NTNU – Trondheim
Norwegian University of
Science and Technology

# Simulation based optimization of petroleum production problems

Development of a special purpose B&B for a non-convex MINLP

Sheri S. Shamlou
Stine Ursin-Holm

Industrial Economics and Technology Management
Submission date: June 2013
Supervisor: Henrik Andersson, IØT
Co-supervisor: Vidar Gunnerud, Institutt for teknisk kybernetikk

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>**Ursin-Holm, Stine** | Fødselsdato<br>**22. jun 1988** |
| E-post<br>**stine.ursin.holm@gmail.com** | Telefon<br>**90197802** |

### 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>**Simulation based optimization of petroleum production problems** | |
| Oppgavetekst/Problembeskrivelse<br>The purpose of this work is to develop a simulation based optimization approach for short-term petroleum production optimization problems.<br><br>The problem instance is generic, so that it will match similar offshore platforms around the world and make the results broadly applicable. The planning horizon is typically one day to one week.<br><br>Main content:<br>1. Describe the optimization problem<br>2. Further develop and better the solution approach<br>3. Implement the developments<br>4. Perform a computational study<br>5. Compare the results to a benchmark approach | |
| Hovedveileder ved institutt<br>**Førsteamanuensis Henrik Andersson** | Medveileder(e) ved institutt |
| Ekstern bedrift/institusjon<br>**Institutt for teknisk kybernetikk** | 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, 23.01.2012
**Sted og dato**

Stine Vesin-Holm
**Student**

Henrik Sh
**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>**Shamlou, Sheri S.** | Fødselsdato<br>**15. okt 1987** |
| E-post<br>**shamlou@stud.ntnu.no** | Telefon<br>**45297899** |

## 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>**Simulation based optimization of petroleum production problems** | |
| Oppgavetekst/Problembeskrivelse<br>The purpose of this work is to develop a simulation based optimization approach for short-term petroleum production optimization problems.<br><br>The problem instance is generic, so that it will match similar offshore platforms around the world and make the results broadly applicable. The planning horizon is typically one day to one week.<br><br>Main content:<br>1. Describe the optimization problem<br>2. Further develop and better the solution approach<br>3. Implement the developments<br>4. Perform a computational study<br>5. Compare the results to a benchmark approach | |
| Hovedveileder ved institutt<br>**Førsteamanuensis Henrik Andersson** | Medveileder(e) ved institutt |
| Ekstern bedrift/institusjon<br>**Institutt for teknisk kybernetikk** | 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, 23.01.2013
........................................
**Sted og dato**

........................................
**Student**

........................................
**Hovedveileder**

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

# Problem Description

The purpose of this work is to develop a simulation based optimization approach for short-term petroleum production optimization problems.

The problem instance is generic, so that it will match similar offshore platforms around the world and make the results broadly applicable. The planning horizon is typically one day to one week.

**Main content:**

1. Describe the optimization problem
2. Further develop and better the solution approach
3. Implement the developments
4. Perform a computational study
5. Compare the results to a benchmark approach

# Preface

This Master thesis is written as a part of the Master Degree in Industrial Economics and Technology Management with specialization in Applied Economics and Optimization at the Department of Industrial Economics and Technology Management at the Norwegian University of Science and Technology (NTNU). The background for this thesis has been the authors' previous experience within petroleum production optimization, conducted for the Center for Integrated Operations (IO-Center) in cooperation with Petrobras and IBM. Some of the ideas presented here are based on the preliminary work done in TIØ-4500 during the fall of 2012.

We have enjoyed working on this thesis, developing solution methods, and testing and formalizing results on a typical oil production problem. This has given us insight into the varying optimization approaches studied and awareness around the challenges of the industry within production.

We would like to thank our supervisor, Associate Professor Henrik Andersson, for his time and for having constructive discussions with us and providing guidance. A special note of gratitude is in order for Post Doctor Vidar Gunnerud, our co-supervisor. He has devoted much of his time in helping us understand the problem, answer our questions and provide direction to this thesis. In addition, we would like to thank Petrobras, particularly Alex Teixeira for providing essential data regarding the P-35 platform.

Trondheim, June 2013.

Stine Ursin-Holm and Sheri Shamlou

*The picture on the cover page is taken from www.viking-systems.net*

# Summary

This work is concerned with the upstream and operational planning of petroleum fields. The short-term production optimization problem is modeled here as a simulation based non-convex MINLP problem. A special purpose B&B algorithm fit for manipulation is developed to allow sophisticated operations on each node of the B&B tree, such as running heuristics or implement non-convexity measures designed specifically for this problem.

Two types of heuristics are included to help obtain good feasible solutions quickly. A generic feasibility pump heuristic is modified to fit non-convex production allocation problems, creating high quality solutions. This heuristic can also function as a standalone solution method. In addition, two problem specific heuristics are made based on problem case knowledge and data analysis, giving good solutions fast. Specific non-convexity measures are included to avoid eliminating interesting parts of the solution space and help push towards finding the global solution. Pruning poses a challenge when solving non-convex problems with B&B, therefore negative gaps are allowed in order to assess the impact on the solution found. Further, the influence of including several starting points is studied.

Two different versions of the algorithm with different emphasis are designed, the first one on solution time, and the second more sophisticated one on solution quality. The results show quite a dramatic increase in solution time for the sophisticated version. The proposed algorithm is compared to other existing methods for optimizing the production allocation problem. In comparing emphasis has been on the tradeoff between solution time and quality, in addition to integration of the method into the daily work process of operating an offshore production platform. The version focusing on solution time is considerably faster, although both versions prove better than the other methods in terms of solution quality.

# Sammendrag

Dette arbeidet gjelder oppstrøms og operasjonell planlegging av petroleumsfelt. Det kortsiktige produksjonsoptimeringsproblemet er her modellert som et simuleringsbasert ikke-konveks MINLP problem. En spesiallaget B&B algoritme er utviklet med mulighet for avanserte operasjoner i hver node av B&B treet. Disse operasjonene består i å inkludere heuristikker eller å implementere tiltak for å håndtere ikke-konveksitet.

To typer heuristikker er inkludert for å oppnå gode tillatte løsninger raskt. En generell heuristikk kalt *feasibility pump* er tilpasset ikke-konvekse produksjonsproblemer, og gir løsninger av høy kvalitet. Denne heuristikken kan også fungere som en frittstående løsningsmetode. I tillegg er to problemspesifikke heuristikker laget basert på problemforståelse og dataanalyse, som gir gode løsninger raskt. Spesifikke tiltak for ikke-konveksitet er inkludert for å unngå å eliminere interessante deler av løsningsrommet. Det er en utfordring å kutte i B&B treet når man løser ikke-konvekse problemer, derfor tillates den øvre grensen å passere den nedre grensen i tresøket. Påvirkningen av å inkludere flere startpunkter er også studert.

To forskjellige versjoner av algoritmen er utformet, den første fokuserer på løsningstid, og den andre mer sofistikerte versjonen fokuserer på løsningskvalitet. Resultatene viser en dramatisk økning i løsningstid for den sofistikerte versjonen. Den foreslåtte algoritmen er så sammenlignet med andre eksisterende løsningsmetoder. I sammenligningen er løsningstid og løsningskvalitet vektlagt. I tillegg er integreringen av metoden i den daglige arbeidsprosessen ved operasjon av en offshore produksjonsplattform vurdert. Versjonen med fokus på løsningstid er betydelig raskere, selv om begge versjoner overgår de andre metodene i form av løsningskvalitet.

# Table of Contents


1. Introduction ..... 1
2. Background ..... 3
   2.1 Reservoir and Well ..... 3
   2.2 Production Principles ..... 4
   2.3 Work Process Offshore Field ..... 6
3. Literature Review ..... 9
4. Problem Description ..... 15
   4.1 The Marlim Field ..... 15
   4.2 The Production Asset ..... 16
5. Modeling the Production System ..... 19
   5.1 Simulation Based Optimization and SmartOpt Concept ..... 19
   5.2 Mathematical Model ..... 22
6. Algorithm ..... 29
   6.1 Branch and Bound ..... 29
   6.2 Heuristics ..... 30
      6.2.1 Feasibility Pump ..... 31
      6.2.2 Diving ..... 32
   6.3 Special Purpose B&B Algorithm ..... 32
      6.3.1 B&B Framework ..... 33
      6.3.2 Feasibility Pump for a Non-convex MINLP ..... 36
      6.3.3 Problem Specific Diving Heuristics ..... 43
      6.3.4 Non-convexity Measures ..... 44
7. Data and Implementation ..... 45
   7.1 Well and Pipeline Data ..... 45
   7.2 Test Cases ..... 46
   7.3 Implementation SmartOpt-B&B ..... 47
      7.3.1 B&B Framework ..... 47
      7.3.2 Heuristics ..... 48

7.3.3 Non-convexity Measures ........ 49
7.3.4 Simulator Implementation ........ 52
7.4 Implementation of Other Methods ........ 54
8. Computational Study ........ 57
8.1 Technical Testing ........ 57
8.1.1 Heuristics ........ 59
8.1.2 B&B Strategies ........ 62
8.2 Case Testing ........ 68
9. Discussion ........ 75
9.1.1 Modeling and Implementation ........ 75
9.1.2 Technical Aspects ........ 76
9.1.3 Comparing Solution Methods ........ 78
9.1.4 Work Process ........ 80
10. Conclusion ........ 81
11. Bibliography ........ 83
12. Appendix ........ 87

# Nomenclature

| | |
|---|---|
| B&B | - Branch and Bound |
| B&P | - Branch and Price |
| ESP | - Electrical Submersible Pumps |
| FP | - Feasibility Pump |
| FP-MILP | - Constraint relaxed MINLP, which becomes a (feasibility pump) MILP |
| FP-NLP | - Integer relaxed MINLP, which becomes a (feasibility pump) NLP |
| FP-NLP-R | - Reduced original MINLP, solved with binary variables locked to discrete values |
| FPSO | - Floating Production Storage and Offloading unit |
| GOR | - Gas-Oil-Ratio |
| H1 | - Problem Specific Heuristic 1 |
| H2 | - Problem Specific Heuristic 2 |
| IP | - Integer Program |
| LB | - Lower Bound |
| LP | - Linear Program |
| MILP | - Mixed Integer Linear Program |
| MINLP | - Mixed Integer Nonlinear Program |
| MP | - Mathematical Program |
| NA | - Nonlinear Approximation model |
| NLP | - Nonlinear Program |
| OFV | - Objective Function Value |
| SOS2 | - Special Ordered Sets of type 2 |
| SQP | - Sequential Quadratic Programming |
| WC | - Water Cut |

# 1. Introduction

Petroleum, also called crude oil, is found under the Earth's surface in reservoirs. It is a naturally occurring flammable liquid made up of hydrocarbons and other liquid organic compounds. Petroleum can have varying chemical configuration, and is therefore refined and separated into a large number of products. Examples are gasoline, kerosene, asphalt and chemical reagents used to make plastics and pharmaceuticals.

The petroleum consumption of the world has increased immensely since the dawn of large scale petroleum production in the early 20$^{th}$ century. It is a global industry with many players present throughout the value chain, normally divided between downstream, midstream and upstream. Many of todays' producing fields are maturing and oil companies often use technology, for example gas lift, to maintain or increase production and remain competitive.

Development of a petroleum field requires planning on multiple horizons. A common division is between strategic, tactical and operational planning. Strategic planning involves long-term planning, usually up to 5-20 years ahead in time and seeks to optimize the total drainage of reservoirs and make investment decisions to facilitate this. Tactical planning is medium-term, and can be planning anywhere between 6 months and 3 years ahead in time. Operational planning deals with the short-term, normally a few days up to a couple of weeks ahead in time.

This work is concerned with the upstream, operational planning of petroleum fields. Short-term production optimization in the oil industry aims at maximizing the oil production right now. Long term strategies regarding reservoir drainage are communicated to the production engineers by setting production targets or limits on well operations. In addition, the platforms usually have limited capacity regarding gas and liquid production, and a limited amount of gas available for gas lifting. Having all these restrictions in mind, the production engineers need to find the optimal schedule which gives the maximum oil production. In addition to a focus on and elaborate discussion about optimization solution methods, this thesis is also concerned with the applicability of the algorithms to the workflow of the users of such software.

The motivation for this Master thesis has been the authors' previous experience with solving short-term petroleum production optimization problems using a range of different models and solution methods. Based on the results of a comparative study of solution methods conducted in preparation of this work (Shamlou and Ursin-Holm, 2012), the most promising method was chosen for further inspection and development in this thesis. SmartOpt is based on dividing larger network simulators into their smaller parts, and bringing the relevant smaller simulators into the optimization directly. In the preparatory work the general Mixed Integer Nonlinear

Programming (MINLP) solver Bonmin was used with its Branch and Bound (B&B) algorithm. In order to develop the SmartOpt solution method further in accordance with the authors' ideas, it is necessary to design a B&B algorithm more fit for manipulation. This is done to allow sophisticated operations on each node of the B&B tree, for instance running heuristics for obtaining good feasible solutions. The combination of SmartOpt with this tailored B&B algorithm is called the SmartOpt-B&B solution method.

The problem case chosen is somewhat fictional, although it is inspired by the Marlim P-35 field. It is constructed in a way as to be generic, so that it might match similar offshore platforms around the world and make the results broadly applicable. It includes both topside wells and subsea wells, routing and gas lift technology. The problem is treated independent of time dynamics. However, the production planning problem is complex, and gives rise to non-convexities in the model formulation. In this work no attempt has been made to decrease the non-convexity of the model by using linearization techniques. Instead, special attention has been given to this adversity when designing the branch and bound algorithm. The model has many nonlinear relations, in addition to binary variables making it a non-convex MINLP problem, perhaps the most complex class of all optimization problems.

To summarize, a generic short-term oil production problem is modeled according to the SmartOpt concept, bringing simulators into the optimization. Because of non-convexities, nonlinearities and binary variables, a special purpose B&B algorithm is designed and developed, including heuristics for faster convergence. This proposed solution method will be analyzed and compared to other solution methods, and discussed in terms of pure optimization benchmarks and the applicability and value it can bring to the users of such algorithms.

The rest of this text is organized as follows. First background material is presented and definitions regarding petroleum production are given in Chapter 2. Chapter 3 contains a literature study conducted in relation with the problem. In Chapter 4 the asset and problem to be solved are described in detail, while Chapter 5 gives the formal mathematical formulation of the problem. General remarks related to the problem type are discussed in Chapter 6, before the proposed solution algorithm is explained in detail. Chapter 7 covers problem data and implementation of the proposed algorithm, followed by short introductions to the varying solution methods included for comparison purposes. In Chapter 8 the results of the computational study are given which are then discussed in Chapter 9. Finally Chapter 10 gives some concluding remarks. The bibliography is in Chapter 11. The Appendix includes a list of attached electronic files.

# 2. Background

This chapter presents terminology to provide a deeper understanding of properties, production principles and dynamics of an oil field. The Marlim Field offshore Rio de Janerio, Brasil, is used as a basis for the information presented in this chapter, and the particular field is considered representative for the general production principles of an offshore oil field. Firstly, an explanation of reservoir and well terminology is provided, and then further details about the production principles and issues are discussed. Further, an explanation of the challenges related to distinct phases of operation and declining productions is given. Finally, the work process of a production engineer in relation to production optimization is discussed.

## 2.1 Reservoir and Well

A reservoir is a rock body where hydrocarbons of several types such as natural gas, condensates, liquid hydrocarbons and water are stored (Alpha Thames Ltd, 2004). The three basic types of reserve phases are gas, oil and water, typically occupying different parts of the reservoir. Production systems are developed with the purpose of extracting the majority of these reserves. Extraction of fluids causes changes in the composition of oil, gas and water within the reservoir. Gas and water production are often described through volume fractions. The ratio of produced gas to produced oil is called gas-oil-ratio and commonly abbreviated GOR, whereas the ratio of produced water to produced liquids is called water cut and commonly abbreviated WC.

Figure 2.1 is an illustration of a production system with labels that will be explained here. A production well is a borehole drilled with the purpose of extracting the fluids from a reservoir. The wellbore constitutes the bottom part of the well, whereas the wellhead is the surface termination of the wellbore, which contains the interface for the drilling and production equipment. The pipelines from the wellbore to the wellhead are called tubing, and the pipelines from the seabed to the platform are called riser pipelines. In the remaining text *pipeline* means such a riser pipeline. Satellite wells are drilled offshore and connected to a platform or a Floating Production Storage and Offloading unit (FPSO), to capture the hydrocarbons in the outskirts of a reservoir. Manifolds are arrangements of pipes and valves by which the flows from several sources are directed to process systems on the platform. Typically, the satellite wells are connected directly to a topside manifold located at the platform, and in this paper the terms satellite well and topside well are used interchangeably to indicate this type of well. The remaining wells are connected to subsea manifolds and are here called subsea wells. At the platform the satellite wells and the riser pipes are routed to separators for processing.

Normally reservoir dynamics are slow with minimal changes over weeks and months, so the properties and compositions within the reservoir change gradually over the production life time.

It may take months before significant changes in the gas-oil-ratios and water cuts occur, typically resulting in an increasing share of gas and/or water over time. If the reservoir dynamics are in-fact slow, GOR and WC can be assumed constant for the short-term planning period with a time horizon of days up to a week. In addition pipeline dynamics are fast and the flow of oil, gas and water through pipelines can be assumed to be steady-state. As a result, when dealing with a short-term production period the influence of time can be neglected.

## 2.2 Production Principles

For a well to be producing such that the hydrocarbons flow from the reservoir through the well and up to the surface, the pressure differences must be sufficient for driving the flow through the whole production system, i.e. the pressure upstream the system must be higher than the pressures downstream.

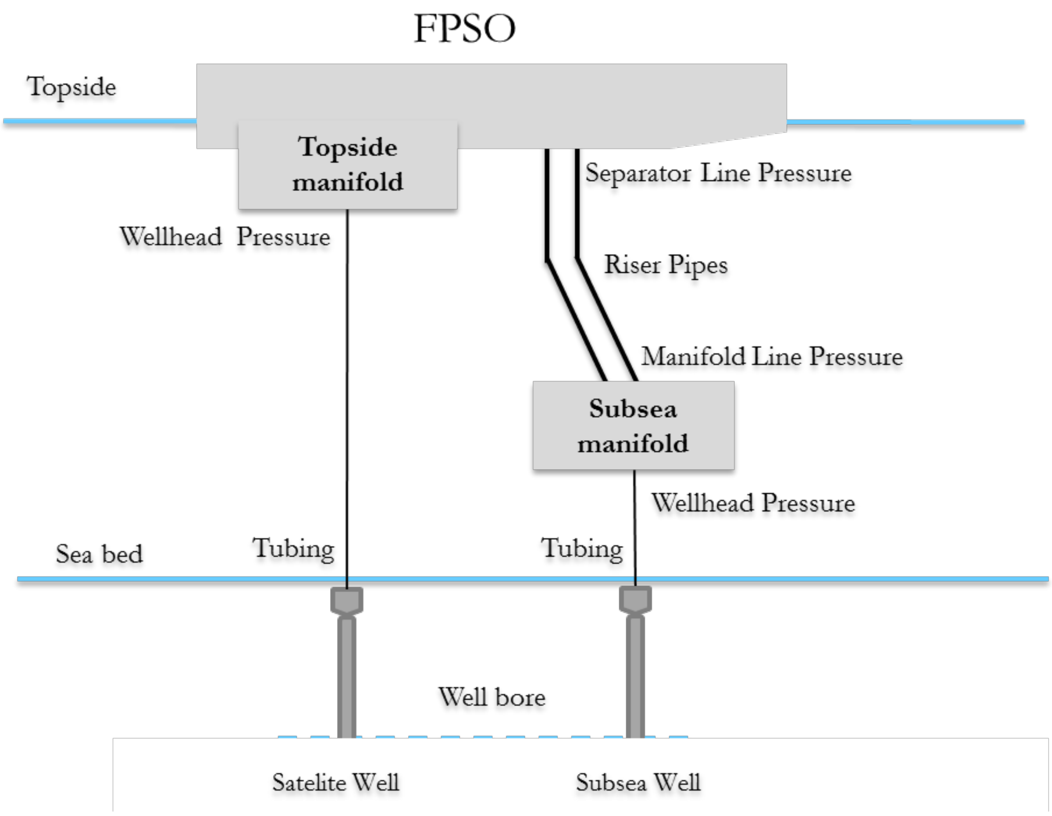


**Figure 2.1: Production System**

The reservoir pressure is defined in the reservoir, whereas the bottom hole pressure is in the wellbore and determines the flow rate. Wellhead pressure is defined at the outlet of the flow line just before it enters the manifold. Manifold line pressure is measured in the pipeline just as the flow leaves the manifold. Pressure losses occur throughout the system, however for the work of this thesis the assumption is that pressure losses only arise over the pipelines, due to friction and declining hydrostatic pressure. Separator line pressure is defined as the pressure over the pipeline just as the flow enters a separator. Each separator has a distinct inlet pressure that must be respected. Chokes control the flow rates, which regulate the pressures in the various parts of the system. Such regulation might be necessary if there are limited capacitates on the platform restraining total production.

