**NTNU – Trondheim**
Norwegian University of
Science and Technology

# A Logic Branch and Bound Algorithm for Petroleum Production Optimization Based on Generalized Disjunctive Programming

Jakob G. Hansen-Tangen
Sindre Dombu Sangnes

Industrial Economics and Technology Management
Submission date: June 2013
Supervisor: Henrik Andersson, IØT
Co-supervisor: Vidar Gunnerud, IO-senteret

Norwegian University of Science and Technology
Department of Industrial Economics and Technology Management

NTNU
Det skapende universitet

# MASTERKONTRAKT
## - uttak av masteroppgave

### 1. Studentens personalia

| Etternavn, fornavn<br>**Sangnes, Sindre Dombu** | Fødselsdato<br>**14. apr 1988** |
|---|---|
| E-post<br>**sindresa@gmail.com** | Telefon<br>**45662357** |

### 2. Studieopplysninger

| Fakultet<br>**Fakultet for samfunnsvitenskap og teknologiledelse** | |
|---|---|
| Institutt<br>**Institutt for industriell økonomi og teknologiledelse** | |
| Studieprogram<br>**Industriell økonomi og teknologiledelse** | Hovedprofil<br>**Anvendt økonomi og optimering** |

### 3. Masteroppgave

| Oppstartsdato<br>**15. jan 2013** | Innleveringsfrist<br>**11. jun 2013** |
|---|---|
| Oppgavens (foreløpige) tittel<br>**A Logic Branch and Bound Algorithm for Petroleum Production Optimization Based on Generalized Disjunctive Programming** | |
| Oppgavetekst/Problembeskrivelse<br>The purpose is to develop and test optimization based models and methods for the day to day petroleum production problem. The main focus will be on developing a special purpose branch and bound algorithm for handling on/off and well routing.<br><br>Main contents:<br><br>- Problem description<br>- Development of mathematical model(s) for the problem<br>- Development of method(s) for solving the problem<br>- Implementation and testing of model(s) and method(s)<br>- Presentation of results and discussion around the obtained results | |
| Hovedveileder ved institutt<br>**Førsteamanuensis Henrik Andersson** | Medveileder(e) ved institutt |
| Ekstern bedrift/institusjon<br>**IO-senteret** | Ekstern veileder ved bedrift/instutisjon<br>**Vidar Gunnerud** |
| Merknader<br>**1 uke ekstra p.g.a påske.** | |

## 4. Underskrift

**Student:** Jeg erklærer herved at jeg har satt meg inn i gjeldende bestemmelser for mastergradsstudiet og at jeg oppfyller kravene for adgang til å påbegynne oppgaven, herunder eventuelle praksiskrav.

Partene er gjort kjent med avtalens vilkår, samt kapitlene i studiehåndboken om generelle regler og aktuell studieplan for masterstudiet.

Trondheim 06.06.2013
**Sted og dato**

Sindre Sørgnes
**Student**

Henrik Sundsen
**Hovedveileder**

Originalen lagres i NTNUs elektroniske arkiv. Kopi av avtalen sendes til instituttet og studenten.

NTNU
Det skapende universitet

# MASTERKONTRAKT
## - uttak av masteroppgave

### 1. Studentens personalia

| Etternavn, fornavn<br>**Hansen-Tangen, Jakob G.** | Fødselsdato<br>**13. nov 1987** |
|---|---|
| E-post<br>**jakobght@gmail.com** | Telefon<br>**99621712** |

### 2. Studieopplysninger

| Fakultet<br>**Fakultet for samfunnsvitenskap og teknologiledelse** | |
|---|---|
| Institutt<br>**Institutt for industriell økonomi og teknologiledelse** | |
| Studieprogram<br>**Industriell økonomi og teknologiledelse** | Hovedprofil<br>**Anvendt økonomi og optimering** |

### 3. Masteroppgave

| Oppstartsdato<br>**15. jan 2013** | Innleveringsfrist<br>**11. jun 2013** |
|---|---|
| Oppgavens (foreløpige) tittel<br>**A Logic Branch and Bound Algorithm for Petroleum Production Optimization Based on Generalized Disjunctive Programming** | |
| Oppgavetekst/Problembeskrivelse<br>The purpose is to develop and test optimization based models and methods for the day to day petroleum production problem. The main focus will be on developing a special purpose branch and bound algorithm for handling on/off and well routing.<br><br>Main contents:<br><br>- Problem description<br>- Development of mathematical model(s) for the problem<br>- Development of method(s) for solving the problem<br>- Implementation and testing of model(s) and method(s)<br>- Presentation of results and discussion around the obtained results | |
| Hovedveileder ved institutt<br>**Førsteamanuensis Henrik Andersson** | Medveileder(e) ved institutt |
| Ekstern bedrift/institusjon<br>**IO-senteret** | Ekstern veileder ved bedrift/instutisjon<br>**Vidar Gunnerud** |
| Merknader<br>**1 uke ekstra p.g.a påske.** | |

## 4. Underskrift

**Student:** Jeg erklærer herved at jeg har satt meg inn i gjeldende bestemmelser for mastergradsstudiet og at jeg oppfyller kravene for adgang til å påbegynne oppgaven, herunder eventuelle praksiskrav.

Partene er gjort kjent med avtalens vilkår, samt kapitlene i studiehåndboken om generelle regler og aktuell studieplan for masterstudiet.

Trondheim 06.06.2013
**Sted og dato**

Jakob Hasn Tøyen
**Student**

Henrik Andersen
**Hovedveileder**

Originalen lagres i NTNUs elektroniske arkiv. Kopi av avtalen sendes til instituttet og studenten.

# Preface

This master thesis was conducted at the Norwegian University of Science and Technology (NTNU), within Managerial Economics and Operations Research at the Department of Industrial Economics and Technology Management.

All text in is written in LaTeX and edited in TeXworks. The models and algorithms presented are written in AMPL and solved using Bonmin and Ipopt.

We would like to thank our supervisor Associate Professor Henrik Andersson at the Department of Industrial Economics and Technology Management for professional guidance and fruitful academic discussions. We would also like to acknowledge the help and technical inputs provided by Post Doc. Vidar Gunnerud at the Department of Engineering Cybernetics. Finally, we would like to thank Alex Teixeira at Petrobras for providing necessary data.

Trondheim, June 7th, 2013

Jakob G. Hansen-Tangen

Sindre D. Sangnes

# Abstract

A new solution method for solving the real time production optimization (RTPO) problem for a petroleum production system is presented in this thesis. The objective function of the problem maximizes oil production and the RTPO handles decision variables at operational level. Including routing of production flows, lift gas allocation, and pressure configurations of the system. It is aimed to give decision support in a time horizon of days to weeks. Such problems require solution methods able to obtain solutions swiftly, as production planners adjust network components frequently to maintain optimal production.

The problem contains binary decision variables combined with nonlinear expressions and is mathematically classified as a nonconvex mixed integer nonlinear problem (MINLP). MINLPs are in general known as computationally expensive and hard to solve to optimality, and when nonconvexities are present, few solvers can guarantee global optimality. The solution method presented deviates from traditional optimization techniques applied to such problems, and introduces logic disjunctions to substitute the binary variables of the MINLP. A specialized branch and bound algorithm (LBB) is developed to utilize the structure of these disjunctions, and as time is of paramount importance for the RTPO, it is aimed to reduce demanded computational effort for the problem. The LBB is given a high degree of user flexibility to be able to tailor the algorithm to different problems.

Results of the LBB show substantial variation in solution efficiency when applied to a real petroleum production system. Only when specific problem knowledge is utilized to customize the algorithm to the current system, the algorithm provides solid reduction in computational effort compared to a recognized commercial solver. Also when applied to variations in system structure the LBB clearly outperforms the applied solver, and the effectiveness and robustness of the proposed algorithm when utilizing problem specific knowledge is confirmed. The fact that the LBB provides the same solution to the problem as the applied solver might also indicate that the nonconvexities of the problem are not as complex as expected, and that the solver is in fact able to find the global optimal solution.

# Sammendrag

En ny løsningsmetode for å løse et real time production optimization (RTPO) problem for et petroleum produksjonssystem er presentert i denne oppgaven. Problemets objektfunksjon maksimerer oljeproduksjon og RTPOen behandler beslutningsvarable på operasjonelt nivå. Dette inkluderer ruting av produksjonsstrømmer, allokering av løftegass og trykkonfigurasjoner i systemet, og gir beslutningsstøtte i en tidshorisont fra dager til uker. Denne typen problemer krever løsningsmetoder som raskt kan generere gode løsninger, ettersom produksjonsplanleggere justerer nettverkskomponenter ofte for å opprettholde optimal produksjon.

Problemet består av både binære beslutningsvariable og ulineære uttrykk, og kan dermed klassifiseres matematisk som et ikkekonvekst ulineært heltallsproblem (MINLP). MINLPer er generelt kjente for å kreve mye datakraft, og som vanskelige problemer å løse til optimalitet. Når problemet også er ikkekonvekst er det lite tilgjengelig software som kan garantere optimalitet. Løsningsmetoden som presenteres i denne oppgaven avviker fra tradisjonelle optimeringsteknikker brukt på denne typen problemer. Den introduserer logiske disjunksjoner som erstatter binærvariablene fra MINLPen. En spesialtilpasset branch and bound algoritme (LBB) utvikles så for å nyttiggjøre strukturen til disse disjunksjonene. Fordi tid er viktig for RTPOen, er dette ment å redusere datakraft brukt på problemet. LBBens design gir den brukerfleksibilitet til å tilpasses ulike problemer.

Resultatene fra LBBen varierer i stor grad når den brukes på et ekte oljeproduksjonssystem. Kun når problemspesifikk kunnskap blir brukt til å skreddersy algoritmen til det gjeldende systemet gir den en solid reduksjon i brukt datakraft sammenlignet med en anerkjent kommersiell solver. Også når den testes på varierte problemer utkonkurrerer LBBen den brukte solveren, og effektiviteten og robustheten til algoritmen, når kombinert med problemspesifikk kunnskap, bekreftes. Det at LBBen finner den samme løsningen som den brukte solveren kan også indikere at problemets ikkekonveksiteter ikke er så komplekse som forventet, og at solveren faktisk finner den globalt optimale løsningen til problemet.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

In a world with increasing energy consumption and demand, the need for efficient utilization of the available resources is clear. In 2011, the global energy consumption rose by 2.5%. Even with increased focus on renewable energy and output from other energy sources, oil remains the leading fuel of the world, with a 33.1% share of global consumption (BP, 2012). The world's population is expected to grow with over 25 percent up to almost 9 billion in 2040, and overall global energy demand is expected to rise with 35 percent. Oil will probably continue to be the primary fuel source, and efficiency in all parts of the value chain is a key part for handling the energy challenges (ExxonMobil, 2012).

With this in mind, and oil companies' focus on profitable production, operations research might play an important part for the energy and petroleum sector. New technology has led to discoveries of new petroleum fields and the possibility of extracting resources that until now has been unavailable. Optimization of the production of these resources might lead to better recovery rates, and hence result in better profitability and more available energy. Fields that have been in production for some time might also show declining production rates, and the need for high recovery is essential to maintain profitable operations.

Petroleum production optimization has lately been given increased attention, and several models describing production systems or fields over varying periods of time have been used. What parts of the production system to emphasize in modeling, and decisions related to these, are greatly dependent on the chosen time horizon of the problem. This thesis will focus on operational planning, i.e. decision support for a short time horizon, typically days to weeks. Such problems are commonly denoted as Real Time Production Optimization (RTPO) problems. Production rates, routing of flows in pipeline networks, and artificial lift gas allocation may be among the decision variables. Optimization models for the RTPO are most commonly formulated as mixed integer and include nonlinear descriptions of reservoir and multiphase flow behavior.

Regardless of the model formulation, the problem at hand is likely to be both hard

and time consuming to solve to optimality. Approaches that can provide faster convergence or better solutions are therefore relevant to investigate. Decomposition techniques have been applied to a number of formulations with promising potential when considering petroleum fields. However, when evaluating a single production system, i.e. one platform, decomposition techniques might not be of that much use. The focus should rather be on modeling the production network and its flows more accurately, and on methods that can enhance solution efficiency of the problem and guarantee its global optimality.

A general petroleum production system holds structures that can be formulated using logic disjunctions in mathematical modeling. The introduction of these simplify the general model by removing the need for binary variables, something that can be utilized in solution approaches. Decisions represented by binary variables can then be imposed through logic disjunctions, and a specially designed solution method can reintroduce the constraints related to them. This thesis will propose a reformulation of the classic mathematical formulation of the problem and present an algorithm that explicitly exploit the structure of the petroleum production system through logic expressions and disjunctive programming. The algorithm introduced is a customized branch and bound approach that branches directly on the problem structure of the system. The implementation will be tested on realistic data from the P35 production asset belonging to Petrobras.

The use of logic in modeling and solution methods has not been properly tested for petroleum production systems. The main scope of this thesis is to formulate a new model, develop a suitable solution algorithm for it, and compare its performance to an implementation of a regular mixed integer nonlinear programming (MINLP) model of the problem. As the combination of binary variables and nonlinear expressions might result in highly nonconvex solution spaces, finding the global optimum is hard. The idea is that by removing the binary variables and utilizing the introduced structure, one may be able to reduce solve time to that of available optimization software and possibly obtain solutions they cannot find. In addition, the development of a tailored algorithm facilitates a high degree of user flexibility that can be utilized through problem specific knowledge. This gives the possibility to control and observe the solution progress of the algorithm more systematically and in detail. This will also ease the process of testing the algorithm efficiency and one may gain valuable information about the optimization problem.

The thesis is organized as follows. First a background on petroleum production and optimization is provided to introduce the reader to common terminology and give a basic understanding of petroleum production. Next, previous work on operational petroleum production optimization and logic, disjunctive programming is presented to form a basis for the model and algorithm development. A regular optimization model for the RTPO is then presented together with techniques for handling nonlinearities, before a reformulation to a logic, disjunctive form is given. Next, the tailor-made algorithm for the problem is developed, with a following presentation of its important properties and characteristics. Subsequently, a computational study with discussion of results is included, followed by conclusions and further work at the end.

# Chapter 2

# Background

Petroleum production is a complex process, especially offshore, and the achieved result is dependent on many aspects. Before going into an optimization scheme, basic understanding of the different components of a production system is essential. The goal is not to describe the system as accurately as possible, as computational effort and solve time is likely to increase with the level of detail. The degree of accuracy required is heavily dependent on the goal and time horizon chosen for the model, but the committed optimizer must in any case have a basic understanding of the problems underlying system. This chapter will present basic technical aspects and expressions that are needed to describe an offshore production system and its dynamics in a simple way. The terminology presented will be used throughout the thesis. To motivate the choice of accuracy in the petroleum production description that later will be used, a short background on the basics of petroleum production optimization is first presented. The production asset that will be used in the computational study will also be presented, to form a basis for the models that will be developed.

## 2.1 Petroleum production optimization

Optimization methods have been applied to many areas within petroleum production. When working with such problems it is important to consider the implications of looking at different time horizons, as these include problems of different sizes and level of detail. Planning on multiple time horizons is needed to develop and operate a production asset, and the decisions to consider are highly reliant on the time horizon evaluated. Classification of different time horizons have been presented for various problems, see e.g. Schlumberger (2005), Saputelli et al. (2007), Nikolaou et al. (2006) and Foss et al. (2009). The alternatives are more or less the same when it comes to which decisions are included for the different time periods. A common way to separate the planning horizons is between strategic, tactical and operational planning, as illustrated in figure 2.1. This is the distinction that will be used in this thesis.

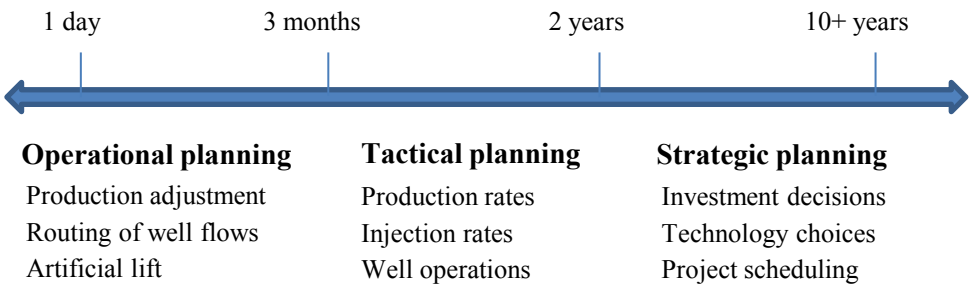


Figure 2.1: Different planning horizons for petroleum production

### 2.1.1 Strategic planning

On a long term horizon, typically several years, strategic decisions regarding development, investment and future scheduling of fields and projects must be made. Evaluations of different production scenarios, technologies, and estimates of future requirements must be made, in order to provide an optimal allocation and utilization of resources. An example of strategic planning in literature can be found in Nygreen et al. (1998). The authors suggest a multiperiod mixed integer deterministic programming model for investment planning for fields and scheduling of projects at the Norwegian continental shelf. The model had at the time been in use by the Norwegian Petroleum Directorate and major oil companies for more than fifteen years, and continuously improved during this period. Decisions are provided regarding production start on fields and design of pipeline systems, with the aim to maximize net present value for the chosen development plan. Carvalho and Pinto (2006) address long term planning of offshore oilfield infrastructure, and propose a mixed integer model for the optimization of exploration of oil and gas in a petroleum field. The model is aimed to support the decisions related to platform installation and to maximize net present value, and includes both discrete and continuous variables representing investment and production decisions, respectively.

The main scope of strategic planning in general is to assist in decision making on a high level. The focus in this type of planning should therefore be on capturing the main forces driving the decisions instead of accurate descriptions of various components of the systems (Nygreen et al., 1998). The need for a more detailed description of components becomes more relevant with shorter planning horizons.

### 2.1.2 Tactical planning

On a medium time horizon, typically 3 months to 2 years, decisions may include production rates to meet demand expectations, injection rates, routing in transportation networks, and the possibility for drilling programs. An example of this type of planning model is presented in Ulstein et al. (2005). They present a mixed integer program with the aim of maximizing profit for petroleum production in the Norwegian sector. The planning includes decisions on regulation of production levels,

production processing and routing of gas (multi-component flows) in transportation networks to meet demand. The model is simply constructed consisting of a few rigid building blocks. These are put together in a nodal network, to simply model the more complicated real world processes. The nodes represent the different processes of production, blending, splitting, and sales. With this simple structure the model is flexible and can easily be expanded or changed.

On the tactical level the target is often to choose a relatively stable production pattern and maintain this for some time. The modeling often includes more components than in strategic planning, but great simplifications are still done to describe the real processes in solid and simple models.

### 2.1.3 Operational planning

The shortest planning horizon is typically defined for days to weeks, and involves decisions on a more detailed level. The modeling therefore requires accurate descriptions of the system. In the case of offshore operations modeling this often includes descriptions of both subsurface and platform components. Important decisions are normally related to detailed production levels, artificial lift technology usage, and routing of well flows (Foss et al., 2009). To be able to provide decisions with more accuracy, the modeling needs to be more exact.

The problem studied in this thesis belongs as mentioned to this class, commonly known as RTPO. The next section will provide some background on petroleum production systems, based on the information needed when modeling and solving this kind of problem.

## 2.2 Petroleum production

A complete offshore petroleum production system consists primarily of a reservoir, wells, flowlines, a platform with process equipment such as separators and pumps, and normally transportation pipelines. A typical structure for a simple system with the most important components is shown in figure 2.2. Production of petroleum is subject to many factors that might impose restrictions to the system, such as reservoir conditions, falling pressure differentials and fluid handling capacities for pipelines and separators. The reservoir is a porous and permeable subsurface formation containing hydrocarbons, enclosed by impermeable rock or water barriers. The reservoir supplies the wellbore with hydrocarbons if the pressure in the reservoir is higher than the bottomhole pressure. The wellbore is the drilled hole in the bottom of the well, including the uncased portion. Measured pressure here is called bottomhole pressure (Schlumberger, 2013*b*).

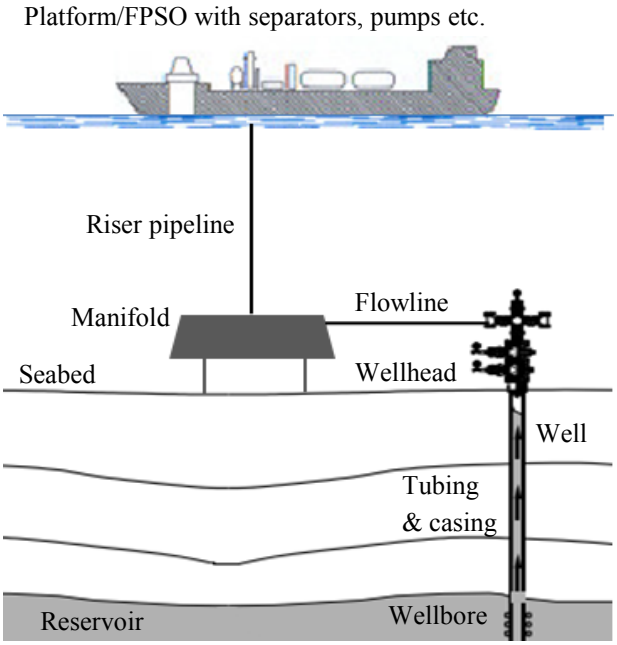


Figure 2.2: A simple petroleum production system

The well provides a connection from the wellbore to the seabed for the crude oil to flow in. This connection consists of inner tubing where the fluids flow, and casing, normally built of metal tubes cemented in the well. The casing works to protect the tubing, strengthen the sides of the well hole and to keep other fluids from leaking out. The well also provides ways that make it possible to control the rate of production, such as choke valves. The wellhead is installed at the top of the well at the seabed, and includes equipment to monitor and regulate the production from the reservoir. It also seals the well, prevents oil or gas from leaking out, and prevents blowout in case of high pressure formations (Devold, 2006). The system attached to the top wellhead is often referred to as a Christmas tree. It connects the well to the flowline, and has a series of functionalities. The tree consists of equipment such as safety valves, nozzles, pressure gauges, chokes and chemical injection points, and provides with this a number of ways to control and monitor the flow (Downey, 2009). From the seabed the produced fluids go via the wellhead and are transported to the platform through flow- and riser pipelines, usually to separators where the multiphase flow is treated. In many cases the well flows are connected to the platform via a manifold, as seen in figure 2.2, where flows from several wells can be routed to a fewer number of pipelines that are connected to the platform. Manifolds can also be placed on the platform; it is then called a topside manifold. Wells connected directly to the topside manifold, i.e. wells that flow directly from the well to the platform, are called satellite wells. The separators primary task is to remove gas and water from the oil. Pumps and compressors are used for purposes such as transporting gas and oil through pipelines, reinjection of water and gas into the reservoir to maintain pressure, or for artificial lift technologies such as lift gas injection (Guo et al., 2011).

A fundamental physical property for production is that fluids move from high pressure

to low pressure. A key element for all conventional oil production is that the pressure on the oil in the reservoir has to be above the bottomhole pressure, and that the pressure has to be falling downstream the system (from wellbore to platform) for the fluids to flow. This must also be included when modeling a production system in an optimization scheme.

### 2.2.1 Production phases

The production profile over the lifetime of an asset has phases with different characteristics. Production commences with the first quantities of hydrocarbons flowing through the well. From a commercial perspective this marks the turning point, as cash starts to be generated from the earlier investments in the field. The exact production cannot be foreseen, but the expected production profile is normally divided in three phases; the build-up period, the plateau period and the decline period. In the build-up period new wells are drilled and brought on stream, and the production gradually increases. In the plateau period the production for the asset is brought on to its peak level with the production facilities at full capacity, and a more or less constant production is maintained. New wells may still be drilled and start production, but the production level of older wells may start to decline. This period typically ranges between two and five years for an average oil field. In the decline phase the wells and the asset in general experience declining production and the profitability is falling. In addition, the decline period is usually the longest of the phases, so allocating resources in the best possible way and producing as efficiently as possible is crucial (Jahn et al., 2008). The use of optimization may increase recovery rates in the decline period, and thus extend the lifetime of a production asset. A typical production profile can be seen in figure 2.3.

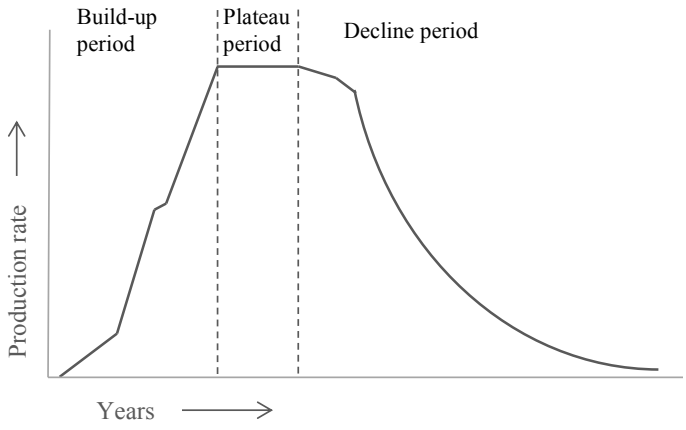


Figure 2.3: A typical production profile from simulation

### 2.2.2 Multiphase flow

The fluid mixtures produced from oil wells are normally complex with hundreds of components. An oil well stream is typically a turbulent and high-velocity mixture of gas and hydrocarbon liquids accompanied by water, various gases and sometimes solids (Gudmestad et al., 2010). To measure and model all these components exactly would be almost impossible. From an optimization point of view it is also both very difficult and not of interest to model all these components. To be able to make intuitive and usable models, simplifications are done to describe the flow. The multiphase flow is therefore described by oil, gas and water flow rate. The relations between the different flows can be described by two properties; the ratio between the amount of gas and oil in the flow is given by the gas to oil ratio (GOR); the liquid properties of the flow are described by the watercut, defined as the percentage of the total amount of liquid flow that is water. These relationships provide valuable information about the flow and about the properties of the well. Given information about the flow of one phase, the multiphase flow can be completely described by the GOR and the watercut. Mathematical representations is given in relations (2.1) and (2.2).

$$GOR = \frac{amount\ of\ gas\ in\ flow}{amount\ of\ oil\ in\ flow} \tag{2.1}$$

$$watercut = \frac{amount\ of\ water\ in\ flow}{total\ liquid\ flow\ (oil\ +\ water)} \tag{2.2}$$

### 2.2.3 Artificial lift gas

Production wells can be said to be free flowing or lifted. A free flowing well has high enough bottomhole pressure to keep a satisfying wellhead pressure and thereby maintain the required pressure drop for liquids to flow through the system. According to Schlumberger (2013*a*), about five percent of the approximately one million producing oil and gas wells in the world are tree flowing. Thus, almost all hydrocarbon production is dependent on some kind of lifting operation to be able to operate efficiently. Water and gas are normally injected into the reservoir to maintain the pressure over time, as the reservoir is being depleted and the pressure declines, but this cannot always guarantee a suitable pressure. Increasing watercuts over the lifetime of the field as it moves through the decline period may also complicate production. Driven by more activity in deep waters and areas where complex well construction is required, the need for high lifting rates has also increased in order to ensure profitable production (Fleshman and Lekic, 1999). If the reservoir pressure cannot provide desired production rates or flow from wells at all, artificial lift technologies can be applied to the well.

The most commonly used artificial lift technologies can be split in two categories. Mechanical lift uses different kinds of pumps to transfer mechanical energy to the

fluids to push them forward, e.g. beam pumps, submersible pumps and hydraulic pumps. The other is gas lift, which will be the lift technology considered in this thesis. Gas lift is the injection of compressed gas to the well flow to increase the production of oil (Economides et al., 2012). The gas is normally transported through separate pipelines from the platform and injected through valves into the lower section of the tubing, as illustrated in figure 2.4. The gas lowers the specific gravity of the fluid by aerating the liquid column, something that helps the well to flow easier. The aim is to bring the fluids to the top at a desirable wellhead pressure, while keeping the bottomhole pressure small enough to give a good pressure differential and driving force in the reservoir. The gas is compressed from the platform and when it is injected it expands, and the energy released in this expansion can also help to push the oil to the surface (Guo et al., 2011).

