e., the batch l in pipeline i can be consumed in period k, only if its predecessor is fully consumed during periods $l-1, \cdots, k$. In addition, since r_i^{kl} defines the proportion of gas extracted from batch l in period k, the sum of r's for the whole horizon cannot exceed 1:

$$\sum_{k \in K} r_i^{kl} \leq 1, \forall i \in L, l = 0, \cdots, \kappa. \tag{9}$$

2.5. Gas quality constraints

A key point of the model proposed in this paper is its capability of keeping trace of energy content and quality of the gas that flows through the pipeline system. These gas properties, also referred to as quality parameters in this work, are imposed by the market and have to be considered by the producer. For instance, the customer may require that the percentage of ethane C_2H_6 and propane C_3H_8 in the natural gas mixture meets certain bounds for industrial or economic purposes, but it may require that the content of carbon dioxide CO_2 and mercury H_g be less than a certain value for environmental reasons. Hence, the model must consider these restrictions when requiring gas from the available supplies. All constraints related to the quality requirements of natural gas are studied next.

Let Γ be the set of gas quality parameters, including contaminants and energy content, that must be measured and tracked during the planning horizon.

For the current period, let the variable λ_{ia}^{kl} represent the quality parameter $a \in \Gamma$ of the batch that in period l entered pipeline i. Similarly, let the variable Λ_{ia}^{k} represent the quality parameter a of the total flow leaving pipeline $i \in L$, and the total flow entering sink node $i \in T$ in period k.

Note first that, as shown in Fig. 3, the total flow stream leaving pipeline i in period k can be composed of several batches with possibly different quality. Hence, the flow streams coming from these batches are first blended into a virtual pool before being sent as a unique gas flow to the corresponding adjacent pipelines. We thus impose the following constraint:

$$\Lambda_{ia}^{k} \cdot \sum_{j \in V_i^-} x_{ij}^{k} = \sum_{l=0}^{k} \lambda_{ia}^{kl} \cdot \alpha_i^{kl}, \forall i \in L, k \in K, a \in \Gamma. \quad (10)$$

Eq. (10) estimates the resulting quality of the gas stream leaving pipeline i, which is used to create the new batches in adjacent pipelines in the current period, as a linear blending of each quality parameter $a \in \Gamma$. More precisely, the gas quality after the blending conducted in pipeline i is estimated as the weighted average quality Λ_{ia}^{k} of all batches supplied from the pipeline in period k.

As a consequence, the quality of the new batch built up in period k in pipeline $j \in L$ is imposed by the weighted average quality of all flow streams that are entering the pipeline (see Fig. 3). We can then estimate the gas quality of batch k by imposing the following constraint:

$$\lambda_{ja}^{kk} \cdot \beta_j^{kk} = \sum_{i \in V_j^+} \Lambda_{ia}^{k} \cdot x_{ij}^{k}, \forall j \in L, k \in K, a \in \Gamma. \quad (11)$$

2.6. Final inventory requirements

First, the minimum line-pack required in pipeline i at the end of the whole planning horizon is imposed by

$$\sum_{l=0}^{\kappa} \beta_i^{\kappa+1,l} = \Delta_i^{final}, \forall i \in L, \quad (12)$$

where $\Delta_i^{final} = \beta_i^{00}$, i.e., the required final inventory must be equal to the initial inventory in the pipeline.

Second, the quality of the final inventory is imposed as follows. Let the variable μ_{ia} represent the expected quality of the final inventory in pipeline i for each quality parameter $a \in \Gamma$. We can then put this constraint for each pipeline i as follows:

$$\sum_{l=0}^{\kappa} \lambda_{ia}^{\kappa+1,l} \beta_i^{\kappa+1,l} = \mu_{ia} \cdot \sum_{l=0}^{\kappa} \beta_i^{\kappa+1,l}, \forall i \in L, a \in \Gamma. \quad (13)$$

As Eq. (10) introduced in the previous section, Eq. (13) estimates the expected quality of the final inventory in pipeline i as the weighted average quality of the gas remaining in the batches of the pipeline in period $\kappa + 1$. Consequently, the following constraint is required:

$$\underline{\mu}_{ia} \leq \mu_{ia} \leq \overline{\mu}_{ia}, \forall i \in L, a \in \Gamma, \quad (14)$$

where $[\underline{\mu}_{ia}, \overline{\mu}_{ia}]$ are given bounds of the quality of the final inventory having at least the same values as the initial one.