The lifetime of a field can be divided into production phases with distinct characteristics and technical challenges, see Figure 2.2. Typically, in the early stage the underground pressures in the reservoir are sufficient for a natural lift of fluids to the surface after drilling has begun. During this stage the total production rates of the field will increase as the system reaches its full potential. After a while the field reaches a mature phase and total oil production remains more or less constant at its target level. In this phase the production might be limited by the topside capacities for handling fluids such as water and gas, or by a general attempt to minimize the wear and tear on the system by limiting factors such as velocities or temperatures. Eventually the pressures will decline and so will the natural lift capacities. Petroleum fields that have been operated for a few decades often experience declining production. At this point meeting production targets might become nontrivial and the objective becomes to maximize oil production while adhering to all relevant restrictions. To increase flows, artificial lifting technologies should be investigated such as water or gas injection, Electrical Submersible Pumps (ESP), gas lift, etc.

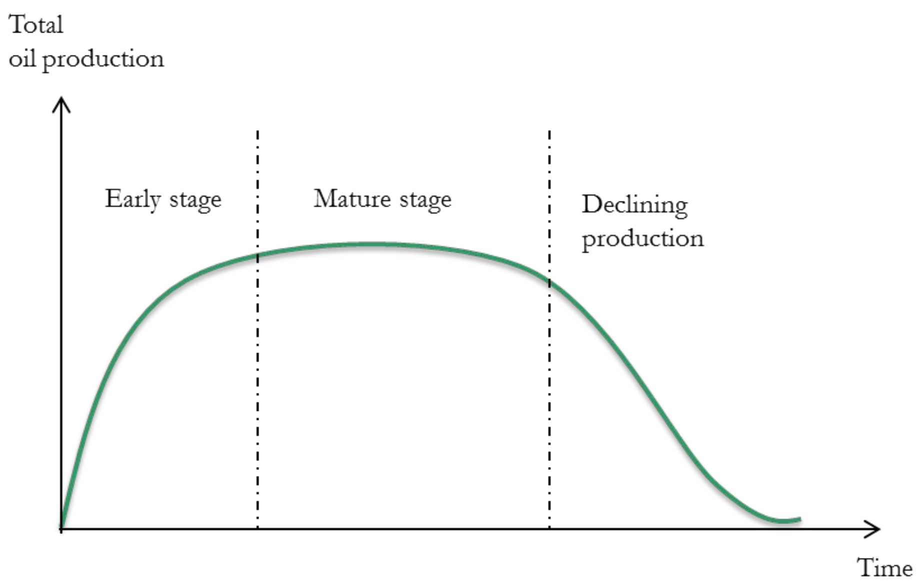


**Figure 2.2: Production Stages**

Artificial gas lift technology can be installed to lift more fluids from the reservoir. In some wells the wellhead pressure is not sufficiently high to support liquid flow up towards the platform. For this reason some wells have a connected gas lift pipe which transports gas from the platform down to the well. Gas is injected into the tubing to reduce fluid density and increase the pressure in the wellhead, resulting in higher pressure differences in the system. The gas is stored in a compressor of limited size and the gas lift might be a restricted resource. Therefore the gas must be carefully distributed between the different wells to maximize the utility value. A producing platform should only allocate gas lift to the wells that have fully open chokes and insufficient natural lift, since whenever a well is choked back the natural lift is adequate and gas lift not really needed. Choking back a well and at the same time using gas lift is comparable to driving a car while both accelerating and breaking. Assuming no pressure losses this indicates that when gas lift is allocated to a well its wellhead pressure should be equal to the successive pressure downstream.

## 2.3 Work Process Offshore Field

Reservoir dynamics and operational conditions determine the frequency of repeating the "work process" when operating a production platform. Depending on these dynamics and conditions, changes in operational settings are typically implemented from once a day up to once a week. Generally, constantly changing operational conditions require daily re-optimization of production strategies.

The daily work process should ensure improvement in production while honoring system and process limitations. A sequence of actions defining a typical daily work process on a modern field is described here, see Figure 2.3. Initially, the onshore operators, normally production engineers, study the production history of the offshore field within the last week, with emphasis on the last 24 hours to identify changes in well performances. Subject to limitations set by the process and reservoir engineers, experience and decision support tools such as simulators and mathematical optimization guide the production engineers in deriving a production strategy for the next period. Further discussions with the offshore operators are then conducted before deciding upon the final strategy which the offshore operators will then implement.

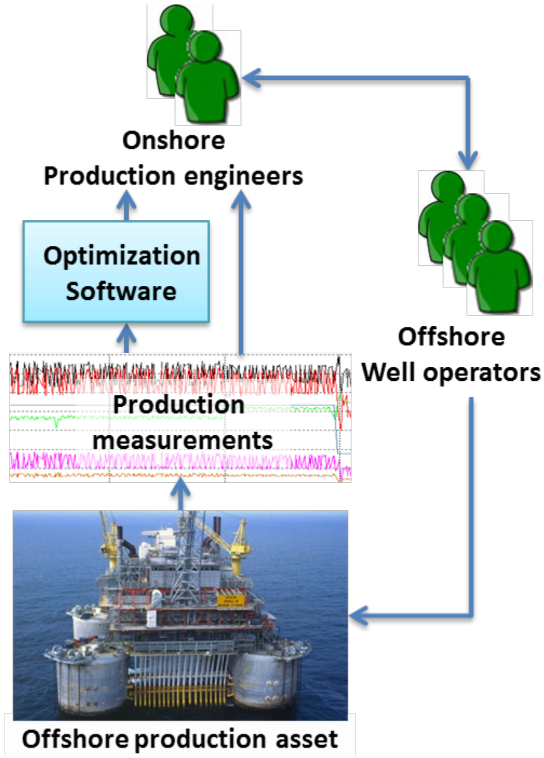


**Figure 2.3: Work process**

These discussions can go through several channels throughout the day, e.g. chat or phone, but scheduled video meetings are common for coordinating the activities of the day and deciding upon what changes to implement. In the context of using mathematical optimization to aid this discussion, solution time is of great importance. If the solution time of an optimization algorithm is relatively low, e.g. under 5 minutes, this will motivate using the optimization in the discussions between on- and offshore. Based on the topics discussed, several runs of the algorithm are possible in order to see different results and hopefully aid the participants in making the right decisions.

If this solution time is long (hours), it is quite natural that the participants will not have the time nor patience to wait for answers from the optimization algorithm during a discussion. The optimization is then often started in the evening when the production engineer ends his day, and the results are ready for him early the next morning. The results can still be valuable and provide decision support for the production engineer, but the optimization is used in a different way than if it is able to provide solutions quickly. In a sense, the solution time will determine the frequency with which optimization will be used and consequently to what degree it contributes as a decision support tool for the users.

In addition to the daily activities of the production engineers and offshore operators, coordination with other departments must also take place regarding production optimization. The frequency of interaction varies, and is highly asset dependent. The most important surface is towards the reservoir engineer, another is towards the topside process engineer. The optimization software can also be a valuable addition to the communication and planning that takes place between these different departments, and the issues related to solution time also apply here (Almklov and Gunnerud, 2013).

An asset's production system model which is used in the optimization cannot, practically speaking, include everything related to the problem. In other words, it is a model and an approximation of reality. If it could, it would not be necessary to run the optimization repeatedly because the global optimal solution would be available after one run. Instead, relevant assumptions and simplifications can lead to a solvable model giving results within a reasonable amount of time. Certain aspects of the problem are then inevitably neglected. This might necessitate several runs of optimization to capture different sides of the problem. The expertise and experience of the on- and offshore workers is very important in such optimization work. It is the integration of the knowledge and capabilities of the production engineer, the offshore operator and the optimization algorithm that provides good feasible solutions to the production optimization problem.

# 3. Literature Review

A substantial amount of literature has been produced regarding petroleum production optimization, focusing on both different planning horizons and varying parts of the value chain. In 1990, a history of Mathematical Programming (MP) in the petroleum industry was written by Bodington and Baker (1990). The article reviews the MP usage from the 1940s until 1990, concluding that most of the effort up until then had been with linear models, but asserting the potential of MP and foreseeing great activity in the field, including nonlinear optimization.

Concentrating on the short-term production optimization problem, there are still many different system descriptions and solution methods available in the literature. Even though the petroleum production optimization problem is highly nonlinear at its core, the bulk of the literature concerns linearizing the problem in some way in order to use Linear Programming (LP) solution methods. Other simplifications and assumptions are widespread, apparently to gain efficient solution methods in terms of computational time.

Glæserud and Syrdalen (2009) model operational production optimization at Troll West. This work includes system pressure and multiphase flow, but has no routing between wells, pipelines or separators and well production rates are crudely linearized. Uncertainty in system capacities and well production outputs are however incorporated, necessitating stochastic solution methods. Gunnerud and Foss (2010) also describe production planning optimization at Troll West, but include gas coning and routing of wells leading to a Mixed Integer Linear Programming (MILP) model. This is achieved by using piecewise linear approximations and Special Ordered Sets of type 2 (SOS 2), solved by using branch and bound and tested using both Lagrangian- and Dantzig-Wolfe decomposition. The paper concludes that the latter decomposition technique is most promising. Foss et al. (2009) give an introduction to decomposition of petroleum optimization problems.

Torgnes et al. (2012) and Gunnerud et al. (2011) also model large scale petroleum production networks with solution methods based on piecewise linearization of the nonlinearities through SOS 2 sets. The allocation problems include routing and become MILP models. Torgnes et al. (2012) show the increased computational efficiency the parallel Dantzig-Wolfe decomposition provides for solving the allocation problem. Gunnerud et al. (2011) decomposes the problem into smaller subproblems, and uses column generation through a Branch and Price (B&P) framework to solve the decomposed problem.

As mentioned earlier, gas lifting technology is becoming more and more common as a way of counteracting the declining oil production of maturing fields. There has been a steady increase of interest and literature adding gas lift modeling into the oil production optimization problem.

Camponogara and Nakashima (2006) aim at solving the gas lift allocation problem, but simplify the model by ignoring the pressures throughout the system. The oil production rate at each well is nonlinearly related to the gas lift injected into that well, which is proposed piecewise linearized by the authors leading to a MILP model. Multiphase flow is not modeled directly, rather the phase fractions are included in a profit function being maximized. In addition, lift gas economics are included in the objective function.

Dzubur and Langvik (2012) also include optimal gas lift allocation into the short-term production optimization problem. This work can be seen as an extension to Gunnerud and Foss (2010), with many of the same problem components and solution techniques, but applied to the Petrobras Marlim field in Brazil. This is a maturing brown field, and using lift gas technology is essential in maintaining the oil production rates.

Literature consisting of nonlinear solution methods is also increasing, with the derivative based optimization algorithm Sequential Quadratic Programming (SQP) being a popular method. Wang (2003) broadly describes optimization methods used within the petroleum industry in his PhD thesis. A short-term production optimization model including gas lift is studied, and several solution methods are suggested. Notably, he develops a rate allocation approach solved by SQP. Díez et al. (2005) also analyzes the use of SQP to solve operational production optimization problems.

A mixed integer nonlinear programming model for daily well scheduling is presented in Kosimidis et al. (2005). Nonlinear reservoir behavior, multiphase flow, gas lifting and platform capacity constraints are taken into account. Routing options account for the discrete decisions, while well production and optimal gas lift allocation comprise the continuous variables. The authors propose to solve the problem using logical constraints, piecewise linearization of some of the nonlinearities and an outer approximation algorithm. This problem formulation is similar to the problem instance in this thesis, although the solution method is different.

The petroleum production problem is inherently a MINLP problem, an especially challenging class of optimization problems covered by significant amounts of literature. Most notably the literature covers the convex MINLP about which good surveys are readily available, examples are Bonami et al. (2012), Leyffer et al. (2009) and Grossmann (2002). Limited literature and experiments are available considering non-convex MINLP problems however, even though this class comprises many real-life applications. An excellent survey is conducted by Burer and Letchford (2012) discussing non-convex MINLP applications, exact approaches and heuristic solution methods.

Several heuristics have been developed for providing feasible solutions to MILPs and further developed to fit MINLPs. Grötschel and Padberg (1979) give a short presentation to heuristics solving the travelling salesman problem, which was one of the first combinatorial problems to be efficiently solved by heuristics. Berthold (2006) gives a presentation of construction heuristics for MILP and convex MINLP problems. An example is the rounding based so called diving heuristic. The heuristic bounds or fixes the binary variables of the IP problem to promising 0/1 values and resolve the linear or Nonlinear Program (NLP) iteratively, until all values are integer. These heuristics quickly go down the branch and bound tree, hereby the name diving.

Bertaccoa et al. (2007) address the problem of finding a solution to a generic MILP problem, and call their heuristic the Feasibility Pump (FP). The purpose of the heuristic is to provide a good feasible solution quickly. The algorithm alternates between rounding of continuous binary values and solving an LP problem until a constraint feasible integer solution is found. Bonami et al. (2009) suggest further development of the FP for providing feasible solutions to convex MINLP problems. They suggest an algorithm that alternates between solving NLPs and MILPs. This algorithm exploits outer approximation techniques and also suggests how the first feasible solution can be improved. Recent efforts have been made in order to develop a FP for non-convex MINLP problems. D'Ambrosio et al. (2012) presents a FP tailored for non-convex MINLP problems. They suggest that the first problem becomes a non-convex NLP problem solved with global optimization techniques whereas the second problem becomes a MILP problem complemented by a tabu list. Sharma (2012) presents the different concepts of the FP for MILPs, convex MINLPs and non-convex MINLPs.

The merging of optimization and simulation technologies has seen rapid growth in recent years, and April et al. (2003) provides an introduction to the realm of optimizing simulated systems. This papers lists some classical approaches for simulation optimization, among these are stochastic approximation (gradient-based approaches), (sequential) response surface and metamodel methodologies and random search algorithms. The paper uses an informal definition of a simulation model taken from Law and Kelton (1991) as a "mechanism that turns input parameters into output performance measures". In other words, this "mechanism" can be thought of as a function relating inputs to outputs, a view that has motivated the family of simulation optimization approaches based on response surfaces and metamodels. A response surface is "in essence a plot that numerically characterizes the unknown (simulator) function" April et al. (2003). This can be obtained by running the simulation model over a list of specified values for the input parameters and recording the responses. Fu et al. (2005) is a more recent paper elaborating on the same issues regarding simulation optimization.

A metamodel is an algebraic model of the simulator function, approximating the response surface. Using this approach with nonlinear approximations can be found in Gunnerud et al. (2013). This paper proposes a solution algorithm for optimizing general process industry type problems, with particular applicability to the oil and gas industry. The algorithm seeks to take advantage of the structure of the problem, and decomposes it into smaller parts whose behavior is nonlinearly approximated using the underlying simulator data. Shamlou et al. (2012) applies this technique to a small operational production optimization problem, using local approximations that are iteratively updated as the algorithm moves around in the solution space. Both of these papers address MINLP formulations.

Another way of approaching the approximation is by building nonlinear functions representing the underlying simulator data globally. This approach is taken in Ursin-Holm (2012), where the real production optimization problem of the P-35 field of Petrobras is modeled. This report uses data generated from the Petrobras in-house simulator Marlim-II, and creates nonlinear functions with global validity that are included in the model and solution method, and which characterize the nonlinear realities of the problem.

Instead of approximating the simulator data, another idea is to include relevant simulators directly. Such an approach might be computationally inefficient however, if the simulators require significant runtime to output results. The upside is increased accuracy in using exact simulator output values. In a report written for the Center for Integrated Operations (IO-Center) at NTNU, Shamlou (2012) solves the real production optimization problem of the P-27 field belonging to Petrobras. In the model the nonlinear relationships are given as exact functions, i.e. simulators connecting inputs to outputs, instead of the usual way of approximating them. When the solution algorithm is run, these functions are evaluated by invoking the relevant simulators. The solution method seems promising, although the particular case studied is rather small and does not include any subsea manifolds and consequently no routing. The problem instance only has topside connected wells, including binary variables for indicating their on or off status.

This literature study was started by referring to Bodington and Baker (1990) and the historical review of using mathematical programming in the petroleum industry. At the end of that article the authors make some predictions about the future of MP in the industry. And quite interestingly, their predictions have more or less come true, something that is evident from this literature study. They predicted more integrated models spanning the value chain, "direct optimization of simulations" and finally that nonlinear optimization will become more widespread.

On that note, and for reasons previously spelled out in the introduction to this text, we see this work as an interesting addition to the literature. We are solving a generic short-term petroleum production planning problem including system pressures and multiphase flow, gas lifting technology, routing options and platform capacity constraints. The model is kept and solved as a non-convex MINLP, where the system as a whole is split up into its smaller components bringing simulators of those components directly into the optimization.

# 4. Problem Description

In this chapter the Marlim field and the petroleum production asset addressed are introduced. The production system is illustrated in Figure 4.2 and is designed to be representative of a typical offshore production system.

## 4.1 The Marlim Field

The Marlim field is a subsea field located in the north-eastern part of the Campos Basin, approximately 110 km offshore Rio de Janeiro State, Brazil, with a varying water depth of 650 to 1050 meters. Explorations started in 1985 while production began in 1991. Today there are approximately 100 wells producing together with 50 injection wells. The field holds 8 operating FPSO units. It is expected that the Marlim field will be generating oil and gas until 2030 (Offshore Technology, 2012).

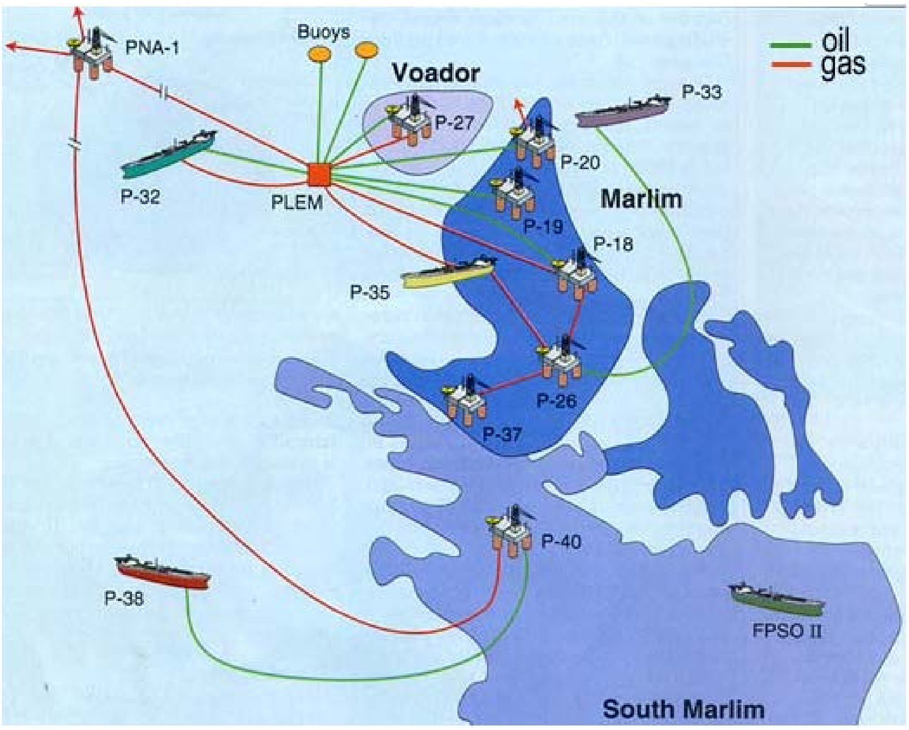


**Figure 4.1: Map Marlim Field**

The reservoirs in the Marlim Field are known to be water driven, thus water replaces the oil within the reservoir as the oil is extracted. As a result the WCs are gradually increasing and the GORs are changing. Furthermore, as the Marlim field is gradually becoming a mature field, the reservoir pressures and production rates are declining.