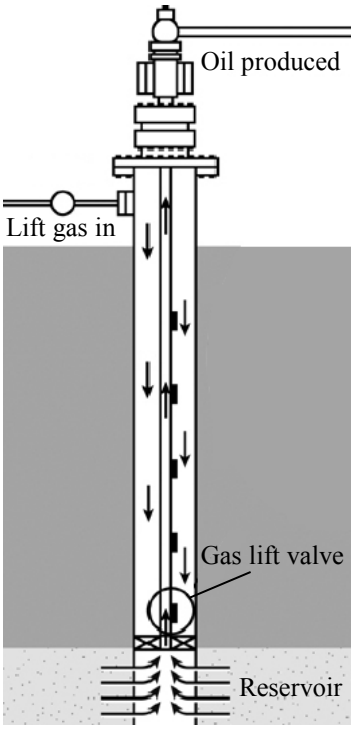


Figure 2.4: Typical well with gas lift injection, (Economides et al., 2012)

### Injection and production limitations

A production system often has a capacity for lift gas injection due to compressor capacities, and the available gas lift must be allocated efficiently between the wells to maximize production. However, the injection is also limited by physical factors. When a well is not free flowing it must be injected with lift gas to be able to produce. In addition, the lift gas injection for a well is often restricted by both upper and lower limits. This means that if a well is open and lift gas is used, the amount of gas injected has to be above a minimum and below a maximum amount. These restrictions combined make the lift gas allocation difficult and vital to maintain high production

rates. There can be many reasons for the injection limits. A minimum amount of gas has to be injected in order to be able to lift any fluids and affect the production. Avoiding slug flow is also a wanted effect. This is a flow impeding phenomenon where gravity causes liquid to accumulate and block the flow. This creates alternating periods of pressure build-up phases without production and periods of high flow rates, something that results in instability for the system (Sausen et al., 2012). To avoid this, a minimum amount of gas injection is required to maintain a continuous flow and avoid unstable flow regimes. The maximum limit for injection can typically be due to consequences in the long run, e.g. one does not want to produce too much because of unfortunate long term reservoir effects.

Production is also highly dependent on the pressures from reservoir and throughout the system, which again are closely connected. The differential between the reservoir and the bottomhole pressure is a driving force for production, and the wellhead pressure can be calculated in relation with the bottomhole pressure. Due to system properties such as reservoir and equipment characteristics, lower and upper bounds also exist for the wellhead pressure.

Lift gas and wellhead pressure are obviously crucial to the production of petroleum. In systems with characteristics described above, the oil rates are dependent on the relation between wellhead pressure and lift gas injection. This gives an area of possible production from the combined limitations on pressure and lift gas that further complicates the production, as depicted in figure 2.5.

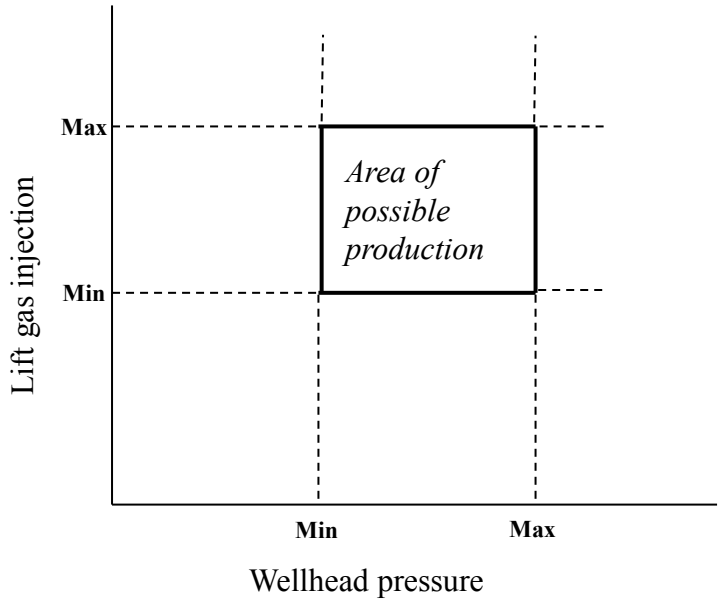


Figure 2.5: Possible area of production

### 2.2.4 Production regulation

Production of oil from a well is dependent on and most often given by the wellhead pressure and amount of lift gas injected. The pipeline behavior is dependent on the multiphase flow, and the flow through the system is dependent on the pressure drop over the pipelines. This is normally given as nonlinear functions of oil, gas and water flow. For controlling the production there are three main decisions that must be made. First, production flow can be pressure regulated, i.e. a well is not fully open. This is controlled with choke valves that can regulate the flow, and thus the pressure, to comply with system constraints. Second, lift gas can be injected to wells. Because of the production restrictions presented in the previous section, these two decision are strongly connected (figure 2.5). If a well is fully open it will be pressure constrained, i.e. no more flow can go through the system due to the required pressure drop through the system. Available lift gas is then injected to increase oil production as much as possible. The last big decision that must be made is on the routing of flows between wells, pipelines and separators. Flows should be routed in a way that takes maximum advantage of the capacities of equipment, such as processing capacities for separators. Proper handling of all these decisions is essential when modeling an optimization problem for a petroleum production system.

## 2.3 The P35 production asset

The P35 is a FPSO (floating production, storage and offloading unit) located in the Marlim field in the Campos Basin, about 110 km outside the east coast of Brazil. The Marlim field is a brown field, meaning that the production is in the decline phase mentioned in section 2.2.1, and the need for production planning is hence important to maintain profitability for the production. The Marlim field was discovered in 1985, and has a current output of 390,000 barrels of oil per day (OilVoice, 2012). Production started in 1991, with a peak production of 616,000 barrels of oil per day occurring in 2002 (Bampi and Costa, 2010).

Due to the declining production, the configuration of the P35 has been and is under changes to maintain production as high as possible. Some wells have been closed because of too low production, while new wells which are more efficient due to reservoir dynamics have been drilled to increase recovery. The P35 has for the time being the possibility of producing from 10 wells, connected to two subsea manifolds at the seabed, and one topside manifold located at the platform. Four wells are connected to subsea manifold 1, two wells to subsea manifold 2, and there are four satellite wells connected to the topside manifold, as shown in figure 2.6. From each of the subsea manifolds there are two pipelines connected to the FPSO. There are three separators to which flows from satellite wells or pipelines can be routed. Gas lift injection is included in all wells. There are also a number of water injection wells to maintain pressure in the reservoir. Production planning include decisions on which wells to produce from, pressure configuration of the system, routing of well flows to

pipelines in subsea manifolds, routing to separators for the pipelines and the satellite wells, and allocation of lift gas to wells. The platform has a production capacity of about 100,000 barrels of oil per day and about 3 million cubic meters of gas per day. It also has storage capacity for two million barrels of oil (Offshore-technology, 2012). To see these numbers in a context, total Norwegian oil consumption was in 2012 about 256,000 barrels of oil per day (Eia, 2013).

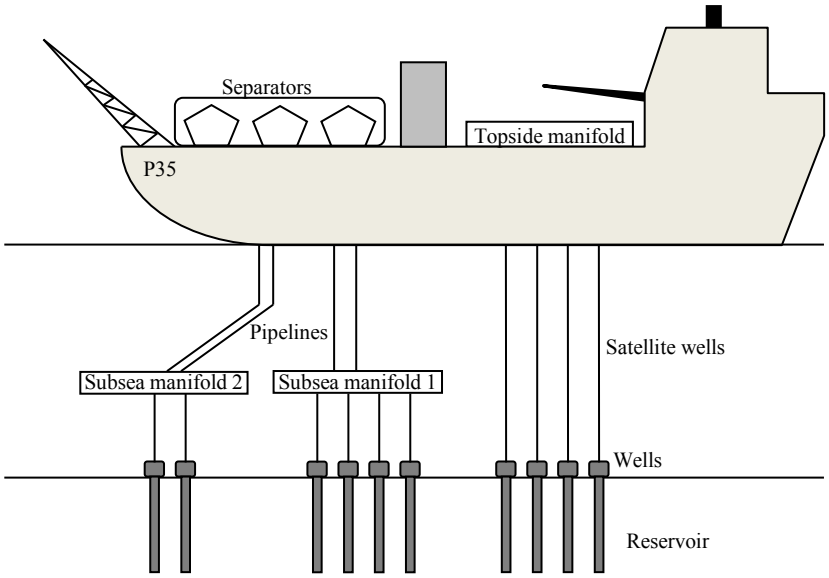


Figure 2.6: The P35 production asset

# Chapter 3

# Literature study

This section will give a brief review of previous work on optimization of operational planning of petroleum production. Since the focus of this thesis is on RTPO, the reviewed literature focuses on this type of problem. Further, to form a basis for the logic based models that will be developed, previous work within logic based and disjunctive programming is presented.

## 3.1 Previous work on operational production optimization

Common for models for RTPO is that they normally are formulated as MINLPs, including binary decision variables on well production and routing in pipelines and nonlinear flow behavior. This is widely accepted as the hardest type of optimization problem, and several methods and algorithms have been developed for the RTPO. Some highlights are presented here.

Wang (2003) gives an overview over models and solutions methods for the RTPO. The main focus is on formulations and solution methods for determining production rates, lift gas rates, and well connections for production systems, in order to maximize the operational objectives. Multiphase flow rates and pressure constraints are modeled. The effect of different simulations on the system is also considered. Both here and in Wang et al. (2002) it is shown that the effects of well interactions and complexity of production systems make a new formulation required, and a new model is proposed. A sequential quadratic programming algorithm is used and compared with the BB solution of a mixed integer linear problem (MILP).

Saputelli et al. (2003) describe the meaning of Real Time Optimization (RTO) to efficiently exploitat hydrocarbons and give forecasts for future technologies. RTO is a widely used and more general version of RTPO. They argue for the incentives for heavier use of RTO and relates the challenges to availability of technology. Saputelli et al. (2005) works further with this, and present a framework for optimizing

petroleum fields while complying with physical as well as business constraints. The focus is on reservoir-management in a multi-level decision making approach with continuously updating of reservoir performance. The proposed approach is not tested, but the authors recommend further validation of the method.

Bieker et al. (2006) provide a description of the information flow associated with the optimization of offshore oil and gas production systems. They include elements from the entire system, such as data, well and reservoir models, production planning and processing. A review of existing optimization models and technologies is presented, and challenges related to RTO are discussed. They conclude that closed loop optimization, where system configurations are reoptimized and imposed directly every time new data is available, should be implemented in a dynamic setting to improve the efficiency of RTO. Chen et al. (2009) presents such a scheme for closed-loop optimization, with geological reservoir model updating combined with production optimization. They show a significant increase in the objective value when testing the model.

Kosmidis et al. (2005) present a MINLP for the daily production planning of wells in petroleum fields. The entire production system is modeled, and both nonlinear reservoir and multiphase flow behavior, constraints from surface facilities and routing of flows to manifolds and separators are considered. A solution strategy involving a mix of logic imposed by binary variables, piecewise linearization and an outer approximation algorithm is presented. Results indicate an increase of up to 10% in oil production when compared to typically used heuristics.

In Gunnerud and Foss (2009) a MILP is proposed for the RTPO at the Troll west asset at the Norwegian continental shelf, and solved with decomposition techniques. The nonlinearities in the model occurring due to multiphase flows and well inflow, and routing decisions, are taken care of by piecewise linearization using special ordered sets of type 2 (SOS2) and a big-M approach. Two decomposition techniques, Dantzig-Wolfe decomposition and Lagrangian relaxation, are tested on the model. Results show that the use of decomposition techniques clearly outperforms normal MILP global solving techniques.

Güyagüler and Byer (2008) also present a MILP for the production allocation problem with the use of piecewise linear functions for the nonlinearities, with the aim to provide optimal production rates. They guarantee the exact solution when it can be found from using their approach, but also suggest an alternative formulation when it is not possible to comply with all system constraints simultaneously. Here they allow certain constraints to be broken and induce penalties when this is done. This is shown to be of great practical significance, since many systems have conflicting constraints, and the approach can then still always provide a solution to the system.

The literature on RTPO has been developed for some time, and methods are getting increasingly more efficient. The structure of RTPO problems and the underlying production system gives a good foundation for using logic more actively in modeling and solving problems. Imposing logic both in the model and solution approach may simplify models by e.g. removing variables and constraints. This has to the authors

of this study not been looked thoroughly into for the petroleum production problem. The next section will therefore provide some information on disjunctive programming in general, to investigate and provide an understanding of the use of logic, before using this on the RTPO.

## 3.2 Logic and disjunctive programming in optimization

MINLPs arise in several situations and are commonly recognized as computationally expensive and hard to solve to optimality. As computational computer power has risen exponentially the last decades, so has the attention to MINLPs. Several algorithms, including B&B (Gupta and Ravindran, 1985), outer-approximation (OA) (Duran and Grossmann, 1986), generalized Benders decomposition (GBD) (Geoffrion, 1972), extended cutting plane (ECP) (Westerlund and Pettersson, 1995), LP/NLP-based branch and bound (Quesada and Grossmann, 1992), and branch and cut (Stubbs and Mehrotra, 1999), use conventional optimization to solve MINLPs. Raman and Grossmann (1994) and Raman and Grossmann (1993) on the other hand, propose logic propositions and generalized disjunctive programming (GDP) as effective tools for handling the complexity of the MINLPs' structures. The scope of the GDP and the use of logic is to reduce solution time, explicitly exploiting the disjunctive structure of the reformulation of the MINLP.

The use of logic in modeling of discrete optimization was first studied by Balas (1974). He developed disjunctive programming (DP) as an alternative way to describe discrete variables in a mixed integer problem (MIP). In this case, the MIP is reformulated to a LP with subsets of constraints presented by disjunctions (sets of constraints of which at least one must be true).

Balas (1985) states that every MIP can be reformulated as a DP, and that every bounded DP can be reformulated as a MIP. Further, Raman and Grossmann (1994) point out that from a modeling standpoint it is more systematic, and often natural, to first give a logic presentation of a discrete problem. They show that it is not always beneficial to transform the logic formulations into equation form and introduce a new theoretical characterization of disjunctive constraints. This characterization is used to keep "poorly behaved" parts of the problem in disjunctive form, and transform the remaining "well behaved" part to mixed-integer equations, e.g. using the big-M method.

Raman and Grossmann (1994) also propose a solution algorithm to exploit the structure of the above formulated problem. They solve a process network problem and a jobshop scheduling problem using the proposed logic and algorithm. Both cases show substantial reduction in number of nodes, iterations, and used CPU-time.

Raman and Grossmann (1993) provide a branch and bound algorithm for solving a mixed integer linear programming (MILP) problem. This algorithm makes logic

decisions in each node and provides a mechanism for fixing subsets of binary variables. It is applied in synthesis of process flow sheets and shows that in comparison to the case with no logic introduced, the number of nodes enumerated is greatly reduced.

Bollapragada et al. (2001) investigate the possibility of using logic-based methods on the nonlinear problem of struss-structure design. They find that when introducing logic, the size of problems that can be solved to optimality significantly increases.

For more detailed description of the logic structure of the GDP, the reader is referred to Vecchietti and Grossmann (2000) and Lee and Grossmann (2000). The former present a framework of building more complex logic relations and describe modeling issues of implementing disjunctive programming. The latter describe algorithms for solving convex GDPs.

All of the above mentioned literature describes convex problems and programs. As the RTPO problem described in the previous chapter contains nonlinear flow approximations and a number of nonlinear expressions, we need to account for the possibility of a nonconvex problem. If nonconvexities are present, conventional solution methods to MINLPs are often trapped in suboptimal solutions. In the following some literature concerning disjunctive programming in combination with nonconvexity is presented.

Bonami et al. (2008) give an algorithmic framework for convex MINLPs. However, they point out that the algorithms may also serve as heuristics for the nonconvex problem. Grossmann (2002) gives a similar overview, but in addition briefly discusses extensions to the nonconvex case. When considering nonconvexities, he focuses on the formulation of a lower-bounding MINLP problem in combination with global optimization techniques. Several branch and bound methods are mentioned, and although computationally expensive, all of them rigorously find the global optimum. Grossmann also points to heuristic strategies as a way to reduce the effect of nonconvexities. While not being rigorous, these techniques require less computational effort.

Lee and Grossmann (2001) propose a global optimization algorithm for nonconvex GDPs. They use convex underestimating functions for the continuous variables and construct the convex hull of the nonlinear disjunction. Lower bounds of the problem is found by discrete branch and bound on the relaxed convex GDP and a spatial branch and bound, that branches in both integer and continuous variables, is used to solve nonconvex NLPs to update the upper bound. The algorithm is used to solve several examples, and is proven to find a global optimal solution where a big-M method cannot, although CPU-time of the algorithm may increase, compared to the big-M method.

The literature presented shold provide a basic understanding of the purpose and application of general disjunctive programming. Background examples with proposed algorithms and results are given, and motivate further investigation of introducing

logic to the RTPO problem due to the complexity and structure of the MINLP. But first, a general mathematical model of the problem is derived and presented in the following chapter. A brief introduction on how to express MINLPs with disjunctions and logic propositions is given in section 5.1.

# Chapter 4

# Model formulation

There are numerous ways of combining wells, manifolds, pipelines, and separators in an oil production network. In this chapter a general mathematical formulation of a petroleum production network, such as Petrobras' asset P35 (figure 2.6), is presented. The model can easily be extended to include a variety of manifolds, wells, pipelines, or separators, but the P35 will later in this study be used as an example case. The model maximizes the total amount of oil extracted from the reservoir over a short period of time. Variable values associated with the optimal objective value may serve as decision support in the goal of increased oil recovery from a petroleum field.

Because of the model structure and the authors' optimization background, the outline of this chapter is as follows. First, assumptions behind the model are discussed and necessary sets, indices, parameters and variables are presented. Then, the model is presented in its full formulation.

## 4.1 Assumptions

To be able to formulate the mentioned example case as a relatively simple mathematical program, and simultaneously keep a highest possible level of reality to the problem, several assumptions are made. The four assumptions with greatest impact on the model are discussed below, two of them are physical and two of them economic.

The first of the physical assumptions is that the formulations throughout this text disregard the element of time. In general, reservoir dynamics of a field change when oil is extracted over time, but if the planning horizon of a model is within a certain time limit, specific properties of the field can be viewed as constant, e.g. flow composition. The RTPO problem covers production problems for a short time interval, i.e. days to weeks, and the reservoir dynamics of the Marlim field are considered to be changing slowly. Thus, it seems reasonable to consider the model as independent of time. A direct result of this assumption is time independent GORs

and watercuts for the production wells.

A second physical factor left out in this text is temperature. The impact of temperature and temperature change in pipelines and other equipment is of great importance when considering hydrates and wax formation. The use of glycol to prevent such formations is a problem of flow assurance in petroleum processing. It is assumed that this has little or no impact on the production flow in this model, and it is therefore left out.

The objective function of the model maximizes total amount of oil extracted from the reservoirs. As oil producers in general want to maximize total profit, the question can be raised if oil production is a relevant production measure. Several answers to that can be argued as the formulation does not include the cost of lift gas or other operational costs related to production. The given formulation is either way chosen because of the economical assumption that oil flow in most cases reflects short term revenue. In addition, it is a common production measure and fairly easy to implement. Another economic assumption is that there is an infinite demand for oil, i.e. there is always a market for the produced commodities.

With these assumptions forming the basis for the problem, sets and indices, parameters and variables necessary to formulate the production network algebraically are introduced in the following sections.

## 4.2 Sets and indices

A complete overview of the sets and indices used in this text is given in table 4.1. Note that some sets are indexed over another set and some are subsets of others. $M$ is the set of all manifolds, while $M^T$ and $M^S$ are subsets of $M$ corresponding to the topside and subsea manifolds, respectively. Sets for wells, $J$, and pipelines, $L$ , have ranges covering all wells and pipelines, respectively. But they also have indexed subsets, $J_m$ and $L_m$ , that ranges over a number of wells or pipelines connected to a certain manifold.

| | | |
|---|---|---|
| $M$ | - | *set of manifolds, indexed by m* |
| $M^T$ | - | *set of topside manifolds, indexed by m* |
| $M^S$ | - | *set of subsea manifolds, indexed by m* |
| $J$ | - | *set of all wells, indexed by j* |
| $J_m$ | - | *set of wells of manifold m, indexed by j* |
| $L$ | - | *set of all pipelines, indexed by l* |
| $L_m$ | - | *set of pipelines of manifold m, indexed by l* |
| $P$ | - | *set of all phases, indexed by p ∈ o,w,g (oil, water, gas)* |
| $S$ | - | *set of separators, indexed by s* |

Table 4.1: Sets and indices

## 4.3 Data

Table 4.2 gives an overview of data and parameters used in the model formulations. They are mostly system capacities, given pressures of system components, or given fractions of phases in the production flow. Take special notice to $f^W_{mjo}(\ )$ and $f^L_{ml}(\ )$ which are nonlinear functions used to describe complex reservoir dynamics and pressure drop through pipelines.

| | | |
|---|---|---|
| $P^S_s$ | - | *separator pressure for separator s* |
| $P^{W,MAX}_{mj}$ | - | *maximum well head pressure in well j of manifold m* |
| $Q^{O,MAX}_{mj}$ | - | *maximum oil production from well j of manifold m* |
| $Q^{I,MAX}_{mj}$ | - | *maximum lift gas injected in well j of manifold m* |
| $Q^{I,MIN}_{mj}$ | - | *minimum lift gas injected in well j of manifold m* |
| $C^G$ | - | *total gas handling capacity for platform* |
| $C^{LIQ}_s$ | - | *liquid handling capacity for separator s* |
| $C^W_s$ | - | *water handling capacity for separator s* |
| $C^{LG}$ | - | *liftgas capacity for platform* |
| $f^W_{mjo}(\ )$ | - | *function for oil flow from well j of manifold m* |
| $f^L_{ml}(\ )$ | - | *function for pressure drop in pipeline l of manifold m* |

Table 4.2: Parameters and data

## 4.4 Variables

All variables used in the formulation of the general production model are presented in table 4.3. These include pressure and flow variables for several production components and binary variables for routing of flow from wells to pipelines and separators, and from pipelines to separators. Figurative sketches with the most common variables are presented in figure 4.1 and 4.2.

Figure 4.1 presents the routing of flow from wells to pipelines for the subsea manifolds. Within the dashed area illustrating the manifold, each well has two routing possibilities of which only one can be chosen. That is, a well can only be routed to one pipeline, but several wells can be routed to the same pipeline. If a well is routed to a pipeline (represented by the solid line in the figure), the binary variable $y_{mjl}$ takes the value of one. Thus, the other routing possibility for the well cannot be chosen and the binary variable for this connection takes the value zero (represented by a dashed line). Closing the well is a third routing possibility, setting both routing variables to zero.

| | | |
|---|---|---|
| $q^W_{mjp}$ | - | *flow of phase p from well j connected to manifold m* |
| $p^W_{mj}$ | - | *pressure at well head for well j connected to manifold m* |
| $q^L_{mlp}$ | - | *flow of phase p in pipeline l connected to manifold m* |
| $p^L_{ml}$ | - | *pressure before separator inlet in pipeline l of manifold m* |
| $p^M_{ml}$ | - | *pressure in pipeline l at manifold m* |
| $q^I_{mj}$ | - | *amount of liftgas injected in well j connected to manifold m* |
| $p^D_{ml}$ | - | *pressure drop in pipeline l connected to manifold m* |
| $y_{mjl}$ | - | *1 if well j is routed to pipeline l through manifold m,0 otherwise* |
| $x^S_{mls}$ | - | *1 if pipeline l of subsea manifold m is routed to separator s,0 otherwise* |
| $x^T_{mjs}$ | - | *1 if well j of topside manifold m is routed to separator s,0 otherwise* |

Table 4.3: Variables

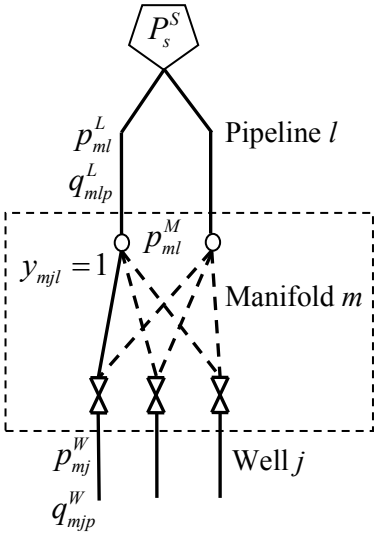


Figure 4.1: Subsea manifold routing

A similar description can be transferred to the routing of flow from wells or pipelines to separators. As there are more separators to choose from for a given well or pipeline than pipelines for a well, separator routing contains more routing possibilities and binary variables, but the idea is the same. In figure 4.2 the separator routing is illustrated. To the left, routing from pipelines to separators is presented. To the right, the routing of satellite wells directly to separators is illustrated. As for the subsea manifold routing in figure 4.1, when a connection is made to a separator, from a well or a pipeline, all other routing alternatives from that well or pipeline are denied. A separator can be connected to several pipelines and wells, but a well or pipeline can only be connected to one separator. For a connection between a well and a separator or a pipeline and a separator, the variables $x^T_{mjs}$ and $x^S_{mls}$ takes the value of one, respectively. If no connection is made, they are zero. It should be noted that well connections to separators can be closed completely, while pipelines must

be routed to a separator if there is flow to the pipeline. This also means that there is no storage capacity in the pipelines.

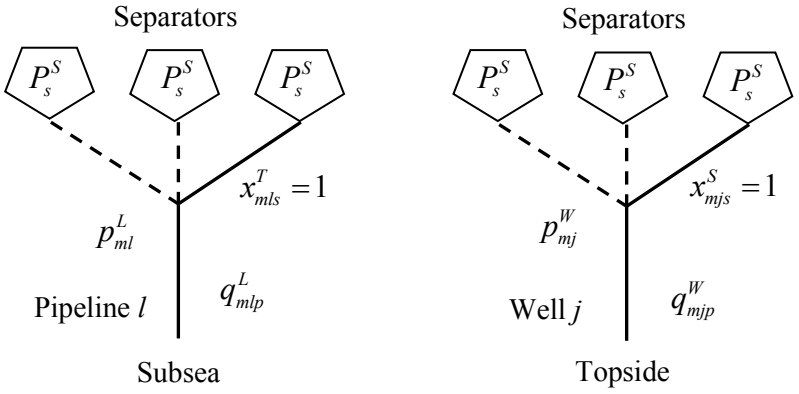


Figure 4.2: Separator routing

Figures 4.1 and 4.2 shows simplified systems containing a few components, but serves their purpose of illustrating the idea of the binary routing variables. Both figures can easily be extended to include more production components or be placed as parts in bigger production network illustrations.

## 4.5 Model

In this chapter a general mathematical MINLP-formulation of the RTPO is presented. The problem includes decision variables such as which wells to keep open, routing of flow from wells through manifolds to pipelines and separators, pressure configurations, and injection of artificial lift gas to allow and increase production. First, the objective function of the formulation is presented. Then, constraints are divided into groups defined by their field of application, and their formulations presented.

### 4.5.1 Objective function

The objective function (4.1) maximizes the sum of oil flow from all wells connected to the topside manifold and all pipelines connected to subsea manifolds. Thus, total oil production is maximized.

$$\max Z = \sum_{s \in S} \sum_{m \in M^T} \sum_{j \in J_m} q_{mjo}^W x_{mjs}^T + \sum_{m \in M^S} \sum_{l \in L_m} q_{mlo}^L \tag{4.1}$$

Production flow from wells connected to the topside manifold is associated with a binary variable, $x_{mjs}^T$, while flow from pipelines connected subsea is not. This is necessary as flows from the satellite wells are not regulated elsewhere, and this is the only possibility to shut down a satellite well.

### 4.5.2 Constraints

The production system is modeled as a network where production flow is routed through manifolds to separators. The network consists of wells, pipelines, manifolds, and separators, as outlined in figure 4.1. Following flow from reservoirs to separators, several elements limit the production flow through the system. These limitations must be included in the problem formulation, and are given below as the constraints in this model. Constraints are categorized by the headlines of this subsection.

#### System specifications and capacities

The following constraints fathom separator handling capacities of phase flows, upper limits of pressure and flow in the wells and regulate the use of artificial lift gas. Constraints (4.2), (4.3), and (4.4) state that the production flow cannot exceed the platform capacity for handling of gas or the separators' capacities for liquid and water. Gas is handled as a total of flow from all separators, while each separator has a capacity for the amount of liquid and water it can process.