2.7. Demand satisfaction in period $k \in K$

In order to meet the multiple contracts, we follow the strategic idea of demand satisfaction, i.e., we assume that no more than what is demanded can be delivered. We thus impose the constraint:

$$\sum_{i \in V_j^+} x_{ij}^k \leq d_j^k, \forall j \in T, k \in K, \tag{15}$$

where d_j^k is the maximum amount of natural gas required by customer $j \in T$ during the period $k \in K$.

Concerning the quality of the gas required at the terminals, we impose the constraint

$$\Lambda_{ja}^k \cdot \sum_{i \in V_j^+} x_{ij}^k = \sum_{i \in V_j^+} \Lambda_{ia}^k \cdot x_{ij}^k, \forall j \in T, a \in \Gamma, k \in K. \tag{16}$$

Eq. (16) estimates the quality at sink node $j \in T$ as the weighted average quality of all incoming flows at the terminal in period k.

Note that quality bounds are solely imposed at sink nodes. Let $[\underline{\Lambda}_{ja}^k, \overline{\Lambda}_{ja}^k]$ be the interval in which the quality of the flow has to be met at sink node $j \in T$. Then the following constraint is required:

$$\underline{\Lambda}_{ja}^k \leq \Lambda_{ja}^{kk} \leq \overline{\Lambda}_{ja}^k, \quad \forall j \in T, k \in K, a \in \Gamma. \tag{17}$$

2.8. Net mass flow constraint at source nodes in period k

Similarly, at the sources there exists a maximum supply of gas that can be used in order to satisfy market demand. Let b_i^k be the maximum amount of gas at source $i \in S$ in period $k \in K$. We thus impose that the sum of all gas streams leaving source node i in period k can no be larger than b_i^k as follows:

$$\sum_{j \in V_i^-} x_{ij}^k \leq b_i^k, \forall i \in S, k \in K. \tag{18}$$

2.9. Resistance of the pipeline $i \in L$ in period k

The Weymouth equation is used to define the relationship between the pressure drop and the flow through a pipeline.

$$\left(\sum_{j \in V_i^-} x_{ij}^k - \sum_{j \in V_i^+} x_{ji}^k \right)^2 = R_i \left(p_i^k - q_i^k \right), \forall i \in L, k \in K. \tag{19}$$

where R_i is the resistance factor of pipeline i, and p_i^k and q_i^k are the squared inlet and outlet pressure variables of the pipeline segment, respectively, in period k.

2.10. Valve constraints

As already mentioned, every link $(i,j) \in A$ has a valve that the model takes into account. An open valve implies that the downstream pressure at the start node and the upstream pressure at the end node are equal. Since a closed valve obviously implies zero flow through the link, we have the constraint

$$x_{ij}^k (q_i^k - p_j^k) = 0, \forall (i,j) \in A, k \in K. \tag{20}$$

Eq. (20) essentially allows the model to open or close a valve between any pair of adjacent nodes in the system. If $x_{ij}^k > 0$, valve (i,j) is open in period k, and (20) implies $p_j^k = q_i^k$. Correspondingly, if the pressures differ, the equation implies zero flow.

Table 1. Decision variables for the MINLP model.

$Obj \in \mathbb{R}$:	*Objective function value*
	Variables defined in period $k \in K$:
$x_{ij}^k \in \mathbb{R}_+$:	*Total flow through link* $(i,j) \in A$
$p_i^k \in \mathbb{R}_+$:	*Squared upstream pressure at node* $i \in L$
$q_i^k \in \mathbb{R}_+$:	*Squared downstream pressure at node* $i \in L$
$\beta_i^{kl} \in \mathbb{R}_+$:	*Amount of flow stored in batch* l *in pipeline* $i \in L$
$r_i^{kl} \in \mathbb{R}_+$:	*Ratio of gas extracted from batch* l *in pipeline* $i \in L$
$y_i^{kl} \in \mathbb{B}$:	*Activation of batch* l *in pipeline* $i \in L$ *for gas extraction*
$\alpha_i^{kl} \in \mathbb{R}_+$:	*Proportion of gas extracted from batch* l *in pipeline* $i \in L$
$\lambda_{ia}^{kl} \in \mathbb{R}_+$:	*Gas quality of batch* l *in pipeline* $i \in L$
$\Lambda_{ia}^k \in \mathbb{R}_+$:	*Weighted average quality of total flow stream leaving pipeline* $i \in L$
$\mu_{ia} \in \mathbb{R}_+$:	*Weighted average quality of the final inventory in pipeline* $i \in L$