## 4.2 The Production Asset

In the following section the production platform addressed in this thesis is presented and described from bottom to top according to Figure 4.2. An FPSO operating in the Marlim field has been used as a template for the structure of this particular production system which consists of 16 wells. Out of these, 6 are satellite wells connected directly to the platform/FPSO, the remaining 10 are wells connected to subsea manifolds, 6 to subsea manifold 1, and 4 to subsea manifold 2. Each subsea manifold is attached to the platform through two riser pipes, and the subsea wells are only routed to one out of these two pipes.

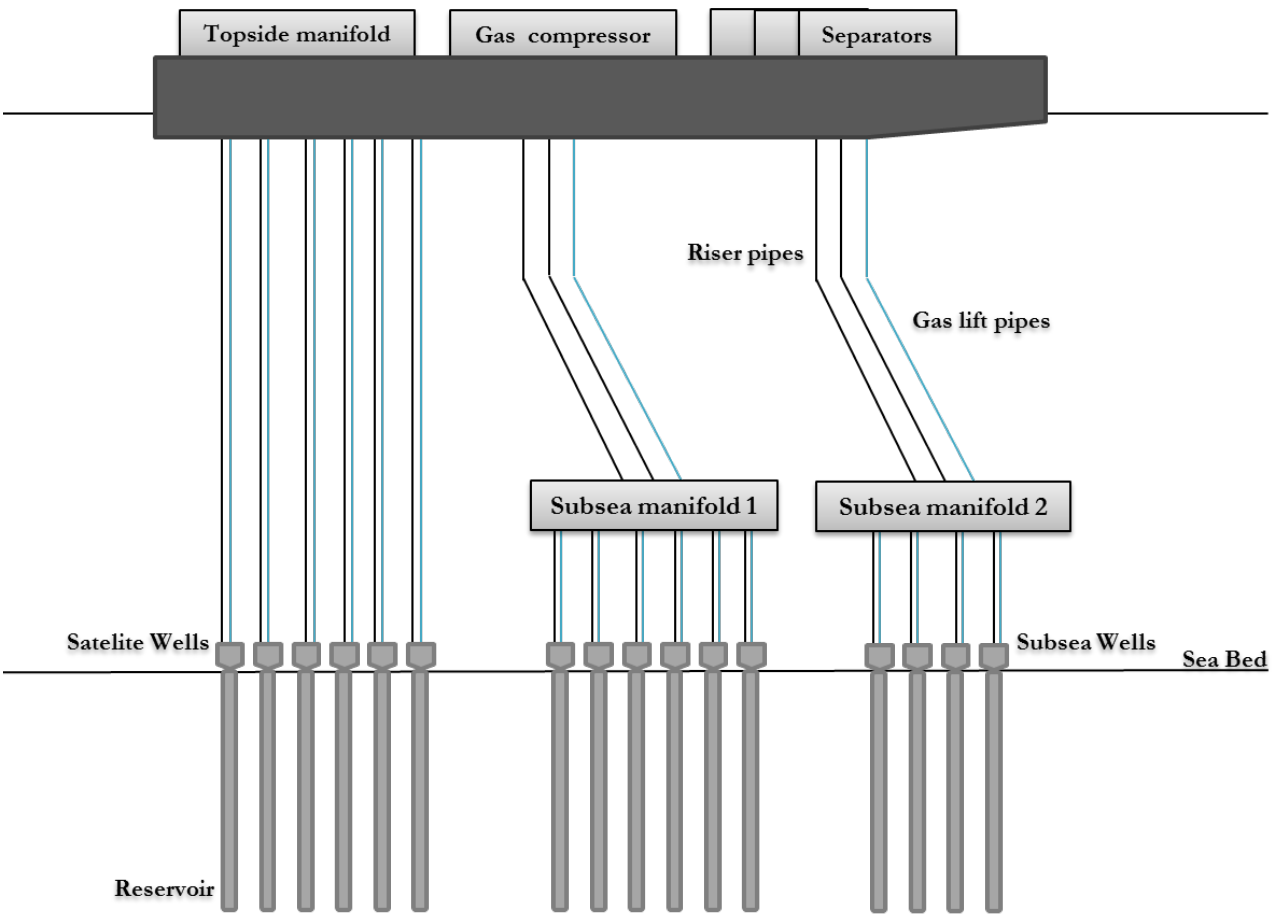


**Figure 4.2: Conceptual Layout of the Production System**

Three separators are available at the platform requiring the flow from the satelite wells and riser pipes to be routed to one of these separators. Wellhead pressures of open satellite wells need to be higher than or equal to the pressure of the respective separator. When such a well is fully open, its wellhead pressure equals the inlet pressure of the separator it is connected to. Otherwise the wellhead pressures are higher than the separator pressures, and the chokes are used to regulate the flow. Flow of producing subsea wells are led to riser pipes, and the wellhead pressures must be higher than or equal to the respective manifold line pressure. If the well is fully open the wellhead pressure will equal the manifold line pressure, while it will be higher in any other case. Pressure losses occur over the pipes, and to ensure flow the separator line pressures must be equal to or greater than the inlet pressure of the separator the pipe is connected to. All the wells have the option of artificial gas lift, thus production is dependent on pressure differentials and gas lift rates.

The platform has capacity limits for handling gas, water and liquid which cannot be exceeded. Additionally, there are restrictions on choke changes for certain wells, and the total amount of gas lift available for allocation is limited. Within the timeframe of this short-term optimization the GORs and WCs are constant for each well respectively.

**Production Optimization**

This petroleum production asset gives rise to an optimization problem. Optimization methods can provide an optimal operating plan for the asset. The method decides upon the optimal values on relevant variables in order to maximize oil production while at the same time fulfills the requirements of the mass balances, capacities and pressure constraints of the system.

Some implicit decision variables are necessary in the optimization model so that the model is correct considering the physics of the system. However, when controlling the system these variables will take values as a result of the explicit decision variables. These implicit decision variables are the well production rates, the pressures in the pipelines as the flow leaves the manifold, the pressure loss over the pipelines and the flow through the pipelines. The explicit decision variables are, starting with the binary variables 1) on/off and routing of satellite wells to the separators, 2) on/off and routing of the subsea wells to pipelines and 3) routing of riser pipes to the separators. The continuous control variables are 4) what wellhead pressures should be set for every well, and finally 5) lift gas allocation.

# 5. Modeling the Production System

In the realm of optimization, it is necessary to formulate the problem to be solved mathematically in an optimization model. The model and notation should also "facilitate the conversion into a modeling language" (Lundgren et al., 2010). This chapter includes first an explanation of the SmartOpt concept used for modeling purposes, which motivates the actual mathematical model chosen to represent the problem. Subsequently the complete mathematical model is given including sets, indices, parameters, variables, objective function and constraints.

## 5.1 Simulation Based Optimization and SmartOpt Concept

Simulators have entered many engineering disciplines, no doubt because of their contribution in modeling complex systems. However, detailed simulators are often complex in terms of size and may require excessive runtime to compute a solution with the required accuracy. They are used in a variety of ways, the most interesting in terms of this thesis is in "what-if" analyses of different solutions or alternative courses of action. If derivatives are available the simulators can be used for sensitivity analysis, as well as in optimization. The importance of accurate derivatives for most optimization algorithms cannot be emphasized enough, as they essentially point the search in the right direction (Gunnerud et al., 2013).

Many simulators depict a larger system as a whole, for instance many oil companies have an all-encompassing black-box simulator representing a production network such as the one in Figure 4.2. The important point to note here is that a production network simulator is a *network* simulator, with some exploitable qualities. For example, it can be difficult to accurately compute the pressure drop through a riser pipe based on the multiphase flow running through it. On the other hand, mass and pressure balances through the network are easily calculated, e.g. a mass balance equation simply states that what goes out of a subsea well goes into a pipeline. Hence, the structure of the network simulator can be formulated with very simple algebraic equations, while each component, e.g. well, pipeline or separator, can be quite complex and comprise thousands of equations and code lines.

Treating the network simulator as a large black-box, which is the standard industry approach, does not exploit the network structure as explained in the previous section. Doing this is fine for evaluating the production network's response to a particular setup (i.e. a single "what-if" analysis), but this approach has large limitations when it comes to computing gradient information. That is because only gradients on the mapping between the input and output values of the network are accessible, and can in most cases only be computed numerically by finite differencing.

SmartOpt is a modeling concept developed by the IO-center at NTNU, where the network structure is taken advantage of by splitting the network simulator into components for each well, pipeline, compressor, separator etc. This means that instead of connecting the components inside the simulator, they are being connected inside the optimization algorithm. The algorithm is thus provided with much more information regarding the problem, which is especially important regarding gradients. Instead of looking at the effect on total oil production when the gas lift rate of one well is changed, one looks at the change in oil production from the well itself. The mapping to the total oil rate is then handled by the algebraic mass balance equations, which can be derived analytically. Another advantage is that one can call the specific component simulators as one needs information about that part of the system, which is more efficient than calling the whole network simulator each time. A necessary condition for using this modeling concept is that the system is divisible into smaller parts (Almklov and Gunnerud, 2013).

The concept is visualized for the problem in Figure 5.1. Each box represents one simulator giving the relations between what goes in and what comes out of that box. There are two types of simulators present in the figure, blue boxes representing pipeline pressure drop simulators, and the green boxes well production simulators. Additionally, there are mass and pressure balances connecting the parts in a feasible manner.

In this thesis a main goal is optimizing short-term oil production by combining sophisticated optimization techniques with the simulation of real-world petroleum fields. It is evident that the SmartOpt approach is suitable for an upstream oil production problem as the one sought to be solved here.

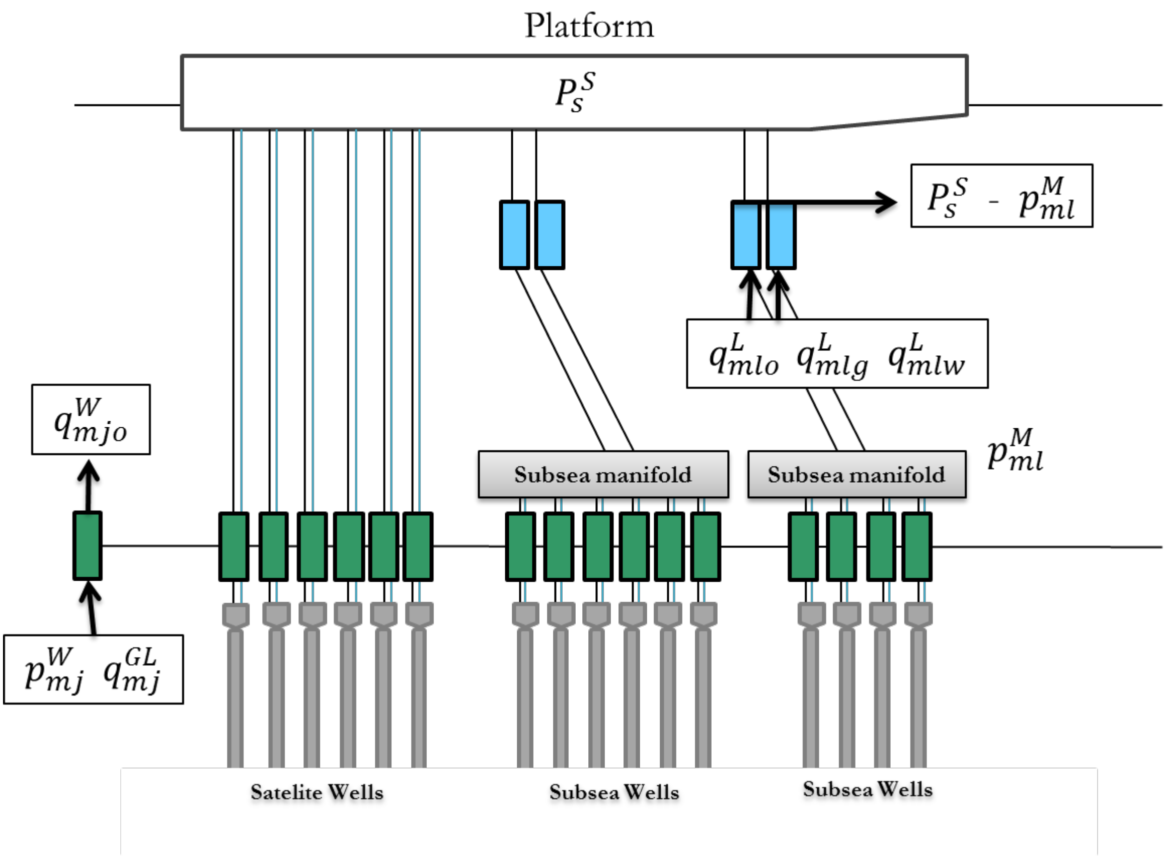


**Figure 5.1: Production system topology showing distinct simulators**

## 5.2 Mathematical Model

Optimization problems arising from production networks composed by connecting pressure drop and well production simulators with mass and pressure balances are often non-convex. Formulating the mathematical models of such problems can be done in several ways. One way is to formulate the problem in a straight forward manner, keeping the nonlinearities of the problem and obtain a compact formulation. Another possibility is to use methods which reduce the number of potentially non-convex relations. Such methods eliminate the non-convexities which might result from the nonlinear relations, e.g. through creating analytical proxy models or through linearizing the nonlinear relations with the big M method. Another example is using piecewise linear functions. The disadvantage is a potentially large increase in number of constraints and number of variables, leading to a bigger model.

Regarding the nonlinear problem at hand the emphasis has been on expressing it in a compact matter. The mathematical model is kept nonlinear and is solved by solution methods tailored for such problems. Solution methods specially designed for non-convex problems are tested together with this mathematical formulation. In this section the sets and their indices are introduced, before the parameters and variables of the problem are given. After that, the objective function as well as the system constraints are listed.

**Table 5.1: Sets**

| | | |
|---|---|---|
| $J$ | - | Set of wells |
| $J_m$ | - | Set of wells connected to manifold $m$ |
| $L_m$ | - | Set of pipelines connected to manifold $m$ |
| $M^S$ | - | Set of subsea manifolds |
| $M$ | - | Set of manifolds ($t$ for topside $\cup$ $M^S$ ) |
| $P$ | - | Set of phases ($g$ for gas, $o$ for oil, $w$ for water) |
| $S$ | - | Set of separators |

**Table 5.2: Indices**

| | | |
|---|---|---|
| $j$ | - | Well |
| $l$ | - | Pipeline |
| $m$ | - | Manifold |
| $p$ | - | Phase |
| $s$ | - | Separator |

**Table 5.3: Parameters**

| | | |
|---|---|---|
| $C_g$ | - | Capacity limit on gas flow to platform |
| $C_{ws}$ | - | Capacity limit on water flow to separator $s$ |
| $C^{GL}$ | - | Capacity limit on gas lift available for allocation |
| $GOR^W_{mj}$ | - | Gas to oil ratio for well $j$ connected to manifold $m$ |
| $WC^W_{mj}$ | - | Water cut for well $j$ connected to manifold $m$ |
| $P^S_s$ | - | Inlet pressure for separator $s$ |
| $\overline{P}^W_{mj}, \underline{P}^W_{mj}$ | - | Upper/lower limit of wellhead pressure at well $j$ connected to manifold $m$ |
| $\overline{Q}^{GL}_{mj}, \underline{Q}^{GL}_{mj}$ | - | Upper/lower limit on gas lift for well $j$ connected to manifold $m$ |

**Table 5.4: Variables**

| | | |
|---|---|---|
| $p^W_{mj}$ | - | Wellhead pressure on well $j$ connected to manifold $m$ |
| $p^M_{ml}$ | - | Pressure in line $l$ at the subsea manifold $m$ |
| $p^P_{ml}$ | - | Pressure in line $l$ connected to manifold $m$ at the platform |
| $q^W_{mjp}$ | - | Flow of phase $p$ from well $j$ connected to manifold $m$ |
| $q^{GL}_{mj}$ | - | Flow of gas lift injected to well $j$ connected to manifold $m$ |
| $q^L_{mlp}$ | - | Flow of phase $p$ in pipeline $l$ connected to manifold $m$ |
| $x_{js}$ | - | 1 if well $j$ connected to topside manifold $t$ is open and routed to separator $s$, 0 otherwise |
| $y_{mjl}$ | - | 1 if well $j$ connected to manifold $m$ is open and connected to line $l$, 0 otherwise |
| $z_{mls}$ | - | 1 if flow from pipeline $l$ connected to manifold $m$ is routed to separator $s$ |

The production- and gas lift variables are mostly semi-continuous. This means that they should either assume the value 0, or lie between nonzero lower and upper bounds. To accommodate this in the modeling language and solver chosen, these variables are incorporated with only their nonzero lower and upper bounds. Consequently, the actual production variable, $q^W_{mjp}$, and the gas lift variable, $q^{GL}_{mj}$, may be nonzero even though a well is not open. However, the routing and on/off variables $x_{js}$ and $y_{mjl}$ ensure that whenever a well is closed, there is zero contribution from this well to total oil, gas and water productions and zero gas lift consumption. This is achieved by multiplying the production or gas lift variables with corresponding binary variables indicating on/off of wells, which can be seen repeatedly in the model below.

**Objective Function**

The objective function seeks to maximize the oil production and can be seen in equation (5.1), which has two parts. The first part relates to the production from the subsea wells, summing the total oil flowing to the separators as given by the pipeline flows. This is possible, because the mass balance equations (5.11) handle the routing and mass balance through the subsea manifolds. The second part sums the oil production from each topside well. Each topside well has $|S|$ binary variables giving routing of flow to the platform separators or closure of the well. As mentioned the production variables $q^W_{tjo}$ may be nonzero for a closed well, and hence $q^W_{tjo}$ are multiplied with the routing variables $x_{js}$ to ensure zero contribution from a closed well in the objective value.

$$Max \sum_{m \in M^S} \sum_{l \in L_m} q^L_{mlo} + \sum_{s \in S} \sum_{j \in J_t} q^W_{tjo}\, x_{js} \tag{5.1}$$

**Gas Lift Capacity**

The gas lift capacity restriction (5.2) requires the total amount of gas lift consumption of all open wells to be less than or equal to the amount of available gas lift. In the summation gas lift variables of satellite wells must be multiplied by their respective separator routing binary variables to avoid contribution to total gas lift consumption from closed wells. Similarly, to avoid contribution from closed subsea wells, the gas lift variable for a certain subsea well must be multiplied by its respective pipeline routing variables.

$$\sum_{m \in M^S} \sum_{j \in J_m} \sum_{l \in L_m} q^{GL}_{mj}\, y_{mjl} + \sum_{s \in S} \sum_{j \in J_t} q^{GL}_{tj}\, x_{js} \le C^{GL} \tag{5.2}$$

**Water Production Capacities**

Inequalities (5.3) give the water production restrictions. Each separator on the platform has a certain water handling capacity, requiring the total amount of water routed to each separator to be less than or equal to its water handling capacity. The first part relates to the subsea wells, and the second to topside wells, summing to total water routed to each separator. The same considerations regarding the multiplication of the binary variables mentioned earlier apply here.

$$\sum_{m \in M^S} \sum_{l \in L_m} q^L_{mlw} z_{mls} + \sum_{j \in J_t} q^W_{tjw}\, x_{js} \le C_{ws} \qquad s \in S \tag{5.3}$$

**Gas Production Capacity**

Inequality (5.4) give the gas production restriction, requiring the total amount of gas produced to be less than or equal to the gas handling capacity of the platform. In the same way as the objective function (5.1), the first part relates to the subsea wells, and the second to the topside wells. The same considerations regarding the multiplication of the binary variables explained for the objective function also apply here.

$$\sum_{s\in S}\sum_{m\in M^S}\sum_{l\in L_m} q^L_{mlg} + \sum_{s\in S}\sum_{j\in J_t} q^W_{tjg}\, x_{js} \leq C_g \tag{5.4}$$

**Pressure Relationships**

If a topside well is turned on, its wellhead pressure must be larger than or equal to the separator inlet pressure it is routed to in order to support flow in the right direction (5.5). In the same way, inequalities (5.6) ensure that the flow will be in the right direction when a producing subsea well is connected to a certain pipeline. This inequality states that if well $j$ is connected to line $l$, then its wellhead pressure must be larger than or equal to the pressure in line $l$ at the manifold $m$. Finally, inequalities (5.7) take care of the pressure relationship between a line $l$ and a separator $s$. When a line $l$ is connected to a separator $s$, the line outlet pressure up by the platform must be larger than or equal to the separator inlet pressure it is routed to.