Constraint (4.2) summarizes the flow of gas from open wells and pipelines, through all separators and limits it to be less or equal to the gas handling capacity for the platform. For the same reason as for the objective function, a binary variable is included for topside flow in this constraint, but not for the pipeline flows since these are regulated by the binary routing variables at the subsea manifold.

$$\sum_{s\in S}\sum_{m\in M^T}\sum_{j\in J_m} q^W_{mjg}x^T_{mjs} + \sum_{m\in M^S}\sum_{l\in L_m} q^L_{mlg} \leq C^G \tag{4.2}$$

Each separator has a handling capacity for liquid flow. Liquid consists in this context of oil and water, and constraints (4.3) limit total liquid flow from all wells and pipelines to a separator to be less or equal to its capacity. A binary variable is included both topside and subsea to handle which of the separators flow is routed to. The binary variables are multiplied by respective flows routed to the separators.

$$\sum_{p\in o,w}\sum_{m\in M^T}\sum_{j\in J_m} q^W_{mjp}x^T_{mjs} + \sum_{p\in o,w}\sum_{m\in M^S}\sum_{l\in L_m} q^L_{mlp}x^S_{mls} \leq C^{LIQ}_s, \quad \forall\, s\in S \tag{4.3}$$

The separators also have limited capacity for handling water alone, and constraints (4.4) limit the water flow to each separator not to exceed its capacity. Also here both terms of the constraints include binary variables.

$$\sum_{m\in M^T}\sum_{j\in J_m} q^W_{mjw}x^T_{mjs} + \sum_{m\in M^S}\sum_{l\in L_m} q^L_{mlw}x^S_{mls} \leq C^W_s, \quad \forall\, s\in S \tag{4.4}$$

Formulations in (4.5), (4.6), and (4.7) are system specifications and introduce well specific maximal values for oil production, lift gas injection, and wellhead pressure, respectively. They are handled as upper limits, and not constraints of the system.

The flow of oil out of a well cannot exceed the maximum amount of flow out of that well allowed by the system.

$$q^W_{mjo} \leq Q^{O,MAX}_{mj}, \quad \forall\, m \in M,\ j \in J_m \tag{4.5}$$

Injection of lift gas in a well is limited to the upper injection limit for that well.

$$q^I_{mj} \leq Q^{I,MAX}_{mj}, \quad \forall\, m \in M,\ j \in J_m \tag{4.6}$$

The wellhead pressure of a well cannot exceed the maximum wellhead pressure for that well.

$$p^W_{mj} \leq P^{W,MAX}_{mj}, \quad \forall\, m \in M,\ j \in J_m \tag{4.7}$$

In addition to the upper limit, the model also contains a lower limit for the amount of injected lift gas in a well. It states that if a well is producing the injected lift gas in that well must be greater than or equal to the well's lower lift gas limit. Constraints (4.8) introduces the lower lift gas limit for open wells connected subsea, and (4.9) the same for satellite wells.

$$q^I_{mj} \geq Q^{I,MIN}_{mj} y_{mjl}, \quad \forall\, m \in M^S,\ j \in J_m,\ l \in L_m \tag{4.8}$$

$$q^I_{mj} \geq Q^{I,MIN}_{mj} x^T_{mjs}, \quad \forall\, m \in M^T,\ j \in J_m,\ s \in S \tag{4.9}$$

Constraint (4.10) is the lift gas capacity constraint for the whole platform. Injected lift gas is summed over all manifolds and all wells connected to their respective manifolds, and must be less than or equal to the available capacity of the platform.

$$\sum_{m \in M} \sum_{j \in J_m} q^I_{mj} \leq C^{LG} \tag{4.10}$$

### Well model

Constraints (4.11) set the flow out of a well equal to $f^W_{mjp}(\,)$. The well flow expression is defined as a function of wellhead pressure and injected lift gas to the well, and is valid for all phases.

$$q^W_{mjp} = f^W_{mjp}(p^W_{mj}, q^I_{mj}), \quad \forall\, m \in M,\ j \in J_m,\ p \in P \tag{4.11}$$

Constraints (4.12) keep the flow of a phase in a pipeline equal to the flow of that phase from all wells connected to that pipeline.

$$q^L_{mlp} = \sum_{j \in J_m} q^W_{mjp} y_{mjl}, \quad \forall\, m \in M^S,\ l \in L_m,\ p \in P \tag{4.12}$$

### Pipeline model

Constraints (4.13) describe the pressure drop in a pipeline as a function, $f^L_{ml}(\,)$, of the flow of each phase in the pipeline.

$$p^D_{ml} = f^L_{ml}(q^L_{mlw}, q^L_{mlg}, q^L_{mlo}), \quad \forall\, m \in M^S,\ l \in L_m \tag{4.13}$$

Constraints (4.14) set the pressure drop of a pipeline equal to the difference between manifold and pipeline pressure. Constraints (4.13) and (4.14) could be modeled as one, but for clarity they are formulated separately.

$$p^D_{ml} = p^L_{ml} - p^M_{ml}, \quad \forall\, m \in M^S,\ l \in L_m \tag{4.14}$$

### Pressure and routing constraints

In this subsection the pressure and routing constraints of the model network are described. These constraints are presented together as the pressure relationships are required feasible only for connections of the network where flow is routed. Constraints (4.16), (4.18), and (4.20) state that for open production lines the pressure must decrease downstream the network, while constraints (4.15), (4.17), and (4.19) limit the routing of flow. Figure 4.1 gives a simple overview of how routing in a subsea manifold is limited.

Constraints (4.15) state that a well can be connected to at most one pipeline through a given subsea manifold. If all binary variables, $y_{mjl}$, are zero for a specific well, the well is routed nowhere, i.e. there is no flow out of the well and the well is closed. This can be seen in context with the mass balance of constraints (4.12), no liquids from the well are flowing downstream the system if all connections from the well is closed.

$$\sum_{l \in L_m} y_{mjl} \leq 1, \quad \forall\, m \in M^S,\ j \in J_m \tag{4.15}$$

If a well is connected to a pipeline through a subsea manifold, then constraints (4.16) makes sure that the wellhead pressure is greater than or equal to the pressure of the manifold.

$$p^M_{ml} y_{mjl} \leq p^W_{mj}, \quad \forall\, m \in M^S,\ j \in J_m,\ l \in L_m \tag{4.16}$$

The same relationships as for subsea manifold routing are set for the separator routing. In constraints (4.17) the flow from a pipeline is routed to exactly one separator. The equality operator is used to express that if there is flow to a pipeline, it must eventually be routed to a separator for processing of the flow.

$$\sum_{s \in S} x^S_{mls} = 1, \quad \forall\, m \in M^S,\ l \in L_m \tag{4.17}$$

When a pipeline is routed to a separator the pressure relationship in constraints (4.18) must hold, i.e. pipeline pressure must be greater than or equal to the separator pressure.

$$P^S_s x^S_{mls} \leq p^L_{ml}, \quad \forall\, s \in S,\ m \in M^S,\ l \in L_m \tag{4.18}$$

Constraints (4.19) regulate the flow from a satellite well to at most one separator. If all separator connections are set to zero, the well is closed.

$$\sum_{s \in S} x^T_{mjs} \leq 1, \quad \forall\, m \in M^T,\ j \in J_m \tag{4.19}$$

If there is flow from a satellite well to a separator, then the inequality of (4.20) must hold. The wellhead pressure must be greater than or equal to the pressure of the separator.

$$P^S_s x^T_{mjs} \leq p^W_{mj}, \quad \forall\, s \in S,\ m \in M^T,\ j \in J_m \tag{4.20}$$

The principle of the separator routing is given in figure 4.2 for both pipelines and wells.

### Non-negativity and binary constraints

All variables in constraints (4.21) are continuous, but can only take non-negative values. Note that all variables describing well behavior is indexed over all wells, but variables for pipelines are only indexed over the subsea manifolds, as there are no pipelines connected topside.

$$\begin{aligned}
&q^W_{mjp} \geq 0, && \forall\, m \in M,\ j \in J_m,\ p \in P \\
&q^L_{mlp} \geq 0, && \forall\, m \in M^S,\ l \in L_m,\ p \in P \\
&p^W_{mj}, q^I_{mj} \geq 0, && \forall\, m \in M,\ j \in J_m \\
&p^L_{ml}, p^M_{ml}, p^D_{ml} \geq 0, && \forall\, m \in M^S,\ l \in L_m
\end{aligned} \tag{4.21}$$

The binary variables are presented in constraints (4.22) and take discrete values of zero or one if a connection is made or not, respectively. As the continuous variables, these are also indexed over subsets of manifolds.

$$\begin{aligned}
&y_{mjl} = \{0,1\}, && \forall\, m \in M^S,\ j \in J_m,\ l \in L_m \\
&x^S_{mls} = \{0,1\}, && \forall\, m \in M^S,\ l \in L_m,\ s \in S \\
&x^T_{mjs} = \{0,1\}, && \forall\, m \in M^T,\ j \in J_m\ s \in S
\end{aligned} \tag{4.22}$$

Together, the objective function in (4.1) and constraints (4.2) through (4.22) form a general MINLP-model for an oil production network. Next, it is described how handling of the nonlinear properties of the model is taken care of. The nonlinear functions of phase flow and pressure drop (constraints (4.11) and (4.13)) are approximated, and assumptions for the flow composition are made.

## 4.6 Nonlinear approximations

Complex reservoir behavior and multiphase flows require nonlinear descriptions and complicate the modeling of the system. While the rest of the system can be explicitly formulated as equations and constraints in the optimization problem, these nonlinearities require special care. To be able to implement the entire optimization problem in a modeling language and combine it with some solver, the well flow and pressure drop are approximated by polynomial functions. These functions are fitted to simulated production data, through a least square approximation scheme performed in Matlab. Data for the P35 asset presented in section 2.3 has been gathered from Petrobras' in-house developed simulator, Marlim II, for both wells and pipelines. In the least square approximations the polynomials serve as proxy models for the nonlinear data describing the flows and pressures. Each polynomial is built by a number of coefficients coupled to the function variables. The least square fitting is in itself an optimization problem, where the target is to find the optimal coefficients so that the sum of the squared residuals of the polynomial values to the actual data is minimized. The type of polynomial chosen is naturally important for how good the fit will be, so different functions should be tested to provide a good fit.

### 4.6.1 Well approximations

In the general MINLP, the flow of each phase from a well is dependent on the wellhead pressure and the lift gas injection. As mentioned in section 4.1, constant reservoir dynamics is assumed within the planning horizon. The data provided by Marlim II contains information about flow composition, and well specific values of gas to oil ratio and watercut are assumed constant. To simplify the nonlinear descriptions, the oil flow is approximated by data, while the flow of gas and water are derived from linear relations to the oil flow. The following new parameters are introduced.

| | | |
|---|---|---|
| $GOR_{mj}$ | - | *gas to oil ratio for well j of manifold m* |
| $WC_{mj}$ | - | *water cut for well j of manifold m* |

Table 4.4: GOR and WC definitions

The flow of gas from a well is stated in (4.23) by the oil flow of the well multiplied by the GOR for that well. The amount of injected lift gas is also added to the gas flow of the well.

$$q^W_{mjg} = q^W_{mjo}\, GOR_{mj} + q^I_{mj}, \quad \forall\, m \in M,\ j \in J_m \tag{4.23}$$

The water fraction of the production flow in a well is formulated in (4.24) as the relationship between flow of oil and the specific watercut for that well.

$$q^W_{mjw} = \frac{q^W_{mjo}\, WC_{mj}}{1 - WC_{mj}}, \quad \forall\, m \in M,\ j \in J_m \tag{4.24}$$

The oil flows are given by the function (4.25), which is the well flow function of constraints (4.11) for $p = o$, are approximated for each well by the proxy model for the nonlinear behavior. The gas and water flows are then calculated explicitly from these. The proxy function is presented in (4.26).

$$q^W_{mjo} = f^W_{mjo}(p^W_{mj}, q^I_{mj}), \quad \forall\, m \in M,\ j \in J_m \tag{4.25}$$

$$q^W_{mjo} = \alpha_1 + \alpha_2\, p^W_{mj} + \alpha_3\, q^I_{mj} + \alpha_4\, p^W_{mj} q^I_{mj} + \alpha_5\, {p^W_{mj}}^2 + \alpha_6\, {q^I_{mj}}^2, \quad \forall\, m \in M,\ j \in J_m \tag{4.26}$$

The second degree polynomial in (4.26) is chosen to serve as the approximation function for the well flows. The data for the well flows can be plotted over a relatively smooth surface, and it is hence no need for more complicated polynomials in order to make good approximations. Figure 4.3 gives an example of the curve fitting for a well connected to a subsea manifold, while figure 4.4 gives the corresponding residual plot for the least square fit.

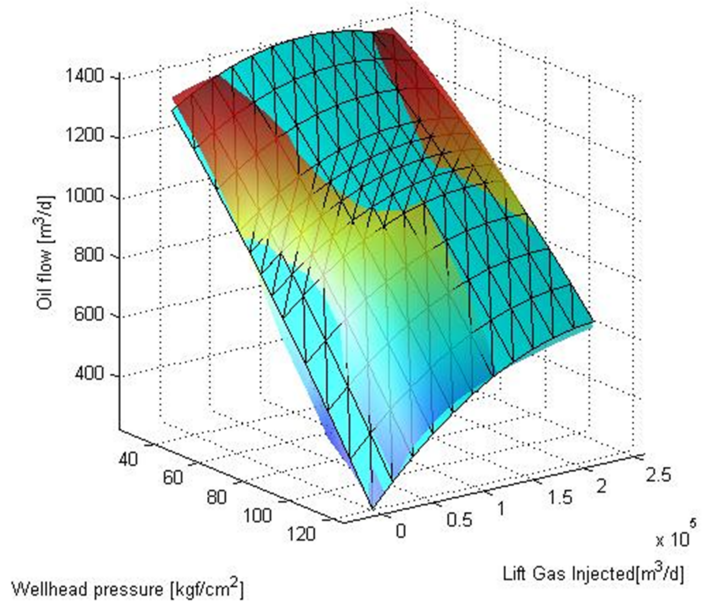


Figure 4.3: Plot of well flow approximation to real data

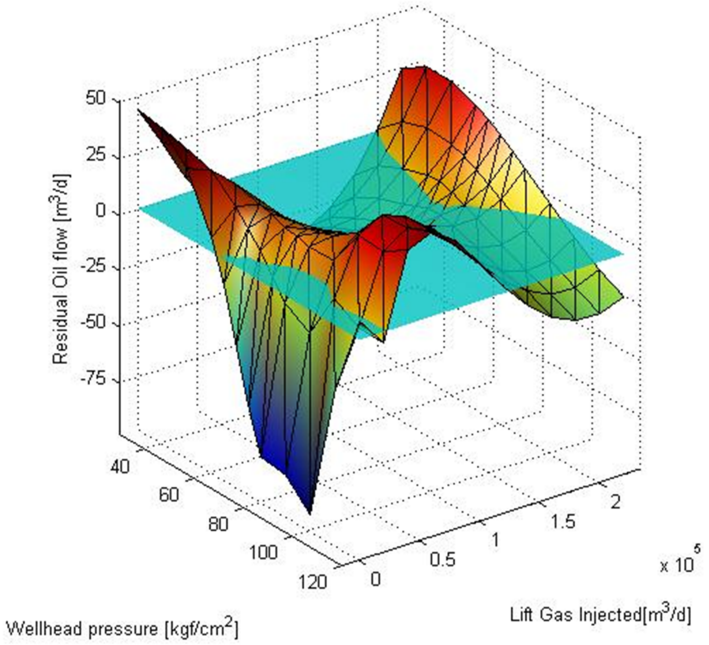


Figure 4.4: Well approximation residuals

In figure 4.3 the turquoise plot shows the approximated oil flows, while the transparent multicolored surface represents the real data. The plot demonstrates that the proxy model gives a satisfying fit for all the relevant data points. In figure 4.4 the transparent turquoise surface indicates the zero level, while the colored plot shows how the approximation deviates from the data. For most of the data points, except for points with very low lift gas injection and high wellhead pressure, the errors are small and within an absolute value of 25 $m^3$ oil/day. The error might seem high for some data ranges, but within the most likely areas of production, around 800 $m^3$ oil/day and more, the relative errors are small and the resulting oil flow should be descriptive. For the purpose of this thesis the approximations should work excellent as the focus is on solution methods rather than describing the nonlinear relations perfectly.

### 4.6.2 Pipeline approximations

The pressure drop in the pipelines is given as a function of the multiphase flow, i.e. the flows of oil, gas and water, as shown in equation (4.13). The pressure drop is dependent on three input variables, and the data has a more complex behavior than that of the well flow. A third degree logarithmic polynomial, shown in (4.27) is therefore chosen as proxy model for the pipeline approximations.

$$p^D_{ml} = f^L_{ml}(q^L_{mlw}, q^L_{mlg}, q^L_{mlo}), \quad \forall\, m \in M^S,\ l \in L_m \tag{4.13}$$

$$\begin{aligned} p^D_{ml} = {} & \beta_1\, ln(q^L_{mlo}+1) + \beta_2\, ln(q^L_{mlg}+1) + \beta_3\, ln(q^L_{mlw}+1) \\ & + \beta_4\, ln(q^L_{mlo}+1)^2 + \beta_5\, ln(q^L_{mlg}+1)^2 + \beta_6\, ln(q^L_{mlw}+1)^2 \\ & + \beta_7\, ln(q^L_{mlo}+1)\, ln(q^L_{mlg}+1) + \beta_8\, ln(q^L_{mlo}+1)\, ln(q^L_{mlw}+1) \\ & + \beta_9\, ln(q^L_{mlg}+1)\, ln(q^L_{mlw}+1) + \beta_{10}\, ln(q^L_{mlo}+1)^3 + \beta_{11}\, ln(q^L_{mlg}+1)^3 \\ & + \beta_{12}\, ln(q^L_{mlw}+1)^3 + \beta_{13}\, ln(q^L_{mlo}+1)\, ln(q^L_{mlg}+1)\, ln(q^L_{mlw}+1), \\ & \qquad \forall\, m \in M^S,\ l \in L_m \end{aligned} \tag{4.27}$$

Applying different functions for the pipeline approximations shows that the logarithmic polynomial clearly outperforms other alternatives, and is thus the preferred choice. The least square fitting is dependent on four dimensions, and it is therefore more difficult to visualize the approximations. However, by plotting parts of the data with fixed watercut, the pipeline behavior can be illustrated as a function of oil and gas flow. The corresponding water flow can easily be found from the oil flows and the watercut. Figure 4.5 shows the approximations for one of the pipelines connected to a subsea manifold with data corresponding to the watercut of zero. Figure 4.6 shows the residual plot belonging to the least square fit.

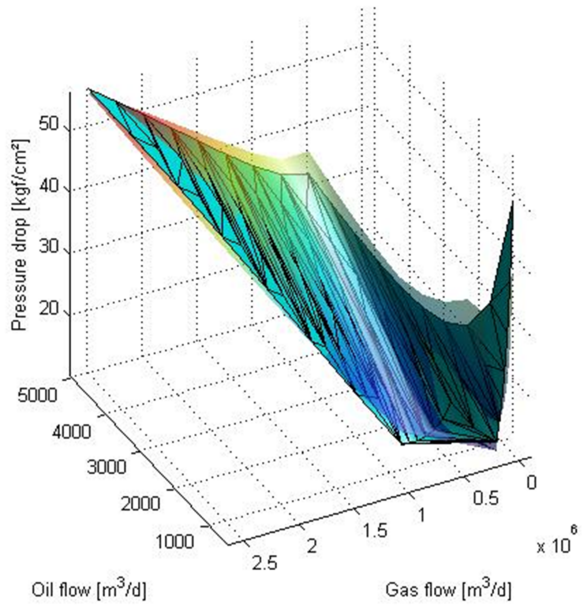


Figure 4.5: Plot of pipeline pressuredrop approximation to real data

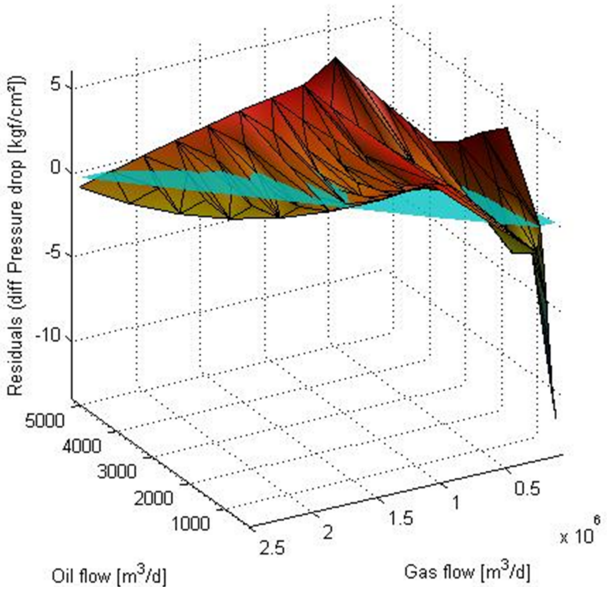


Figure 4.6: Pressure drop approximation residuals

Figure 4.5 shows the approximated data plotted in turquoise while the real data is given as the transparent multicolored one. Also here the polynomial gives a satisfying approximation to the simulated data. Figure 4.6 shows the residuals in the same way as for the well approximations. Except for the results where all flows are very close to or equal to zero the error terms are small, and always less than 5 $kgf/cm^2$. When the flows are zero the function will in any case set the pressure drop to zero. The results indicate that the proxy model should work fine as a representation of the pipeline behavior. Due to the more complex data behavior for the pipelines it is naturally harder to get a good match. If a more detailed representation is wanted the data should probably be approximated with several functions representing different operating areas for the system. The suggested functions should in any case provide good enough representations of the nonlinear functions for the system to be able to measure possible effects from the proposed solution method.

# Chapter 5

# Logic reformulation

In this chapter, the goal is to derive a model formulation that can provide a basis for an efficient solution algorithm to the RTPO-problem, presented in chapter 2. To make this possible, reformulations to parts of the MINLP-model presented in the previous chapter are done by introducing logic to the formulation. The aim with this is to provide a formulation that is equivalent to the MINLP-model in all physical properties, but where some of the challenging characteristics are expressed in alternative ways. The main result of this is that a model without binary variables is obtained, where the properties lost with this relaxation later can be handled through use of disjunctions and logic in a solution approach.

Parts of the original model are reformulated to facilitate the introduction of logic propositions and disjunctive constraints to the problem. The introduced disjunctions hold special structures that can be utilized in solution methods. In chapter 3, it is argued by Raman and Grossmann (1994) that a natural order of formulation may be to first give a full disjunctive formulation of a problem, and then reformulate "well behaved" parts to algebraic form. However, because of the structure of the RTPO-problem at hand, and the authors' optimization background, the order of formulation is turned in this thesis. The full algebraic MINLP is relaxed, allowing infeasible routing flow, before logic is added to restore feasibility. The resulting logic formulation is in itself a complete model of the problem at hand, and could be presented as the main model for this study, but as its formulation is closely related to the solution algorithm, it is thoroughly derived and presented here. Before the MINLP is reformulated, a basic introduction on how to reformulate an algebraic optimization problem using logic based programming is presented.

## 5.1 Logic based programming

Logic based programming can trivially be summarized from Raman and Grossmann (1994) through their formulation of generalized disjunctive programming (GDP). They state that the basic idea of GDP is to represent discrete decisions in the continuous

space as disjunctions, and constraints in the discrete space as logic propositions. While a MINLP is based entirely on algebraic equations and inequalities, the GDP facilitates representation of discrete decisions by allowing combinations of algebraic and logic equations. Raman and Grossmann give a good overview of notation and structure of the GDP, and in the following a brief introduction of the principles of disjunctive programming is given using their notation. A general linear or nonlinear program is stated below in the classic form of a mixed integer problem.

$$\begin{aligned} \min\ & Z = c^T y + f(x) \\ \text{s.t.}\ & g(x) \leq 0 \\ & h(x) + My \leq 0 \\ & Ay \leq b \\ & x \in R^n,\ y \in \{0,1\}^m \end{aligned} \tag{MIP}$$

Here, $x$ is a vector of continuous variables and $y$ a vector of binary variables. $M$, $A$, $c$ and $b$ are coefficient matrices and vectors. $f(x)$, $g(x)$ and $h(x)$ are linear or nonlinear functions of the continuous variable. To formulate the above program as a GDP, several structural changes are done. First, the binary variables, $y$, are removed, resulting in a model strictly consisting of the continuous variables, $x$. This alteration makes the problem simpler, but also deprives it of important properties. To restore these and simultaneously keep the obtained simplicity, auxiliary Boolean variables are introduced along with disjunctive constraints and logic propositions. The reformulation of the MIP can be expressed in disjunctive form the following way.

$$\begin{aligned} \min\ & Z = \sum_i \sum_k c_{ik} + f(x) \\ \text{s.t.}\ & g(x) \leq 0 \\ & \bigvee_{i \in D_k} \begin{pmatrix} Y_{ik} \\ h_{ik}(x, c_{ik}) \leq 0 \end{pmatrix},\ k \in SD \\ & \Omega(Y) = \text{true} \\ & x \in R^n,\ c \in R^m,\ Y \in \{\text{true,false}\}^m \end{aligned} \tag{GDP}$$

Here, some notational elaboration is in order. $SD$ is the set of disjunctions dividing the original solution space into subspaces, and k denotes a disjunction in this set. $D_k$ is the set of constraints associated with a disjunction $k$, and $i$ denotes a constraint in this set. For each subspace $k$ in $SD$, the V-shaped or-operator introduces logic relations between the Boolean variables, $Y_{ik}$, corresponding to algebraic constraints, $i \in D_k$. $Y_{ik}$ state whether a given term in a disjunction is true ($h_ik(x, c_{ik}) \leq 0$) or false ($h_{ik}(x, c_{ik}) > 0$). $h_{ik}(x, c_{ik})$ and $c_{ik}$ relate to $h(x)$ and $c$ from the MIP, respectively, by being their representation in a given disjunction and constraint within the disjunction in the GDP. $g(x)$ represent constraints valid over the entire search space, and $\Omega(Y)$ represents propositional logic expressing the relationship

between the Boolean variables. Note that the system contains no binary variables, only the continuous variables, $x$, and the auxiliary variables, $Y_{ik}$. For clarity, a simple example of disjunctions and logic propositions is given below.

If the set of disjunctions is $SD = \{1, 2\}$, and the set of constraints belonging to each disjunction are $D_1 = \{h_{11}(x, c_{11}) \leq 0\}$, $D_2 = \{h_{12}(x, c_{12}) \leq 0\}$, respectively, then discrete decisions in the continuous space can be expressed the following way.

$$\begin{bmatrix} Y_{11} \\ h_{11}(x, c_{11}) \leq 0 \end{bmatrix} \bigvee \begin{bmatrix} Y_{12} \\ h_{12}(x, c_{12}) \leq 0 \end{bmatrix} \qquad \text{(Disjunction)}$$

Here the search space of the original problem is divided in two. Each disjunction contains one Boolean variable and one constraint valid over the subspace defined by the disjunction. If the cardinality of the set $SD$ is greater than two it is usually necessary to apply logic propositions in addition to the disjunctions. For this example such a proposition could be

$$Y_{11} \neq Y_{12} \qquad \text{(Proposition)}$$

This proposition states that if constraint $i = 1$, in disjunction $k = 1$ is true, then the corresponding constraint for disjunction $k = 2$ cannot be true. Certainly, if the disjunctions in (Disjunction) are mutually exclusive, (Proposition) follows implicitly, but is shown here as an illustrating example of notation.

## 5.2 Relaxation and logic formulation

By removing all binary variables in (4.22) from the MINLP, and remove or rewrite all constraints containing binary variables, the resulting model formulation is simpler and can be solved by a continuous solver. However, as the convexity of the formulation cannot be verified, optimality to this solution can still not be guaranteed. Compared to a traditional NLP-relaxation where binary variables are relaxed to continuity between zero and one, the relaxation presented here might seem strange to the reader, well-travelled in the world of nonlinear optimization. Strange as it might seem, it is argued in the following that its structure can be utilized by introducing logic to the formulation, and it will serve as the root node subproblem in the later proposed solution algorithm.