2.11. A MINLP Model

Summarizing all the above, we can now formulate a *mixed-integer non-linear programming* model, $\mathcal{M}_1$, as follows.

$$(\mathcal{M}_1) \qquad Obj = \max \sum_{k \in K} \sum_{j \in T} \sum_{i \in V_j^+} x_{ij}^k \tag{21}$$

$$\begin{aligned}
s.t.\ (x,p,q,\beta,r,y,\alpha,\lambda,\Lambda,\mu) &\in \Omega, && && (22)\\
p_i^L \le p_i^k &\le p_i^U, && \forall i \in N, k \in K, && (23)\\
q_i^L \le q_i^k &\le q_i^U, && \forall i \in N, k \in K, && (24)\\
x_{ij}^k &\ge 0, && \forall (i,j) \in A, k \in K, && (25)\\
p_i^k, q_i^k &\ge 0, && \forall i \in N, k \in K, && (26)\\
\beta_i^{kl}, r_i^{kl}, \alpha_i^{kl}, \lambda_i^{kl} &\ge 0, && \forall i \in L, 0 \ge l \ge k \ge \kappa, && (27)\\
y_i^{kl} &\in B, && \forall i \in L, 0 \ge l \ge k \ge \kappa, && (28)\\
\Lambda_{ia}^k &\ge 0, && \forall i \in N \backslash S, a \in \Gamma, k \in K, && (29)\\
\mu_{ia} &\ge 0, && \forall i \in L, a \in \Gamma. && (30)
\end{aligned}$$

where $\Omega \subset \mathbb{R}_+ \times \mathbb{B}$ is the set of feasible solutions defined as:

$$\Omega = \{x,p,q,\beta,r,y,\alpha,\lambda,\Lambda,\mu |\ \textit{Eqs. } (1)\text{–}(20) \textit{ are satisfied.}\ \}$$

Table 1 shows a complete list of decision variables for the MINLP model, where all but variable $y_i^{kl} \in \mathbb{B}$ are continuous decision variables.

3. Numerical experiments

In this section, we present a computational evaluation carried out on the MINLP model proposed in the previous section. The aim is to examine the computability of the model, and analyze what features make the model more difficult to solve. This is accomplished by applying a global optimizer, BARON [6] on a wide set of test instances. BARON, which stands for *Branch And Reduce Optimization Navigator*, is an implementation of a variant of branch-and-bound where a convex program is solved in each node of the search tree. Note that BARON is set to call MINOS [5] to solve the convex subproblems. MINOS is an NLP local solver that iteratively solves subproblems with linearized constraints and an augmented Lagrangian objective function.

We impose a time limit of 3600 CPU-seconds on each application of BARON. A relative optimality tolerance, $\varepsilon = 10^{-2}$, is set to BARON; this implies that any feasible solution is

Table 2. Test instances.