$$P^S_s x_{js} \leq p^W_{tj} \qquad j \in J_t \;\; s \in S \tag{5.5}$$

$$p^M_{ml} y_{mjl} \leq p^W_{mj} \qquad m \in M^S \;\; j \in J_m \;\; l \in L_m \tag{5.6}$$

$$P^S_s z_{mls} \leq p^P_{ml} \qquad m \in M^S \;\; l \in L_m \;\; s \in S \tag{5.7}$$

**Routing**

In order to correctly route flow from the subsea wells into one of the pipelines, binary variables are included. Each subsea well $j$ has $|L|$ binary variables, and if variable $y_{mjl}$ takes the value 1 it signifies that well $j$ is routed to line $l$. In other words, inequalities (5.8) prevent a well from being routed to several lines by stating that at most one of the binary variables belonging to a certain well can take the value 1.

$$\sum_{l\in L_m} y_{mjl} \leq 1 \qquad m\in M^S \;\; j \in J_m \tag{5.8}$$

Additionally, binary variables must be included in order to correctly route flow from topside wells and subsea pipelines to the separators on the platform. Each topside well $j$ has $|S|$ binary variables, and if variable $x_{js}$ takes the value 1 it signifies that well $j$ is routed to separator $s$. Hence, inequalities (5.9) prevent a topside well from being routed to several separators by stating that at most one of the binary variables belonging to that well can take the value 1. Equally, each pipeline $l$ has $|S|$ binary variables, and if variable $z_{mls}$ takes the value 1 it signifies that pipeline $l$ is routed to separator $s$. Note that (5.10) are equalities, preventing pipelines from being routed to several separators, but ensuring that each is indeed connected to one of them.

$$\sum_{s \epsilon S} x_{js} \leq 1 \qquad j \in J_t \tag{5.9}$$

$$\sum_{s \epsilon S} z_{mls} = 1 \qquad m \in M^S \;\; l \in L_m \tag{5.10}$$

**Mass Balances**

The subsea well mass balances are given by equations (5.11), stating that the flow of phase $p$ through line $l$ is the sum of produced rates of phase $p$ from all subsea wells that are routed to that line. This is obtained by multiplying the well production rates by the corresponding binary variables.

$$\sum_{j \epsilon J_m} q^W_{mjp}\, y_{mjl} = q^L_{mlp} \qquad m \in M^S \;\; l \in L_m \;\; p \in P \tag{5.11}$$

**Well Model**

The function $f^W_{mjo}$ in (5.12) represents the fact that the nonlinear relationship between the oil flow and the wellhead pressure of each well is given by simulation. As mentioned earlier the GOR and WC are considered constant, so that the gas and water rates can be obtained from the oil rate, as presented in (5.13) and (5.14). Notice in equations (5.13) that the gas production rate is a sum of the gas flow from the reservoir and the gas lift used at this well. Notice also that the triplet of equations below exists for each and every well.

$$q^W_{mjo} = f^W_{mjo}\left(p^W_{mj}, q^{GL}_{mj}\right) \qquad m \in M \;\; j \in J_m \tag{5.12}$$

$$q^W_{mjg} = GOR^W_{mj} q^W_{mjo} + q^{GL}_{mj} \qquad m \in M \;\; j \in J_m \tag{5.13}$$

$$q^W_{mjw} = q^W_{mjo}\left(\frac{WC^W_{mj}}{1 - WC^W_{mj}}\right) \qquad m \in M \;\; j \in J_m \tag{5.14}$$

**Pressure Drop Model**

The pipeline pressure drop is a function of the oil, gas and water that flow through it. This relationship can be quite complex, and is essentially found by simulation. One such simulator or relationship exists for each pipeline, and is given in equation (5.15).

$$p_{ml}^{P} - p_{ml}^{M} = f_{ml}^{L}\left(q_{mlo}^{L}, q_{mlg}^{L}, q_{mlw}^{L}\right) \qquad l \in L_m \tag{5.15}$$

**Variable Bounds**

$$\begin{array}{ll}
q_{mjp}^{W} \geq 0 & m \in M \quad j \in J_m \quad p \in P \\
p_{ml}^{M} \geq 0 & m \in M^S \quad l \in L_m \\
p_{ml}^{P} \geq 0 & m \in M^S \quad l \in L_m \\
q_{mjp}^{W} \geq 0 & m \in M \quad j \in J_m \quad p \in P \\
\underline{Q}_{mj}^{GL} \leq q_{mj}^{GL} \leq \overline{Q}_{mj}^{GL} & m \in M \quad j \in J_m \\
\underline{P}_{mj}^{W} \leq p_{mj}^{W} \leq \overline{P}_{mj}^{W} & m \in M \quad j \in J_m \\
x_j \in (0,1) & j \in J_t \\
y_{mjl} \in (0,1) & m \in M^S \quad j \in J_m \quad l \in L_m \\
z_{mls} \in (0,1) & m \in M^S \quad j \in J_m \quad l \in L_m
\end{array} \tag{5.16}$$

# 6. Algorithm

Having formulated the mathematical problem and recognizing the existence of both nonlinearities and integer variables, the problem to be solved can be categorized as a MINLP problem. This chapter starts by discussing branch and bound solution methods and problem convexity, before giving an introduction to heuristics. The last section of the chapter gives the proposed algorithm designed by the authors to handle the petroleum production optimization problem.

## 6.1 Branch and Bound

Branch and bound is a solution strategy for tackling the integer variables in both mixed integer linear programming problems and mixed integer nonlinear programming problems. The basic B&B algorithm is quite elementary and is based on splitting the feasible region into smaller regions, and solving a relaxed problem in each subproblem. It can however be seen as a solution paradigm (Clausen, 1999) with various possible strategic and parametric choices and add-ons, which can be tailored to specific problem types.

**Linear Programming**

The B&B algorithm was originally proposed as a solution method for MILP problems by Land and Doig (1960). The algorithm starts by relaxing all integer requirements of the MILP leading to a LP problem in the root node typically solved with the simplex method. Branching is then done on single variables with integer requirements, by adding one variable specific constraint in each branch. This will divide the relaxed problem into subproblems throughout the search, which is structured in a search tree.

Nodes represent the subproblems, while the edges represent the added constraints designed to eliminate infeasible parts of the solution space and push the integer or binary variables to take discrete values. For each constraint added, a new node representing a different relaxation of the original MILP problem is solved to optimality. If a feasible solution to the LP problem is found, it provides an optimistic bound. If the solution to the subproblem provides a feasible *integer* solution to the MILP, the search from that node is terminated and provides a pessimistic bound. The search continues as long as the gap, defined as the difference between the optimistic and pessimistic bounds, is larger than a predefined tolerance.

The solution to a subproblem solved at a deeper level of the tree can only be equal to or worse than the solution in a previous node, which enables pruning. If the solution to the LP relaxation of a subproblem is worse than the best pessimistic bound, no better solution can be found in this branch and the search from that node can be terminated or pruned.

The basic B&B algorithm requires the user to make certain choices. Two strategic decisions are the choice of the variable to branch on at each node and the choice of the next node to process during the tree search i.e search strategies (Bonami et al., 2011). Another possibility is to create more than the traditional two branches out of each node in the tree. Common variable choice schemes are most fractional, closest to 0.5 and strong branching. Best-first, depth-first and breadth-first are possible search strategies, with the former two being most popular.

**Nonlinear Programming**

The branch and bound method is also applicable for MINLP problems. In the root node all integer restrictions are relaxed, and the resulting NLP problem is solved with an appropriate algorithm, e.g. an interior point solver. Integer variables must be selected to branch on, and a new NLP problem must be solved in each node.

In the case of NLPs, there is no single method that can be used to solve a general nonlinear model, unlike that of LP problems. Solution methods must be adopted to the structure of the problem and to the functions used. Many different classifications of nonlinear problems exist. A broad distinction is usually made between convex and non-convex problems.

When seeking to solve nonlinear problems, it is important to know if the problem is convex or not. In a convex problem, each local optimum is also a global optimum. This means that the convexity characteristics must be taken into account when deciding which solution methods are suitable. Most nonlinear solution methods are search algorithms that only guarantee finding local optima, so the convexity of the problem decides if the solution found can be claimed to be the global optimum or not. Informally, a maximization problem can be defined to be convex if its objective function is concave and the feasible region (mapped by the problem constraints) is a convex set. All LP problems are convex, nonlinear problems are often non-convex and Integer Programming (IP) problems are always non-convex (Lundgren et al., 2010).

If the NLP is non-convex, there is no guarantee that the NLP problem solved in each node provides the global optimum of the subproblem. The B&B algorithm works as a heuristic and there is no promise that the solution to a subproblem solved at a deeper level will be equal to or worse than the solution in the previous node. Hence there are more difficulties as to deciding when to prune and stop branching. As such, inclusion of heuristics for MINLP problems may contribute positively.

## 6.2 Heuristics

Heuristics are a class of solution methods that are often applied to complex optimization problems. Typically, the purpose is to generate feasible solutions quickly, however heuristics belong to a group of solution methods that are not exact and there is usually no guarantee for the quality or optimality of a provided solution. They are usually developed for a specific class of optimization problems, and further tailored to fit the exact structure of a particular problem.

In view of mixed integer programs, the purpose of a construction heuristic is finding a good feasible integer solution. As part of a branch and bound algorithm the incumbent solution of the heuristic provides a pessimistic bound which may result in earlier pruning of inferior nodes, and might also reduce the size of the branch and bound three. Two common heuristics used in branch and bound algorithms are the feasibility pump and the diving heuristic.

### 6.2.1 Feasibility Pump

As presented in the literature review, a feasibility pump is a construction heuristic which was first developed for handling generic MILP problems to provide a feasible solution quickly. Initially this heuristic solves the LP relaxation of the original problem. If the integer variables take continuous values they are bounded to the nearest integer value to provide a, although possibly constraint infeasible, integer feasible solution. This integer solution is further used in the LP problem. The objective function is now minimizing the distance from the sought continuous solution to the obtained integer solution through bounding up or down. The algorithm then alternates between solving the LP problem with this objective function and bounding of the continuous integer variable values. Hopefully, the solution converges towards being both integer and constraint feasible (Bertaccoa et al., 2007).

A feasibility pump heuristic correspondingly exists for providing feasible solutions to convex MINLP problems. This FP iterates between solving two relaxations of the underlying MINLP problem, where the intention is for the solutions of the two relaxations to converge towards the same feasible solution. The main approach consists of solving the two relaxations of the underlying problem given in Table 6.1.

**Table 6.1: Relaxations of the Underlying MINLP Problem used in the Feasibility Pump**

| | | |
|---|---|---|
| FP-NLP | - | Integer relaxed MINLP, which becomes a (feasibility pump) NLP |
| FP-MILP | - | Constraint relaxed MINLP, which becomes a (feasibility pump) MILP |

The FP-NLP provides constraint feasible solutions to the original problem, while FP-MILP provides integer feasible solutions. The FP-MILP is only composed by linear outer approximations of the solution space of the original MINLP problem. One new approximation is added to the FP-MILP each time a new FP-NLP solution is found, by using the solution points provided by the FP-NLP. The algorithm alternates between solving these two problems until convergence. Convergence is sought through modifications of the original objective functions, and if a feasible solution to the MINLP problem exists, the algorithm will converge.

The objective function of the MINLP problem is used as a basis for the objective functions in both the FP-NLP and FP-MILP, however an extra element is added. In the FP-NLP the objective function becomes a weighted average between optimizing the original objective function and minimizing the distance to the previously obtained solution of the FP-MILP. Similarly for the FP-MILP the distance minimized is to the previously obtained solution of the FP-NLP. In both problems the weighting starts by emphasizing the original objective function and gradually shifts towards minimizing the distance to the previously obtained solutions. This enables the solutions of the two relaxations to converge towards a feasible solution (Bonami et al., 2009).

### 6.2.2 Diving

Diving heuristics are heuristics specially designed for use in branch and bound algorithms. This heuristic explores a path from the root node or current node in the branch and bound tree to a leaf node. The purpose is to provide an integer feasible solution to the MILP or MINLP problem quickly. Initially the heuristic solves a NLP relaxation of the original problem, and depending on where the diving heuristic is used, the number of previously locked integer variables varies. In the root node, no integer variables are locked, downwards the tree the numbers of locked integer variables increase. During the diving heuristic all relaxed integer variables are locked to an integer value according to some predefined rules. Thus, if a integer variable takes a fractional value in the relaxed solution, it is bounded either up or down. When the heuristic is finished, all integer variables have integer values, and if the solution is constraint feasible, a MINLP feasible solution is obtained.

## 6.3 Special Purpose B&B Algorithm

In analyzing the convexity of the problem, the existence of binary variables automatically renders the problem non-convex. It is precisely this non-convexity that makes IP problems difficult to solve and require special solution methods such as B&B. But it is also important to analyze the NLP relaxation of the problem, meaning the same problem, but with all binary variables relaxed to be continuous. Because it is the convexity of these NLPs solved in the B&B nodes that determines if the resulting solution can be taken as the global optimum.

Equality constraints can also contribute to problems becoming non-convex. Constraints of the form $f(x) = b$ define a convex set if $f(x)$ is linear, while the opposite is true for nonlinear $f(x)$. There are a number of these in the problem formulation, e.g. equation (5.11). In addition, examination shows non-convex well- and pipeline data. Thus it is safe to conclude that the NLP is indeed non-convex, meaning that a B&B algorithm can only be taken as a heuristic method, not guaranteeing global optimality.

A special purpose B&B algorithm has been designed targeting the adversities of the problem at hand. Tailored heuristics are included with varying frequencies to help obtain good lower bounds fast, in addition to some special features that are put in place to tackle the non-convexity of the problem. The first of these is a multiple start option, enabling the algorithm to start with several starting points in order to search a larger part of the solution space. The other measures target pruning and the extent to which negative gaps are allowed. The gap coefficient gives how negative the gap can be, while a predefined limit gives how many times (given by the node level) a node with a negative gap is allowed to branch down the B&B tree.

In the remaining of this chapter the proposed B&B algorithm is presented in a general manner for overview, before each distinct part of the method is explained in greater detail. These distinct parts comprise of the feasibility pump adapted to non-convex problems, two problem specific heuristics and finally the non-convexity measures implemented in the code.

### 6.3.1 B&B Framework

The branch and bound consists of one main list of nodes that must be solved. Initially, this list contains the root node that the algorithm will branch out of. In this work a multi-start option enables using multiple starting points for the root node. This option is explained further towards the end of this chapter, while the implementation is described in the next chapter. Current node means the node being processed in this iteration. The algorithm terminates if the gap reaches below a set tolerance, or if the list of nodes to solve is emptied. The latter means setting the gap tolerance to zero.

In the remaining explanation the loop of the branch and bound is described according to Algorithm 1. Each numbered part in the algorithm is discussed separately, and has a corresponding number in the text. Objective function value is abbreviated OFV, and lower bound is given as LB.

**Algorithm 1 – B&B**

Create root node, give starting points according to multi-start option
Create main list of nodes and put the root node in the list
Set current node
If the feasibility pump is used, it provides a decent LB

```
1:    repeat
2a:       Solve current node
          if (NLP feasible)
3:            Test for integer feasibility
4:            if  (integer solution)
                  Update LB
                  Erase current node
5:            else
6:                Run Problem Heuristics 1 and/or 2 → Update LB
7:
8:                if (current node→OFV > LB)
                      Clear level
                      Choose branching variable and make next generation nodes
                      Erase current node
9:                else if (current node→OFV > gap coefficient*LB) AND
                          (current node→level < predefined limit)
                      Increment level
                      Choose branching variable and make next generation nodes
                      Erase current node
10:               else
                      Erase current node
                  end
              end
          else
2b:           Erase current node
          end
11:       Reset current node based on search strategy
      until Gap tolerance is reached
```

1) The far outer loop will continue as long the gap is above the tolerance.
2) Solve the current node. If a NLP feasible solution is obtained, continue to next block. But if a solution was unattainable, given in 2b, simply delete the current node just solved. Go to point 11.
3) Test the solution for integer feasibility.
4) If the solution is also integer feasible, it is eligible to update the lower bound. The objective function is checked against the LB, which gets updated if this solution is better. Current node is subsequently erased.
5) If the solution is not integer feasible, the algorithm goes into this block containing heuristics and non-convexity measures.
6) Heuristics step. Problem heuristics 1 and/or 2 are run. This will give MINLP feasible solutions. Update the lower bound if a better one is gained from the heuristics.
7) Branching step. Since the solution is not integer feasible, current node must be considered for branching. Here there are 3 alternative courses:
    8) If the objective function value of the node just solved is greater than the lower bound. The level of the node just solved will be cleared. This is so that the child nodes will inherit the right level. The branching variable is identified (branching strategy) and child nodes are created, with the starting point of the child nodes set to be the solution of the parent node, and then added to the main list. The current node just solved is deleted from the main list.
    9) If the objective function value of the node just solved is above $c$ times the lower bound, and if the level of this node is less than the level limit. The branching variable is identified (branching strategy) and the child nodes are created, but their levels are increased, this will be explained in Section 7.3.3. The starting point of the child nodes is set to be the solution of the parent node, and they are added to the main list. The current node just solved is deleted from the main list.
    10) The third alternative means that either the objective function of the node just solved is below $c$ times the lower bound, or the level of the node just solved is at or above the level limit, or both. If the node enters this code block it will simply be deleted, and will not be branched on.
11) Current node must be set to point to the node to be solved in the next iteration, given by the search strategy.

### 6.3.2 Feasibility Pump for a Non-convex MINLP

The concept of the FP for convex MINLPs is explained in Section 6.2.1. As stated the heuristic iterates between solving two relaxations of the underlying MINLP problem, where the intention is for the solutions of the two relaxations to converge towards the same feasible solution.

In this thesis a special purpose FP is developed for the non-convex MINLP problem addressed. The algorithm is explained in relation to the model in 5.2, but applies correspondingly to similar allocation problems. The FP-NLP and the FP-MILP problems of this algorithm are slightly modified from the generic FP. The proposed FP algorithm also includes an additional problem. This is a reduced NLP, abbreviated FP-NLP-R. It equals the original MINLP problem, however it is solved with discrete values assigned to all binary variables, which means the problem contains fewer variables. Detailed descriptions to the three problems are given in Table 6.2.

**Table 6.2: Relaxations of the Underlying non-convex MINLP Problem used in the tailored Feasibility Pump**

| | **FP-NLP** | **FP-MILP** | **FP-NLP-R** |
|---|---|---|---|
| **Relaxation** | Integer relaxed original MINLP | Constraint relaxed original MINLP | Reduced original MINLP, solved with binary variables locked to discrete values |
| **Problem type** | NLP | MILP | NLP |
| **Objective function** | Modified version of the objective function of original MINLP | Modified version of the objective function of original MINLP | Objective function equal to the original MINLP |
| **Solution** | Constraint feasible with regards to the integer relaxed original MINLP | Integer feasible with regards to the original MINLP | Potentially constraint and integer feasible with regards to the integer relaxed original MINLP |
| **Additional information** | | MINLP problem structure retained. Wells and pipelines approximated through tangent planes. | The binary variables are locked to the previously obtained integer solution from the FP-MILP |

The proposed feasibility pump is explained according to Algorithm 2. All numbered parts are briefly described below. Further elaborations on features of the FP are given afterwards. This includes the modified objective functions and tangent plane generation. In addition, comments about handling non-convexities are given throughout the explanations.

**Algorithm 2 - Feasibility Pump**

---

**1:** Start by solving the integer relaxed original MINLP problem to get a continuous solution
Update the FP-MILP with the tangent planes from the obtained solution

**2:** **repeat**
**3:** Solve FP-MILP
**if (original MINLP feasible)**
Return the current solution
**end**
**4:** Solve the FP-NLP-R with binary variables locked to the FP-MILP solution values
**if (NLP feasible)**
Integer and constraint feasible solution obtained
**end**
**5:** Solve the FP-NLP
**if (NLP feasible) AND (integer solution)**
Return current solution
**end**
**6:** Update the FP-MILP with the new tangent planes
**until** Maximum number of iterations is reached or convergence happens

---

1) The FP is initiated by solving the integer relaxed original MINLP problem from Section 5.2 to get continuous values on the binary variables. If used in a B&B tree, this solution is obtained by the node NLP. If the continuous solution provided is constraint feasible, tangent planes based on the obtained solution are added to the FP-MILP. The composing of the FP-MILP is further explained later in this section.
2) At each iteration the FP solves all three problems provided in Table 6.2; the FP-MILP, the FP-NLP and the FP-NLP-R. The FP will iterate until a sufficiently good integer and constraint feasible solution is obtained, the maximum number of iterations is reached or the algorithm converges.