Without the binary variables, many of the constraints presented in the MINLP-formulation make little sense, and are removed. These are the pressure, routing and mass balance constraints, and lower limit constraints for lift gas. Other constraints may be rewritten, e.g. the capacity constraints for the separators. In the NLP-relaxation, the possibility of continuous flow to more than one routing alternative is allowed (previously denied by binary variables and routing constraints). To obtain this property, new variables are introduced to describe continuous flow in the

routing connections. These flows will later be referred to as routing flows. Figure 5.1 illustrates the application of the new variables and a full overview of them is presented in table 5.1.

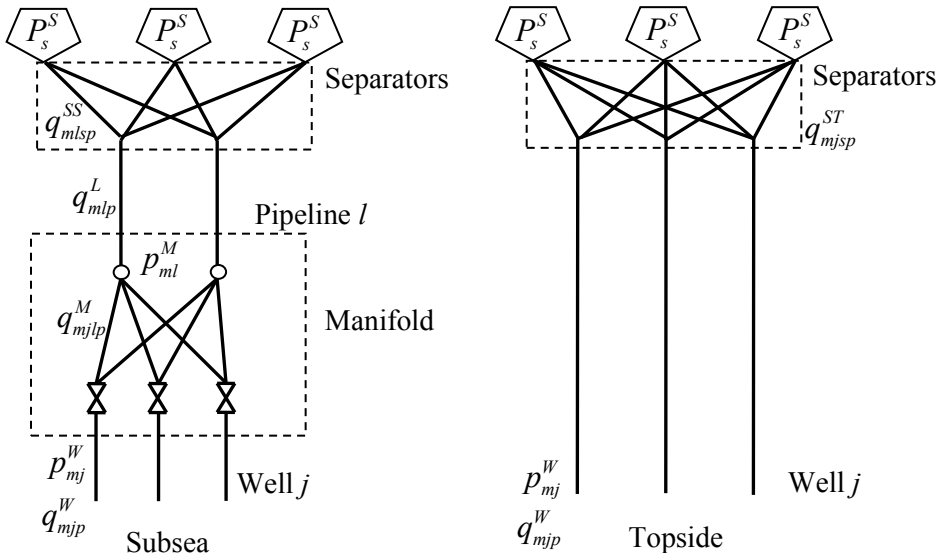


Figure 5.1: New variables

The dashed squares represents separator and pipeline routing, and previously dashed connections (from figures 4.1 and 4.2) are now replaced by solid ones. This means that flow from one component now can be routed to several components downstream the network. To the left, routing of flow from wells connected subsea is illustrated, and to the right, flow from wells connected topside.

| | | |
|---|---|---|
| $q_{mjlp}^{M}$ | - | *flow of phase p from well j to pipeline l through subsea manifold m* |
| $q_{mlsp}^{SS}$ | - | *flow of phase p to separator s from pipeline l of subsea manifold m* |
| $q_{mjsp}^{ST}$ | - | *flow of phase p into separator s from well j of topside manifold m* |
| $q_{mjl}^{I}$ | - | *flow of liftgas from well j to pipeline l* |
| $q_{mjs}^{I}$ | - | *flow of liftgas from well j to separator s* |

Table 5.1: New variables to reformulated NLP problem

### 5.2.1 New objective function

As the previous objective function contains binary variables, it is rewritten to (5.1).

$$\max Z = \sum_{m \in M^T} \sum_{j \in J_m} q_{mjo}^{W} + \sum_{m \in M^S} \sum_{l \in L_m} q_{mlo}^{L} \tag{5.1}$$

It is the exact same production measure as before, except that the binary variable in the first part is removed along with the summation over all separators.

### 5.2.2 New constraints

In the following, all changes to the model constraints are presented. Constraints added to give meaning to the new variables are also introduced through mass balances. New capacity constraints are obtained by replacing the previous product of binary and continuous variables with the new flow variables. All capacity constraints state the same as in the MINLP formulation, but through other variables. Platform gas handling capacity, previously presented by constraint (4.2), now takes the form of constraint (5.2).

$$\sum_{s\in S}\sum_{m\in M^T}\sum_{j\in J_m} q^{ST}_{mjsg} + \sum_{s\in S}\sum_{m\in M^S}\sum_{l\in L_m} q^{SS}_{mlsg} \leq C^G \tag{5.2}$$

Similarly, the same flow variables are included in the rewritten formulation of constraints (4.3). The new liquid capacity constraints are presented in (5.3).

$$\sum_{p\in o,w}\sum_{m\in M^T}\sum_{j\in J_m} q^{ST}_{mjsp} + \sum_{p\in o,w}\sum_{m\in M^S}\sum_{l\in L_m} q^{SS}_{mlsp} \leq C^{LIQ}_s, \quad \forall\, s \in S \tag{5.3}$$

The water capacity constraints of (4.4) is now written as constraints (5.4).

$$\sum_{m\in M^T}\sum_{j\in J_m} q^{W}_{mjsw} + \sum_{m\in M^S}\sum_{l\in L_m} q^{SS}_{mlsw} \leq C^W_s, \quad \forall\, s \in S \tag{5.4}$$

With the new variables follow six new mass balances that make sure that the flow of a phase in wells, pipelines, or separators equals the flow into or out of them. These balances also link the original variables to the new ones. In addition, the derivation of flow composition from chapter 4.6 is introduced in constraints (5.11) through (5.14).

Constraints (5.5) and (5.6) are reformulations of the mass balance constraints (4.12). As the flow of gas and water now are calculated from the oil flow through GOR and watercut, respectively, they are presented only for $p = o$.

Constraints (5.5) set the flow in a pipeline equal to flows from all wells connected to the same subsea manifold into that pipeline.

$$q^L_{mlo} = \sum_{j\in J_m} q^M_{mjlo}, \quad \forall\, m \in M^S,\ l \in L_m \tag{5.5}$$

Constraints (5.6) state that the flow out of a well connected to a subsea manifold must be equal to the sum of the flows into pipelines from that well.

$$q^W_{mjo} = \sum_{l \in L_m} q^M_{mjlo}, \quad \forall\, m \in M^S,\ j \in J_m \tag{5.6}$$

The mass balance in constraints (5.7) demands equality between the flow in a pipeline and the sum of flows from that pipeline to all separators.

$$q^L_{mlp} = \sum_{s \in S} q^{SS}_{mlsp}, \quad \forall\, m \in M^S,\ l \in L_m,\ p \in P \tag{5.7}$$

The same argument as for subsea manifolds is applied to flow from wells connected topside. In constraints (5.8) the equality between the flow out of a satellite well and the sum of flows from that well to all separators is introduced. Also here, well flows are only defined for $p = o$.

$$q^W_{mjo} = \sum_{s \in S} q^{ST}_{mjso}, \quad \forall\, m \in M^T,\ j \in J_m \tag{5.8}$$

In the last two mass balances of constraints (5.9) and (5.10) it is made sure that the injected lift gas to a well equals the flow of lift gas from that well to all pipelines or separators for subsea and topside connected wells, respectively.

$$q^I_{mj} = \sum_{l \in L_m} q^I_{mjl}, \quad \forall\, m \in M^S,\ j \in J_m \tag{5.9}$$

$$q^I_{mj} = \sum_{s \in S} q^I_{mjs}, \quad \forall\, m \in M^T,\ j \in J_m \tag{5.10}$$

With the assumption of modeling phase flows from the wells with constant GORs and watercuts, the gas and water fractions of the flows are now calculated from the new variables. This is to maintain a reasonable flow composition when flows from wells are split to pipelines or separators. The reformulation below implies that the fractions of phases gas and water are calculated in the manifold flows for wells connected subsea, between wells and pipelines. The fractions in flow from wells connected topside are calculated in the manifold routing between wells and separators.

In constraints (5.11) the flow of gas from a well to a pipeline, routed through a subsea manifold, is calculated by the oil flow routed the same way multiplied by the GOR for that well. The amount of lift gas injected in the well, and routed to the same pipeline, is also added.

$$q^M_{mjlg} = q^M_{mjlo}\, GOR_{mj} + q^I_{mjl}, \quad \forall\, m \in M^S,\ j \in J_m \tag{5.11}$$

The water fraction in the production flow from a well is calculated in a similar way, and formulated in constraints (5.12) as the oil flow multiplied by its relationship to the watercut of the well.

$$q^M_{mjlw} = \frac{q^M_{mjlo}\, WC_{mj}}{1 - WC_{mj}}, \quad \forall\, m \in M^S,\ j \in J_m \tag{5.12}$$

For the separator routing for flows from satellite wells the fractional flow of gas and water is handled the same way as for subsea manifolds and are formulated in constraints (5.13) and (5.14), respectively.

$$q^S_{mjsg} = q^S_{mjso}\, GOR_{mj} + q^I_{mjs}, \quad \forall\, m \in M^T,\ j \in J_m \tag{5.13}$$

$$q^S_{mjsw} = \frac{q^S_{mjlo}\, WC_{mj}}{1 - WC_{mj}}, \quad \forall\, m \in M^T,\ j \in J_m \tag{5.14}$$

All new flow variables are continuous and nonnegative, as stated in constraints (5.15). The variables are indexed over subsets of manifolds, wells, and pipelines.

$$\begin{aligned} q^M_{mjlp} &\geq 0, \quad \forall\, m \in M^S,\ j \in J_m,\ l \in L_m,\ p \in P \\ q^{SS}_{mslp} &\geq 0, \quad \forall\, m \in M^S,\ l \in L_m,\ s \in S,\ p \in P \\ q^{ST}_{mjsp} &\geq 0, \quad \forall\, m \in M^T,\ j \in J_m,\ s \in S,\ p \in P \end{aligned} \tag{5.15}$$

In the above NLP-formulation there are no constraints for lower lift gas limits for the wells, or pressure constraints downstream the production lines. This allows for physically impossible flows, and infeasible flow routing and lift gas injection (infeasible solutions to the original problem) are allowed in all parts of the system. The following section introduces conditions that cope with infeasible flow routing by denying all other routing alternatives for a well or pipeline, and implement pressure constraints, once one alternative is chosen.

### 5.2.3 Logic variables and constraints

To deny all infeasible routing of flow, conditions that restore the decision properties of the binary variables, without complicating the NLP-relaxation, are needed. Through logic conditions based on GDP, jointly exhaustive and mutually exclusive disjunctions that facilitate this without the help of binary variables are introduced.

Three Boolean variables and three disjunctive constraints that regulate routing in the production network are formulated below. Pairwise, a variable and a constraint is added for subsea manifold routing, separator routing from pipelines, and separator

routing from satellite wells, respectively. These parts in the production network are later referred to as levels of routing, or routing levels. For the subsea manifolds, the Boolean variable $Y_{mjl}$ controls the routing from wells to pipelines.

$$Y_{mjl} = \begin{cases} Y_{mj1} - \textit{true if well } j \textit{ is routed to pipeline 1 of manifold } m \\ Y_{mj2} - \textit{true if well } j \textit{ is routed to pipeline 2 of manifold } m \end{cases}$$

$$Y_{mjl} = \{true, false\}, \quad \forall\, m \in M^S,\ j \in J_m,\ l \in L_m$$

With the use of $Y_{mjl}$, the disjunctive constraints (5.16) are presented. They state that flow from a well connected to a subsea manifold can only be routed to one or no pipelines connected to the same manifold. If a well is routed to a pipeline, i.e. the Boolean variable for this connection is true, the constraints associated with this disjunction become valid. This means that a pressure constraint is added for this connection only, and a lower limit constraint for lift gas is added for the current well. Flow to any other pipeline is denied. As the lower limit for lift gas is introduced to the system as a well constraint, both disjunctions with flow contain the same lift gas constraint. The formulation $\neg Y_{mjl}$ means that $Y_{mjl} = false$ and well $j$ of manifold $m$ is not routed to pipeline $l$. In the third disjunction, the well is not routed to either pipeline, and the well is closed. As no wells of the system are free flowing, the last disjunction also set the flow of lift gas injected into the current well to zero. Constraints (5.16) is only defined for $p = o$ as flow of the other phases are derived from the oil flow inside the manifold.

$$\begin{bmatrix} Y_{mj1} \\ q_{mj2o} = 0 \\ p^W_{mj} \geq p^M_{m1} \\ q^I_{mj} \geq Q^{I,MIN}_{mj} \end{bmatrix} \bigvee \begin{bmatrix} Y_{mj2} \\ q_{mj1o} = 0 \\ p^W_{mj} \geq p^M_{m2} \\ q^I_{mj} \geq Q^{I,MIN}_{mj} \end{bmatrix} \bigvee \begin{bmatrix} \neg Y_{mj1} \wedge \neg Y_{mj2} \\ \sum_{l \in L_m} q_{mjlo} = 0 \\ q^I_{mj} = 0 \end{bmatrix}, \quad \forall\, m \in M^S,\ j \in J_m \tag{5.16}$$

For the routing of pipelines to separators, $X^S_{mls}$ is introduced.

$$X^S_{mls} = \begin{cases} X^S_{ml1} - \textit{true if separator 1 is open for routing from pipeline } l \\ X^S_{ml2} - \textit{true if separator 2 is open for routing from pipeline } l \\ X^S_{ml3} - \textit{true if separator 3 is open for routing from pipeline } l \end{cases}$$

$$X^S_{mls} = \{true, false\}, \quad \forall\, m \in M^S,\ l \in L_m,\ s \in S$$

Also for the routing from pipeline to separator a constraint with three disjunctions is introduced. Even though this routing level seemingly consists of four routing

possibilities, one for each of the three separators and one for routing to no separators, we present only disjunctions with possibility of flow. This is based on the system limitation that if there is flow to a pipeline, flow cannot be stored and must be processed through the separators (see constraints (4.17) of the MINLP model). Due to the third disjunction in (5.16) all flow can be shut down at manifold level and the need to stop production at separator level becomes redundant. When no flow is present, no pressure drop is needed to get the flow through the system. The logic separator routing constraint for pipelines connected to a subsea manifold is formulated in (5.17).

$$\begin{pmatrix} X^S_{ml1} \\ q^{SS}_{ml2p} = 0 \\ q^{SS}_{ml3p} = 0 \\ p^S_1 \leq p^L_{ml} \end{pmatrix} \bigvee \begin{pmatrix} X^S_{ml2} \\ q^{SS}_{ml1p} = 0 \\ q^{SS}_{ml3p} = 0 \\ p^S_2 \leq p^L_{ml} \end{pmatrix} \bigvee \begin{pmatrix} X^S_{ml3} \\ q^{SS}_{ml1p} = 0 \\ q^{SS}_{ml2p} = 0 \\ p^S_3 \leq p^L_{ml} \end{pmatrix}, \quad \forall\, m \in M^S,\ l \in L_m,\ p \in P \tag{5.17}$$

The last of the Boolean variables, $X^T_{mjs}$, is presented for the routing of wells connected to the topside manifold to the separators.

$$X^T_{mjs} = \begin{cases} X^T_{mj1} - \textit{true if separator 1 is open for routing from well } j \\ X^T_{mj2} - \textit{true if separator 2 is open for routing from well } j \\ X^T_{mj3} - \textit{true if separator 3 is open for routing from well } j \end{cases}$$

$$X^T_{mjs} = \{true, false\}, \quad \forall\, m \in M^T,\ j \in J_m,\ s \in S$$

In (5.18) four disjunctions are presented as the routing possibilities for wells connected topside to the separators. As there is no other routing level to shut down flow for the satellite wells, the possibility of no flow is included in this separator routing. Concerning lift gas, the same concepts applies here as for the subsea manifold routing. Also constraints (5.18) are defined only for $p = o$.

$$\begin{pmatrix} X^T_{mj1} \\ q^{ST}_{mj2o} = 0 \\ q^{ST}_{mj3o} = 0 \\ p^S_1 \leq p^W_{mj} \\ q^I_{mj} \geq Q^{I,MIN}_{mj} \end{pmatrix} \bigvee \begin{pmatrix} X^T_{mj2} \\ q^{ST}_{mj1o} = 0 \\ q^{ST}_{mj3o} = 0 \\ p^S_2 \leq p^W_{mj} \\ q^I_{mj} \geq Q^{I,MIN}_{mj} \end{pmatrix} \bigvee \begin{pmatrix} X^T_{mj3} \\ q^{ST}_{mj1o} = 0 \\ q^{ST}_{mj2o} = 0 \\ p^S_3 \leq p^W_{mj} \\ q^I_{mj} \geq Q^{I,MIN}_{mj} \end{pmatrix} \bigvee \begin{pmatrix} \neg X^T_{mj1} \wedge \neg X^T_{mj2} \wedge \neg X^T_{mj3} \\ \sum_{s \in S} q^{ST}_{mjso} = 0 \\ q^I_{mj} = 0 \end{pmatrix}, \quad \forall\, m \in M^T,\ j \in J_m \tag{5.18}$$

The logic constraints and variables formulated above represent specific constraints for the problem at hand, where there are two pipelines per subsea separator and three separators. This gives the number of routing alternatives for each well or pipeline and thus, the number of disjunctions per constraint. This can with little effort be expanded to include more alternatives, i.e. more wells, pipelines, and separators. If desirable, similar problems can also be presented more general as in the GDP presented previously in this chapter. It is however kept here in this state as it illustrates the exploitable structure of the disjunctions well. When considering the new constraints and variables, it is important to distinguish between the binary variables used in the MINLP-model and the Boolean variables presented here. $Y_{mj}$, $X^S_{mls}$, and $X^T_{mjs}$ are strictly auxiliary variables, used to relate constraints to certain parts of the solution space, i.e. the disjunctions. Propositional logic constraints, as $\Omega(Y)$ in the GDP-problem from section 5.1, can be formulated for relationships between the Boolean variables. However, as the disjunctions are formulated as dichotomies and written in their complete form, such logic is implicitly present in the disjunctions because if one of them is true, the others are false.

By including the logic conditions formulated above to the NLP-relaxation, the desired logic reformulation of the MINLP-model is completed. In the relaxation, binary variables were removed to allow continuous flow in all routing levels, i.e. infeasible with respect to the original model. The logic conditions force the flow to feasibility when including associated constraints as different routing alternatives are chosen. The logic model formulation can be found in its entirety in Appendix A.3. It makes the foundation of the specialized branch and bound algorithm presented in the following chapter. The NLP-relaxation creates an upper bound to the problem, while nodes are made through the logic disjunctions.

# Chapter 6

# Algorithm

In this chapter, the main goal is to derive an efficient solution algorithm to the RTPO-problem, by using the reformulated model. First, some introduction to the use of branch and bound (from here on called BB) and how it can be coupled with a logic implementation scheme is presented, before a specialized branch and bound algorithm is derived. Then, some important characteristics of the implemented algorithm are discussed in detail, to demonstrate some of the advantages of the problem structure and the flexibility that lies within it. Branching criteria, problem specific branching levels, and branching strategies are presented and illustrated by conceptual pseudocodes, and the relationship between them discussed. Next, some important synergies between the problem structure and the solution method that can be utilized in the algorithm are pointed out and shown in a restructuring of the disjunctions. At the end of the chapter, a complete structural overview of the implementation of the algorithm is presented, to demonstrate how the algorithm is built up.

## 6.1 Branch and bound

This section outlines a general framework of the standard BB method, before its differences to logic BB methods are highlighted. Then the structure of the envisioned specialized BB algorithm, from here on called the logic branch and bound algorithm (LBB), is derived.

### 6.1.1 Standard branch and bound

BB is generally a solution method for solving integer programming (IP) problems. It divides the search space of the original problem into smaller subspaces and solves a relaxed problem, called a subproblem, in each subspace. The aggregated information from all subproblems can give an optimal solution to the original problem. As a relaxation of the original problem is solved in each subproblem, the solution to

the subproblem when solved to optimality provides an upper bound to the original problem (when solving a maximization problem). When a solution to a subproblem yields a feasible solution to the original problem, a lower bound is generated.

BB exploits the structure of an IP problem by only enumerating subproblems that promise good solutions. This means that subproblems that provide solutions worse than the obtained lower bound to the problem are pruned, i.e. not investigated further. This way, the method implicitly enumerates all subproblems. A subproblem is created by adding new constraints to the relaxation or by fixing variables. A search tree (or BB tree) provides information about fixed variables and added constraints in each subproblem, together with solution information about the subproblems. Each subproblem is presented by a node and the tree is expanded successively by branching on these, i.e. adding additional constraints to a node. The first branch is done in the root node (usually the LP-relaxation of the IP) and thus, the whole tree expands from this node. If the LP solution contains fractional values of relaxed variables, the solution method chooses one of them and makes two branches, one where the variable is zero and one where it is one. A node is called the parent node to all nodes that expand directly from it, and any node (except the root node) is called a child node of its parent. An example of a search tree is given in figure 6.1.

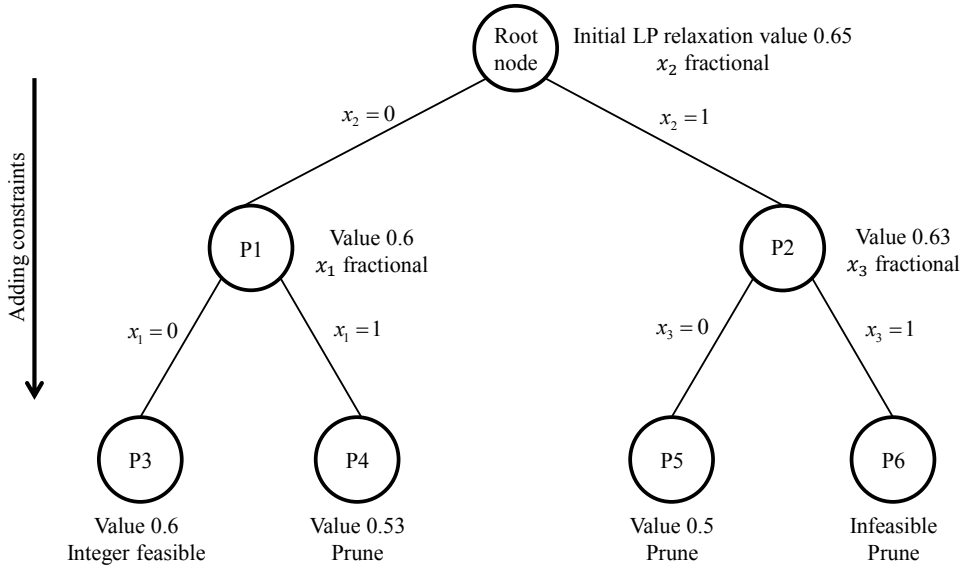


Figure 6.1: Example of a standard BB search tree

As the root node is solved, a first upper bound to the problem is found. This solution is however not integer infeasible and contains a fractional value of the integer variable $x_2$. This variable is then branched on into two new nodes, $P1$ and $P2$, containing new constraints $x_2 = 0$ and $x_2 = 1$, respectively. When solving the two new nodes fractional values of other integer variables occur. The solution method continues to branch and makes two new child nodes from each parent. Node $P3$ yields an integer feasible solution and provides the original problem with a lower bound to the problem. Nodes $P4$ and $P5$ provides solutions lower than the lower bound of the problem. This means that a better solution cannot be obtained by further investigation of

any of these nodes. A better solution cannot be obtained by adding constraints to a problem that already has a value lower than that of an integer solution of the original problem, and both nodes are pruned. Considering the last node of the BB tree, $P6$, an infeasible problem has occurred, and also this node is pruned. Thus, the optimal solution to the original problem is found in node $P3$.

In general, as the BB algorithm moves deeper down the tree, more constraints are added and tighter bounds may be found, i.e. the algorithm finds solutions closer to the optimal objective value of the original problem. The efficiency of a BB algorithm may depend greatly on the quality of the bounds it is able to produce, as this limits what parts of the search tree that does not need explicit enumeration and can be pruned. A BB algorithm should terminate when no nodes can produce a solution better than a known integer solution (as in the trivial example above), or by the following termination criterion for a maximization problem.

$$\frac{UBD - LBD}{LBD} \leq \epsilon$$

Here $UBD$ is the upper bound of the problem, $LBD$ the lower bound and $\epsilon$ a parameter chosen by the decision maker, with a typical value of $\epsilon = 0.01$ (Lundgren et al., 2010).

For most practical applications of the BB method there exist more than one fractional variable in the solution of a subproblem and more than one branching possibility. A criterion by which the possibilities is prioritized must be implemented in the solution algorithm. In general one can branch on both the largest or smallest fractional part, depending on the search strategy of the algorithm.

When choosing which node to evaluate next, different strategies called search strategies, can be implemented. They describe different orders of node enumeration in the search tree. Depth first branches on the node deepest in the tree. This usually gives a feasible solution relatively fast. Breadth first solves all nodes at a level of the tree before it moves on to branching in the next level. This strategy is often used where few constraints are preferable in the subproblems. Best first branches on the enumerated node with the most promising upper bound, i.e. best solution to the relaxed subproblems.

### 6.1.2 Logic branch and bound

When considering logic BB, several parallels can be drawn to the classic case. Both methods require a relaxed formulation of the original problem to branch on, and in both cases the solution to the relaxation serves as an upper bound to the original problem. Classic BB, as outlined above, relaxes an IP- or MIP-problem by making the binary variables continuous between zero and one. By doing this, a LP-relaxation is obtained and branching is done by fixing one variable at a time to zero and one, each representing a branch in the tree, i.e. two new nodes are generated when another

is branched on. In the logic case, the number of branches from a node depends on the number of disjunctions in the current logic constraint. Referring to the constraints (5.16), (5.17), and (5.18), representing each of the routing levels in the model, they yield three, three, and four new nodes when branched on, respectively. As these contain no binary variables they facilitate branching directly on the continuous variables by utilizing the structure of the disjunctions. In both methods pruning can be done, comparing a node's objective value to the obtained bounds. This is true for LPs and some NLPs, but when handling nonconvexities, caution should be exercised when deriving upper bounds. As is clear by now, the two methods have both similarities and dissimilarities, but they share the most important property of BB; aggregating information of solutions of simpler problems, solved over subspaces of the original feasible space. This information is in both cases used to construct an optimal solution to the original problem.

So far, notational and structural differences between traditional and logic BB have been pointed out. Another important point is that of solvability. There exist many solvers able of solving both LPs and NLPs formulated algebraically by the use of BB. On the other hand, no general solver exists able to read the logic structure of a GDP. Thus, a non-trivial degree of programming and customization is required when solving GDPs and logic structures.

## 6.2 LBB algorithm

In this section, a conceptual outline of the logic branch and bound (LBB) algorithm, designed for solving the RTPO problem of the example case of P35, is presented. The algorithm is carefully developed and customized to utilize the logic and problem specific structures introduced in the previous chapters, and a pseudocode is presented to describe its main purpose. More detailed descriptions of some parts of the algorithm follows in section 6.3.

### Conceptual outline

Algorithm 1 illustrates the main principles of the suggested LBB algorithm. It begins by solving the NLP relaxation presented in section 5.2 in its root node. This node contains no pressure, routing, or lower limit lift gas constraints and finds the upper bound to the problem. Forbidden flows are then iteratively restored to feasibility with respect to the original problem by branching on and solving nodes. Generally, a queue of problems is made for the algorithm to solve, and it will continue to do this until the queue is empty, and the optimal solution hopefully is found. The queue starts with the root node, and extends as new subproblems are added. Each subproblem is presented by a node in the search tree. If the solution to a node is either infeasible or yields a solution lower than the current lower bound for the problem, the node is pruned. No more nodes lower in the tree with connection to this node will be investigated. If the solution to a node is integer feasible with respect to

the original problem, i.e. all routing, pressure and lift gas constraints are complied with, and the solution is higher than the current lower bound, the lower bound of the problem is updated. On the other hand, if a continuous NLP solution with a promising objective value is obtained, new subproblems are added to the queue as the node is branched on. In either case, after one of these scenarios has occured, a new node from the queue is selected, solved, and removed from the queue. Since the algorithm continues until the queue of problems is empty, all generated nodes or subproblems will be explored.