	Size of the problem						Model Statistics			
								#Variables		#NLP
Ref	$\|S\|$	$\|L\|$	$\|T\|$	$\|A\|$	$\|K\|$	$\|\Gamma\|$	#Consts	Total	Discrete	terms
1A	2	2	3	10	1	4	172	139	4	240
1B	2	2	3	10	3	4	507	411	24	732
1C	2	2	3	10	6	4	1227	1041	84	1620
2A	3	4	5	14	1	7	480	367	8	697
2B	3	4	5	14	3	7	1430	1139	48	2115
2C	3	4	5	14	6	7	3455	2897	168	4722
3A	4	3	4	13	1	3	201	165	6	268
3B	4	3	4	13	3	3	619	523	36	822
3C	4	3	4	13	6	3	1516	1330	126	1833
4A	6	4	5	45	1	5	349	293	8	823
4B	6	4	5	45	3	5	1061	917	48	2493
4C	6	4	5	45	6	5	2543	2267	168	5232
5A	8	5	6	70	1	1	243	219	10	506
5B	8	5	6	70	3	1	777	705	60	1548
5C	8	5	6	70	6	1	1878	1734	210	3261
6A	10	6	7	102	1	2	390	346	12	1022
6B	10	6	7	102	3	2	1216	1096	72	3102
6C	10	6	7	102	6	2	2905	2671	252	6494
7A	12	7	8	140	1	11	1321	1109	14	5212
7B	12	7	8	140	3	11	3891	3395	84	15678
7C	12	7	8	140	6	11	9216	8294	294	32637
8A	14	8	9	176	1	13	1743	1467	16	7741
8B	14	8	9	176	3	13	5115	4479	96	23271
8C	14	8	9	176	6	13	12093	10917	336	48246
9A	18	11	14	53	1	2	1645	1335	22	2564
9B	18	11	14	53	3	2	4867	4113	132	7758
9C	18	11	14	53	6	2	11680	10260	462	17199
10A	10	10	10	33	1	4	754	614	20	1018
10B	10	10	10	33	3	4	2300	1940	120	3114
10C	10	10	10	33	6	4	5669	4979	420	7008

considered to be optimal if the gap between the objective function value and its upper bound is below 1% of the objective function value. In instances where BARON fails to compute the global optimum, it may still provide an upper bound on the maximum flow to give some indications on the quality of the output.

Our experiments were run on a 2.4 GHz Intel(R) processor with 2 GByte RAM under Linux Red Hat operating system, and the mathematical model was formulated in GAMS release [4] while using version 8.1.5 of BARON and version 5.51 of MINOS.

The test instances are shown in Table 2, where an instance identifier is given in the first column. Note that the instance identifier is composed of a number and a letter. The number represents the network topology designed as test instance, and the letter: {A, B, C} distinguishes the three planning horizons used for the test instances, namely $|K| = \{1, 3, 6\}$. Columns 2-6 show the size of the instance in terms of number of sources, pipelines, terminals, arcs, periods, and quality parameters, respectively. Furthermore, the model statistics for each instance, that is number of constraints, total number of variables, discrete variables and non-linear terms are given in the four last columns of Table 2.

Table 3 shows the results achieved by applying BARON to the MINLP model. Instance identifiers are given in the first column. Columns 2-6 show the CPU time (in seconds) spent by BARON, number of iterations of the branching, number of open branch-and-bound nodes, value of the objective function, and the upper bound for the problem, respectively.

TABLE 3. Performance of BARON when applied to $\mathcal{M}_1$.

Ref	BARON v.8.1.5 CPU	Its	Nodes	Y_b	UB	Gap (%) from UB
1A	2	10	8	10.04	10.04	0
2A	5	109	56	1.07	1.07	0
3A	0	1	1	0.41	0.41	0
4A	1	10	9	1.14	1.14	0
5A	1	1	1	1.01	1.01	0
6A	10	1	1	1.22	1.22	0
7A	52	1	1	1.23	1.23	0
8A	46	19	14	0.79	0.79	0
9A	32	64	44	7.28	7.28	0
10A	20	158	16	1.48	1.48	0
1B	4	28	24	15.33	15.33	0
2B	219	448	153	3.20	3.21	0.3
3B	18	91	60	1.15	1.15	0
4B	38	10	9	0.84	0.84	0
5B	172	379	273	3.02	3.03	0.3
6B	210	442	326	1.88	1.88	0
7B	1074	532	100	2.46	2.46	0
8B	463	46	24	1.58	1.58	0
9B	1631	667	253	8.98	8.98	0
10B	3600	1900	1296	1.99	2.34	14.9
1C	52	208	113	36.09	36.09	0
2C	3600	2944	1109	1.39	1.52	8.5
3C	108	253	191	1.70	1.70	0
4C	1675	2557	1710	1.31	1.31	0
5C	881	1117	845	3.01	3.01	0
6C	236	19	16	1.22	1.22	0
7C	3600	181	43	1.28	1.42	9.8
8C	3232	136	54	2.37	2.37	0
9C	3600	442	415	4.73	4.90	3.5
10C	3600	577	94	3.15	3.85	18.2

For the sake of illustration, Table 3 has been arranged with respect to the three planning horizons considered in the experiments for each test instance: $|K| = \{1, 3, 6\}$.