3) The FP-MILP is solved to obtain integer values for the binary variables, $\tilde{x}_{js}^{i}, \tilde{y}_{mjl}^{i}, \tilde{z}_{mls}^{i}$, which might violate some constraints of the MINLP, but will satisfy the integer requirements. If the integer solution obtained is constraint feasible the algorithm has converged is ended.
4) The FP-NLP-R is solved with the binary variables locked to the previously obtained FP-MILP solution. If the solution is feasible, an integer and constraint feasible solution is obtained. The algorithm may be terminated here if the solution is satisfactory.
5) The FP-NLP is solved and continuous values to the binary variables are obtained for iteration $i$, $\hat{x}_{js}^{i}, \hat{y}_{mjl}^{i}, \hat{z}_{mls}^{i}$. These will satisfy all problem constraints of the MINLP, but might violate the integer requirements. If the solution is constraint feasible and all binary variables take discrete values, the algorithm has converged and is ended.
6) If the FP-NLP solution is not integer feasible, tangent planes based on the obtained FP-NLP solution is added to the FP-MILP.

The FP-MILP of the proposed feasibility pump differs from the FP outlined in Section 6.2.1. As opposed to the general version, the FP-MILP of the proposed FP is tailored to the problem at hand and the structure of the original problem is retained. Formulated linearly, this problem consists of several constraints, just as the original MINLP. The mass balance equations (5.11), the capacity constraints (5.2)-(5.4) and the pressure constraints (5.5)-(5.7) are converted to linear expressions through big M reformulation. Maximum flow values from wells and through pipelines, maximum wellhead pressure values and maximum pressure drops over pipelines are used as the big M values. Furthermore, the problem is relaxed by reformulating all equalities assuring feasible behaviors of wells and pipelines, i.e. equations (5.12) and (5.15). This feasible range is approximated by inequalities made by tangent planes generated with information from the FP-NLP solution. Each well and pipeline is represented by multiple tangent planes. A comprehensive description on the derivation of tangent planes is given later.

As previously explained, assuming that feasible solutions to the underlying problem actually do exist, the aim of the FP is for the two problems, FP-MILP and FP-NLP, to converge towards the same feasible solution. To direct the two solutions towards convergence the objective functions differ from the one given in Section 5.2, and change throughout the iterative process. Detailed descriptions of the objective functions are given in the next section.

In principal convergence should be ensured by these modified objective functions, however a feature has been added to the FP to improve the chances of generating feasible solutions fast. As shown in Step 4 in the FP algorithm, in each iteration the problem called FP-NLP-R is solved with the binary variables locked to the obtained integer feasible solution from the FP-MILP solution. If this provides a feasible solution it is both integer and constraint feasible and the FP can be ended before the FP converges. However, several more iterations may be run either to seek convergence or to look for a superior feasible solution.

**Objective function**

The objective function, equation (6.2), of the linearized model becomes the weighted average of maximizing total oil production and minimizing the sum of the distances between the integer feasible points (which the FP-MILP seeks to find), and the previously obtained constraint feasible points (the FP-NLP solution). In this model these distances must be expressed linearly. Thus, for each binary variable of the FP-MILP two new auxiliary variables are introduced. As an example, binary variable $x_{js}$ is chosen, with the new associated auxiliary variables, $x_{js}^{a}$ and $x_{js}^{b}$. Equation (6.1) is added to the FP-MILP, which expresses the relationship between the new auxiliary variables, the previously obtained fractional value $\hat{x}_{js}^{i-1}$ and binary variable $x_{js}$.

$$x_{js} = \hat{x}_{js}^{i-1} + x_{js}^{a} - x_{js}^{b} \tag{6.1}$$

Obviously the binary variable should be given value 0 or 1. Thus, together with equation (6.1), minimizing the sum of the two auxiliary variables in the objective function corresponds to minimizing the distance between the integer feasible point, $x_{js}$, and the previously obtained constraint feasible point provided by the FP-NLP solution, $\hat{x}_{js}^{i-1}$. Only one of the two auxiliary variables will take value. The same relations are true for all binary variables $y_{js}$ and $z_{js}$. The FP-MILP objective function is given in equation (6.2).

$$\begin{aligned} &Min \\ &(1-\alpha^{i})\left(\sum_{j\in J_t}\sum_{s\in S}\left(x_{js}^{a}+x_{js}^{b}\right)+\sum_{m\in M^S}\sum_{j\in J_m}\sum_{l\in L_m}\left(y_{mjl}^{a}+y_{mjl}^{b}\right)+\sum_{m\in M^S}\sum_{l\in L_m}\sum_{s\in S}\left(z_{mls}^{a}+z_{mls}^{b}\right)\right) \\ &-\frac{\alpha^{i}}{D}\left(\sum_{m\in M^S}\sum_{l\in L_m} q_{mlo}^{L}+\sum_{j\in J_t}\sum_{s\in S} q_{tjo}^{W}\right) \end{aligned} \tag{6.2}$$

In both objective functions, $\alpha^{i}$ represents the weighting of the original objective function of maximizing total production from Section 5.2 in iteration $i$. In the objective functions the impact of maximizing total production is reduced for each iteration. $D$ indicates a scalar used for scaling.

Similarly, the objective function of the FP-NLP problem becomes a sum of the weighted average of maximizing total oil production and minimizing the sum of the distances between the sought solution points and the previously obtained integer feasible points. The objective function is given in equation (6.3).

$$Min$$

$$(1-\alpha^i)\left(\sum_{j\in J_t}\sum_{s\in S}\sqrt{\left(x_{js}-\tilde{x}^i_{js}\right)^2}+\sum_{m\in M^S}\sum_{j\in J_m}\sum_{l\in L_m}\sqrt{\left(y_{mjl}-\tilde{y}^i_{mjl}\right)^2}+\sum_{m\in M^S}\sum_{l\in L_m}\sum_{s\in S}\sqrt{\left(z_{mls}-\tilde{z}^i_{mls}\right)^2}\right)$$
$$-\frac{\alpha^i}{D}\left(\sum_{m\in M^S}\sum_{l\in L_m}q^L_{mlo}+\sum_{j\in J_t}\sum_{s\in S}q^W_{tjo}x_{js}\right) \quad (6.3)$$

**Tangent planes**

Configuration of the FP-MILP model requires the formulation of inequalities made by tangent planes to approximate the feasible range for wells and pipelines. The tangent planes are generated by inserting the solution points and the corresponding derivative information from the previous obtained FP-NLP solution in a general tangent plane equation, see inequalities (6.4) and (6.5). Table 6.3 gives explanations of all parameters in these inequalities.

**Table 6.3: Tangent Plane Inequality Parameters**

| | | |
|---|---|---|
| $\hat{p}^{Wk}_{mj}$ | - | Pressure value for well $j$ connected to manifold $m$ at the FP-NLP solution in iteration $k$ |
| $\hat{q}^{GLk}_{mj}$ | - | Gas lift value for well $j$ connected to manifold $m$ at the FP-NLP solution in iteration $k$ |
| $\frac{\partial f^W_{mjo}}{\partial p^W_{mj}}\left(\hat{p}^{Wk}_{mj},\hat{q}^{GLk}_{mj}\right)$ | - | Partial derivative value of the oil production function with respect to the wellhead pressure of well $j$ connected to manifold $m$, at the FP-NLP solution in iteration $k$ |
| $\frac{\partial f^W_{mjo}}{\partial q^{GL}_{mj}}\left(\hat{p}^{Wk}_{mj},\hat{q}^{GLk}_{mj}\right)$ | - | Partial derivative value of the oil production function with respect to the gas lift rate of well $j$ connected to manifold $m$, at the FP-NLP solution in iteration $k$ |
| $\hat{q}^{Lk}_{mlp}$ | - | Flow of phase $p$ through pipeline $l$ connected to manifold $m$ at the FP-NLP solution in iteration $k$ |
| $\frac{\partial f^L_{ml}}{\partial q^L_{mlp}}\left(\hat{q}^{Lk}_{mlo},\hat{q}^{Lk}_{mlg},\hat{q}^{Lk}_{mlw}\right)$ | - | Partial derivative value of the pressure drop function with respect to the flow of phase $p$ of pipeline $l$ connected to manifold $m$, at the FP-NLP solution in iteration $k$ |

At every iteration one tangent plane is generated for every pipeline and every well, giving a total of 20 tangent planes per iteration. The inequalities approximating the feasible operating area for wells can then be formulated as a 1st order Taylor expansion. The total oil production for each well must be less than or equal to the values of all tangent planes for that particular well.

$$
\begin{aligned}
q_{mjo}^{W} \leq \quad & f_{mjo}^{W}\left(\hat{p}_{mj}^{Wk},\hat{q}_{mj}^{GLk}\right) \\
& + \frac{\partial f_{mjo}^{W}}{\partial p_{mj}^{W}}\left(\hat{p}_{mj}^{Wk},\hat{q}_{mj}^{GLk}\right)\left(p_{mj}^{W}-\hat{p}_{mj}^{Wk}\right) \qquad m \in M \quad j \in J_m \; k \in I \\
& + \frac{\partial f_{mjo}^{W}}{\partial q_{mj}^{GL}}\left(\hat{p}_{mj}^{Wk},\hat{q}_{mj}^{GLk}\right)\left(q_{mj}^{GL}-\hat{q}_{mj}^{GLk}\right)
\end{aligned}
\tag{6.4}
$$

Similarly, the feasible area for pressure drop through the pipelines is estimated by inequalities (6.5). The pressure drop over one pipeline must be more than or equal to the values of tangent planes for that particular pipeline.

$$
\begin{aligned}
p_{ml}^{P} - p_{ml}^{M} \geq \quad & f_{ml}^{L}\left(\hat{q}_{mlo}^{Lk},\hat{q}_{mlg}^{Lk},\hat{q}_{mlw}^{Lk}\right) \\
& + \frac{\partial f_{ml}^{L}}{\partial q_{mlo}^{L}}\left(\hat{q}_{mlo}^{Lk},\hat{q}_{mlg}^{Lk},\hat{q}_{mlw}^{Lk}\right)\left(q_{mlo}^{L}-\hat{q}_{mlo}^{Lk}\right) \\
& + \frac{\partial f_{ml}^{L}}{\partial q_{mlg}^{L}}\left(\hat{q}_{mlo}^{Lk},\hat{q}_{mlg}^{Lk},\hat{q}_{mlw}^{Lk}\right)\left(q_{mlg}^{L}-\hat{q}_{mlg}^{Lk}\right) \qquad m \in M \quad l \in L_m \; k \in I \\
& + \frac{\partial f_{ml}^{L}}{\partial q_{mlw}^{L}}\left(\hat{q}_{mlo}^{Lk},\hat{q}_{mlg}^{Lk},\hat{q}_{mlw}^{Lk}\right)\left(q_{mlw}^{L}-\hat{q}_{mlw}^{Lk}\right)
\end{aligned}
\tag{6.5}
$$

Necessary measures are taken in the generation of tangent planes to assure that good feasible solutions are not cut away due to non-convexity issues. The 3- and 4-dimensional data sets for wells and pipelines are somewhat non-convex. Figure 6.1 illustrates how the generation of tangent planes might cut away parts of the feasible area. In this example the solid black line represents the upper bound of a feasible area and the solution should be below or on this line. A tangent plane is generated using the current solution point, and might become as the dashed line illustrated to the left in Figure 6.1. This line cuts away a substantial part of the feasible area. To avoid such situations throughout the algorithm, excessively restrictive tangents are moved to the bound of the feasible area, as illustrated to the right in the figure. The slope of the initial tangent plane is kept as it is moved stepwise until no parts of the feasible area lie above the tangent plane.

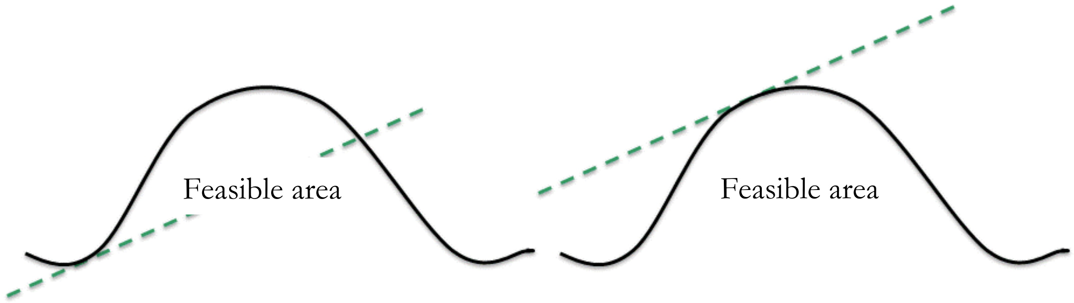


**Figure 6.1: Tangent Plane Modification**

As mentioned the aim is for the FP-MILP and FP-NLP solutions to converge. However, in the case of a highly non-convex problem, this will become a challenge as the abovementioned actions may cause insufficiently tightening tangent planes. This motivates further the inclusion of the FP-NLP-R problem in the algorithm.

Tangent planes for a well are presented in Figure 6.2. The light green surface in the left figure indicates the original data giving oil production as a function of gas lift and wellhead pressure. In the figure to the right tangent planes are added approximating the feasible area, represented by the blue and green colored squares.

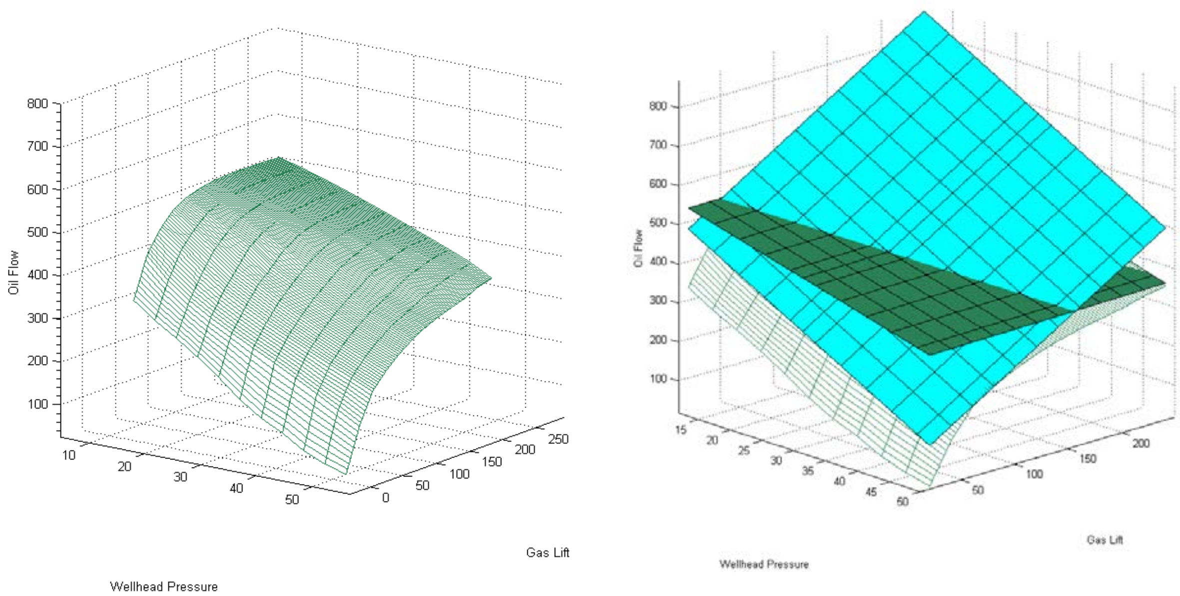


**Figure 6.2: Tangent Planes Approximating Well Surface**

### 6.3.3 Problem Specific Diving Heuristics

Two problem specific diving heuristics are also created to provide good lower bounds fast. These heuristics are construction heuristics based on the conditions that the properties of the asset are known and that wells should be prioritized according to gas and water content as well as total production potential. Rounding rules are created based on this knowledge.

The general ideas of the algorithms are fairly similar, moreover the one is simply an extension of the other. Initially the integer relaxed MINLP is solved in each node of the B&B to obtain continuous values for the binary variables. Further, an integer variable is selected according to a predefined priority list and bounded up or down. During the algorithm all relaxed binary variables are locked to an integer value according to fractional values and rounding rules, hence a diving heuristic.

The difference between the two heuristics is essentially that Heuristic 1 (H1) locks all variables after solving the NLP problem once, while Heuristic 2 (H2) is a slightly extended version which locks one routing decision at a time and re-solves the problem to get a new NLP solution after each bounding decision. For both heuristics the integer relaxed MINLP solution obtained from a B&B node is used as initial values for the binary variables. Algorithm 3 and 4 represent H1 and H2 respectively.

**Algorithm 3 – Problem Specific Heuristic 1**

---

Bound all binary variables with continuous values to discrete values of 0 or 1 according to variable priority and rounding rules.

---

**Algorithm 4 – Problem Specific Heuristic 2**

---

**repeat**

Evaluate the most recent NLP solution, choose the binary variable of highest priority with continuous value. Bound the variable up/down.

Solve the integer relaxed MINLP with this and previously bounded binary variables locked to integer value to obtain a new solution.

**until** Integer solution

---

### 6.3.4 Non-convexity Measures

As mentioned in the general discussion about B&B solution methods, there are certain challenges associated with solving non-convex MINLP problems with this approach. Since the solution of the non-convex NLP problem solved in each node is not guaranteed to be the global optimal solution of that problem, it cannot be taken as the actual optimistic bound and consequently makes pruning difficult. In such cases B&B can generally be seen as a heuristic to solving the non-convex MINLP. However, there are certain measures to be taken in order to upgrade the B&B to better tackle the non-convex MINLP case. The processes devised for this purpose will be explained in the next subsections.

**Multi-start**

Since the problem is non-convex, general understanding suggests that starting points can have significant impact on which (local) optimal solutions the solver obtains. In order to remove the risk of starting at a disproportionally bad point and widen the search, an option here called multi-start is incorporated. This option allows the root node of the B&B algorithm to begin with several starting points instead of the traditional one starting point. One can visualize this as several separate branch and bound trees solved in parallel, except that the lower and upper bounds are shared among all the trees which might be more efficient than solving each such tree independently. Details about how this is implemented are given in the next chapter.

**Pruning**

In B&B for convex problems a node is pruned if its optimistic bound is worse than the best pessimistic bound obtained so far. That is possible because the optimistic bound is proven to be the global optimal solution to that problem. In non-convex problems, such pruning might lead to the elimination of good local solutions, or even worse the global optimal solution. But if no pruning is done at all the search tree will grow immensely in size and results in explicit enumeration of the solution space.

Designing a scheme that lies between the extremes of pruning and not pruning seems promising. This can be done by allowing nodes that get slightly worse objective functions than the best lower bound (which is a pessimistic bound in a maximization problem) to be branched on a few times. If an improvement is observed, which well might be the case in the non-convex space, the node and its subsequent child nodes are kept. If however, no improvement is detected after allowing such a node to branch a few times this part of the tree will be pruned. The extent to which the gap can be negative is controlled by the gap coefficient, while the number of times such a node is allowed to branch is restricted by the level limit parameter. Additional implementation details are included in the next chapter.

# 7. Data and Implementation

This chapter presents the data applied for the production optimization of the asset addressed. Firstly, background and information about well and pipeline data are provided. Further, the different test cases are introduced. Finally, details concerning coding and implementation of the solution methods are given.

## 7.1 Well and Pipeline Data

SmartOpt is a simulator based model, and the relevant simulators are called directly as the optimization algorithm needs values throughout the search. This can either be the simulator itself, or some form of approximation of the simulator behavior. As long as the component simulators are kept updated to portray the real physical phenomena they represent, the data can be taken as fairly accurate. Oil companies normally have this updating and simulator infrastructure in place, an assumption that has been made in the designing of this solution method. In this thesis data tables are generated upfront, and the simulator response is estimated by interpolation.