**Algorithm 1** Main algorithm

**Start:** Solve $Node = RootNode$ (NLP-formulation): $\overline{z} = z_{NLP}$, $\underline{z} = -\infty$.
**Initialize** queue of nodes $ActiveNodes = Node$
**repeat**
  **if** subproblem in $Node$ is infeasible OR $z(Node) \leq \underline{z}$ **then**
    Prune $Node$.
  **else if** subproblem in $Node$ is unallowed(wrt. routing/LG/pressure) **then**
    Branch $Node$.
    Include set of nodes $LeafNodes(Node)$ in $ActiveNodes$.
  **else**
    Integer feasible, allowed solution found.
    **if** $z(Node) \geq \underline{z}$ **then**
      $\underline{z} = z(Node)$.
    **end if**
  **end if**
  Pick new $Node$ from $ActiveNodes$.
  Solve $Node$.
  Remove $Node$ from $ActiveNodes$.
**until** $ActiveNodes = \emptyset$

It is in general hard to predict the outcome of the proposed algorithm because of nonconvexities and the structure of the problem. The solution time of the LBB algorithm will strongly depend on the quality of the bounds it is able to produce. This may in turn be highly correlated to the NLP-solver used to calculate the NLP subproblems. As the NLP solved in each node might be nonconvex, safe upper bounds cannot be updated each time one of them is solved. In fact, as long as global optimality cannot be guaranteed for the solutions of the subproblems, a valid upper bound might never be found. This means that one cannot with certainty give upper bounds to the original problem, and the LBB algorithm relies solely on the quality of the feasible solutions found, i.e. the lower bounds of the problem. Even though the nonconvexities preclude the verification of a global optimal solution, one can guarantee the feasibility of a solution with respect to the original problem, and thus also the quality of the lower bounds the algorithm provides. The presented solution approach will remove parts of the problem that contribute to the nonconvexities through elimination of the binary variables, and could theoretically be able to find better solutions than that of a MINLP-solver, if they exist. The LBB algorithm also contains user flexibility that may be utilized to make it able to solve a given problem

faster. In the following section some of the flexible modules of the algorithm are discussed in detail.

## 6.3 Algorithm configuration

Between the main features of Algorithm 1, there are several aspects to consider that might prove important for the success of the solution method. Not only must certain parts be explicitly defined to get a BB algorithm to work at all, but the LBB algorithm facilitates user flexibility in combining several vital building blocks in different configurations. This allows detailed knowledge of the problem and the production system to be utilized to the full extent. As the configuration of these modules may prove paramount in finding a good or optimal solution swiftly, some technicalities are elaborated on and presented in this section. Three main factors must be considered when implementing the LBB. First, a decision must be made on what to branch on, i.e. a suitable branching criterion must be defined. Second, one must consider where in the production network to apply the branching criterion, i.e. in what order the different routing flows (illustrated in figure 5.1) should be considered. Third, an effective search strategy must be defined, i.e. a general rule of which node to evaluate next.

### 6.3.1 Branching criteria

When a relaxed node is solved, the solution may contain several flows physically forbidden by the actual production system. To be able to eliminate such flows and continue the algorithm by successively adding new constraints to formulate new subproblems, some branching criterion must be included in the algorithm. When faced with a problem with as many decision variables as in the example case of this thesis, there can be many seemingly appropriate branching criteria. As described in section 6.1.1, a common branching criteria in standard BB is related to the fractional values of binary variables for IPs and MIPs. However, the reformulated MINLP presented in this thesis contains no binary variables, and a completely different criterion based on some measure of the value of continuous variables must be chosen. The criterion should force the algorithm in the direction of the optimal solution as fast as possible, as achieving good solutions early may prove beneficial for the ability to prune nodes and thus reduce the size of the search tree.

A huge number of different branching criteria could easily be made for the LBB algorithm, but only a selection of them is presented, as an attempt to test all of them would be time consuming and unnecessary. It is however relevant to include branching criteria of different properties to explore the breadth of possibilities, but the chosen criteria must also be considered promising in finding good solutions fast. As the problem's objective function maximizes oil production, it seems reasonable to relate the criterion to some kind of measure of the oil flows of the system, either

the flow out of the wells, $q^W_{mjo}$, or the routing flows, such as flow between pipelines and separators, $q^L_{mlso}$. Since physical limitations force the wells of the production system to only produce when lift gas is injected into them, a branching criterion dependent on the lower limit of lift gas for a well, $Q^{I,MIN}_{mj}$, or the amount of lift gas injected, $q^I_{mj}$, might also be a good measure. In any case, several alternatives must be implemented and evaluated to get an overview of the impact of different branching criteria on the algorithm.

In table 6.1 seven branching criteria are presented. They can roughly be divided in two groups; one branching on the oil flows of the system and the other on amounts of lift gas injected in the wells. The former is thus applied on outputs of the system and the latter on inputs to it. They are all chosen as they are hoped to find optimal output or input allocation fast, or alter the solution of the root node as little or as much as possible when successively branching on it.

| Criterion | Choosing what | Reason |
|---|---|---|
| *Lowest* | Lowest routing flow | Little alteration to relaxed solution |
| *Highest* | Highest routing flow | Fix dominating flows from relaxed solution early |
| *Most equal* | Routing with most equally distributed flows | Much alteration to obtain feasible solution |
| *Most unequal* | Routing with most unequally distributed flows | Little alteration to obtain feasible solution |
| *Flow to LG ratio* | Well with highest ratio oil flow to lift gas lower limit | Use little lift gas per unit oil flow |
| *Max gap LG* | Well with highest difference lift gas flow and lower limit | Identify wells that should be closed |
| *Min gap LG* | Well with lowest difference lift gas flow and lower limit | Identify wells that should be kept open |

Table 6.1: Branching criteria

Lowest, Highest, Most equal, and Most unequal are the branching criteria applied to oil flows of the system, i.e system outputs. Lowest is implemented to identify the lowest forbidden oil flow of the relaxed system. The assumption is that by closing the smallest forbidden flow of a component, the solution of the root node is altered as little as possible. This again is expected to derive good or optimal solutions fast. Highest on the other hand prioritizes and chooses the biggest forbidden routing flow of the system. The underlying assumption is that the biggest flows of the relaxation also will be dominating in the optimal solution and that they should be fixed early to find the optimal solution as fast as possible. Most equal is evaluated as it identifies the relaxed flows that demand most alteration to eventually obtain a feasible solution to the original problem, i.e. it finds the relaxed component with the most evenly distributed routing flows. Most unequal is similarly applied as it potentially identifies relaxed flows that need the least alteration to hopefully find a feasible solution.

While the former consider outputs of the system alone, Flow to LG ratio, Max gap LG, and Min gap LG consider inputs, i.e the injection of lift gas, or a combination of parameters and outputs. Flow to LG ratio is the ratio between oil flow from a well and the lower limit of lift gas injection to that well. This branching criterion is applied based on the assumption that it will find the optimal lift gas allocation of the problem fast by identifying wells with the most "oil per unit lift gas". Max gap LG identifies the wells with lift gas amounts that in a relaxed solution lie furthest from their allowed interval, and thus may be reasonable to close. Similarly, Min gap LG identifies the wells with lift gas amounts that in a relaxed solution lie closest to their allowed interval, and thus may be reasonable to keep open in an optimal solution. All of the lift gas branching criteria are applied to find a good lift gas allocation fast.

To illustrate the process of choosing based on branching criteria, Algorithm 2 shows the basic steps for an implementation of criterion Highest, where the well with the highest routing flow is chosen.

**Algorithm 2** Branching criterion - highest well flow

```
Solve Node.
for WELLS do
    if Forbidden routing flow in Well then
        Store flow of Well.
        if Flow of Well ≥ Highest forbidden routing flow then
            Store Well.
        end if
    end if
end for
Branch on Well.
```

### 6.3.2 Routing levels

When a suitable branching criterion is chosen, the order of where in the system to apply it becomes a decision in itself. As the system consists of flow between different components with varying flow characteristics and the subproblems are nonconvex, the right choice is far from evident. The choices for the user in the implemented LBB is to consider the whole system as one, branch in a predetermined order of the routing levels introduced in section 5.2, or merge some of the levels and consider parts of the system together. In either case, the logic constraints in (5.16), (5.17), and (5.18) are used iteratively to generate new nodes and expand the search tree. When considering the whole system as one, some sort of average weighing of the production flows could be useful to implement when choosing what to branch on, as the system components has varying dimensions and lead flows of different magnitude. Examples of this are criteria *Most equal* and *Most unequal* in table 6.1. When considering predetermined orders of the branching levels such weighing is not relevant as it simplifies the algorithm to consider only one level at the time. In

table 6.2 an overview of all branching level orders, called branching alternatives, are presented.

| | 1. | 2. | 3. |
|---|---|---|---|
| *Alt. 1* | Well to pipeline | Pipeline to separator | Well to separator |
| *Alt. 2* | Well to separator | Well to pipeline | Pipeline to separator |
| *Alt. 3* | Well to pipeline | Well to separator | Pipeline to separator |
| *Alt. 4* | Pipeline to separator | Well to pipeline | Well to separator |
| *Alt. 5* | Pipeline to separator | Well to separator | Well to pipeline |
| *Alt. 6* | Well to separator | Pipeline to separator | Well to pipeline |
| *Well/pipe* | Well routing | Pipeline to separator | - |
| *Pipe/well* | Pipeline to separator | Well routing | - |
| *Global* | All routing possibilities | - | - |

Table 6.2: Branching alternatives

Alt.1 to 6 represent fully predetermined orders that consider well to pipeline, well to separator, and pipeline to separator completely separated and in the order following the numbers of the top row of table 6.2. Well/pipe and Pipe/well both have levels of well to pipeline and well to separator routing merged. The former first considers the routing of wells, and the latter pipeline to separator routing first. Finally, Global considers the whole system at the same time, merging all branching levels. As the continuous flows and the complexity of the production system may make testing chaotic, the routing levels constitute helpful tools in systematizing the development of an efficient solution algorithm.

When applying any of the branching alternatives in table 6.2, the solution algorithm will first branch on forbidden routing flow detected in the first column of the table for the alternative. When no more infeasible routing is detected in this column, the algorithm will succeed to the next. The algorithm will successively add constraints corresponding to necessary columns and levels until no infeasible flows are detected anywhere in the system, and a feasible solution is found. Any of the branching criteria from table 6.1 may be used to sort flows to branch on within the branching levels. To demonstrate how a branching alternative is implemented in the LBB, a simple overview of branching alternative 3 is outlined in Algorithm 3.

**Algorithm 3** Predetermined order of branching levels

**if** *well to pipeline* routing level contains forbidden routing **then**
  Branch on *well to pipeline* routing level.
**else if** *well to separator* routing level contains forbidden routing **then**
  Branch on *well to separator* routing level.
**else if** *pipeline to separator* routing level contains forbidden routing **then**
  Branch on *pipeline to separator* routing level.
**else**
  Integer feasible solution found.
**end if**

### 6.3.3 Search strategy

Many technicalities can be included in the LBB algorithm, but after choosing branching criteria and alternative, only the choice of search strategy remains to make it able to search the solution space of the problem with some efficiency. This means that a general rule of which node to branch on after an iteration must be chosen. In the general outline of the LBB in Algorithm 1, no such strategy is implemented in particular. All nodes added to the queue of subproblems are solved, and when a node is branched on (the first node detected with forbidden routing flow) a number of branches, equivalent to the number of disjunctions in the current logic constraint, are made from this node and the same number of nodes are added to the queue. The choice of strategy depends highly on the size and structure of the given problem, and the general nature of the algorithm. In addition, the preferred strategy may be highly dependent on different combinations of branching criteria and order of branching levels.

In the LBB implementation, depth and best first are chosen as search strategies. The breadth first strategy is not considered, as it is assumed that the payoff with respect to increased oil production when pressure, routing, and lift gas restrictions are relaxed, is so big that a nontrivial number of constraints must be added to the relaxation for feasible solutions to occur. Thus, it makes little sense to solve many weakly constrained subproblems at the beginning of the search tree, and this strategy is omitted. To avoid solving many such subproblems, the depth first strategy is investigated thoroughly, as it goes "deep" in the search tree and hopefully may be able to find possible solutions fast. Depth first also allows for a high degree of customization of the search tree. When implementing depth first, the algorithm will choose the last node added to the queue of subproblems, i.e. the last generated node, and thus the order of which the nodes are generated may prove important. Based on this, the nodes with the assumed best objective value are generated last. An educated guess is that this node will be the one that alters the relaxed solution as little as possible. For a given routing flow branching, the last node generated is the one where the highest routing flow is kept open and the smallest ones closed. The expression "educated guess" is used as it is hard to predict the outcome of a search in an unknown, nonconvex space, but based on general knowledge of the problem at

hand and experience with BB one can hope to make a better attempt than random chance.

Best first is in many cases considered as the most effective, and thus preferred search strategy. In this case, on the other hand, the problem consists only of continuous variables that may take marginally different values to provide slightly varying solutions. This may induce a huge number of nodes with objective values higher than that of the optimal solution, and the best first strategy may prove ineffective as it must evaluate all of them. This, and the extra degree of specialization possible for depth first, makes depth first the assumed best strategy for the problem at hand. It will later be applied to several configurations of the LBB, while best first will be tested on the configurations proven successful for depth first. Algorithm 4 gives a simple overview of how the depth first strategy is implemented in the LBB.

**Algorithm 4** Search strategy - depth first

```
Parameter NewNode = 0.
repeat
    Choose Node with highest node number that has not been solved.
    if Node has valid node number then
        if z(Parent(Node)) ≤ z̲ then
            Prune Node.
        else
            Parameter NewNode = 1.
            Continue algorithm with Node.
        end if
    else
        Parameter NewNode = 1.
        No more nodes to evaluate.
        Terminate algorithm.
    end if
until NewNode = 1
```

In light of the presented properties and the pseudocodes of Algorithms 2, 3, and 4, the LBB implementation, simply described in the main algorithm in section 6.2, can be summarized. When a node is solved and the subproblem solution contains forbidden routing, pressure, or lift gas injection, branching is done and a choice of branching alternative from table 6.2 is included to decide where in the system to branch. Combined with the branching alternative a suitable branching criterion of table 6.1 is chosen, to effectively evaluate which flow in the given level to branch on. New nodes can then be made and added to the set *ActiveNodes*. A new node is picked from this set, e.g. with the depth first strategy where the node with the highest node number is chosen. If best first is implemented, the node with the highest objective value is chosen. If the parent of the node displays a promising objective value it is selected as the new node to branch from. Then it is solved and removed from *ActiveNodes*. When the subproblem solved is either infeasible, unpromising, or

displays a solution feasible to the original problem the node is pruned. No branching occurs and a new node is picked directly. The process is repeated iteratively until feasible solutions and eventually the optimal solution is found, and the queue is empty.

Needless to say, there are numerous possibilities when including different versions of the parts of the algorithm illustrated in this chapter. A number of different branching alternatives can be combined with several branching criteria, which again can be combined with search strategies. To limit the number of possible algorithm configurations, this thesis will include only combinations of the alternatives, criteria, and strategies presented in this section. A wide selection of properties has been chosen to include different scenarios that may have an impact on the algorithm. These should provide a sufficient basis for testing the efficiency of the LBB.

## 6.4 Algorithm and model synergy

Utilization of problem specific knowledge is a vital part of this thesis as it is believed to enhance the solution efficiency of the LBB considerably. Some structures of the logic problem formulation presented in section **??** that may enable such utilization are highlighted and reformulated in this section. This is done to introduce additional flexibility to the algorithm, without altering the physical properties of the problem structure.

#### Restructuring well disjunctions

The purpose of the logic constraints (5.16), (5.17), and (5.18), introduced in section 5.2, is to restore feasibility to the NLP-formulation with respect to all the relaxed constraints of the MINLP. Since the disjunctions of these constraints represent all branches made and all constraints added to the subproblems of the LBB, it is obvious that they are crucial to the development of the BB tree. In addition, the limitation that denies wells to produce unless being supplied with lift gas within defined intervals complicates the problem as semi-continuous flow variables are introduced. This is a problem characteristic with potentially great impact on the system that may be computationally expensive to handle when implemented in a general MINLP-formulation. For the LBB on the other hand, the correlation between lift gas and producing wells might be utilized to find optimal lift gas allocation early and thus reduce the size of the search tree significantly. As lift gas constraints currently are added to subproblems along with routing and pressure constraints, a restructuring of disjunctive constraints (5.16) and (5.18), isolating lift gas decisions to a separate branching level, is presented below.

Since the injection of lift gas and the configuration of opened wells are factors considered to be highly related to the success of the algorithm, the restructuring constitutes the possibility of regulating lift gas and closing of wells in an isolated

branching level. To do this, a new auxiliary variable must be introduced to synchronize opened wells with regulation of lift gas injection. The Boolean variable $W_{mj}^{LG,OPEN}$ controls which wells are open, and is true if a well is open and false if it is closed.

$$W_{mj}^{LG,OPEN} \quad - \quad \textit{true if well } j \textit{ of manifold } m \textit{ is open}$$

$$W_{mj}^{LG,OPEN} = \{true, false\}, \quad \forall\, m \in M,\ j \in J_m$$

With the introduction of the additional Boolean variable, the new routing level of lift gas is written as presented in constraints (6.1). The constraints consist of two main disjunctions where one implements the lower limit lift gas constraint for a certain well, and the well can produce, i.e. $W_{mj}^{LG,OPEN} = true$. The other closes the well, i.e. $W_{mj}^{LG,OPEN} = false$, and sets lift gas injection to zero. Take special notice to the second disjunction, where the well is closed, as it contains disjunctions within itself. These additional disjunctions are simply a way to distinguish between wells connected subsea and topside, as wells connected to different types of manifolds are routed to different components down stream the production system. The left one states that if the current well is connected to a subsea manifold, flows to all pipelines from the well is closed. The right one states that if the current well is a satellite well, flow to all separators from the well is set to zero. In either case, injected lift gas is set to zero and the well is closed. It should also be noted that the additional disjunctions related to closing the well does not include a Boolean variable. This is because they do not represent a choice in the algorithm, but are determined by the value of a parameter, i.e. which manifold the current well is connected to. Constraints (6.1) are after the proposed restructuring the only place in the LBB where a well can be closed and lift gas regulated.

$$\begin{pmatrix} W_{mj}^{LG,OPEN} \\ q_{mj}^{I} \geq Q_{mj}^{I,MIN} \end{pmatrix} \bigvee \begin{pmatrix} \neg W_{mj}^{LG,OPEN} \\ q_{mj}^{I} = 0 \\ \left(\sum_{l \in L_m} q_{mjlo} = 0\right) \bigvee \left(\sum_{s \in S} q_{mjso}^{ST} = 0\right) \end{pmatrix}, \qquad \forall\, m \in M,\ j \in J_m \tag{6.1}$$

After the introduction of (6.1), the disjunctions presented in section 5.2 must be rewritten as they no longer contain the possibility of closing a well or regulating its lift gas consumption. A well connected subsea can only be routed to one of the pipelines, and constraints (5.16) are rewritten to (6.2). A satellite well is routed to one of the separators, and constraints (5.18) are rewritten to (6.3). The disjunctions for pipeline to separator routing, presented in constraints (5.17), do not include either lift gas regulation or the possibility of shutting down and can be left in their original form.

$$\begin{pmatrix} Y_{mj1} \\ q_{mj2o} = 0 \\ p^W_{mj} \geq p^M_{m1} \end{pmatrix} \bigvee \begin{pmatrix} Y_{mj2} \\ q_{mj1o} = 0 \\ p^W_{mj} \geq p^M_{m2} \end{pmatrix}, \quad \forall\, m \in M^S,\ j \in J_m \tag{6.2}$$

$$\begin{pmatrix} X^T_{mj1} \\ q^{ST}_{mj2o} = 0 \\ q^{ST}_{mj3o} = 0 \\ p^S_1 \leq p^W_{mj} \end{pmatrix} \bigvee \begin{pmatrix} X^T_{mj2} \\ q^{ST}_{mj1o} = 0 \\ q^{ST}_{mj3o} = 0 \\ p^S_2 \leq p^W_{mj} \end{pmatrix} \bigvee \begin{pmatrix} X^T_{mj3} \\ q^{ST}_{mj1o} = 0 \\ q^{ST}_{mj2o} = 0 \\ p^S_3 \leq p^W_{mj} \end{pmatrix}, \quad \forall\, m \in M^T,\ j \in J_m \tag{6.3}$$

The action of releasing lift gas routing and closing of wells from the other routing levels gives an additional degree of flexibility to the algorithm while preserving the original functionality of the production system. The system now has four distinct routing levels that can be combined; routing from wells connected subsea to pipelines, represented by constraints (6.2); routing from pipelines to separators, represented by constraints (5.17); routing from satellite wells to separators, represented by constraints (6.3); and lift gas routing, represented by constraints (6.1).

The number of predetermined branching level orders does with the new branching level increase from six to twenty four, as the lift gas branching level introduces four level orders for each of the existing six presented in table 6.2. In addition, the new lift gas level could be introduced to versions of the merged branching orders. The possibilities are many, but as the reason for implementing the presented restructuring was to increase the flexibility of the algorithm to make it possible to choose lift gas level as the first level to branch on. This is justified by the assumption that the lift gas allocation is paramount in finding the optimal solution, and when derived early it can reduce the computational effort of the algorithm substantially. Following the lift gas branching level, the alternatives of table 6.2 are implemented as they are. It should also be noted that the the lift gas branching level must be applied with some of the last three branching criteria of table 6.1. Later both the original and the reformulated logic formulations are tested with flow branching criteria, and flow and lift gas branching criteria combined.

## 6.5 Implementation

In the implementation of the LBB algorithm different code files are used in order to make it operate properly. Separate model and data files are needed for loading of common problem parameters and restrictions in the beginning of the algorithm. In addition it could be wise to split up some parts of the algorithm code, such as the implementation of the branching levels, to increase the readability of the code and make it easier to implement structural changes when testing different cases. The

model file contains the relaxed NLP-formulation described in section 5.2, except for the logic disjunctions (these are implemented in the LBB). The data file includes all data and the parameters generated for the problem, such as separator capacities, lift gas limits and parameters for approximations.

A simple overview over the basic processes of the LBB-algorithm for the depth first strategy is shown in figure 6.2. (A best first implementation requires only minor modifications and runs on the same foundation.) After the model and data file has been read and loaded, new parameters, sets and constraints are defined for the LBB. These include all the needed tools for implementing the logic disjunctions, such as parameters for storing indexes on well and pipelines, sets for storing new constraints, and lists containing all nodes with information telling whether or not they have been solved or branched on. The relaxed problem is then solved (the root node) before going into the repeating BB-environment. After the initial solution is found, the code itself is a repeat loop that will go on until there are no more nodes that can improve the best objective value found, as simply described in Algorithm 1. First, a simple test on the subproblem's feasibility and its objective value is performed. If the problem is feasible and the objective value is promising, i.e. there exists a solution to the problem that is higher than the current lower bound of the problem, a test on the variable values will be executed. If it is not promising or feasible, the node will be pruned, and the algorithm jumps to searching for a new node to solve. The test on the variable values is performed searching for forbidden routing flow, pressures and lift gas amounts, and includes storing of indexes for the forbidden values based on the criterion chosen for branching. The indexes that are stored also have to be in correspondence with the branching alternative that is chosen. If the solution to the node is allowed for both routing, pressure and lift gas injection, a feasible solution to the original problem is found and the values of this solution is stored in the current node. If it is higher than the current lower bound it is also updated as the best node and new lower bound. The algorithm then prunes and searches for a new node to solve.

If some constraints of any of the logical disjunctions are broken anywhere in the system, i.e. the solution of any node is either pressure, routing, or lift gas infeasible with respect to the original problem, the algorithm continues to the branching process. This is done in separate files that are included when needed, in order to facilitate structural changes in the testing, such as swapping between the alternatives mentioned in table 6.2. There are different branching files, one for all of the routing levels presented in section 5.2 and 6.4. They can be combined to include all the routing alternatives mentioned in table 6.2. The files are included based on parameters that are set from the branching criterion and that decide where in the system the next branching will occur. When branching is done, new nodes are added to the node list. They include information about their parent and the restrictions added to the nodes to handle the broken constraints. (The former information is important to be able to properly activate the correct constraints when solving the subproblems.) After branching is done, the algorithm proceeds to searching for a new node.

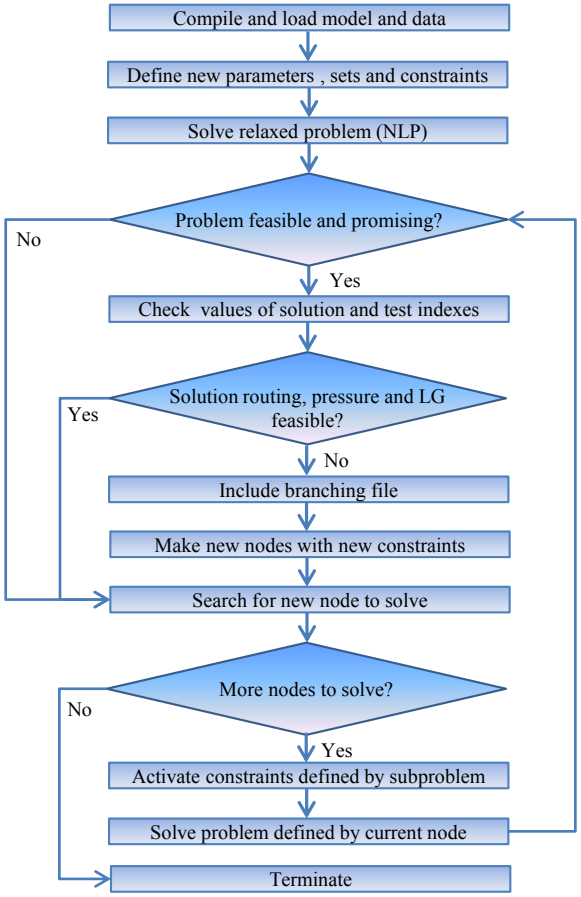


Figure 6.2: Flow chart of the LBB-algorithm implementation

The search for a new node is done by searching through the node list (which is expanding for each new iteration in the repeat loop), for the nodes that have not yet been solved or pruned. In depth first, the unsolved node with the highest node number in the list is chosen, while the node with the highest objective value is chosen in best first. When a node is chosen, the algorithm continues by dropping all currently defined constraints and activating the constraints defined by the new node. This is done by searching through all the parent nodes starting from the new node going all the way up to the root node. All constraints defined for the current node can in this way be "located" and activated, and the new subproblem is solved. The algorithm will then go up to the start of the repeat loop and check the new subproblem's feasibility and objective value before continuing the process. The algorithm terminates after the search process when there are no more nodes in the list that can be solved.

# Chapter 7

# Computational study

In this chapter, a computational study of the LBB algorithm used to solve the logic reformulation of the MINLP is presented and discussed. First, the test case of P35 and the asset's main characteristics are presented. Then, the MINLP in its complete algebraic form is solved with a suitable commercial solver and detailed results are presented. These will serve as a benchmark for later comparisons. The LBB is then implemented with different configurations of the algorithm elements from chapter 6 in the search for an effective solution algorithm. Finally, a sensitivity analysis is carried out to test the robustness of the algorithm, and the LBB is tested on an extended case and once again compared to the benchmark.

All models are implemented in AMPL, and solved using the solvers Bonmin and Ipopt. AMPL is chosen because of its simplicity and flexibility with respect to choice of solvers. Bonmin in itself utilizes different solvers and is chosen as the solver for the original MINLP, as it has shown promising records of solving other MINLPs. It cannot guarantee global optimal solutions for nonconvex MINLPs, but the results will in any case serve as a good benchmark for comparison to the proposed solution method. The comparisons can hopefully provide insight both with respect to computational results and serve as a measure of the "degree" of nonconvexity of the original problem. As all subproblems of the LBB are NLPs, only Bonmin's nonlinear solver, Ipopt, will be used in the LBB algorithm. All models are run on an Ubuntu based computer with eight Intel Core i7-2600 3.40GHz processors and 15.6 GB RAM.

The most relevant solution characteristics to consider for each implemented version of the algorithm are number of nodes enumerated, node number of best node, objective value of best solution found, and solve time. The total number of nodes gives a measure of the computational effort the algorithm demands, and the node number of the best node reveals how efficiently the algorithm is able to obtain the optimal solution. The best objective value, along with more detailed system configurations, is an obvious measure when evaluating the quality of the solution the current algorithm is able to find. When considering computational time, AMPL provides several measures which mainly can be divided in system user time and solve time. Solve

time is the actual time spent solving problems in the algorithm, while system time is the overhead time spent in the algorithm, outside the solver. As AMPL mainly is a modeling language, it is considered irrelevant to provide system time in this thesis as the algorithm could be implemented in a more suitable language, nearly eliminating the overhead. Hence, only solve time is presented for the following results.