We can make several observations from Table 3. First, BARON could provide optimality in 25 out of 30 cases. Second, from the instances where it failed to do so, we observe that the optimality gap was no larger than 10% in 3 cases (2C, 7C, and 9C) and up to 20% in the remaining cases (10B and 10C). As we can observe, 4 out of 5 cases in which BARON failed to provide optimality correspond to the planning horizon C. This is an indication of the possible limitation of BARON to deal with network instances with a large number of time periods. On the bright side, we can also observe that BARON was capable to solve to optimality a model with more than 12,000 constraints, 10,000 variables and 48,000 non-linear terms (instance 8C).

Concerning the CPU time, we observe that BARON required less than 1 minute to provide optimality in all test cases with a planning horizon A (first group), less than 5 minutes in 6 out of 9 cases with a planning horizon B (second group), no more than 20 minutes in 2 cases, and up to 28 minutes in the remaining case (instance 9B). Regarding the test instances with a planning horizon C (third group), we can observe that BARON required less than 4 minutes to provide optimality in 3 out of 6 cases, no more than 30 minutes in 2 cases, and up to 54 minutes in the remaining case (instance 8C).

A second set of experiments was also conducted by applying a local optimizer, MINOS [5], to an alternative NLP model of $\mathcal{M}_1$. Note that MINOS relies significantly on the starting-point provided by the user. Finding a good starting-point is not a trivial task but it may lead to a better algorithmic performance of MINOS. We thus proceed to solve the NLP model by calling MINOS in a multi-local search procedure, i.e., we run this experiment by choosing 1000 starting points for MINOS. Different strategies were applied for providing such starting points. However, although MINOS performs well when applied to small network instances, it does not stand up when applied to more challenging networks, thus providing the zero solution in most of the moderate size instances, and similarly in all bigger networks.

4. Concluding remarks

As insurance against unforeseen or scheduled events that may affect the production or delivery of natural gas, a process known as line packing, where gas is temporarily stored in the pipeline system itself, is applied. By conducting this, a greater amount of natural gas can be supplied to delivery points during a given period of high demand than what it is currently injected at sources.

In this paper, we have presented a MINLP mathematical formulation for a multi-time period horizon to tackle the line-packing problem by building up heterogeneous batches in pipelines. The model maximizes the flow of natural gas in a transmission pipeline system and keeps track of energy content and the quality at the nodes of the network to maintain the reliability of supply needed to meet clients' demand. We have also presented a computational evaluation on a wide set of test instances to assess the computability of the model by applying a global optimizer, which has proved to be efficient for a moderate number of time periods.

Acknowledgements

This work was financed by the Norwegian Research Council, Gassco and Statoil under contract 175967/S30.

References

[1] R. G. Carter and H. H. Rachford Jr. Optimizing line-pack management to hedge against future load uncertainty. PSIG 0306, 2003.

[2] Council of European Energy Regulators. *Recommendations on implementation of Third Party Access to Storage and Linepack.* Report at the Madrid Forum Joint Working Group (2003/55/EC), July, 2003.

[3] L. Frimannslund and D. Haugland. Line pack management for improved regularity in pipeline gas transportation networks. In: Safety, Reliability and Risk Analysis. Theory, Methods and Applications. CRC Press 2009 ISBN 9780415485135, pp. 2963-2969, 2008.

[4] GAMS Development Corporation. GAMS: The Solver Manuals, Washington, DC, USA, 2008.

[5] B. A. Murtaugh and M. A. Saunders. MINOS 5.1 User's guide. Tech Report SOL-83-20R, Systems Optimization Laboratory, Stanford University, Stanford, CA, USA, 1983.

[6] M. Tawarmalani and N. V. Sahinidis. Global optimization of mixed-integer nonlinear programs: A theoretical and computational study, Mathematical Programming, 99(3):563-591, 2004.