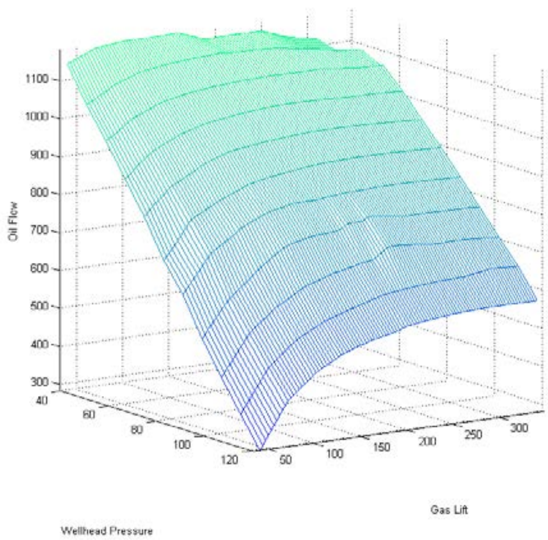


Figure 7.1: Well Example Data

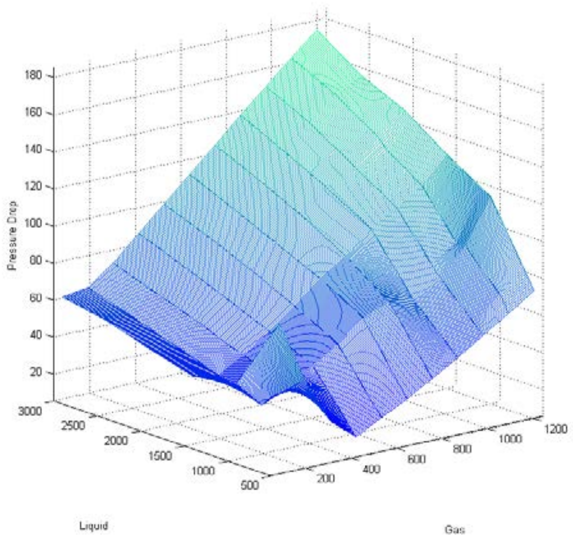


Figure 7.2: Pipeline Pressure Drop Example Data

The oil production rate of each well depends on its wellhead pressure and the amount of gas lift allocated to it. For most wells total production as a function of the wellhead pressure and gas lift rate, equation (5.12), becomes a fairly smooth 3 dimensional surface with increasing production rates for decreasing wellhead pressures and increasing gas lift rates, see Figure 7.1. This relationship was obtained by using the Petrobras in-house Marlim II simulator, and tables were generated over a reasonable range of combinations of input values. The oil rate is given for 144 combinations of gas lift rates and wellhead pressures. Because the GOR and WC are considered constant for this field, the gas and water rates can be calculated from the oil value directly.

Pressure drop over pipelines are dependent on three variables, the rates of gas, oil and water that flows through the pipeline (equation (5.15)). The data are provided by the GAP simulation and optimization package by Petroleum Experts and are adjusted to fit the well data to create a system which preferably resembles a real production platform. The pipeline pressure drop is given for 27000 combinations of oil, gas and water rates. Furthermore, the surfaces are 4 dimensional and appear quite non-convex. Figure 7.2 illustrates the pressure drop over line 1 connected to subsea manifold 1 as a function of liquid and gas. The graph is plotted for a fixed water cut of 44 %.

The oil and water values are given in $m^3/_d$, while the gas values are scaled, and therefore given in 1000 $m^3/_d$. Pressures are given in bar.

## 7.2 Test Cases

Necessary measures have been taken in order to avoid publicizing sensitive information and data. All values in this document have been scaled and/or offset so that sensitive information is not compromised.

As mentioned previously, the lifetime of a field is often divided into production phases with distinct characteristics. As the Marlim field is gradually becoming more mature the WC and GOR are increasing. Test cases are divided into three production phases, hence three data sets of GOR and WC have been created to simulate oil field properties for the early part, an intermediary period and the late stage of the field lifetime. For the younger field, the GORs and the WCs are the lowest. The values increase randomly with a total average increase of about 20% going from one stage to the next.

The production platform has a total capacity limit on produced gas, while each separator has a separate capacity limit on produced water. In addition, the amount of available gas lift is restricted. At each production phase the production optimization is done for three different platforms/FPSOs. The composition of platform capacities is characteristic for each platform respectively, and given in Table 7.1.

Table 7.1: Case Descriptions

| | Platform 1 | Platform 2 | Platform 3 |
|---|---|---|---|
| **Gas Capacity** | 2000 | 2400 | 2000 |
| **Gas Lift Available** | 1100 | 1600 | 1100 |
| **Water Capacity Separator 1** | 2400 | 2400 | 3000 |
| **Water Capacity Separator 2** | 2400 | 2400 | 3000 |
| **Water Capacity Separator 3** | 2000 | 2000 | 2500 |

These capacity schemes are created in an attempt to configure platforms such that for each platform different capacities are restrictive. Platform 1 was compiled to provide strict limits on all capacities. Further, platform 2 provides strict limits on water capacities while platform 3 provides strict limits on gas and gas lift capacities. It should be emphasized that the pressures in all separators are different, see Table 7.2, and although the water capacities of separator 1 and 2 are similar, no symmetry can be exploited.

Table 7.2: Separator Pressure

| | Platform 1,2,3 |
|---|---|
| **Pressure Separator 1** | 10 |
| **Pressure Separator 2** | 12 |
| **Pressure Separator 3** | 15 |

## 7.3 Implementation SmartOpt-B&B

In the following, details about coding and implementations of all the parts of the proposed algorithm will be given. First, general comments about the coding and solvers used in the B&B framework are given. This is followed by similar comments about the heuristics. Then the implementation of the non-convexity measures is elaborated on, before the simulator implementation is explained.

### 7.3.1 B&B Framework

Two open-source C++ solvers are used and incorporated in the proposed algorithm. Ipopt is a software package for large-scale nonlinear optimization using an interior point algorithm (COIN-OR, Ipopt, 2012). Bonmin is a code for solving general MINLP problems. Ipopt is used by Bonmin to solve NLPs. Both are available from the COIN-OR initative (COIN-OR, Bonmin, 2012). The computer used is a 64-bit workstation under the Linux operating system with 8 Intel processors running at 3.40 GHz and with 15.6 GB of RAM. This is a multiple-user machine at

NTNU, where several users can connect remotely to run their applications, sometimes leading to incomparable solution times as a result of varying load.

The model and B&B algorithm are coded in C++. All NLPs are solved using Ipopt. The software needs both first and second order derivatives of all equations used in the model formulation. Ipopt includes an option for approximating the second order derivative matrix with a limited-memory quasi-Newton method, which is enabled in this implementation. Regarding the first order derivatives, Ipopt also includes the option of numerically computing these gradients, although this option was not enabled in this work. The model formulation (Section 5.2) mainly consists of explicit equations, therefore analytical first derivatives are provided to Ipopt for those. This is more accurate than numerical computation. The well- and pipeline simulator functions are not explicit, so a differentiating scheme was created especially for these and the details are given at the end of this chapter.

When problems are solved with Ipopt, the software returns a status. This number indicates how the solution process went and if Ipopt was able to converge to a (NLP) feasible solution. Status 0 signifies success and solver convergence, while status 1 indicates that Ipopt was not able to converge to a solution within the permitted number of iterations set in Ipopt options. In other words, status 0 means that Ipopt has found a NLP feasible solution. When Ipopt returns status 1, the point it has stopped at might still be interesting, and is treated in the algorithm as the nodes that were allowed negative gaps.

### 7.3.2 Heuristics

The feasibility pump is coded in C++ as a part of the root node, with the continuous root node solution being used as initial values for the variables. Because of implementation issues it has not been within the scope of this thesis to implement the feasibility pump in any other nodes. In addition more implementation difficulties have led to the use of Bonmin, which uses interior point and a branch and bound algorithm, to solve the MILPs of the feasibility pump. This evidently causes inferior solution times compared to a special purpose MILP solver which uses the simplex method to solve the continuous problem. The NLPs of the feasibility pump are solved with Ipopt, in the same way as each NLP in the branch and bound algorithm. The problem specific heuristics are coded in C++ as standalone applications that can be called and applied to any node in the B&B tree.

Some bounding rules are defined for the problem specific heuristics in order to provide good feasible solutions based on the fractional values. By analyzing the problem data, one can identify wells and routing decisions that are superior to others and apply this knowledge in the heuristics. Specific rules are set for the asset addressed. A priority list of wells is defined according to production potential, gas-oil-ratios and water cuts of the wells. Routing decisions can be divided into four groups and the bounding decisions are completed in the order shown in Table 7.3. Subsea wells can be routed to two pipelines, however there is set limitations to how many wells can be routed to the same pipeline simultaneously. In addition, a limit is set to the number of wells that are allowed be routed to the same pipeline and a limit is set to the total number of pipelines and satellite wells which are allowed to be routed to the same separator. Only certain wells are allowed to be closed. Example rules are summarized in Table 7.3, these are specific for the case later in the thesis referred to as the base case.

**Table 7.3: Rounding Rules of Problem Specific Heuristics**

| **Evaluation order** | Manifold 1 wells to pipelines | Manifold 2 wells to pipelines | Pipeline to separator | Topside wells to separator |
|---|---|---|---|---|
| **Maximum number of wells to pipelines** | 4 wells | 3 wells | | |
| **Maximum number of pipelines to separators** | | | 2 pipelines | |
| **Maximum total number of pipelines and wells to separators** | | | | 2 pipelines+2wells or 1 pipeline+3wells or 4 wells |
| **Illegal combinations wells to the same pipeline** | The 4 wells of highest priority | The 2 wells of highest priority | | |
| **Allowed to close** | Wells of lowest priority | Wells of lowest priority | | Wells of lowest priority |

### 7.3.3 Non-convexity Measures

The multi-start generator is initiated by declaring the number $ns$ of different starting points the code should generate, for example one thousand. The code then generates that number of random starting points. For each such starting point, the code randomly sets the continuous control variable values, namely the gas lift rate and wellhead pressure of each well. The upper and lower bounds on these variables are respected in the generation. The binary control variables, namely the routing and on/off variables, are also set. These are randomly generated to be between 0 and 1. The remaining variables of the model are implicit variables, and get their values when all of the control variables are set. Note that when the starting points are created in this way, they are likely to be both NLP- and integer infeasible.

After the starting points are generated, the problem is solved in Ipopt with each of these starting points. The obtained solutions are mainly NLP feasible but IP infeasible, resulting from solver convergence. Some solutions are also NLP infeasible, which is often the case when Ipopt reaches the limit on number of iterations. Note the distinction between the NLP feasibility of the starting points and resulting (root node) solutions.

The solutions are sorted according to objective function value and subsequently clustered in $nc$ groups. This means that the list of sorted solutions is analyzed so as to identify the solutions that lie close together in terms of control variable values. A threshold value is set for each of the three categories of control variables, namely the gas lift rate, the wellhead pressure and the binary variables. If the absolute value of the difference between two solutions in the sorted list is within the threshold value, the two solutions are considered similar with regards to that control variable. E.g. if one solution has a wellhead pressure of 50 bar on well 1 and the threshold value is 5, then another solution with a wellhead pressure between 45 and 55 on well 1 will be deemed similar to the first solution with regards to wellhead pressure of well 1. Finally, if a high percentage (e.g. 80%) of the control variables is similar between two solutions, they are considered to be in the same cluster.

The purpose of including *several* starting points for the root node is to search a wider area of the solution space. But if the corresponding solutions are simply sorted by objective function value and chosen based on their potential, the solutions might be rather similar in regards of variable values. This similarity is not exposed by simply analyzing the objective functions. So in an attempt to avoid choosing starting points with high potential solutions that are fundamentally the same, the clustering scheme is designed.

The randomly generated starting point corresponding to the solution with highest objective function value in each cluster is then extracted. These $nc$ starting points make up the potential root node starting points if the multi-start option is used. The searching strategy implemented is best-first. Note that the branch and bound algorithm will then have $nc$ starting points for the root node instead of the traditional one. This can be visualized as $nc$ separate branch and bound trees, except that the best-first search strategy will always choose the most promising node to branch on. This means the algorithm will jump among the $nc$ "separate branch and bound trees" if it finds the highest objective function value in different trees each time. In the same way the best lower bound obtained so far is shared among the trees.

It is possible to only cluster the start points that lead to solver convergence and NLP feasible solutions, or also include the starting points that lead to the solver hitting maximum iterations. Including the latter will lead to higher upper bounds and consequently larger trees. But it might also lead to searching areas of the solution space which the algorithm would never enter if only the former were considered, which again might lead to finding better solutions.

Further, the algorithm allows branching on nodes that have an objective function value that is above a certain fraction of the best lower bound, e.g. $c$=0.95, see Figure 7.3. In the implementation, nodes are given levels which essentially indicates how many times their ancestry has obtained objective functions worse than the best lower bound. Since the level indicator is taken into account in the branch and bound code, this branching on “slightly bad” nodes will not continue without limit. In the figure, the level limit is set to 2, so that “bad” nodes can only continue in two steps down the tree if their child nodes continue to have worse objective functions than the best lower bound. Note that the level indicator is inherited from parent to child nodes.

In the example illustrated in Figure 7.3 suppose the tree is part of a larger tree where a lower bound of 100 has already been obtained. Node 1 is a zero level node. When this node is solved, its objective function is 98, meaning it is above the boundary of $c$=0.95 of the lower bound. If this was B&B for a convex problem, this node 1 would have been pruned. Here this node is branched on, but the levels of its child nodes are increased as can be seen in nodes 2 and 3. Following the right side of the figure, node 3 again obtains an objective above the $c$=0.95 limit, so it can be branched on while its child nodes get a higher level. Node 7 then has level 2, and since its objective function still has not increased above the lower bound, this node will be pruned. Looking at the left side of the figure, node 2 is level 1, but since its objective function improves above the lower bound of 100, the child nodes 4 and 5 will get a clear level, meaning level 0. Node 4 is pruned because the objective function here is below the $c$=0.95 which can be called regular pruning and not a prune because of the node level. Note that the level limit of 2 can be adjusted in the algorithm, and is set to 2 here for simplicity.

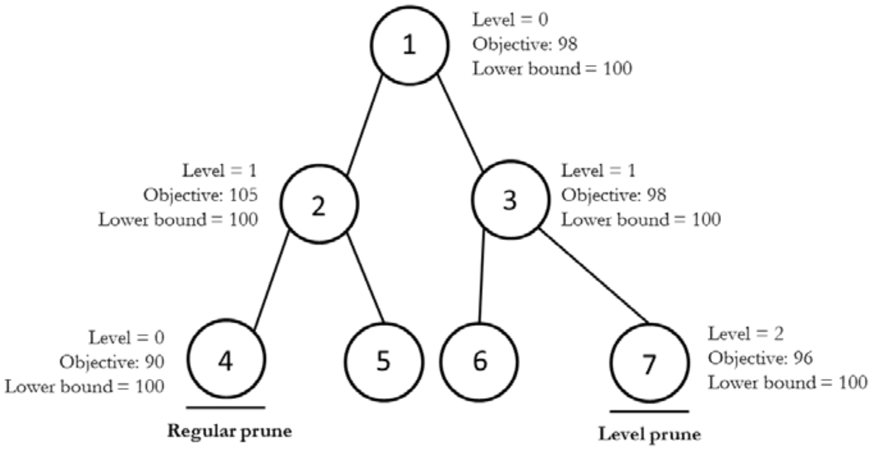


**Figure 7.3: B&B tree demonstrating regular prune versus level prune**

### 7.3.4 Simulator Implementation

The well and pipeline functions (5.12) and (5.15) represent simulator behavior, making the extraction of both the function value and its derivative nontrivial. If interfacing is possible, one option is to call the relevant simulators directly in the optimization. Another possibility is to build some form of approximation of the simulator data, obtained by running the simulator over a reasonable range of input values and recording the output.

Derivatives of the simulator functions must also be supplied to the solver. Some recent simulators supply derivative information, which can be used if the simulators are called directly. If the simulators are approximated, a suitable differentiating scheme must be designed dependent on the approximation used. The well and pipeline simulators are here approximated by interpolating in data tables, while derivatives are obtained by finite differencing. In the following the implementation of the interpolation and the finite differencing is explained in detail.

**Well and Pipeline Pressure Drop Simulator Implementation**

First, the implementation of equations (5.12) is discussed. The interpolation code itself is a standalone bit of code that is invoked by the optimization algorithm with the necessary input values. The output is the resulting oil rate. Because the GOR and WC are considered constant for this field, it is mainly the oil production rate of each well that is in question from an implementation point of view.

Bilinear interpolation (Wikipedia, 2012) is applied to obtain the corresponding oil production rate. This method consists of finding the function value $p = f(x, y)$when the function values are only known at a grid surrounding this point, namely at the points $f_1 = (x_1, y_1), f_2 = (x_2, y_1), f_3 = (x_2, y_2), f_4 = (x_1, y_2)$ in Figure 7.4. Two linear interpolations are done in the x-direction to find the function values at the points $r_1 = (x, y_1)$ and $r_2 = (x, y_2)$. Then a final linear interpolation is done in the $y$-direction between the points $r_1$ and $r_2$ to get the function value at the desired point $(x, y)$. Notice also that in this example the interpolation was first done in the x-direction. The result will be the same if the interpolation is first done in the $y$-direction.

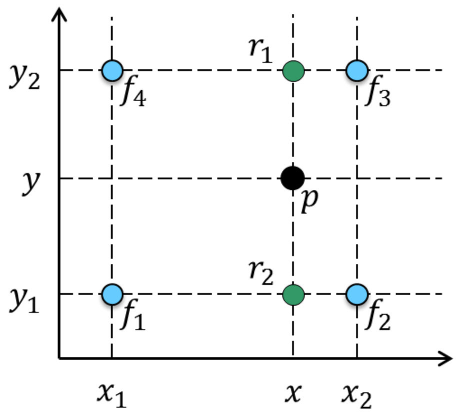


**Figure 7.4: Bilinear Interpolation**

Since the well oil production rate is a function given by two input values, the bilinear interpolation technique is well suited. Bilinear interpolation generally gives a quadratic interpolant along any straight line inside the quadrant of values. The exception is if that straight line is parallel to the x- or y-direction, then the interpolant is linear. In other words, if either the x or y value is held constant, the interpolation function is linear. This is what happens when we calculate the derivatives.

Partial derivatives must be given to Ipopt for this equation, namely the change in the oil output as a result of a marginal increase of the wellhead pressure at, or the gas lift allocated to, each particular well. This is done using finite differencing, and a suitable perturbation value is chosen for each input value. The change in oil output is then recorded when a certain input variable is increased with its perturbation value as opposed to the original run. It is the previously explained interpolation code that is called again with one input variable increased marginally. This becomes the partial derivative of oil rate with respect to that input variable. This partial derivative will be constant inside the current quadrant, because the interpolation function is linear.

As the interpolating point wanders from quadrant to quadrant, the interpolated function value changes continuously. But the gradient of the interpolated function changes discontinuously at the boundaries of each quadrant, see Figure 7.5. The solid black line represents the interpolated function moving through three quadrants. The derivative, given by the dotted line will be discontinuous when the quadrant changes. This might cause problems for the convergence of the search algorithm.

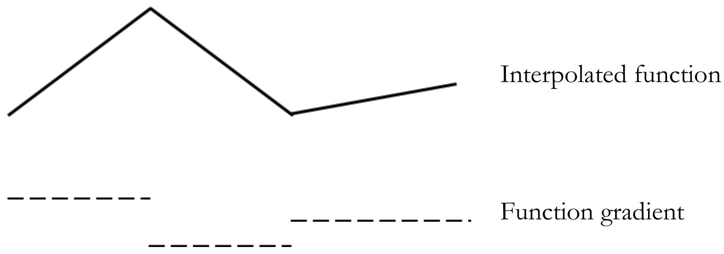


**Figure 7.5: Interpolated example function and its derivative**

This section explains the implementation of the pipeline pressure drop equations (5.15). In the same way as for the well functions, an interpolation code is made. Notice here that the pressure drop is a function of three input values. Consequently the interpolation is done in three dimensions using trilinear interpolation (Wikipedia, 2012). The concept is the same as that of bilinear interpolation, except that now the interpolation is done inside a cube surrounding the point $p$ instead of a grid. The same concepts explained for the two-dimensional case also apply for this three-dimensional case.

Partial derivatives are also obtained for this function by finite differencing. Each of the input variables are marginally increased one at a time, and the function is evaluated again at this point. The difference in pressure drop value will give the partial derivative of the function with respect to the increased input variable.

## 7.4 Implementation of Other Methods

In this section short introductions are given to the additional solution methods and the solvers applied, further elaborations of the developments and approaches can be found in the project work done by the authors (Shamlou and Ursin-Holm, 2012).