## 7.1 Test case

The model is tested on real production data from the P35 asset, described in section 2.3. All the data presented and used has been scaled to ensure protection of sensitive information, while still being able to attain comparable results. Table 7.1 presents common platform capacities, table 7.2 the separator specific properties, and table 7.3 contains well specific values of limits for lift gas injection. The data displayed in these tables are used in the implementations of both the MINLP and the logic formulation. All values are presented with the units used in the data provided by Petrobras. Flows are given in standard cubic meters per day [$Sm^3/d$] and the pressure values are given in kilogram-force per square centimeter [$kgf/cm^2$]. One standard cubic meter is a cubic meter at standard pressure and temperature conditions, usually 288.15 K and 101.325 kPa, and 1 $kgf/cm^2$ equals approximately 98.0667 kPa.

| | | | |
|---|---|---|---|
| Gas processing capacity | [$Sm^3/d$] | $C^G$ | 2 400 000 |
| Lift gas capacity | [$Sm^3/d$] | $C^{LG}$ | 1 000 000 |

Table 7.1: Platform capacities

| | | | |
|---|---|---|---|
| Liquid handling capacity [$Sm^3/d$] | Separator 1 | $C_1^{LIQ}$ | 7 000 |
| | Separator 2 | $C_2^{LIQ}$ | 7 000 |
| | Separator 3 | $C_3^{LIQ}$ | 5 078 |
| Water handling capacity [$Sm^3/d$] | Separator 1 | $C_1^W$ | 5 000 |
| | Separator 2 | $C_2^W$ | 5 000 |
| | Separator 3 | $C_3^W$ | 2 500 |
| Separator pressure [$kgf/cm^2$] | Separator 1 | $P_1^S$ | 7 |
| | Separator 2 | $P_2^S$ | 7 |
| | Separator 3 | $P_3^S$ | 7 |

Table 7.2: Separator pressures and capacities

| Manifold | Well | Lower lift gas limit for production [$Sm^3/d$] | Upper lift gas limit for production [$Sm^3/d$] |
|---|---|---|---|
| Sub 1 | 1 | 190 000 | 212 000 |
| | 2 | 97 000 | 161 000 |
| | 3 | 180 000 | 195 100 |
| | 4 | 131 000 | 168 000 |
| Sub 2 | 5 | 95 000 | 137 000 |
| | 6 | 136 000 | 165 000 |
| Top | 7 | 120 000 | 160 000 |
| | 8 | 154 000 | 165 000 |
| | 9 | 122 730 | 156 747 |
| | 10 | 111 000 | 178 000 |

Table 7.3: Lift gas lower limits for wells

## 7.2 Algorithm benchmark

Table 7.4 shows general solution characteristics from the solution of the MINLP using Bonmin. The benchmark objective value is found in 182.3 seconds and is 9339.02 $Sm^3/d$ of oil. The solution is lift gas and pressure constrained. This means that the total lift gas constraint of (4.10) and at least one of pressure constraints (4.16) and (4.20) hold by equality. These are common binding constraints in an oil production asset, such as the P35.

| | |
|---|---|
| Oil production [$Sm^3/d$] | 9339.02 |
| Solve time [s] | 182.3 |
| Number of nodes | 6786 |
| Best node | 202 |
| Number of open wells | 8 |
| Pressure constrained | YES |
| Gas flow constrained | NO |
| Water flow constrained | NO |
| Liquid constrained | NO |
| Lift gas constrained | YES |

Table 7.4: Optimal characteristics of Bonmin MINLP solution

To be able to give a complete comparison between solutions of the MINLP and the logic formulation of the LBB, specific system configurations are presented for the benchmark solution. In addition to objective value and solution time, system information of which wells to keep open or closed, routing decisions, allocation of lift gas, and pressure configurations in the system are presented. All these results may prove relevant when comparing Bonmins' solution to that of the LBB. Table 7.5 presents phase fractions of the production flow and injected lift gas for each well associated with the optimal solution. Table 7.6 presents routing and pressure

configurations. The latter also shows which wells are pressure constrained, i.e. which of constraints (4.16), (4.18) and (4.20) that are held by equality. All flows are presented with one decimal, and pressure with three since these take smaller relative values and are more sensitive to changes.

| Manifold | Well | Oil flow [$Sm^3/d$] | Gas flow [$Sm^3/d$] | Water flow [$Sm^3/d$] | Injected lift gas [$Sm^3/d$] |
|---|---|---|---|---|---|
| Sub 1 | 1 | - | - | - | - |
| | 2 | 533.7 | 140550 | 1087.5 | 97000 |
| | 3 | 1377.4 | 290192 | 302.4 | 180000 |
| | 4 | 1320 | 257060 | 1202.5 | 131000 |
| Sub 2 | 5 | 1201.9 | 199625 | 884.7 | 102270 |
| | 6 | 2256 | 295499 | 207.7 | 136000 |
| Top | 7 | 1337.9 | 225293 | 1162.4 | 120000 |
| | 8 | - | - | - | - |
| | 9 | 757.8 | 194873 | 1190.3 | 122730 |
| | 10 | 554.3 | 149801 | 1051.4 | 111000 |

Table 7.5: Well characteristics of optimal solution

| Manifold | Well | Well head pressure [$kgf/cm^2$] | Routed to pipeline | Routed to separator | Manifold pressure [$kgf/cm^2$] | Pressure constrained |
|---|---|---|---|---|---|---|
| Sub 1 | 1 | - | - | - | - | - |
| | 2 | 46.284 | test1 | sep3 | 46.284 | YES |
| | 3 | 49.233 | prod1 | sep2 | 42.377 | NO |
| | 4 | 44.322 | prod1 | sep2 | 42.377 | NO |
| Sub 2 | 5 | 55.810 | test2 | sep1 | 55.810 | YES |
| | 6 | 47.779 | prod2 | sep2 | 47.224 | NO |
| Top | 7 | 9.194 | - | sep3 | - | NO |
| | 8 | - | - | - | - | - |
| | 9 | 9.409 | - | sep1 | - | NO |
| | 10 | 7.985 | - | sep1 | - | NO |

Table 7.6: Routing and pressure characteristics of optimal solution

The results presented in tables 7.4, 7.5, and 7.6 seem reasonable with respect to actual characteristics of the production at the example case of P35. The system is constrained on total lift gas capacity, being able to open eight wells, of which two are constrained by the pressure configurations of downstream components. All open wells except for one are constrained by their lower limits for lift gas. It is not by chance that the well not constrained by its lower lift gas capacity is one of the pressure constrained ones, as it makes little sense to choke the valve of the well and at the same time aerate the flow by injecting more lift gas than required by the lower limit. Simply put, this can be compared to using the break and speed pedal in a car at the same time, and is why additional lift gas is injected into a fully opened, or

pressure constrained well. The choice of which of the pressure constrained wells to utilize the additional lift gas in is dependent on the function of constraints (4.11) that give the well flow as a function of injected lift gas and the wellhead pressure.

## 7.3 Results from the LBB algorithm

As described in chapter 6, there are many ways of combining elements of the structure in the presented LBB algorithm to enhance its solution efficiency. As the LBB solves nonconvex NLPs and its solution space is so intangible, it seems that an efficient configuration of the algorithm only can be found by the method of trial and error. But by utilizing detailed knowledge of the problem at hand it is possible to systematize the search of a comprehensive and robust algorithm. In this section several versions of the LBB is tested and then the most promising versions are explored more thoroughly. Finally, the effectiveness and robustness of the complete algorithm is tested and compared to its benchmark.

### 7.3.1 Disjunctions including lift gas constraints

Here the algorithm is tested in its most basic form, with regulation of flow and lift gas integrated in the disjunctive constraints. That is, all results presented in table 7.7 are derived from branching on logic constraints (5.16), (5.17), and (5.18) from section 5.2. This is combined with several of the oil flow branching criteria and level orders from chapter 6. The argumentation behind implementing the algorithm in its basic form is to fix production configurations based on output flows of the system, i.e. the oil flows, and that the lift gas is regulated as a result of the system's routing configuration. As lift gas regulation is implemented in the disjunctive constraints of the algorithm, only the oil flow branching criteria of table 6.1 are tested here. The algorithm is currently tested with the depth first strategy and all versions of the algorithm is forced to terminate after 30 000 nodes (marked by -).

| Branching criterion | Branching alternative | Number of nodes | Best node | Best solution [$Sm^3/d$] | Solve time [s] |
|---|---|---|---|---|---|
| Lowest | Alt.1 | - | 6440 | 9213.02 | - |
| | Alt.2 | - | 15901 | 8607.07 | - |
| | Alt.3 | 10925 | 8358 | 9339.02 | 748.2 |
| | Alt.4 | - | 1596 | 9339.02 | - |
| | Alt.5 | - | 11969 | 8607.07 | - |
| | Alt.6 | - | 9811 | 8363.93 | - |
| | Global | - | 28818 | 8317.44 | - |
| | Pipe/well | - | 12020 | 8607.07 | - |
| | Well/pipe | - | 16756 | 8607.07 | - |
| Highest | Alt.1 | - | 27432 | 9207.39 | - |
| | Alt.2 | 19925 | 1279 | 9339.02 | 733.1 |
| | Alt.3 | 16197 | 10904 | 9339.02 | 474.1 |
| | Alt.4 | - | 1411 | 9339.02 | - |
| | Alt.5 | - | 938 | 9339.02 | - |
| | Alt.6 | - | 20074 | 9339.02 | - |
| | Global | - | 1631 | 9207.39 | - |
| | Pipe/well | - | 1336 | 9339.02 | - |
| | Well/pipe | 16201 | 10906 | 9339.02 | 474.0 |
| Most equal | Alt.1 | - | 15641 | 9213.02 | - |
| | Alt.2 | - | 16726 | 8607.07 | - |
| | Alt.3 | 10356 | 6272 | 9339.02 | 567.3 |
| | Alt.4 | - | 782 | 9339.02 | - |
| | Alt.5 | - | 10252 | 8607.07 | - |
| | Alt.6 | - | 6193 | 8363.93 | - |
| | Global | - | 28683 | 8425.96 | - |
| | Pipe/well | - | 13188 | 8607.07 | - |
| | Well/pipe | - | 23735 | 9339.02 | - |
| Most unequal | Alt.1 | - | 26176 | 9207.39 | - |
| | Alt.2 | 25064 | 1699 | 9339.02 | 859.6 |
| | Alt.3 | 16506 | 10448 | 9339.02 | 548.4 |
| | Alt.4 | - | 1468 | 9339.02 | - |
| | Alt.5 | - | 1595 | 9339.02 | - |
| | Alt.6 | - | 28039 | 9339.02 | - |
| | Global | - | 5547 | 9339.02 | - |
| | Pipe/well | - | 946 | 9339.02 | - |
| | Well/pipe | 25070 | 1699 | 9339.02 | 878.3 |

Table 7.7: Original LBB algorithm with various branching alternatives and criteria

Considering the results of table 7.7, most versions of the LBB are not able to finish after enumeration of over 30 000 nodes. When comparing the results of the versions that are able to find the same solution as that of the benchmark of section 7.2, none of them are able to compete with Bonmin with respect to the number of nodes enumerated or computational time. The best version of the algorithm with respect to number of nodes, is currently that of flow branching criterion Most equal, combined with branching order alternative 3. It finishes after 10356 nodes and 567.3 seconds of solve time, which is 53 percent more nodes and three times the solve time as Bonmin. This might seem damaging for the motivation of further investigation of the LBB, but several important findings are made from the results presented here. Firstly, the fact that all versions of the algorithm that are able to finish find the same solution as that of the benchmark (9339.02 $Sm^3/d$) may indicate that the nonconvexities of the problem are not as nasty as expected. This may also mean that Bonmin is able to find the global optimal solution of the problem. Secondly, the LBB algorithm is able to find a solution at least as good as Bonmin. And thirdly, several versions of the LBB find the optimal solution after relatively few enumerated nodes, i.e. in a node with low node number. The third finding may be of importance when considering the restructuring of the logic constraints, as outlined in section 6.4. This restructuring is designed to utilize the problem's dependency to lift gas and to be able to find the optimal solution early and hopefully prune large parts of the search three.

In the next section, the restructuring of the disjunctions, described in section 6.4, is applied to the LBB with combinations of the algorithm modules representing promising results in table 7.7. All finished versions are again evaluated, along with versions that does not finish, but find the assumed optimal solution of 9339.02 $Sm^3/d$ in a node with node number below 2 000. E.g. the branching order level of Alt.3 is again considered for all flow branching criteria, as it finds the optimal solution and finishes in all current versions of the algorithm. Another example of further investigation is that of Alt.4, as it finds the optimal solution early for all branching criteria. Some alternatives are only evaluated for some of the flow branching criteria as their results vary greatly, e.g. Well/pipe. Alternatives Alt.1, Alt.6, and Global are excluded from further consideration, as they are consistently outperformed by others.

### 7.3.2 Separated lift gas branching level

Based on the reformulation of the disjunctive constraints that isolates lift gas regulation and closing of wells to a separate branching level, an extra degree of flexibility is added to the algorithm. The new implementation does not change the physical characteristics of the system from that of the original version, but one is now able to customize the algorithm to utilize the system's dependency to lift gas to produce oil. The new dimension of flexibility also demands new decisions from the user, and in addition to the previously tested flow branching criteria, the lift gas branching criteria of table 6.1 is here used to branch efficiently on wells injected with forbidden amounts of lift gas. In the following three tables, results from the

most promising combinations of branching alternatives and flow branching criteria, combined with different lift gas branching criteria, are presented. When introducing lift gas as a separate branching level, the number of possible branching orders increases, but since the lift gas configuration is considered vital to the algorithm's ability to find the optimal solution fast, all runs are done with the branching level of lift gas allocation first. Knowing that wells can only produce when injected with lift gas, this also implicitly means finding which wells to open for production and which to close, before altering the the routing of oil flows. After lift gas is distributed between the wells by respective lift gas branching criteria, branching order alternatives and flow branching criteria are included as they were. Technically, the new implementation replaces the disjunctive constraints (5.16) and (5.18) with (6.2) and (6.3), respectively, and introduces the new disjunctive constraints of (6.1).

### Highest ratio between oil flow and lower limit of lift gas

In table 7.8 results from the lift gas branching criterion of Flow to LG ratio is combined with promising algorithm configurations from table 7.7. As the problem is lift gas constrained, this branching criterion is applied to find the optimal lift gas allocation early. Thus, it is hoped to help the algorithm to be able to prune large parts of the search tree.

| Flow branching criterion | Branching alternative | Number of nodes | Best node | Best solution [$Sm^3/d$] | Solve time [s] |
|---|---|---|---|---|---|
| Lowest | Alt.3 | 3285 | 80 | 9339.02 | 121.3 |
| | Alt.4 | 3120 | 54 | 9339.02 | 151.2 |
| Highest | Alt.2 | 4547 | 271 | 9339.02 | 182.3 |
| | Alt.3 | 7703 | 4564 | 9339.02 | 267.7 |
| | Alt.4 | 1708 | 94 | 9339.02 | 84.0 |
| | Alt.5 | 11562 | 69 | 9339.02 | 636.9 |
| | Pipe/well | 2288 | 109 | 9339.02 | 118.0 |
| | Well/pipe | 7703 | 4564 | 9339.02 | 267.7 |
| Most equal | Alt.3 | 3322 | 80 | 9339.02 | 121.7 |
| | Alt.4 | 3051 | 73 | 9339.02 | 157.5 |
| Most unequal | Alt.2 | 5320 | 287 | 9339.02 | 267.2 |
| | Alt.3 | 7879 | 4645 | 9339.02 | 276.7 |
| | Alt.4 | 3292 | 116 | 9339.02 | 202.0 |
| | Alt.5 | 20071 | 86 | 9339.02 | 1258.2 |
| | Pipe/well | 13886 | 108 | 9339.02 | 786.9 |
| | Well/pipe | 5320 | 287 | 9339.02 | 267.0 |

Table 7.8: Highest rate between oil flow and lower limit of lift gas

As several of the algorithm versions tested here improves the number of nodes enumerated and solve time considerably from previous versions of the LBB, the

results of table 7.8 demonstrates the importance of finding the optimal lift gas allocation fast. Some versions even beat the benchmark of Bonmin clearly. The best algorithm configuration is currently that of flow branching criterion Highest and branching order alternative 4. It finishes after enumerating 1708 nodes in 84 seconds of solve time. Comparing it to the number of nodes enumerated in the best results of table 7.7 and of the benchmark solution, a decrease of 83.5 and 74.8 percent is found, respectively.

It is hard to make any absolute conclusions when considering the results of the LBB algorithm in table 7.8, as different order strategies performs best combined with different flow branching criteria. On the other hand, some important findings can be pointed out. First, with respect to number of nodes evaluated, branching alternative 4 outperforms all other orders of branching levels when combined with any of the proposed flow branching criteria. Second, the flow branching criterion of Most unequal gives the largest number of nodes enumerated compared to other flow branching criteria, when combined with all but one branching alternative. Finally, when considering the branching alternatives Pipe/well and Well/pipe, their structure is very similar to that of other branching alternatives. Pipe/well first fixes pipeline to separator routing before considering the routing of all wells, connected both topside and subsea. This is similar to both Alt.4 and Alt.5, that first fix pipelines and then subsea and topside connected wells, respectively. An interesting observation is that Alt.5 is consistently beaten by Pipe/well, which again is beaten by Alt.4. The same way, Well/pipe is similar to both Alt.2 and Alt.3, but in this case the results are varying. Well/pipe shows as good, or better results than Alt.3 when applied with the same branching criteria, but Well/pipe was proven inconsistent in finding the optimal solution in table 7.7. Thus, we cannot make any conclusions about the two. The same goes for that of Alt.2. In general, it seems that fixing the routing levels one at the time works better than merging these together.

### Lift gas flow furthest from its allowed interval

While the previously tested branching criteria rely solely on output flows and parameters of the system, the version of the algorithm tested here first considers the flow of injected lift gas when branching, before restoring forbidden routing flows. The branching criterion chooses the well with the biggest difference between its lift gas injection and its lower injection limit. That is, the branching criterion restores the relaxed lift gas flow variables with values furthest from their allowed range to feasibility with respect to the original problem. Results are presented in table 7.9.

| Flow branching criterion | Branching alternative | Number of nodes | Best node | Best solution [$Sm^3/d$] | Solve time [s] |
|---|---|---|---|---|---|
| Lowest | Alt.3 | 17414 | 14162 | 9339.02 | 510.1 |
| | Alt.4 | 13351 | 11620 | 9339.02 | 594.4 |
| Highest | Alt.2 | 15946 | 12191 | 9339.02 | 568.8 |
| | Alt.3 | 26930 | 24365 | 9339.02 | 845.6 |
| | Alt.4 | 9312 | 8286 | 9339.02 | 423.5 |
| | Alt.5 | - | 18631 | 7650.17 | - |
| | Pipe/well | 10479 | 8846 | 9339.02 | 492.7 |
| | Well/pipe | 26930 | 24365 | 9339.02 | 963.6 |
| Most equal | Alt.3 | 13719 | 10461 | 9339.02 | 520.4 |
| | Alt.4 | 13829 | 12054 | 9339.02 | 747.5 |
| Most unequal | Alt.2 | 18320 | 14346 | 9339.02 | 908.5 |
| | Alt.3 | - | 27665 | 9285.79 | - |
| | Alt.4 | 16032 | 14346 | 9339.02 | 699.4 |
| | Alt.5 | - | 26562 | 7650.17 | - |
| | Pipe/well | - | 20110 | 8290.71 | - |
| | Well/pipe | 18500 | 14526 | 9339.02 | 753.7 |

Table 7.9: Lift gas rate furthest from its allowed interval

Table 7.9 presents far worse results than that of table 7.8 and Bonmin. This is considered sufficient evidence to exclude the lift gas branching criterion of Max gap LG from further investigation, but some results should be highlighted. A greater part of the algorithm versions tested here are able to finish after finding the optimal solution than that of the versions tested in table 7.7, with lift gas regulation included in the disjunctive constraints. This may provide proof that separating lift gas to a new branching level makes the algorithm more robust, even when combined with an apparently inefficient lift gas branching criterion. Another point is that even though no excellent results are obtained using the current lift gas branching criterion, Alt. 4 again distinguishes itself as the preferred alternative for all flow branching criteria. Argumentation for the robustness of this branching alternative is thus supported.

### Lift gas flow closest to its allowed interval

In table 7.10 results from applying the lift gas branching criterion of choosing the relaxed lift gas flow variable closest to its allowed interval are presented. The argumentation behind including this alternative is to fix the wells with lift gas injection closest to their possible level of operation to make as little alteration to the solution of the root node solution as possible, and by that finding the optimal lift gas allocation as fast as possible.

| Flow branching criterion | Branching alternative | Number of nodes | Best node | Best solution [$Sm^3/d$] | Solve time [s] |
|---|---|---|---|---|---|
| Lowest | Alt.3 | 3364 | 80 | 9339.02 | 122.8 |
| | Alt.4 | 3162 | 54 | 9339.02 | 152.2 |
| Highest | Alt.2 | 4588 | 271 | 9339.02 | 183.5 |
| | Alt.3 | 7741 | 4564 | 9339.02 | 270.3 |
| | Alt.4 | 1746 | 94 | 9339.02 | 86.2 |
| | Alt.5 | 11598 | 69 | 9339.02 | 586.5 |
| | Pipe/well | 2324 | 109 | 9339.02 | 120.3 |
| | Well/pipe | 7741 | 4564 | 9339.02 | 269.2 |
| Most equal | Alt.3 | 3360 | 80 | 9339.02 | 122.9 |
| | Alt.4 | 3089 | 73 | 9339.02 | 159.3 |
| Most unequal | Alt.2 | 5358 | 287 | 9339.02 | 268.1 |
| | Alt.3 | 7917 | 4645 | 9339.02 | 277.5 |
| | Alt.4 | 3312 | 116 | 9339.02 | 211.3 |
| | Alt.5 | 20127 | 86 | 9339.02 | 1084.9 |
| | Pipe/well | 13918 | 108 | 9339.02 | 761.9 |
| | Well/pipe | 5358 | 287 | 9339.02 | 268.4 |

Table 7.10: Lift gas rate closest to its allowed interval

Table 7.10, presenting results using lift gas flow criterion of Min gap LG, mainly displays the same results as that of table 7.8. The results of the two tables follow the same pattern of number of nodes evaluated through the different algorithm configurations, with the former consistently following a few nodes behind the latter. This indicates that versions of the algorithm implemented with lift gas branching criterion Min gap LG uses a small number of additional nodes before finding the optimal gas lift allocation of the system, compared to the versions implemented with Flow to LG ratio, before it finishes the algorithm using the same path through the search tree. It seems that the lift gas branching criterion of Min gap LG always will find its superior in Flow to LG ratio, when implemented in the LBB in its current form. In any case, the results of table 7.10 rigorously find the assumed optimal solution of the problem in few enumerated nodes and little solve time, and present efficient algorithm configurations compared to the results of the algorithm with lift gas constraints included in the disjunctive constraints and the benchmark. This supports the argumentation of releasing lift gas regulation to a separate branching level, and the importance of lift gas allocation further. Otherwise, the same conclusions as from table 7.8 can be made with respect to combinations of flow branching criteria and level orders when considering lift gas branching criterion Min gap LG.

Considering the overall results from the LBB presented so far, a lot of valuable insight to the solution method is gained. Although proof of its potential effectiveness when applying problem specific knowledge already is evident, there is still potential in the solution method. In addition, the robustness of the algorithm when applied to other systems, or when parameters of the current system is changed, remains to

be tested. For that, some of the main findings in the current results are summarized here.

Results from table 7.7, with algorithm versions with lift gas included in the disjunctive constraints, show that all versions of the LBB algorithm that are able to finish within 30 000 nodes find the same optimal solution as Bonmin. This is used to argue that Bonmin actually finds the global optimal solution to the problem, a belief that is strengthened by the results of later versions of the algorithm. When introducing lift gas injection as a separate branching level the importance of finding optimal lift gas allocation early becomes evident. In table 7.8 it is also found that some branching alternatives are preferred to others. Alt. 4 consistently dominates both Alt.5 and Pipe/well, and the two latter alternatives can be excluded from further evaluation. Although Alt.4 rigorously finishes with fewer nodes than *all* other alternatives, no more are excluded. This is done to exclude only those alternatives that are dominated by a similar one, i.e. that is assumed to choose the same path in the search tree, and to keep combinations of branching criteria and level orders that are expected to make other paths through the search tree, as that might prove efficient when combined with e.g. other search strategies. Furthermore, the flow branching criterion of Most unequal provides slowest solutions, both with respect to solve time and number of nodes, when combined with most of the level order alternatives, and is excluded from further consideration. Finally, the results from tables 7.8, 7.9, and 7.10 indicates that Flow to LG ratio is the preferred lift gas branching criterion. It considerably outperforms Max gap LG and marginally dominates Min gap LG. In any case the results from all tables highlights the importance of utilizing the system's dependency to lift gas, and strengthen the argument of branching on forbidden lift gas flows first.

### 7.3.3 Best first

When applying the best first strategy to the LBB algorithm, it is interesting to measure the impact a change of search strategy has on different configurations of the code. The results so far have shown that the LBB with a separate branching level for lift gas clearly outperforms the one with the standard disjunctions when applied with the depth first strategy. Of this reason, the best first strategy will only be tested on the LBB with this structure of the disjunctive constraints. To limit the amount of presented results from the LBB, only chosen algorithm configurations will be tested. Following the argumentation at the end of the previous section, Alt.3 and 4 are tested with flow branching criterion Lowest, Highest, and Most equal, Alt.2 and Well/pipe are tested only in combination with Highest. All other configurations are disregarded.

In table 7.11 results from applying the best first strategy with the separated disjunction for lift gas is presented. All results are derived using the lift gas branching criterion of Flow to LG ratio, following the argumentation at the end of section 7.3.2.

| Branching criterion | Branching alternative | Number of nodes | Best node | Best solution [$Sm^3/d$] | Solve time [s] |
|---|---|---|---|---|---|
| Lowest | Alt.3 | 4300 | 4038 | 9339.02 | 184.4 |
| | Alt.4 | 3900 | 3519 | 9339.02 | 201.3 |
| Highest | Alt.2 | 4516 | 3310 | 9339.02 | 181.3 |
| | Alt.3 | 3168 | 1967 | 9339.02 | 135.5 |
| | Alt.4 | 1396 | 1345 | 9339.02 | 66.7 |
| | Well/pipe | 3168 | 1967 | 9339.02 | 134.6 |
| Most equal | Alt.3 | 4298 | 4036 | 9339.02 | 183.2 |
| | Alt.4 | 3768 | 3417 | 9339.02 | 196.4 |

Table 7.11: Best first with separate branching level for lift gas

Also for the best first search strategy the LBB algorithm proves to be efficient when lift gas is regulated in a separate disjunctive constraint. Table 7.11 shows that all tested versions of the algorithm finish within a reasonable number of nodes and solution time. Alt.4, combined with the flow branching criterion of Highest, stands out as the most promising algorithm configuration, as it did with depth first. This can be said to somewhat as expected. Alt.4 has consistently throughout the results found the optimal solution to the problem in an early node. As this seems to be the case also for best first, the algorithm can evidently take advantage of this and is thus able to terminate early. Considering the results in table 7.11, it should be noted that all versions of the algorithm terminate relatively shortly after finding the optimal solution, i.e. the best node is relatively close to the number of nodes. This is as expected for the best first strategy due to the fact that it searches after the greatest objective value among all generated nodes. As nodes are solved, the current objective value decreases successively from that of the previous node, before eventually the optimal solution is found as the first feasible solution to the original problem. When considering the best first strategy, the most interesting is to compare the results with the depth first implementation, to see which strategy yields the best results. In table 7.12 results from the two strategies are compared through a percentage change in number of nodes enumerated for each of the current versions of the LBB.

| Branching criterion | Branching alternative | Depth first | Best first | % change |
|---|---|---|---|---|
| Lowest | Alt.3 | 3285 | 4300 | 30.9 |
| | Alt.4 | 3120 | 3900 | 25.0 |
| Highest | Alt.2 | 4547 | 4516 | -0.68 |
| | Alt.3 | 7703 | 3168 | -58.9 |
| | Alt.4 | 1708 | 1396 | -18.3 |
| | Well/pipe | 7703 | 3168 | -58.9 |
| Most equal | Alt.3 | 3322 | 4298 | 29.4 |
| | Alt.4 | 3051 | 3768 | 23.5 |

Table 7.12: Change by implementing a best first strategy

All tested algorithm configurations show promising results when combined with both depth and best first. Considering the results, one can hardly claim a general tendency of increased effectiveness when applying the best first strategy, as they differ substantially. Best first do however seem to perform better when flow branching criterion of Highest is used and depth better when combined with Lowest and Most equal. The reason for this is hard to affirm, but it may be that Highest is able to find the optimal routing of flows fastest, and thus converge to the optimal solution in shorter time. This makes Highest the preferred choice when applying the best first strategy. Another important finding is that the same algorithm configuration as with depth first performs best with best first. The belief that the possibility for increased customization in depth first would provide best results is not verified. Even though best and depth first changes on being best with different configurations, best first is the fastest with the best configuration.

Based on the results presented here and in previous sections of this chapter, several promising configurations of the LBB algorithm is found. It is proven paramount to both solution time and number of nodes enumerated to identify optimal lift gas allocation to the problem early, and many of the proposed algorithm structures beat the benchmark solution from Bonmin when this is done. Combined with a good branching criterion and routing level decision the suggested approach has decreased the solution time significantly. Both the depth and best first search strategy has proven efficient when applied to the LBB, with the best first as the most promising, 18.3% better than the most efficient version of depth first when considering the number of nodes. From this point on, all implementations of the LBB refers to the most promising algorithm configuration, namely that of separated lift gas regulation with lift gas branching criterion of Flow to LG ratio, flow branching criterion of Highest and branching order level Alt.4. Even though the best first strategy has shown the best results, also depth first has shown great potential and will be tested further with the best configuration.

### 7.3.4 Symmetric solutions

Symmetric solutions may occur in optimization problems with identical system components. As these components hold equal properties, it is possible for the solution method to provide additional solutions per solution involving one of them. These solutions are practically the same as their originals and are derived by switching the identical components in one solution of the problem to the other, producing the same objective value. This increases the complexity of the problem, as the size of the search space increases. Since the solutions are equal in sense of the objective value, or symmetric, it is not necessary to investigate them all.

The test case that the LBB is applied to contains three separators of which two hold equal capacities. This will lead to symmetric solutions as all subproblem solutions are routing phase flow from pipelines and wells to separators. All flow routed to one of the identical separators in one solution could easily be swapped with the flows

routed to the other identical separator to provide a symmetric solution. This should be considered in implementations of both the logic formulation solved by the LBB and the MINLP-formulation solved by Bonmin.

Removing all symmetry in the problem is hard, but a proof of the effect of removing some symmetric solutions can be obtained through a simple manual preprocessing. By selecting one pipeline and saying that if it is routed to one of the identical separators, it shall be routed to a separator chosen by the modeler, the routing possibilities for the pipeline is reduced from three to two. This reduces the the size of the solution space, but no potentially optimal solutions are cut away as the pipeline could have been routed to either separator in any solution. Practically, this limitation is implemented by denying flow from one pipeline to one of the identical separators. Since pipelines potentially include flows from several wells, the effect of reducing the routing possibilities of a pipeline is assumed greater than that of limiting the routing of a satellite well, and is thus the only option considered. The effect from implementing this simple preprocessing to the MINLP implementation and the LBB algorithm with branching strategies of best and depth first is shown in table 7.13.

| | Original | Symmetry | % change |
|---|---|---|---|
| MINLP | 6786 | 5606 | -17.4 |
| Depth first | 1708 | 1154 | -32.4 |
| Best first | 1396 | 1026 | -26.5 |

Table 7.13: Improvement by reducing symmetric solutions

The results clearly demonstrates a positive impact of removing some symmetry with the simple preprocessing. For the LBB, both depth and best first has a decrease in the number of nodes of around 30%. Bonmin also reduces the number of nodes, but less than the LBB, with about 17% decrease. This might suggest that the LBB algorithm is better suited for this kind of symmetry reduction. The reason for this is probably that the symmetry limitation needs to be implemented more explicitly in the MINLP as an extra constraint, denying one pipeline to be routed to a certain separator, applied directly in the model formulation. As the system contains four pipelines and two identical separators, there exist eight different such constraints relevant for application. The greatest impact on the number of nodes is found when pipeline "prod1" is denied routed to separator "sep2", and is what is presented in table 7.13 for the MINLP. The symmetry limitation is implemented by setting the corresponding binary variable to zero.

In the LBB implementation, the symmetry limitation can be integrated in the development of the search tree. Normally, when branching occurs in the level of pipeline to separator routing, three nodes are made; one for routing to each of the separators. With the symmetry removal, only two nodes are made the first time this branching level is visited by the algorithm; one for routing to one of the equal separators and one to the last separator. This way the pipeline that is closed to one of the equal separators is chosen based on the configuration of the algorithm

(branching criteria and branching alternative). The symmetry limitation will always happen as early as possible in the search tree, and hence reducing the size of the solution space as much as possible. As Bonmin utilizes its own solution method and algorithms, such an implementation is not possible for the MINLP.

Generally, the possibility of symmetric solutions is detected in the actual production system or in the parameter data for the problem formulation, and should probably be considered from the start of the model development and testing. However, the focus of this thesis is on algorithm development and application of logic structures in petroleum production optimization, while symmetry is considered a modeling tool that can be detected and implemented in any solution method or commercial solver. It is therefore considered sufficient to point out its presence and potential importance here, after a thorough testing of different configurations has been done.

## 7.4 Computational analysis

The implemented LBB algorithm has achieved promising results in the previous sections, but its robustness and general applicability can not yet be guaranteed. The algorithm has been tested on a specific test case with real production data from Petrobras, but the effect of changing the system could have a vast impact on its usability. Increasing the number of wells, or changing some capacities, limits, or other parameters of the system could all influence the efficiency of the algorithm. This section will provide a computational analysis, testing how the LBB will respond to such alterations. First, the lift gas capacity for the platform is varied, before the algorithm is applied to a bigger system consisting of more wells. It is important to note that no economic analysis is done of the results in this thesis, as it is a purely methodical contribution. Thus, the computational analysis presented below is only carried out to test the robustness of the presented LBB algorithm.

### 7.4.1 Increased lift gas capacity

The lift gas allocation, and thus the configuration of open wells, has proven to be a vital property of the system production. In the presumed optimal solution to the problem the model opens as many wells as it can without exceeding the lift gas capacity of the platform and at the same time complying with the well specific intervals for lift gas injection. Increasing the platform's total capacity for lift gas is the first change that will be tested on the system. The amount of increased capacity will correspond to the possibility of opening one more well. It will hence be interesting to see whether the model will maximize the production for the wells already opened by injecting more lift gas into them, or if it opens another well and keep lift gas injection to the already producing wells close to their lower limits for injection. Increasing the lift gas capacity is also interesting when it comes to other system capacities such as those for separators. More available lift gas means higher

production levels, which again can make the system constrained also by separator capacities. This may potentially have a great impact on the solution efficiency of the algorithm.

The capacity is increased to allow all combinations of a maximum of nine producing wells. Table 7.14 shows the main characteristics of the solution found with the increased lift gas capacity, while table 7.15 shows computational specifications for the LBB with best and depth first and for the MINLP.

| | |
|---|---|
| Oil production [$Sm^3/d$] | 10394.8 |
| Number of open wells | 9 |
| Pressure constrained | YES |
| Gas flow constrained | NO |
| Water flow constrained | NO |
| Liquid constrained | NO |
| Lift gas constrained | YES |

Table 7.14: Solution with increased lift gas capacity

| | MINLP | Depth first | Best first |
|---|---|---|---|
| Solve time [s] | 84.5 | 58.1 | 44.0 |
| Number of nodes | 3677 | 1620 | 1064 |
| Best node | 572 | 794 | 1043 |

Table 7.15: Computational characteristics with increased lift gas capacity

Both versions of the LBB find the same solution as that of the MINLP implementation. The increased lift gas capacity is utilized by opening an additional well compared to the original case, increasing the oil production substantially. Also with the increased capacity, the problem is pressure and lift gas constrained, and production levels are kept below the separator capacities. Most wells are injected with lift gas amounts corresponding to their lower limits. Excess lift gas is injected to pressure constrained wells. If the capacity had been increased to the point where the problem no longer is constrained by lift gas capacity, but rather by some other parameter such as the separator capacity, the LBB could be greatly affected. Fixing all routing could become the essential decision for the problem, and the fastest choice of routing levels could suddenly be the opposite of the current best performing.

Considering both the number of nodes and the solve time in table 7.15, the LBB algorithm still outperforms the implementation of the MINLP, but some new observations can be made. When the increased capacity is applied, the LBB shows only a small reduction in the number of nodes while the MINLP implementation nearly bisects it. This might indicate that the MINLP has more difficulties when the combinatorial difficulty of the problem is higher, and that it thus is able to converge faster to the optimal solution when more wells can be opened. As for the previous applications of the LBB, the best first strategy still outperforms depth first.

### 7.4.2 Reduced lift gas capacity

Reducing the platform capacity for lift gas injection is also a modification that could have unforeseen effects on the LBB algorithm. If the capacity is reduced to a level where at least one additional well has to be closed, the combinatorial challenge for the algorithm will increase as there are more combinations of few wells than many when selected from the same number of wells. It is of great interest to test how the LBB algorithm handles such challenges compared to the implementation of the MINLP.

By decreasing the platform capacity to right below the point of where it is possible for eight wells to produce, the system allows for production in a maximum of seven wells. In table 7.16 characteristics of the optimal solution found with such a decrease is presented, while table 7.17 presents key properties of the achieved efficiency.

| | |
|---|---|
| Oil production [$Sm^3/d$] | 9065.54 |
| Number of open wells | 7 |
| Pressure constrained | YES |
| Gas flow constrained | NO |
| Water flow constrained | NO |
| Liquid constrained | NO |
| Lift gas constrained | YES |

Table 7.16: Solution with reduced lift gas capacity

| | MINLP | Depth first | Best first |
|---|---|---|---|
| Solve time [s] | 182.4 | 71.9 | 43.3 |
| Number of nodes | 6257 | 1967 | 1048 |
| Best node | 520 | 1444 | 1024 |

Table 7.17: Computational characteristics with reduced lift gas capacity

As with the other cases, the same objective value is found in all implementations. Except for some routing that is swapped (symmetric solutions), the solutions are equal in all decision variables. Due to the decrease in lift gas capacity, the oil production is decreased, and two of the low producing wells from the standard case are closed while one of the closed wells from this solution is opened, giving a total of seven producing wells. Also here the solution consists of finding the best combination of opening as many wells as possible with respect to the lower lift gas injection limits, and allocating the remainder of available lift gas to the well that increases the production the most with this gas. Opening fewer wells with more allocated lift gas to increase production from these wells is hence neither in this case an improving action.

The LBB implementations again show good adaptability to the changed system parameters. Both the depth and best first strategy outperform the MINLP implementation in Bonmin. All implementations finish fast, with best first using only

about 43 seconds of solve time. The increased combinatorial challenge of combining fewer open wells is evidently handled a lot better by the LBB than by Bonmin. This strengthens the basis for being able to establish the LBB as a flexible and widely applicable solution algorithm.

### 7.4.3 Expanded system

The size of the search tree of the LBB expands exponentially as the algorithm is applied to bigger systems. Considering the relatively limited size of the test case, evaluating the algorithm performance when applied to an increased production system will be of great interest. Keeping the common component data on original levels, i.e. system capacities etc., also this will imply an increased combinatorial challenge, as the number of open wells will decrease relative to the total number of wells.

As the P35 platform in fact has been through several changes with respect to the number of producing wells due to changing reservoir dynamics and well characteristics, a restructuring of the well setup is a realistic scenario that should be tested. Using old well data for a former system composition with 15 wells together with the current data for common components of the system, a realistic implementation of an expanded system is tested. The main characteristics of the solution for the expanded system is presented in table 7.18, while table 7.19 shows the differences in computational effort demanded from the different models.

| | |
|---|---|
| Oil production [$Sm^3/d$] | 10025.3 |
| Number of open wells | 8 |
| Pressure constrained | YES |
| Gas flow constrained | NO |
| Water flow constrained | NO |
| Liquid constrained | NO |
| Lift gas constrained | YES |

Table 7.18: Solution with expanded system

| | MINLP | Depth first | Best first |
|---|---|---|---|
| Solve time [s] | 1618.4 | 304.7 | 372.1 |
| Number of nodes | 23490 | 8555 | 8855 |
| Best node | 2617 | 1580 | 8807 |

Table 7.19: Computational characteristics with expanded system

Also for the expanded system, all models find the same optimal solution. The number of wells has been increased from 10 to 15, but the system still produces only from eight wells. This is because the lift gas capacity is left at its original level and the problem is still lift gas constrained, and as before, also pressure constrained. The oil

production is increased considerably compared to that of the original system and shows that some of the new wells are included in the current solution. The new wells are selected over the original ones either because they provide higher production levels at similar lift gas amounts or similar production levels at lower lift gas levels. Either way, they are assumed to give higher oil flow per unit lift gas and thus increase total production.

A different solution than that of the original problem is expected, but what really is interesting is the results presented in table 7.19. Still, the LBB outperforms the implementation of the MINLP. While the MINLP implementation uses over 23000 nodes, both LBB models use less than 9000. Comparing these numbers to those of the original system, the number of nodes for the MINLP increases to about four times as many, while the LBB increases to around seven and eight times as many for the depth and best first strategy, respectively. In spite of this, the absolute increase in the number of nodes is a lot higher for the MINLP compared to both LBB implementations, rising with over 10 000 nodes more. In addition, the efficiency of the MINLP is highly dependent on which routing is closed in the symmetry removal. The result presented here is from the best alternative, while other implementations displayed node numbers over twice as high. This contributes further to the argumentation that the LBB is less vulnerable than the MINLP to combinatorial difficulties.

Another finding is that the depth first strategy now performs marginally better than best first. This is somewhat surprising as best first consistently has outperformed depth first after it was introduced. A certain reason for this is hard to provide, but the LBB algorithm structure as it now is configured was developed specifically for the depth first strategy, as discussed in section 6.3.3. It is possible that the major expansion of the search tree provides many slightly different solutions, and that the best first strategy is forced to solve such a high number of nodes before finding the optimal solution that it is not able to beat depth first. In that case, one can assume that the best first strategy will be the preferred branching alternative when considering smaller systems, while depth first asserts its right when applied to greater systems. The results for the expanded system provides further reason to believe that the flexibility of the LBB can be utilized and tailored to fit different problem structures and sizes with promising results.

## 7.5 Summary and usability of algorithm

After thorough testing it is evident that the LBB holds potential in reducing computational effort compared to traditional optimization methods used in commercial software. However, the efficiency of the algorithm varies greatly when different configurations are implemented. The main findings of the LBB are summarized and discussed in this section. Figure 7.1 demonstrates the increasing efficiency (decreasing number of nodes) of different configurations of the LBB compared to the MINLP implementation.

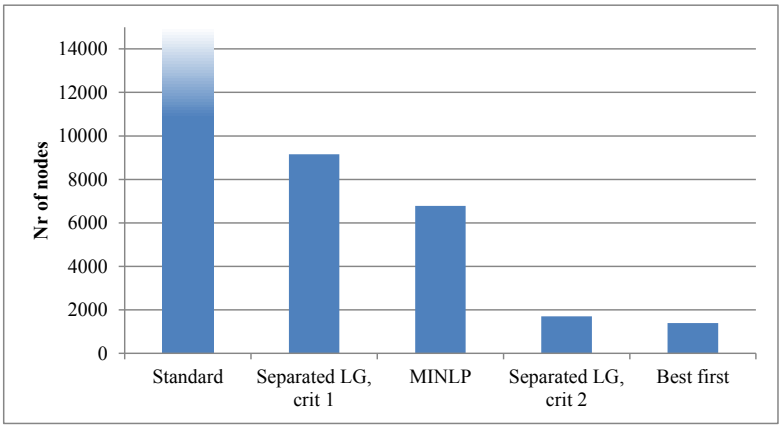


Figure 7.1: Improvement of algorithm configurations

Moving from left to right in figure 7.1 it is clear that the configuration and choice of properties of the algorithm has a tremendous impact on the achieved efficiency. When implemented in its most basic form the LBB exhibits poor utilization of the logic problem structure, and in most cases it is not able to converge to an optimal solution within reasonable time. This is illustrated by the leftmost column in the figure. Although this version of the algorithm performs poorly in general, a clear indication of the correlation between algorithm configuration and efficiency is found. This correlation also holds for implementations of the algorithm where lift gas allocation and well handling is separated to an isolated routing level, With this approach most of the configurations also find the optimal solution and terminate within an acceptable number of nodes and level of solution time. In addition, the number of nodes is reduced considerably, far below that of the MINLP implementation, when a more suitable branching criterion for lift gas is included. Ultimately the depth first strategy of the algorithm is replaced with best first, and the number of nodes is reduced even further.

As of now the LBB satisfies the first and most important requirement for a solution method for a RTPO problem, by providing good solutions fast. In addition, the optimal solution found in all configurations of the LBB is the same solution found by the commercial software solving the MINLP. This may indicate that the nonconvexities of the problem does not make it as hard to solve as feared, and that the algorithm is able to provide the global optimal solution. Thus, the quality of the obtained solution should be sufficient for the RTPO. Although this seems well and good it should be noted that these results are achieved through extensive testing, adaptation, and specialization, and are not easily obtained. If regular commercial solvers can provide the same solution within required time frames, it should be considered if the "juice is worth the squeeze," i.e. if the improved solution efficiency is worth the considerable amount of work that is needed to attain more efficient algorithms.

Furthermore, testing the algorithm on an expanded or altered system supports the assertion about the LBB as an adaptable and flexible algorithm. Even on a problem with 50% increased system size, the LBB's efficiency consistently outperforms the

solved MINLP implementation. Thus, when changing existing parameters, e.g. separator capacities, or changing the number of system components that will not influence the current branching possibilities, e.g. the number of wells or manifolds, the LBB could be implemented with minor alterations with indications of good results. However, when increasing or decreasing the number of separators or pipelines, the flow routing alternatives of some current components are correspondingly increased or decreased, and the current branching possibilities are affected. Although the LBB will cope with such changes, thorough knowledge of the algorithm and a great deal of technical alteration is required, and the results cannot be guaranteed. Ultimately, this means that the LBB is not able to solve general mathematical models of petroleum systems without reformulation of the model and potentially extensive customization of the algorithm.

The need for customization can be said to be both advantageous and unfavorable to the algorithm usability. Advantageous because adjusting the algorithm to each specific problem facilitates a high degree of algorithm customization and may ensure that it is implemented properly. Unfavorable because adjustment to a problem may be a complex and time consuming process. Adjusting the LBB to new problems also requires extensive knowledge of the algorithm structure, which may complicate the use and weaken the general versatility of the LBB. In addition, applying a nontrivial degree of problem specific knowledge has proven essential to the success of the algorithm and this might not always be available.

Looking at the test case of this thesis, such knowledge is utilized by handling the system's lift gas limitation in a specific way and is essential for the results of the LBB. E.g. if the system not was lift gas constrained and wells were free flowing, the success of the LBB would be far from certain. The algorithm would also require modification in order to work properly. If the system also had been constrained on some other limit than the total lift gas capacity, such as the separator capacities, it is also far from certain that the same algorithm configuration would give the best results. The general efficiency of the LBB is in any case dependent on problem structure and conditions, and absolute conclusions are hard to make.

Concerning the applicability of the algorithm, one could discuss whether the LBB should be developed on a more general level. This would ease implementation and broaden the application to problems with different structures and physical characteristics. However, the proven degree of dependency to customization to obtain good results supports the development of a strongly problem specific algorithm. If desirable, elements of the derived algorithm could provide a framework that easily can be utilized to develop a more general algorithm. The general algorithm may later serve as a starting point when fitting the algorithm to a specific problem, to provide the necessary degree of customization to attain desired results.

Independent on the usability of logic based algorithms in general; the LBB developed in this thesis has shown to be a robust and flexible algorithm for the problem type it is developed for. Utilization of logic combined with problem specific knowledge in modeling and algorithm development clearly holds a potential. To say something

conclusive about the overall usability is difficult as the results will be strongly problem dependent. However, many problems have a structure that may be suited for incorporation of logic, and exploiting this more actively could have a positive impact on many optimization problems.

# Chapter 8

# Conclusion

The operational planning for petroleum production problems is in steady advancement. The focus of both modeling and solution approaches is however mostly centered around classic optimization methods. This sufficiently allows precise modeling of all parts of the system, but the problems are often complex and therefore hard and computationally expensive to solve to global optimality. However, most production assets hold certain properties and structures that with some "thinking outside the box" could be utilized more precisely in an optimization scheme. The scope of this thesis has been to utilize such properties in a problem specific model and take explicit advantage of the structure in a customized solution approach.

The main contributions of this thesis are twofold. First, a logic reformulation of the classic MINLP for production operation is derived, which includes disjunctive constraints for handling decisions normally managed by binary variables. This gives a simpler and maybe more intuitive mathematical model of the production system. Second, a customized branch and bound algorithm (LBB) is developed to explicitly utilize the structure and logic conditions of the model, in an attempt to reduce computational effort required to solve the problem or possibly find better solutions. The proposed solution method is designed for wide configuration flexibility to facilitate utilization of problem specific characteristics.

Results show that the proposed algorithm outperforms a recognized MINLP solver significantly, when applied with appropriate configuration. As such solvers are developed over a number of years, while the LBB still is in its developing phase, this clearly shows the powerful tools that can be utilized in disjunctive programming when combined with tailored algorithms. Although the MINLP is implemented only as a basic model tested in a general solver while the logic formulation utilizes the solver in combination with a customized algorithm, an evident potential is demonstrated. It is also argued that the incorporation of logic could have a positive impact for other problems with similar properties, and that it should be given more attention when modeling optimization problems in general.

The proposed algorithm provides the same optimal solution as a regular MINLP

solver. Due to the nonlinear reservoir behavior, multiphase flow and the use of binary variables, the solution space of the MINLP formulation is highly nonconvex. Current software cannot guarantee global optimal solutions for such problems. Since the binary variables partly cause the nonconvexities and are removed in the logic formulation, the LBB should provide solutions at least as good as the solver for the MINLP. The fact that both solution methods find the same optimum may indicate that the nonconvexities does not make the original problem as hard to solve as feared. This again might imply that solutions obtained from existing solvers might be trusted with more confidence. It should however be noted that the NLPs solved in each subproblem of the LBB also are nonconvex, and the solver used, Ipopt, works as a heuristic also in these problems. For this reason the proposed method cannot guarantee global optimum any more than its underlying solvers, but is probably in any case closer to finding it than the original MINLP implementation.

Furthermore, the success of the LBB has proven highly dependent on the algorithm configuration and the degree of customization applied. The obtained results vary widely in both solution quality and computational effort required, and illustrate the importance of profound knowledge of the system. By utilizing this knowledge and tailor the algorithm for the problem at hand, a huge potential in efficiency may be realized. It is however important to notice that the potential success of such an algorithm most likely is limited to a specific problem, or a group of similar problems, and algorithm development should probably not be controlled by general rules.

# Chapter 9

# Further work

Suggestions for further work on the LBB is presented in this chapter. Even though the proposed formulation and implementation of the operational planning problem has shown great potential, the work is far from finished and could be supplemented with several interesting extensions. These include measures for improved efficiency and tools to provide guarantee of finding global optimal solutions.

The proposed method cannot in its current state guarantee finding the global optimum of a nonconvex problem, but should find solutions at least as good as that of standard methods such as a MINLP implementation. To be able to provide solutions with guaranteed global optimality, the model and algorithm should be expanded to include total convexification of the problem's feasible region, e.g. trough piecewise linearization. Linear approximations could be derived from surface estimates of the nonlinear functions, and used to model the convex hull of the solution space. Such approximations could be included and solved in all nodes of the BB search tree, providing certain upper bounds to the problem. This would result in a gap in each subproblem, between the solutions of problems solved in the nonconvex and convex space. If the algorithm is able to close the gap entirely, guarantee of global optimality can be provided.

Different configurations of the LBB has been tested extensively, and the potential of investigating additional configurations is considered limited. However, two more possibilities should be tested. First, when considering a predetermined order of routing levels, i.e. a branching alternative, a priority scheme between routing levels could be deployed. The implemented priority will override the branching level order if the magnitude of infeasible flows in other parts of the system are sufficiently greater than that of the current branching level. This is easy to implement, but may prove hard to test effectively as any such priority scheme could provide infinitely many possibilities based on continuous parameters. Second, combining the depth and best first search strategies could hold a potential. Starting with depth first, a good and feasible solution to the problem might be found after few enumerated nodes, before the algorithm could switch to best first and continue the search. This may be beneficial as the algorithm might be able to prune nodes from a good solution early

and thus be able to converge more quickly.

The impact of other modeling or algorithm modifications might also be of interest. One such example is making a tighter formulation of the root node of the algorithm, i.e. the special NLP relaxation of the problem. This would result in a lower objective value in the root node, and thus a tighter upper bound to the problem. This would be of interest in systems where a nontrivial number of subproblems are solved to high objective values (between the root node and the global optimal solution), as all nodes with objective values currently above the potential upper bound would be removed from the search space. For the current problem however, the algorithm seems to be able to converge efficiently from the root node solution. In addition, the root node of the MINLP solved by Bonmin and the root node of the LBB provide only marginally differentiating objective values in the current implementations. Thus, a drastic improvement should not be expected, at least not without significant effort. One should also consider using heuristics to derive both upper and lower bounds to the problem from the beginning. This would reduce the size of the solution space, and could result in faster convergence. Caution should however be exercised when implementing bounds to the problem when nonconvexities are present, as good or optimal solutions could be excluded from the feasible region of the problem.

Another suggestion to improve the solution time of the algorithm is parallelization. Its relevance has risen in later years along with the available CPU power. Parallelization utilizes several CPUs at the same time and can thus solve many problems in parallel at the same time. Since the LBB makes several subproblems when branching on a node, the use of parallelization could hold great value, as solving several or all of the subproblems simultaneously might prove beneficial. Parallelization might prove especially useful in combination with the separated lift gas branching level of the LBB. This would provide the possibility of generating separate subtrees for all lift gas allocation possibilities, comparing them to each other as they expand, and pruning some of them based on the results of others. This could give rapid convergence and a considerable increase in solution efficiency, especially for larger problem cases.

All of the above are methods that hold potential of reducing computational effort of the LBB or in guaranteeing its solutions' global optimality. Another interesting area would be to compare the algorithm directly to other solution approaches for MINLPs, applying them to the same problems. This could include classic methods as piecewise linearization by SOS2 for the nonlinear relations and the big-M method for binary variables. In addition, literature in petroleum optimization often include decomposition methods for networks of several production systems. Thus, the applicability of disjunctive programming should also be investigated when combined with such methods. Extending the LBB to other types of problems than the RTPO would also be interesting, and would further test its usability and flexibility.

Even though the LBB holds evident potential in solution efficiency, guarantee of success cannot be given when it is applied to a random production system. Therefore, the algorithm should be tested on cases with other properties than that of the test case provided in this thesis. These should include cases where other constraints than

the lift gas capacity are binding. This might change the basis for the algorithm, as it possibly no longer can utilize LG to achieve fast convergence, and an extensive reconfiguration of the LBB would most likely be required to achieve satisfying results.

Finally, the algorithm should be reimplemented in a more advanced programming language. AMPL is a basic modeling language and is not fit for large scale algorithm development. By implementing the LBB in a more suited programming language, e.g. C++, one could practically remove overhead time used in the algorithm.

# Bibliography


Balas, E. (1974), Disjunctive programming: Properties of the convex hull of feasible points, Technical report, Carnegie Mellon University.

Balas, E. (1985), 'Disjunctive programming and a hierarchy of relaxations for discrete optimization problems', *SIAM Journal on Algebraic Discrete Methods* **6**, 466–486.

Bampi, D. and J.C. Costa (2010), 'Marlin field: An optimization study for a mature field'. *This paper was prepared for presentation at the SPELatin America & Caribbean Engineering Conference held in Lima, Peru, 1-3 December 2010, SPE 139376.*

Bieker, H.P., O. Slupphaug and T.A. Johansen (2006), 'Real time production optimization of offshore oil and gas production systems: A technology survey'. *This paper was prepared for presentation at the 2006 SPE Intelligent Energy Conference and Exhibition held in Amsterdam, The Netherlands, 11-13 April 2006, SPE 99451.*

Bollapragada, S., O. Ghattas and J. N. Hooker (2001), 'Optimal design of truss structures by logic-based branch and cut', *Operations Research* **49**, 42–51.