**MILP**

The MILP is a linearized version of the model in Section 5.2, with big M and SOS 2 linearization techniques. It is solved using the mathematical programming language Mosel. Together with the solver Xpress it is commonly used to solve large scale linear optimization problems. Xpress can handle linear and quadratic objective functions and linear constraints, hence it can only be used for a piecewise linear approach. The solutions provided by the MILP model are independent of starting points.

**Black-Box**

This Black-Box method is a framework developed by the Center for Integrated Operations (IO-Center) treating the whole production network as one all-encompassing black box. This framework includes handling of wells and pipelines with a NOMAD software integrated on top making up the Black-Box optimization algorithm, which is a derivative free search method (Digabel, 2011). The MADS algorithm used by NOMAD iteratively evaluates the black-box function(s) at trial points lying on a mesh around a *current* point. Based on these evaluations, the algorithm may choose to move to another better point, where the process is repeated. Each such evaluation is one iteration. The method is a realistic portrayal of the underlying production network as the wells and pipelines can be incorporated by connecting simulators directly into the framework or by loading tables generated upfront with the simulator data. In the testing conducted here, the latter is used.

**Nonlinear Approximations Approach**

Incorporated in this Nonlinear Approximation model (NA) the behaviors of wells and pipelines are represented by proxy models, while the remaining equations of Section 5.2 apply as previously stated. One or several nonlinear functions are created for all wells and pipelines approximating oil flow and pressure drop respectively. AMPL is used together with the solver Bonmin. AMPL is a modeling language for describing and solving large scale optimization problems (AMPL Optimization LLC, 2012). AMPL supports many solvers, like Bonmin and Knitro, and can be used to solve both linear and nonlinear problems, in discrete and continuous variables.

# 8. Computational Study

In this chapter the results of the computational study will be given. The chapter starts with analyzing the heuristics, after which the testing and tuning of the B&B algorithm and relevant parameters is outlined. When the best parameters and strategies are chosen based on this technical testing, the optimal versions of the algorithm are tested on 9 case scenarios in order to confirm the robustness of the solution method. Finally, results are also obtained from other solution methods for comparison purposes and to support the final discussion about the SmartOpt-B&B algorithm.

## 8.1 Technical Testing

The case instance used during the development and technical testing of the heuristics and B&B algorithm is composed by platform 1 and the GORs and WCs for the intermediate phase. This case will be called the base case. In the following the heuristics are analyzed, and then all B&B strategies mentioned and non-convexity measures are tested. Based on these results, two versions of the algorithm with different emphases, one on solution time and the second on solution quality will be designed and brought on to the next subchapter for robustness testing.

**Start Points**

One thousand random points are generated and solved with Ipopt for further investigation. The resulting solutions sorted by their objective function value can be seen in Figure 8.1. These are subsequently clustered, and the starting points resulting in the 4 most promising NLP feasible solutions and the 4 most promising solutions where Ipopt hit maximum iterations are extracted. The solutions obtained from the chosen starting points are labeled in the graph and are the same ones as given in Table 8.1. The clustering is stricter regarding the NLP infeasible solutions, while a more lenient clustering is applied on the feasible ones. This is evidenced in the graph by the chosen infeasible solutions being quite close to each other, while the feasible ones are more spread out. The creation and clustering of the starting points took about 15 minutes.

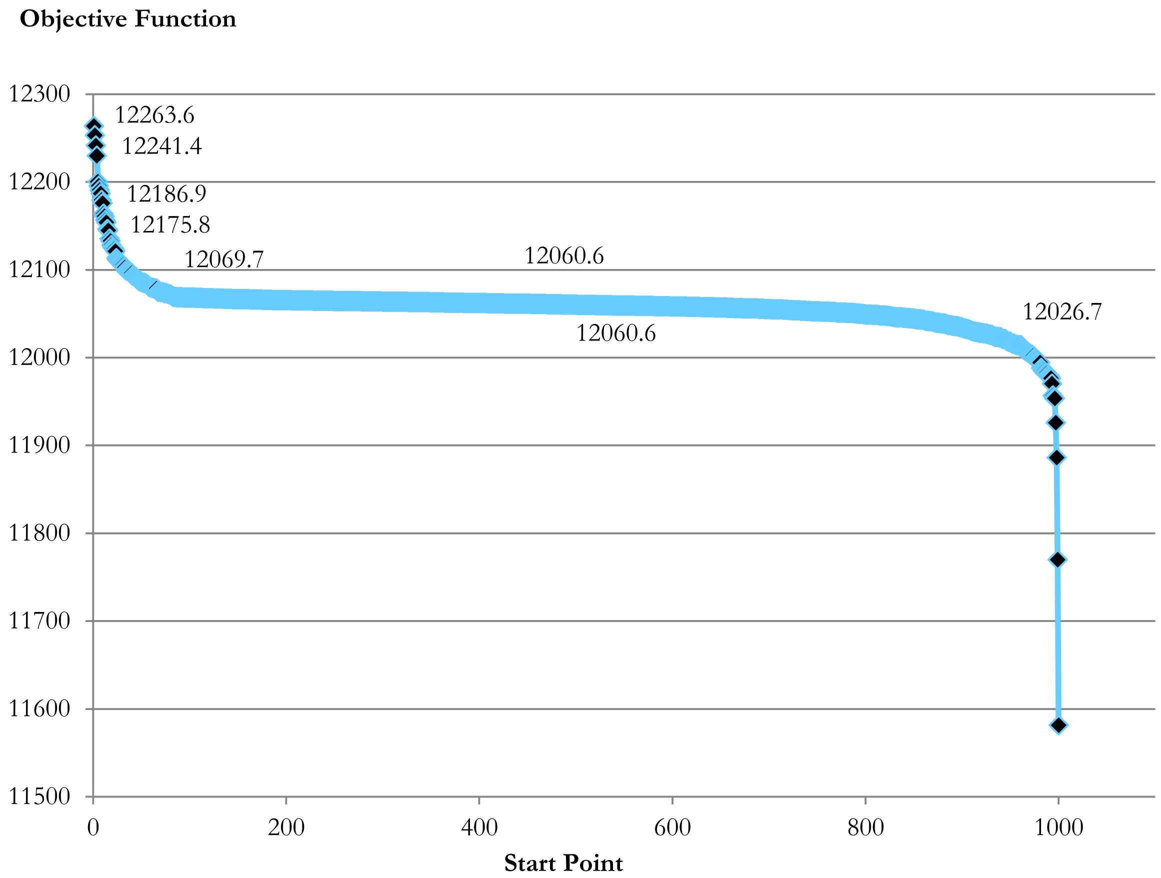


**Figure 8.1: One Thousand Randomly Generated Start Points**

In Table 8.1, starting points 2-9 are obtained from the multi-start generator. Note that these are NLP infeasible. Starting points 2-5 result in NLP feasible solutions, while the solver reaches maximum iterations in the remaining. In addition, one NLP feasible start point is calculated, which also results in a NLP feasible solution when the problem is solved. This is the first numbered starting point given in Table 8.1.

**Table 8.1: Generated Start Points and Corresponding Solutions**

| Start Point Number | Start Point NLP Feasibility | Solution NLP Feasibility | NLP Solution |
|---|---|---|---|
| 1 | Feasible | Feasible | 12057.5 |
| 2 | Infeasible | Feasible | 12069.7 |
| 3 | Infeasible | Feasible | 12060.6 |
| 4 | Infeasible | Feasible | 12060.6 |
| 5 | Infeasible | Feasible | 12026.7 |
| 6 | Infeasible | Infeasible | 12263.6 |
| 7 | Infeasible | Infeasible | 12241.4 |
| 8 | Infeasible | Infeasible | 12186.9 |
| 9 | Infeasible | Infeasible | 12175.8 |

### 8.1.1 Heuristics

Three heuristics are incorporated, and the design choice is whether or not to use the different heuristics and with what frequency. Firstly, the heuristics are tested outside the B&B algorithm on the base case. All heuristics are initiated by solving the integer relaxed MINLP. The testing is done with starting points 1-5 as given in Table 8.1 and is conducted to demonstrate the characteristics of each heuristic on the same problem, and compare them with each other.

Initially, the feasibility pump is tested to identify its performance and the impact of different combinations of the parameter values in the objective functions. As elaborated in Section 6.3.2, in the objective functions $\alpha$ is the weighted average of maximizing total oil production and $1-\alpha$ is the weighted average of minimizing the sum of the distances between the previously obtained solution and the solution sought to be found. Parameter $D$ is the scaling parameter, scaling the term for maximizing total oil production in the objective function.

The FP is tested for three values of the parameter $D$ with associated regimes of $\alpha$. The values of $\alpha$ are strictly decreasing from 1 to 0 throughout the iterations. Note that the decrease in $\alpha$ is somewhat dependent on $D$. This means that for $D$=100, $\alpha$ has a higher rate of decrease (blue line) than for $D$=1000, however for $D$=10 $\alpha$ decreases even faster. Hence, for all values of $D$ the term minimizing distance between the previously obtained solution and the solution sought to be found becomes increasingly important, this again increases the possibility for the FP to converge.

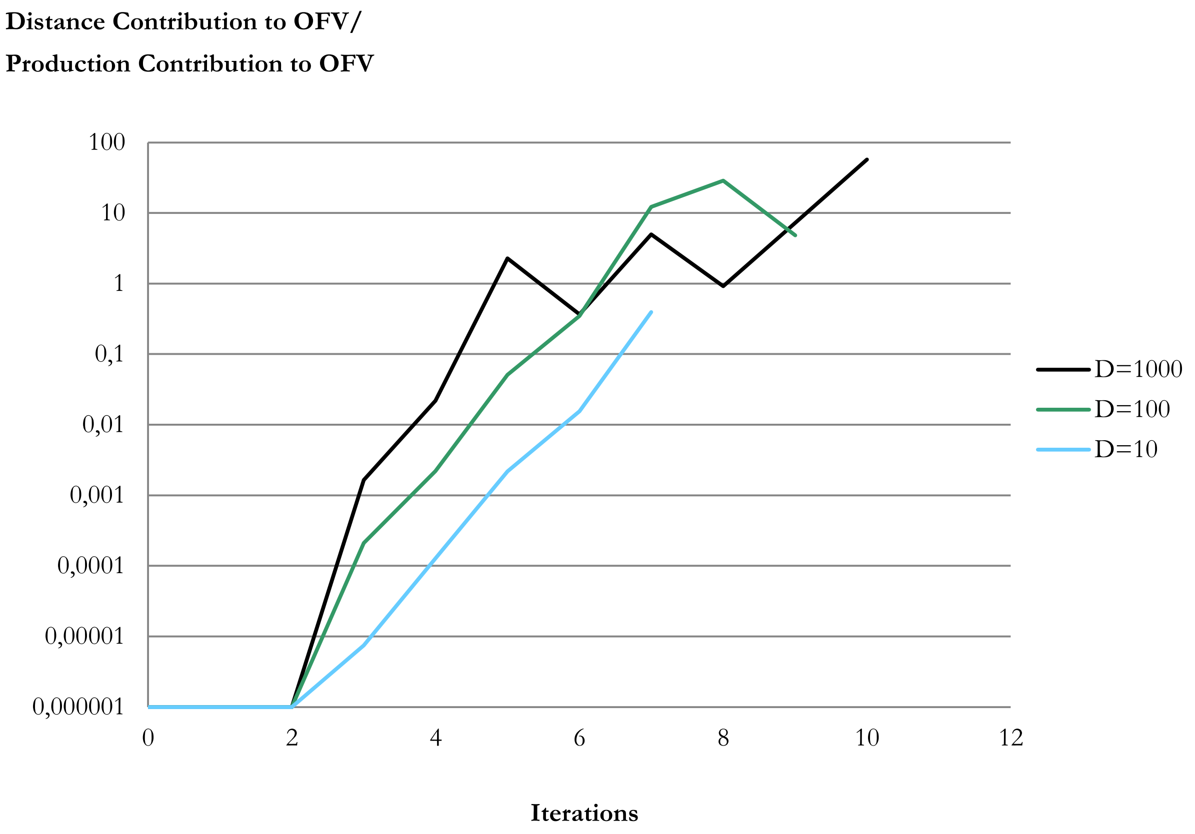


**Figure 8.2: Contribution Ratio of Distance to Production in the Objective Function**

Figure 8.2 is a result of the FP tested with starting point 1. It illustrates how the objective function weighting starts by emphasizing the original objective function and gradually shifts towards minimizing the distance to the previously obtained solution. The contribution of minimizing distance in the objective function is given relative to the contribution of maximizing production, and the ratio is given for several iterations. As can be noticed, the contribution of minimizing distance is almost 0 for the first few iterations. Even though this contribution does not strictly increase the trend is increasing. Note that after a given number of iterations $\alpha$ will take the value 0, meaning that only minimizing distance will contribute to the objective function value. However, in all cases tested the FP converges before this happens. When the lines associated with each $D$ stop in the figure, the FP has converged.

**Table 8.2: Solutions Obtained from the Feasibility Pump**

| | $D = 1000$ | | | | $D = 100$ | | | | $D = 10$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Start point | Objective | $i$ | NLP | FP-MILP | Objective | $i$ | NLP | FP-MILP | Objective | $i$ | NLP | FP-MILP |
| 1 | 10403.2 | 6 | 12 | 6 | 10377.3 | 9 | 18 | 9 | 10399.6 | 6 | 12 | 6 |
| 2 | 10406.9 | 7 | 14 | 7 | 10433.1 | 8 | 16 | 8 | 10334.6 | 5 | 10 | 5 |
| 3 | 10372.9 | 5 | 10 | 5 | 10399.6 | 10 | 20 | 10 | 10263.8 | 6 | 12 | 6 |
| 4 | 10376.4 | 6 | 12 | 6 | 10399.9 | 9 | 18 | 9 | 10440.0 | 6 | 12 | 6 |
| 5 | 10447.2 | 7 | 14 | 7 | 10447.3 | 7 | 14 | 7 | 10327.1 | 7 | 14 | 7 |

The FP is also tested with starting points 2-5. Table 8.2 shows all the resulting MINLP solutions. The column marked $i$ counts the number of iterations until convergence, while the column marked NLP counts the total number of full FP-NLPs and FP-NLP-Rs solved. FP-NLP-R is a reduced NLP which means solving the problem with all the binary variables locked, eliminating integer variables and reducing the total number of variables in the problem. The column marked FP-MILP shows how many FP-MILP problems the FP solves.

Due to values of parameter $\alpha$ all tests of the FP appear to converge at around 6-10 iterations. Note that the total number of problems solved to obtain convergence is low. As elaborated in Section 6.3.2 feasible solutions may be found before convergence through solving the FP-NLP-Rs of Step 4 in the FP algorithm. In the table, the best feasible solution from each test is presented, which is sometimes found during the first few iterations. Note that the iteration count in the table gives iterations until *convergence*. It does not seem to be any correlation between the number of problems solved and the obtained solution value.

The FP provides good solutions for all parameters tested and all starting points. Further, the values of $D$ and $\alpha$ seem to affect the objective function such that the solutions result from different parts of the solution space. However, it is difficult to identify a trend in how the values for $D$ and $\alpha$ impact the solution value, and what combinations of $D$ and $\alpha$ are superior. Nevertheless for further tests of the FP incorporated in the SmartOpt-B&B, the parameter $D$ is set to 1000, with the belonging $\alpha$ values. This is because it provides the best solution for the NLP feasible starting point and good solutions for the remaining ones, better than $D$ =10 in virtually all cases except one. In addition, it always solves the problem within less iterations than when $D$=100.

H1 and H2 are also tested for the 5 continuous NLP solutions stemming from the same starting points 1-5. Note that for these 5 continuous solutions, no binary variables are locked. However the heuristics can handle locked binary variables when included in the branch and bound algorithm. The solutions are presented in Table 8.3.

**Table 8.3: Solutions Obtained from the Problem Specific Heuristics**

| | H1 | | H2 | |
|---|---|---|---|---|
| **Start point** | **Objective** | **NLP** | **Objective** | **NLP** |
| 1 | 9 732.4 | 1 | 9 606.0 | 15 |
| 2 | 8 918.8 | 1 | 9 606.0 | 15 |
| 3 | 9 732.4 | 1 | 9 669.3 | 14 |
| 4 | 9 523.4 | 1 | 9 859.5 | 16 |
| 5 | 9 732.4 | 1 | 9 606.0 | 15 |

The column marked NLP presents the total number of solved NLPs and reduced NLPs. H2 solves a number of NLPs each time it is called depending on the characteristics of the continuous solution, whereas H1 only solves one reduced NLP every time. The two heuristics find inferior solutions to the FP, however in all the cases both heuristics find good MINLP solutions within only a few seconds. It is worth mentioning that when H1 is applied to 5 different continuous solutions the heuristic generates identical integer solutions for several of them, the same is true for H2. Although all NLP solutions are in principal different some characteristics are similar causing this to happen. This can be seen for the H1 results provided for the continuous solutions of starting point 1, and starting points 3 and 5. H1 finds the best solution in 3 out of 5 cases, thus H1 and H2 may be seen as substitutes.

### 8.1.2 B&B Strategies

Initially, the B&B algorithm is tested for various branching strategies, before different heuristic configurations are included in the algorithm. After this, the non-convexity measures are tested.

Regarding the technical B&B testing, varying load on the workstation leads the solution times to vary (often up to +/- 20%). But all results appearing in the same table are run at the same time, so the load is uniform for all those instances making the solution times comparable. Therefore only a percentage is given relative to the quickest test instance of that table for easy comparison. In addition, all tests in this section have a runtime limit of 24 hours, so if a test does not converge during this time it is stopped and the corresponding upper bound is given.

Variable choice when branching is tested first. The $c$ is set to 1.0 meaning that the testing is a standard B&B where negative gaps are not allowed, which also means that the level limit is of no consequence here. Additionally, only starting point 1 is used. Gap tolerance is set to 5, which is about 0.04% of the root node solution and initial upper bound. The searching strategy implemented is best-first.

Three different branching strategies are implemented. The first is to choose the binary variable with the highest fractional value, the second is to choose the variable closest to either 0 or 1. Finally, the variable value closest to 0.5 is chosen. The first strategy implements a small change in the 1-branch since the variable chosen has a high fractional value. In the 0-branch a large change is executed since the variable is locked to zero even though the previous solution suggested a high fractional value. The second strategy gets similar disparity in the change implemented in each branch, although the small and large changes will be more evenly distributed among the 1- and 0-branches. Closest to 0.5 implements a more balanced change in the branches as it locks the variable with fractional value closest to 0.5 to 0 and 1 respectively. The results are given in Table 8.4.

**Table 8.4: Results Branching Strategy**

| Test | Branching Strategy | Search Strategy | Time | Nodes | Objective | Upper Bound |
|---|---|---|---|---|---|---|
| 1 | 1 | Best | 100.0% | 95255 | 10499.5 | - |
| 2 | 0.5 | Best | 970.9 % | 116649 | 10473.2 | 11009.6 |
| 3 | 0/1 | Best | 970.9 % | 80949 | 10502.8 | 12152.9 |

As the results indicate, the first strategy proves much more successful for this problem instance than the two latter strategies. Considering solution time most fractional performs much better and is able to converge in a reasonable amount of time. The other strategies are stopped after 24 hours, with the upper bounds given in the table. Note also that test number 3 finds a marginally better objective function value. Consequently, most fractional is used from here on in an attempt to keep the solution times at their minimum.

The heuristics are also tested in a B&B tree to decide on the best configuration. Still negative gaps are not allowed, the B&B is started with the single NLP feasible starting point and branching is done on the variable with highest fractional value. The FP is only run on the root node, H1 is applied to every or every $10^{th}$ solved node, while H2 is only used on every $10^{th}$ node that is solved.

The column marked NLP counts the number of times a full size NLP version of the problem is solved. This number is a summation of the total number of nodes solved, and when relevant the number of NLP problems solved in the FP and H2. Note that the column reduced NLP counts the number of problems solved with all the binary variables locked, which is only relevant when H1 is applied.

The results can be seen in Table 8.5. According to Table 8.2, the FP runs 6 NLPs, 6 reduced NLPs and 6 MILPs when applied to the root node. Looking at test number 5, these 3 columns are added a value of 6 compared to test number 4 where the FP is not used. This addition also takes place for all subsequent runs where the FP is included.