Bonami, P., L.T. Biegler, A.R. Conn, G. Cornuéjols, I.E. Grossmann, C.D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya and A. Wächter (2008), 'An algorithmic framework for convex mixed integer nonlinear programs', *Discrete Optimization* **5**, 186–204.

BP (2012), 'Bp statistical review of world energy june 2012'.

Carvalho, M. C. A. and J. M. Pinto (2006), 'An milp model and solution technique for the planning of infrastructure in offshore oilfields', *Journal of Petroleum Science and Engineering* **51**, 97–110.

Chen, Y, D.S. Oliver and D. Zhang (2009), 'Efficient ensemble-based closed-loop production optimization', *SPE Journal* **14**, 634–645.

Devold, H. (2006), Oil and gas production handbook, Technical report, ABB ATPA Oil and Gas.

Downey, M (2009), *OIL 101*, Vol. 1, Wooden Table Press LLC.

Duran, M.A and I.E. Grossmann (1986), 'An outer-approximation algorithm for a class of mixed-integer nonlinear programs', *Mathematical Programming* **36**, 307–339.

Economides, M. J., A.D. Hill, C. Ehlig-Economides and D Zhu (2012), *Petroleum Production Systems*, 2 edn, Prentice Hall.

Eia, U.S. Energy Information Administration (2013), 'Norway, country analysis', [Online]. Available: `http://www.eia.gov/countries/country-data.cfm?fips=NO&trk=c`. [Accessed: 2013-05-07].

ExxonMobil (2012), 'The outlook for energy: a view to 2040'.

Fleshman, R. and H.O. Lekic (1999), 'Artificial lift for high-volume production', *Oilfield Review* **11**, 49–63.

Foss, B., V. Gunnerud and M.D. Díez (2009), 'Lagrangian decomposition of oil-production optimization applied to the troll west oil rim', *SPE Journal* **14**, 646–652.

Fourer, R., D.M. Gay and B.W. Kernighan (2003), *AMPL: A Modeling Language for Mathematical Programming*, 2 edn, Duxbury Press / Brooks/Cole Publishing Company.

Geoffrion, A.M. (1972), 'Generalized benders decomposition', *Journal of Optimization Theory and Applications* **10**, 237–260.

Grossmann, I.E. (2002), 'Review of nonlinear mixed-integer and disjunctive programming techniques', *Optimization and Engineering* **3**, 227–252.

Gudmestad, O.T., A.B. Zolotukhin and E.T. Jarlsby (2010), *Petroleum Resources with Emphasis on Offshore Fields*, WIT Press.

Gunnerud, V. and B. Foss (2009), 'Oil production optimization—a piecewise linear model, solved with two decomposition strategies', *Computers & Chemical Engineering* **34**, 1803–1812.

Guo, B., W.C. Lyons and A. Ghalambor (2011), *Petroleum Production Engineering : A Computer-Assisted Approach*, Elsevier Science, Burlington.

Gupta, O.K. and A. Ravindran (1985), 'Branch and bound experiments in convex nonlinear integer programming', *Management Science* **31**, 1533–1546.

Güyagüler, B. and T. Byer (2008), 'A new rate-allocation-optimization framework', *SPE Production & Operations* **23**, 448–457.

Jahn, F., M. Cook and M. Graham (2008), *Hydrocarbon Exploration & Production*, 2 edn, Elsevier Science, Oxford.

Kosmidis, V.D., J.D. Perkins and E.N. Pistikopoulos (2005), 'A mixed integer optimization formulation for the well scheduling problem on petroleum fields', *Computers & Chemical Engineering* **29**, 1523–1541.

Lee, S. and I.E. Grossmann (2000), 'New algorithms for nonlinear generalized disjunctive programming', *Computers & Chemical Engineering* **24**, 2125–2141.

Lee, S. and I.E. Grossmann (2001), 'A global optimization algorithm for nonconvex generalized disjunctive programming and applications to process systems', *Computers & Chemical Engineering* **25**, 1675–1697.

Lundgren, J., M. Rönnqvist and P. Värbrand (2010), *Optimization*, Vol. 1, Studentlitteratur AB.

Nikolaou, M., A.S. Cullick, L. Saputelli, G. Mijares, S. Sankaran and L. Reis (2006), 'A consistent approach towards reservoir simulation at different time scales'. *This paper was prepared for presentation at the 2006 SPE Intelligent Energy Conference and Exhibition held in Amsterdam, The Netherlands, 11-13 April 2006, SPE 99451.*

Nygreen, B., M. Christiansen, K. Haugen, T. Bjørkvoll and Ø. Kristiansen (1998), 'Modeling norwegian petroleum productionand transportation', *Annals of Operations Research* **82**, 251–268.

Offshore-technology (2012), 'Marlim oil field, brazil', [Online]. Available: `http://www.offshore-technology.com/projects/marlimpetro/`. [Accessed: 2013-05-09].

OilVoice (2012), 'Marlim oil field', [Online]. Available: `http://www.oilvoice.com/well/Marlim_Oil_Field/d3ece023f4ad.aspx`. [Accessed: 2013-05-09].

Quesada, I. and I. E. Grossmann (1992), 'An lp/nlp based branch and bound algorithm for convex minlp optimization problems', *Computers & Chemical Engineering* **16**, 937–947.

Raman, R. and I. E. Grossmann (1993), 'Symbolic integration of logic in mixed-integer linear programming techniques for process synthesis', *Computers & Chemical Engineering* **17**, 909–927.

Raman, R. and I. E. Grossmann (1994), 'Modelling and computational techniques for logic based integer programming', *Computers & Chemical Engineering* **18**, 563–578.

Saputelli, L., G. Mijares, J.A. Rodriguez, S. Sankaran, L. Garibaldi and K. Revana (2007), 'Managing risk and uncertainty in the visualization of production scenarios'. *This paper was prepared for presentation at the 2007 Hydrocarbon Economics and Evaluation Symposium to be held from April 1-3, 2007 in Dallas, TX, SPE 107562.*

Saputelli, L., M. Nikolaou and M.J. Economides (2005), 'Self-learning reservoir management', *SPE Reservoir Evaluation & Engineering* **8**, 534–547.

Saputelli, L.A., S. Mochizuki, L. Hutchins, R. Cramer, M.B. Anderson, J.B. Mueller, A. Escorcia, A.L. Harms, C.D. Sisk, S. Pennebaker, J.T. Han, A. Brown, C.S. Kabir, R.D. Reese, G.J. Nunez, K.M. Landgren, C.J. McKie and C. Airlie (2003), 'Promoting real-time optimization of hydrocarbon producing systems'. *This paper*

*was prepared for presentation at the Offshore Europe 2003 held in Aberdeen, UK, 2-5 September 2003, SPE 83978.*

Sausen, A., P. Sausen and M. de Campos (2012), *The Slug Flow Problem in Oil Industry and Pi Level Control*, *in* J. S.Gomes, ed., 'New Technologies in the Oil and Gas Industry', InTech.

Schlumberger (2005), 'Acting in time to make the most of hydrocarbon resources', *Oilfield Review* **17**, 4–13.

Schlumberger (2013*a*), 'Artifical lift', [Online]. Available: `http://www.slb.com/services/production/artificial_lift.aspx`. [Accessed: 2013-05-09].

Schlumberger (2013*b*), 'The schlumberger oilfield glossary', [Online]. Available: `http://www.glossary.oilfield.slb.com`. [Accessed: 2013-05-09].

Stubbs, R.A. and S. Mehrotra (1999), 'A branch-and-cut method for 0-1 mixed convex programming', *Mathematical Programming* **86**, 515–532.

Ulstein, N.L., Nygreen. B. and J.R. Sagli (2005), 'Tactical planning of offshore petroleum production', *European Journal of Operational Research* **176**, 550–564.

Vecchietti, A. and I.E. Grossmann (2000), 'Modeling issues and implementation of language for disjunctive programming', *Computers & Chemical Engineering* **24**, 2143–2155.

Wang, P. (2003), Development and applications of production optimization techniques for petroleum fields, Phd dissertation.

Wang, P., M. Litvak and K. Aziz (2002), 'Optimization of production operations in petroleum fields'. *This paper was prepared for presentation at the SPE Annual Technical Conference and Exhibition held in San Antionio, Texas, 2p September - 2 October 2002, SPE 77658.*

Westerlund, T. and F. Pettersson (1995), 'An extended cutting plane method for solving convex minlp problems', *Computers & Chemical Engineering* **19**, 131–136.

# Appendix A

# Mathematical models

## A.1 Declarations

### Sets and indices

| | | |
|---|---|---|
| $M$ | - | *set of manifolds, indexed by m* |
| $M^T$ | - | *set of topside manifolds, indexed by m* |
| $M^S$ | - | *set of subsea manifolds, indexed by m* |
| $J$ | - | *set of all wells, indexed by j* |
| $J_m$ | - | *set of wells of manifold m, indexed by j* |
| $L$ | - | *set of all pipelines, indexed by l* |
| $L_m$ | - | *set of pipelines of manifold m, indexed by l* |
| $P$ | - | *set of all phases, indexed by p ∈ o,w,g (oil, water, gas)* |
| $S$ | - | *set of separators, indexed by s* |

### Data

$P_s^S$ - *separator pressure for separator s*
$P_{mj}^{W,MAX}$ - *maximum well head pressure in well j of manifold m*
$Q_{mj}^{O,MAX}$ - *maximum oil production from well j of manifold m*
$Q_{mj}^{I,MAX}$ - *maximum lift gas injected in well j of manifold m*
$Q_{mj}^{I,MIN}$ - *minimum lift gas injected in well j of manifold m*
$C^G$ - *total gas handling capacity for platform*
$C_s^{LIQ}$ - *liquid handling capacity for separator s*
$C_s^W$ - *water handling capacity for separator s*
$C^{LG}$ - *liftgas capacity for platform*
$f_{mjo}^W(\,)$ - *function for oil flow from well j of manifold m*
$f_{ml}^L(\,)$ - *function for pressure drop in pipeline l of manifold m*

### Variables

Common variables

$q_{mjp}^W$ - *flow of phase p from well j connected to manifold m*
$p_{mj}^W$ - *pressure at well head for well j connected to manifold m*
$q_{mlp}^L$ - *flow of phase p in pipeline l connected to manifold m*
$p_{ml}^L$ - *pressure before separator inlet in pipeline l of manifold m*
$p_{ml}^M$ - *pressure in pipeline l at manifold m*
$q_{mj}^I$ - *amount of liftgas injected in well j connected to manifold m*
$p_{ml}^D$ - *pressure drop in pipeline l connected to manifold m*
$y_{mjl}$ - *1 if well j is routed to pipeline l through manifold m,0 otherwise*
$x_{mls}^S$ - *1 if pipeline l of subsea manifold m is routed to separator s,0 otherwise*
$x_{mjs}^T$ - *1 if well j of topside manifold m is routed to separator s,0 otherwise*

Additional flow variables for logic formulation

$q^{M}_{mjlp}$ - *flow of phase p from well j to pipeline l through subsea manifold m*
$q^{SS}_{mlsp}$ - *flow of phase p to separator s from pipeline l of subsea manifold m*
$q^{ST}_{mjsp}$ - *flow of phase p into separator s from well j of topside manifold m*
$q^{I}_{mjl}$ - *flow of liftgas from well j to pipeline l*
$q^{I}_{mjs}$ - *flow of liftgas from well j to separator s*

Boolean variables

$Y_{mjl}$ - *true if well j is routed to pipeline l of manifold m*
$X^{S}_{mls}$ - *true if separator s is open for routing from pipeline l*
$X^{T}_{mjs}$ - *true if separator s is open for routing from well j*
$W^{LG,OPEN}_{mj}$ - *true if well j of manifold m is open*

## A.2 MINLP formulation

**Objective function**

$$\max Z = \sum_{s\in S}\sum_{m\in M^T}\sum_{j\in J_m} q^W_{mjo}x^T_{mjs} + \sum_{m\in M^S}\sum_{l\in L_m} q^L_{mlo} \tag{A.1}$$

**s.t.**

**System specifications and capacities**

$$\sum_{s\in S}\sum_{m\in M^T}\sum_{j\in J_m} q^W_{mjg}x^T_{mjs} + \sum_{m\in M^S}\sum_{l\in L_m} q^L_{mlg} \leq C^G \tag{A.2}$$

$$\sum_{p\in\{o,w\}}\sum_{m\in M^T}\sum_{j\in J_m} q^W_{mjp}x^T_{mjs} + \sum_{p\in\{o,w\}}\sum_{m\in M^S}\sum_{l\in L_m} q^L_{mlp}x^S_{mls} \leq C^{LIQ}_s, \quad \forall\, s\in S \tag{A.3}$$

$$\sum_{m\in M^T}\sum_{j\in J_m} q^W_{mjw}x^T_{mjs} + \sum_{m\in M^S}\sum_{l\in L_m} q^L_{mlw}x^S_{mls} \leq C^W_s, \quad \forall\, s\in S \tag{A.4}$$

$$q^W_{mjo} \leq Q^{O,MAX}_{mj}, \quad \forall\, m\in M,\ j\in J_m \tag{A.5}$$

$$q^I_{mj} \leq Q^{I,MAX}_{mj}, \quad \forall\, m\in M,\ j\in J_m \tag{A.6}$$

$$p^W_{mj} \leq P^{W,MAX}_{mj}, \quad \forall\, m\in M,\ j\in J_m \tag{A.7}$$

$$q^I_{mj} \geq Q^{I,MIN}_{mj}y_{mjl}, \quad \forall\, m\in M^S,\ j\in J_m,\ l\in L_m \tag{A.8}$$

$$q^I_{mj} \geq Q^{I,MIN}_{mj}x^T_{mjs}, \quad \forall\, m\in M^T,\ j\in J_m,\ s\in S \tag{A.9}$$

$$\sum_{m\in M}\sum_{j\in J_m} q^I_{mj} \leq C^{LG} \tag{A.10}$$

**Well model**

$$q^W_{mjo} = f^W_{mjo}(p^W_{mj}, q^I_{mj}), \quad \forall\, m \in M,\ j \in J_m,\ p \in P \tag{A.11}$$

$$q^W_{mjg} = q^W_{mjo}\, GOR_{mj} + q^I_{mj}, \quad \forall\, m \in M,\ j \in J_m \tag{A.12}$$

$$q^W_{mjw} = \frac{q^W_{mjo}\, WC_{mj}}{1 - WC_{mj}}, \quad \forall\, m \in M,\ j \in J_m \tag{A.13}$$

$$q^L_{mlp} = \sum_{j \in J_m} q^W_{mjp} y_{mjl}, \quad \forall\, m \in M^S,\ l \in L_m,\ p \in P \tag{A.14}$$

**Pipeline model**

$$p^D_{ml} = f^L_{ml}(q^L_{mlw}, q^L_{mlg}, q^L_{mlo}), \quad \forall\, m \in M^S,\ l \in L_m \tag{A.15}$$

$$p^D_{ml} = p^L_{ml} - p^M_{ml}, \quad \forall\, m \in M^S,\ l \in L_m \tag{A.16}$$

**Pressure and routing constraints**

$$\sum_{l \in L_m} y_{mjl} \leq 1, \quad \forall\, m \in M^S,\ j \in J_m \tag{A.17}$$

$$p^M_{ml} y_{mjl} \leq p^W_{mj}, \quad \forall\, m \in M^S,\ j \in J_m,\ l \in L_m \tag{A.18}$$

$$\sum_{s \in S} x^S_{mls} = 1, \quad \forall\, m \in M^S,\ l \in L_m \tag{A.19}$$

$$P^S_s x^S_{mls} \leq p^L_{ml}, \quad \forall\, s \in S,\ m \in M^S,\ l \in L_m \tag{A.20}$$

$$\sum_{s \in S} x^T_{mjs} \leq 1, \quad \forall\, m \in M^T,\ j \in J_m \tag{A.21}$$

$$P^S_s x^T_{mjs} \leq p^W_{mj}, \quad \forall\, s \in S,\ m \in M^T,\ j \in J_m \tag{A.22}$$

**Non-negativity and binary constraints**

$$\begin{aligned}
&q^W_{mjp} \geq 0, && \forall\ m \in M,\ j \in J_m,\ p \in P \\
&q^L_{mlp} \geq 0, && \forall\ m \in M^S,\ l \in L_m,\ p \in P \\
&p^W_{mj}, q^I_{mj} \geq 0, && \forall\ m \in M,\ j \in J_m \\
&p^L_{ml}, p^M_{ml}, p^D_{ml} \geq 0, && \forall\ m \in M^S,\ l \in L_m
\end{aligned} \tag{A.23}$$

$$\begin{aligned}
&y_{mjl} = \{0,1\}, && \forall\ m \in M^S,\ j \in J_m,\ l \in L_m \\
&x^S_{mls} = \{0,1\}, && \forall\ m \in M^S,\ l \in L_m,\ s \in S \\
&x^T_{mjs} = \{0,1\}, && \forall\ m \in M^T,\ j \in J_m\ s \in S
\end{aligned} \tag{A.24}$$

## A.3 Logic formulation

**Objective function**

$$\max Z = \sum_{m \in M^T} \sum_{j \in J_m} q^W_{mjo} + \sum_{m \in M^S} \sum_{l \in L_m} q^L_{mlo} \tag{A.25}$$

**s.t.**

**System specifications and capacities**

$$\sum_{s \in S} \sum_{m \in M^T} \sum_{j \in J_m} q^{ST}_{mjsg} + \sum_{s \in S} \sum_{m \in M^S} \sum_{l \in L_m} q^{SS}_{mlsg} \leq C^G \tag{A.26}$$

$$\sum_{p \in \{o,w\}} \sum_{m \in M^T} \sum_{j \in J_m} q^{ST}_{mjsp} + \sum_{p \in \{o,w\}} \sum_{m \in M^S} \sum_{l \in L_m} q^{SS}_{mlsp} \leq C^{LIQ}_s, \quad \forall\, s \in S \tag{A.27}$$

$$\sum_{m \in M^T} \sum_{j \in J_m} q^W_{mjsw} + \sum_{m \in M^S} \sum_{l \in L_m} q^{SS}_{mlsw} \leq C^W_s, \quad \forall\, s \in S \tag{A.28}$$

$$q^W_{mjo} \leq Q^{O,MAX}_{mj}, \quad \forall\, m \in M,\ j \in J_m \tag{A.29}$$

$$q^I_{mj} \leq Q^{I,MAX}_{mj}, \quad \forall\, m \in M,\ j \in J_m \tag{A.30}$$

$$p^W_{mj} \leq P^{W,MAX}_{mj}, \quad \forall\, m \in M,\ j \in J_m \tag{A.31}$$

$$q^I_{mj} \geq Q^{I,MIN}_{mj}, \quad \forall\, m \in M^S,\ j \in J_m,\ l \in L_m \tag{A.32}$$

$$q^I_{mj} \geq Q^{I,MIN}_{mj}, \quad \forall\, m \in M^T,\ j \in J_m,\ s \in S \tag{A.33}$$

$$\sum_{m \in M} \sum_{j \in J_m} q^I_{mj} \leq C^{LG} \tag{A.34}$$

**Mass balances**

$$q^L_{mlo} = \sum_{j \in J_m} q^M_{mjlo}, \quad \forall\, m \in M^S,\ l \in L_m \tag{A.35}$$

$$q^W_{mjo} = \sum_{l \in L_m} q^M_{mjlo}, \quad \forall\, m \in M^S,\ j \in J_m \tag{A.36}$$

$$q^L_{mlp} = \sum_{s \in S} q^{SS}_{mlsp}, \quad \forall\, m \in M^S,\ l \in L_m,\ p \in P \tag{A.37}$$

$$q^W_{mjo} = \sum_{s \in S} q^{ST}_{mjso}, \quad \forall\, m \in M^T,\ j \in J_m \tag{A.38}$$

$$q^I_{mj} = \sum_{l \in L_m} q^I_{mjl}, \quad \forall\, m \in M^S,\ j \in J_m \tag{A.39}$$

$$q^I_{mj} = \sum_{s \in S} q^I_{mjs}, \quad \forall\, m \in M^T,\ j \in J_m \tag{A.40}$$

**Well model**

$$q^W_{mjo} = f^W_{mjo}(p^W_{mj}, q^I_{mj}), \quad \forall\, m \in M,\ j \in J_m,\ p \in P \tag{A.41}$$

$$q^M_{mjlg} = q^M_{mjlo}\, GOR_{mj} + q^I_{mjl}, \quad \forall\, m \in M^S,\ j \in J_m \tag{A.42}$$

$$q^M_{mjlw} = \frac{q^M_{mjlo}\, WC_{mj}}{1 - WC_{mj}}, \quad \forall\, m \in M^S,\ j \in J_m \tag{A.43}$$

$$q^S_{mjsg} = q^S_{mjso}\, GOR_{mj} + q^I_{mjs}, \quad \forall\, m \in M^T,\ j \in J_m \tag{A.44}$$

$$q^S_{mjsw} = \frac{q^S_{mjlo}\, WC_{mj}}{1 - WC_{mj}}, \quad \forall\, m \in M^T,\ j \in J_m \tag{A.45}$$

**Pipeline model**

$$p^D_{ml} = f^L_{ml}(q^L_{mlw}, q^L_{mlg}, q^L_{mlo}), \quad \forall\, m \in M^S,\ l \in L_m \tag{A.46}$$

$$p^D_{ml} = p^L_{ml} - p^M_{ml}, \quad \forall\, m \in M^S,\ l \in L_m \tag{A.47}$$

**Disjunctive constraints**

$$\begin{pmatrix} Y_{mj1} \\ q_{mj2o} = 0 \\ p^W_{mj} \geq p^M_{m1} \\ q^I_{mj} \geq Q^{I,MIN}_{mj} \end{pmatrix} \bigvee \begin{pmatrix} Y_{mj2} \\ q_{mj1o} = 0 \\ p^W_{mj} \geq p^M_{m2} \\ q^I_{mj} \geq Q^{I,MIN}_{mj} \end{pmatrix} \bigvee \begin{pmatrix} \neg Y_{mj1} \wedge \neg Y_{mj2} \\ \sum\limits_{l \in L_m} q_{mjlo} = 0 \\ q^I_{mj} = 0 \end{pmatrix}, \quad \forall\, m \in M^S,\ j \in J_m \tag{A.48}$$

$$\begin{pmatrix} X^S_{ml1} \\ q^{SS}_{ml2p} = 0 \\ q^{SS}_{ml3p} = 0 \\ p^S_1 \leq p^L_{ml} \end{pmatrix} \bigvee \begin{pmatrix} X^S_{ml2} \\ q^{SS}_{ml1p} = 0 \\ q^{SS}_{ml3p} = 0 \\ p^S_2 \leq p^L_{ml} \end{pmatrix} \bigvee \begin{pmatrix} X^S_{ml3} \\ q^{SS}_{ml1p} = 0 \\ q^{SS}_{ml2p} = 0 \\ p^S_3 \leq p^L_{ml} \end{pmatrix}, \quad \forall\, m \in M^S,\ l \in L_m \tag{A.49}$$

$$\begin{pmatrix} X^T_{mj1} \\ q^{ST}_{mj2o} = 0 \\ q^{ST}_{mj3o} = 0 \\ p^S_1 \leq p^W_{mj} \\ q^I_{mj} \geq Q^{I,MIN}_{mj} \end{pmatrix} \bigvee \begin{pmatrix} X^T_{mj2} \\ q^{ST}_{mj1o} = 0 \\ q^{ST}_{mj3o} = 0 \\ p^S_2 \leq p^W_{mj} \\ q^I_{mj} \geq Q^{I,MIN}_{mj} \end{pmatrix} \bigvee \begin{pmatrix} X^T_{mj3} \\ q^{ST}_{mj1o} = 0 \\ q^{ST}_{mj2o} = 0 \\ p^S_3 \leq p^W_{mj} \\ q^I_{mj} \geq Q^{I,MIN}_{mj} \end{pmatrix} \bigvee \begin{pmatrix} \neg X^T_{mj1} \wedge \neg X^T_{mj2} \wedge \neg X^T_{mj3} \\ \sum\limits_{s \in S} q^{ST}_{mjso} = 0 \\ q^I_{mj} = 0 \end{pmatrix}, \quad \forall\, m \in M^T,\ j \in J_m \tag{A.50}$$

**Non-negativity constraints**

$$\begin{aligned} &q^W_{mjp} \geq 0, && \forall\, m \in M,\ j \in J_m,\ p \in P \\ &q^L_{mlp} \geq 0, && \forall\, m \in M^S,\ l \in L_m,\ p \in P \\ &p^W_{mj}, q^I_{mj} \geq 0, && \forall\, m \in M,\ j \in J_m \\ &p^L_{ml}, p^M_{ml}, p^D_{ml} \geq 0, && \forall\, m \in M^S,\ l \in L_m \end{aligned} \tag{A.51}$$

$$\begin{aligned} &q^M_{mjlp} \geq 0, && \forall\, m \in M^S,\ j \in J_m,\ l \in L_m,\ p \in P \\ &q^{SS}_{mlsp} \geq 0, && \forall\, m \in M^S,\ l \in L_m,\ s \in S,\ p \in P \\ &q^{ST}_{mjsp} \geq 0, && \forall\, m \in M^T,\ j \in J_m,\ s \in S,\ p \in P \\ &q^I_{mjl} \geq 0, && \forall\, m \in M^S,\ j \in J_m,\ l \in L_m \\ &q^I_{mjs} \geq 0, && \forall\, m \in M^T,\ j \in J_m,\ s \in S \end{aligned} \tag{A.52}$$

**Boolean variables**

$$\begin{aligned} Y_{mjl} &= \{true, false\}, \quad \forall\, m \in M^S,\ j \in J_m,\ l \in L_m \\ X^S_{mls} &= \{true, false\}, \quad \forall\, m \in M^S,\ l \in L_m,\ s \in S \\ X^T_{mjs} &= \{true, false\}, \quad \forall\, m \in M^T,\ j \in J_m,\ s \in S \end{aligned} \tag{A.53}$$

## A.4 Reformulation of the logic formulation

**New disjunctive constraints**

$$\begin{pmatrix} W_{mj}^{LG,OPEN} \\ q_{mj}^{I} \geq Q_{mj}^{I,MIN} \end{pmatrix} \bigvee \begin{pmatrix} \neg W_{mj}^{LG,OPEN} \\ q_{mj}^{I} = 0 \\ \left( \sum_{l \in L_m} q_{mjlo} = 0 \right) \bigvee \left( \sum_{s \in S} q_{mjso}^{ST} = 0 \right) \end{pmatrix}, \quad \forall\, m \in M,\ j \in J_m \tag{A.54}$$

$$\begin{pmatrix} Y_{mj1} \\ q_{mj2o} = 0 \\ p_{mj}^{W} \geq p_{m1}^{M} \end{pmatrix} \bigvee \begin{pmatrix} Y_{mj2} \\ q_{mj1o} = 0 \\ p_{mj}^{W} \geq p_{m2}^{M} \end{pmatrix}, \quad \forall\, m \in M^S,\ j \in J_m \tag{A.55}$$

$$\begin{pmatrix} X_{mj1}^{T} \\ q_{mj2o}^{ST} = 0 \\ q_{mj3o}^{ST} = 0 \\ p_1^S \leq p_{mj}^{W} \end{pmatrix} \bigvee \begin{pmatrix} X_{mj2}^{T} \\ q_{mj1o}^{ST} = 0 \\ q_{mj3o}^{ST} = 0 \\ p_2^S \leq p_{mj}^{W} \end{pmatrix} \bigvee \begin{pmatrix} X_{mj3}^{T} \\ q_{mj1o}^{ST} = 0 \\ q_{mj2o}^{ST} = 0 \\ p_3^S \leq p_{mj}^{W} \end{pmatrix}, \quad \forall\, m \in M^T,\ j \in J_m \tag{A.56}$$

**Additional Boolean variable**

$$W_{mj}^{LG,OPEN} = \{true, false\}, \quad \forall\, m \in M,\ j \in J_m \tag{A.57}$$

# Appendix B

# Attached files

## B.1 Code files

AMPL code for MINLP formulation
AMPL code for NLP formulation
AMPL code for the logic branch and bound algorithm (LBB algorithm)
Matlab code for nonlinear approximations of well flows
Matlab code for nonlinear approximations of pipeline pressure drops
Data for the test case of P35

## B.2 Documents

PDF document of this thesis
readme.doc