**Table 8.5: Results Heuristic Configurations**

| | | | | | Iterations | | | LB Updates | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Test | Heuristics | Relative Time | Objective | Nodes | NLP | Reduced NLP | MILP | FP | H1 | H2 | B&B |
| 4 | - | 108.1 % | 10499.5 | 93399 | 93399 | 0 | 0 | 0 | 0 | 0 | 11 |
| 5 | FP | 100.0 % | 10499.5 | 93399 | 93405 | 6 | 6 | 1 | 0 | 0 | 3 |
| 6 | H1(10) | 100.5 % | 10499.5 | 93399 | 93399 | 9339 | 0 | 0 | 14 | 0 | 1 |
| 7 | H2(10) | 116.5 % | 10499.5 | 88499 | 142188 | 0 | 0 | 0 | 0 | 15 | 0 |
| 8 | FP<br>H1(1) | 128.9 % | 10499.5 | 90999 | 91005 | 91005 | 6 | 1 | 6 | 0 | 1 |
| 9 | FP<br>H1(10) | 100.5 % | 10499.5 | 93399 | 93405 | 9345 | 6 | 1 | 2 | 0 | 1 |
| 10 | FP<br>H1(1)<br>H2(10) | 133.0 % | 10499.5 | 90999 | 107155 | 91005 | 6 | 1 | 6 | 0 | 0 |
| 11 | FP<br>H1(10)<br>H2(10) | 127.4 % | 10499.5 | 93399 | 141317 | 9345 | 6 | 1 | 1 | 2 | 1 |

Looking at the results, all test runs found the same optimal solution. Test number 5 uses less time than test 4, which is a result of applying the feasibility pump and obtaining a lower bound initially in the tree search. This reduces the number of nodes created and put into the list of nodes to be solved, which is evident by step 8 in the B&B Algorithm 1 given in Section 6.3. Note that number of nodes created is not given in the tables, only number of nodes solved. Some of the configurations have smaller trees, a result of obtaining good lower bounds provided by heuristics. But as the iteration columns indicate tests 7, 8 and 10 end up solving more problems, and thus obtain higher solution times. Heuristic 2 leads to a smaller tree in test 7 than in 8 and 10 only because it is started in a different node, and consequently is applied to another set of nodes throughout the search. Since H1 only solves reduced NLPs, and both heuristics more or less end up with the same solutions, configurations 7, 10 and 11 are deemed inferior.

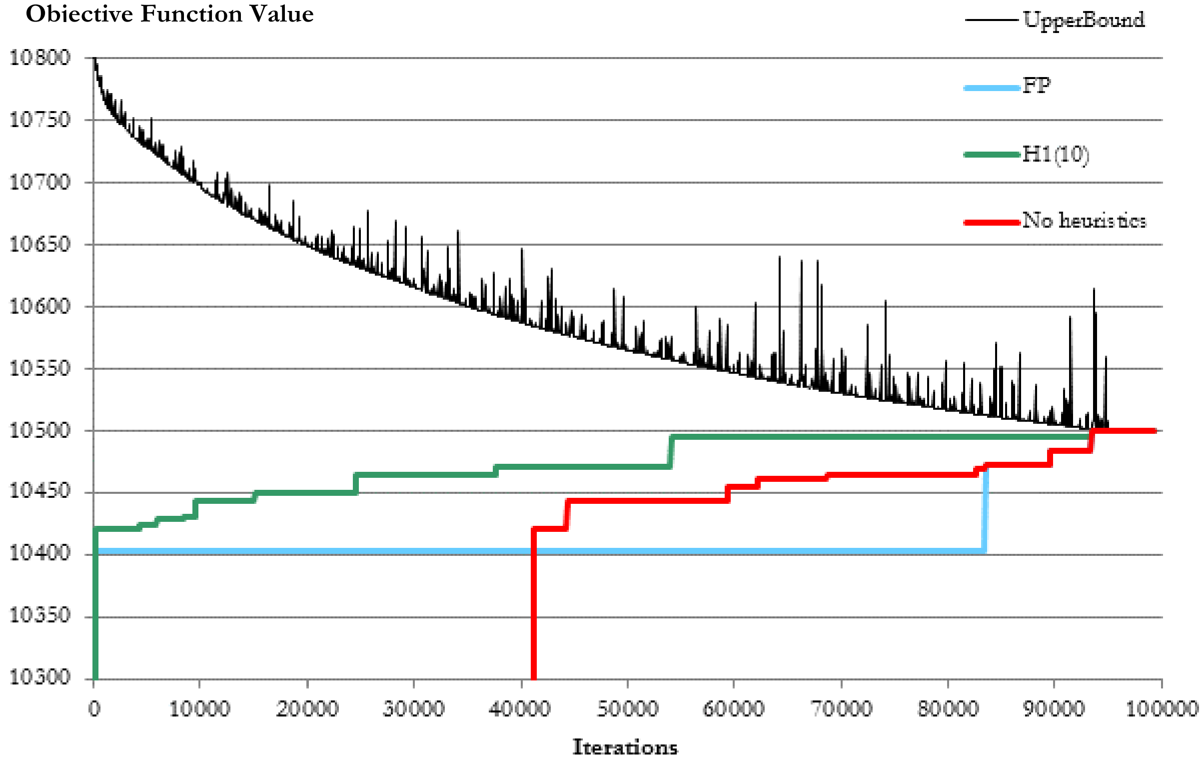


**Figure 8.3: Gap and Lower Bound Updates**

Figure 8.3 shows the decrease of the upper bound when using the best-first search strategy for the base case. The fluctuating upper bound clearly indicates a non-convex problem. Lower bound updates for tests 4, 5 and 6 are also shown in the figure to assess the effect of including the different types of heuristics. Heuristic 2 updates in test 7 show similar characteristics as test 6, and are therefore not included. The vertical axis is limited in both directions in order to get a more detailed view of the lower bound updates. In reality, the upper bound starts at roughly 12000. Furthermore, the first lower bound update for the green line is obtained in the root node and has a value of 9732.4. Regarding the red line, the lower bound is first updated as shown in the figure. So if it is important to find good feasible solutions early, it seems promising to include heuristics in the B&B tree. This is evident by the green and blue lines.

### Non-convexity Measures

The number of starting points, the level limit and the parameter $c$ are all values designed to tackle the non-convexity of the problem. So if one expects a highly non-convex problem these values must be carefully tested and set. In addition, the gap termination tolerance is set to zero, which means that the B&B stops when the list of nodes to solve is empty. The FP is included because it gives a lower bound initially, which leads to fewer nodes being created and considered, which reduces the solution time somewhat.

Beginning with the multiple starting point option, it will be used to test how the B&B algorithm performs if started with more than one starting point, in addition to assessing the effect of using starting points that lead to NLP infeasible versus NLP feasible (root node) solutions.

Table 8.6 includes starting points 2-5 leading to NLP feasible solutions in the root node, while Table 8.7 uses starting points 6-9 leading to the solver reaching maximum iterations. Starting point 1 is always included. Hence, the column marked start points gives the start points that were included for the root node of the B&B *in addition* to the NLP feasible starting point.

**Table 8.6: Results Multiple Starting Points – NLP Feasible Root Node Solutions**

| Test | Gap Tol | $c$ | Level Limit | Heuristics | Start Points | Relative Time | Objective | Nodes |
|---|---|---|---|---|---|---|---|---|
| 12 | 0 | 1.0 | 0 | FP | 2 | 100.00 % | 10499.5 | 197502 |
| 13 | 0 | 1.0 | 0 | FP | 2,3 | 176.95 % | 10499.6 | 315559 |
| 14 | 0 | 1.0 | 0 | FP | 2,3,4 | 226.94 % | 10507.7 | 396922 |
| 15 | 0 | 1.0 | 0 | FP | 2,3,4,5 | 331.51 % | 10507.7 | 545733 |

**Table 8.7: Results Multiple Starting Points – Max Iterations Root Node Solutions**

| Test | Gap Tol | $c$ | Level Limit | Heuristics | Start Points | Relative Time | Objective | Nodes |
|---|---|---|---|---|---|---|---|---|
| 16 | 0 | 1.0 | 0 | FP | 6 | 100.0 % | 10502.4 | 340900 |
| 17 | 0 | 1.0 | 0 | FP | 6,7 | 215.1 % | 10504.1 | 691849 |
| 18 | 0 | 1.0 | 0 | FP | 6,7,8 | 274.4 % | 10514.6 | 817414 |
| 19 | 0 | 1.0 | 0 | FP | 6,7,8,9 | 339.5 % | 10514.6 | 978057 |

The optimal objective function values become marginally better by including additional starting points. On the other hand, the size of the trees and consequently the solution times increase quite dramatically. The highest oil productions are found using the points that lead to the solver hitting maximum iterations, test numbers 18 and 19 find the best objective function value, although number 18 uses less time doing so. The solution times increase more or less proportionally with the number of additional starting points. In the following testing only the starting point leading to the best NLP feasible root node solution is included in addition to start point 1.

Moving on to the parameter $c$, this value is tested for four different values, namely 1.00, 0.98, 0.96 and 0.94. The results can be seen in Table 8.8. The level limit is held constant at 1, in addition to the design concluded on from the heuristic step.

**Table 8.8: Results Negative Gap Coefficient**

| Test | Gap Tol | c | Level Limit | Heuristic | Start Points | Time | Objective | Nodes |
|---|---|---|---|---|---|---|---|---|
| 20 | 0 | 1.00 | 1.0 | FP | 1,2 | 100.00 % | 10499.5 | 197502 |
| 21 | 0 | 0.98 | 1.0 | FP | 1,2 | 248.82 % | 10505.8 | 379338 |
| 22 | 0 | 0.96 | 1.0 | FP | 1,2 | 290.04 % | 10505.8 | 423052 |
| 23 | 0 | 0.94 | 1.0 | FP | 1,2 | 297.42 % | 10505.8 | 431202 |

In Table 8.8, a large increase in solution time is evidenced when changing $c$ from 1.00 to 0.98. The solution time increase is natural considering the much larger tree (see number of nodes), as this will increase the total number of NLPs solved during the search considerably. Unfortunately, this increase in solution time only leads to a very small increase in the amount of oil produced. Going further to a $c$ of 0.96 and then to 0.94, the solution times increase somewhat, but no better solution is found.

Next the level limit is tested for values of 0, 1, 2 and 3. The design from the previous steps are kept, with the $c$=0.98 and starting points 1 and 2. The results are given in Table 8.9.

**Table 8.9: Results Level Limit**

| Test | Gap Tol | $c$ | Level Limit | Heuristic | Start Points | Time | Objective | Nodes |
|---|---|---|---|---|---|---|---|---|
| 24 | 0 | 0.98 | 0 | FP | 1,2 | 100.00 % | 10499.5 | 217782 |
| 25 | 0 | 0.98 | 1 | FP | 1,2 | 234.48 % | 10505.8 | 379338 |
| 26 | 0 | 0.98 | 2 | FP | 1,2 | 373.09 % | 10505.8 | 576294 |
| 27 | 0 | 0.98 | 3 | FP | 1,2 | 502.63% | 10505.8 | 757402 |

Table 8.9 again shows large increases in solution time and number of nodes when allowing negative gaps for several levels to continue in the B&B tree. And again only marginal improvements of the oil production are observed when allowing the level limit to increase. Further, no improvement is seen after raising the limit from 1 to 2 and then to 3.

The given optimal solutions in all the tables have also been analyzed in terms of control variable values to assess their similarity. Some solutions with essentially the same oil production rate, e.g. tests 12 and 13, are fundamentally different in their network routing, gas lift allocation and wellhead pressure values. In other cases, optimal solutions with different values, such as tests 12 and 14, show very similar control variable values.

**SmartOpt-B&B Versions**

In this section the two versions of SmartOpt-B&B are designed as a result of the previous testing. The first is with emphasis on solution time, and is run with a gap termination tolerance of

5. This makes the inclusion of heuristics important. Thus, the first design is composed of using FP plus H1 in every $10^{th}$ node without any non-convexity measures. In the second design the focus is on solution quality, so the B&B will terminate on a zero gap tolerance, rather it is run until all nodes in the list are solved. Starting points 1-3 are used, with $c$=0.98 and using a level limit of 1. The FP is again included because it gives a lower bound initially, which leads to fewer nodes being created and considered, which reduces the solution time somewhat. The two algorithms will be called SmartOpt-B&B version 1 and 2 respectively. An overview of the different versions is given in Table 8.10.

**Table 8.10: SmartOpt-B&B Designs**

| | **Version 1**<br>**Solution Time** | **Version 2**<br>**Solution Quality** |
|---|---|---|
| **Search Strategy** | Best-First | Best-first |
| **Gap Termination Tolerance** | 5 | 0 |
| **Branching Strategy** | Most fractional | Most fractional |
| **Heuristics** | FP + H1(10) | FP |
| **Starting Points** | Start point 1 | Start points 1,2,3 |
| $c$ | 1.0 | 0.98 |
| **Level Limit** | 0 | 1 |

## 8.2 Case Testing

In this section the results provided by the two versions of the algorithm developed are presented and discussed. Subsequently, the results of the methods from preparatory work are provided (Shamlou and Ursin-Holm, 2012), in order to compare the qualities of SmartOpt-B&B in further discussions. Here solution times are given in seconds, although they are not directly comparable as some of the methods are run on a different computer. They are only included to give a sense of how long it might take to run the algorithms. All the algorithms are allowed to run their course until convergence, unless otherwise stated. Some algorithms are prematurely stopped because of very long solution times, in order to allow complete case testing of all the methods within the timeframe of this thesis.

Initially, a base case was defined and applied for the development of the branch and bound algorithm. To get a sense of the robustness the algorithms are tested for 8 additional scenarios. All 9 scenarios are composed by the 3 production stages and the three platforms given in Chapter

7. The case numbering is given in Table 8.11 below, as a combination of platform and production stage. Note that case 2 is the base case.

**Table 8.11: Cases**

| Platform\Stage | 1 | 2 | 3 |
|---|---|---|---|
| **1** | 1 | 2 | 3 |
| **2** | 4 | 5 | 6 |
| **3** | 7 | 8 | 9 |

**SmartOpt-B&B Version 1 and 2**

The results of solving the 9 case instances with the SmartOpt-B&B are given in Table 8.12 for both versions. The solutions obtained are quite similar, only marginal improvements are observed. However, the solution times and tree sizes differ considerably. Note that test number 8 is particularly challenging for both versions, version 2 is not able to converge within a reasonable amount of time and is thus terminated. The corresponding upper bound is given in the table. The algorithm is able to find solutions for all cases. The table gives seconds until convergence, but note that for version 1 all optimal solutions presented are obtained in less than 3 hours, except case 8.

**Table 8.12: Results SmartOpt-B&B Algorithms**

| | FP | Version 1 | | | Version 2 | | | |
|---|---|---|---|---|---|---|---|---|
| Case | Objective | Objective | Time (sec) | NLP Nodes | Objective | Time (sec) | NLP Nodes | Upper bound |
| 1 | 10885.2 | 11132.5 | 18496 | 102049 | 11132.5 | 181568 | 812675 | - |
| 2 | 10403.2 | 10499.5 | 6340 | 93399 | 10505.8 | 63064 | 597287 | - |
| 3 | 9074.42 | 9218.2 | 6441 | 34849 | 9219.3 | 27123 | 222201 | - |
| 4 | 10982.3 | 11196.2 | 24678 | 199699 | 11206.4 | 256831 | 1204859 | - |
| 5 | 10466.4 | 10517.2 | 12258 | 106799 | 10520.9 | 137901 | 935337 | - |
| 6 | 9071.0 | 9218.6 | 3604 | 52260 | 9219.0 | 24007 | 248367 | - |
| 7 | 10876.4 | 11166.8 | 15227 | 71830 | 11170.3 | 101908 | 410505 | - |
| 8 | 10514.6 | 10682.7 | 85978 | 751487 | 10667.5 | 183600 | 922621 | 10769.5 |
| 9 | 9523.9 | 9688.6 | 8790 | 98899 | 9688.9 | 85722 | 637173 | - |

### MILP Model

Table 8.13 shows the results provided for all 9 test cases. The MILP method is run with two time limits. Firstly, a time limit of 3 hours is included to determine how well the solutions become within this time frame, as solution quality versus solution time is of interest. Further, a maximum time limit of 12 hours is set as a result of running initial tests which indicates that the solutions generally change very little after 12 hours.

**Table 8.13: Results MILP**

| | 3 h | | | | 12 h | | | |
|---|---|---|---|---|---|---|---|---|
| Case | Objective | Best Bound | Gap (%) | MILP Nodes | Objective | Best Bound | Gap (%) | MILP Nodes |
| 1 | 11113.00 | 12443.60 | 11.97 | 39338 | 11124.50 | 11641.50 | 4.65 | 235600 |
| 2 | 10502.80 | 11168.30 | 6.34 | 34955 | 10519.50 | 10772.80 | 2.41 | 233700 |
| 3 | 9213.05 | 10299.10 | 11.79 | 66231 | 9247.36 | 10175.03 | 10.03 | 264436 |
| 4 | 11158.40 | 12461.50 | 11.68 | 40357 | 11171.60 | 11745.70 | 5.14 | 231125 |
| 5 | 10526.10 | 11096.00 | 5.41 | 77718 | 10528.70 | 10760.40 | 2.20 | 400625 |
| 6 | 9254.26 | 9812.45 | 6.03 | 48752 | 9254.66 | 9326.16 | 6.77 | 288544 |
| 7 | 11155.10 | 11732.90 | 5.18 | 73552 | 11155.10 | 11409.10 | 2.28 | 314179 |
| 8 | 10721.90 | 12247.4 | 14.23 | 24849 | 10735.80 | 11150.09 | 3.89 | 140276 |
| 9 | 9716.58 | 10396.10 | 6.99 | 42358 | 9719.39 | 9930.19 | 2.19 | 362900 |

Table 8.13 presents the solutions provided for all cases as well as the number of nodes solved. The solutions are good, however the branch and bound algorithm fails to converge within both timeframes with a varying gap of 2-14 %. This is due to the extensive amount of binary variables caused by the SOS 2 formulations, and correspondingly the numbers of nodes solved are high. Both the upper bounds and the gaps are included in the table. Significant decreases in the global upper bounds are detected after 3 and 12 hours respectively, however only slight improvements in integer solutions are recognized.

### SmartOpt-Bonmin

The problem is modeled according to the SmartOpt concept, but solved with Bonmin with results given in Table 8.14. Starting point 1 is used in the Bonmin testing. Interestingly, Bonmin only branches in one of the cases, while it ends the search after gaining a solution from its diving heuristic in the rest. The diving heuristic gives low quality solutions for the case instances.

**Table 8.14: Result SmartOpt-Bonmin**

| Case | Objective | Time (sec) | NLP Nodes |
|---|---|---|---|
| 1 | 10587.1 | 8 | 0 |
| 2 | 10138.6 | 25 | 47 |
| 3 | 8233.8 | 2 | 0 |
| 4 | 10132.4 | 3 | 0 |
| 5 | 9929.8 | 8 | 0 |
| 6 | 8488.8 | 3 | 0 |
| 7 | 10789.2 | 12 | 0 |
| 8 | 9722.9 | 4 | 0 |
| 9 | 8876.18 | 8 | 0 |

## Black-Box

The Black-Box method is dependent on starting points, the same NLP feasible starting point applied for SmartOpt-Bonmin and SmartOpt-B&B is given to the framework, and the solutions are given in Table 8.15 for all 9 cases.

**Table 8.15: Results Black-Box**

| Case | Objective | Time (sec) | Iterations |
|---|---|---|---|
| 1 | 10070.3 | 252 | 12900 |
| 2 | 9487.6 | 170 | 27819 |
| 3 | 6661.4 | 112 | 28253 |
| 4 | 9954.0 | 144 | 18064 |
| 5 | 9401.0 | 133 | 16225 |
| 6 | 7445.8 | 148 | 17552 |
| 7 | 9963.0 | 232 | 13688 |
| 8 | 9540.8 | 228 | 20721 |
| 9 | 7915.2 | 106 | 30700 |

Table 8.15 shows that in all cases, the solutions provided by the Black-Box method are fairly poor. However, the solution method provides solutions within a short amount of time. Also given in the table is the number of iterations run for each case. The amount of allowed iterations is unlimited, this means that the algorithm converges for all solutions presented in the table.

## Nonlinear Approximations Model

This method uses Bonmin, which is also given start point 1. In Table 8.16 the objective value column gives the value of the objective function when variable and parameter values from the solutions obtained from each of the different methods are plugged into the Black-Box algorithm

(Shamlou and Ursin-Holm, 2012). Running the Black-Box algorithm one iteration in this way, the values of variables remain the same. However, a measure of the feasibility or possibly infeasibility of the solution is indicated by the Black-Box results, specifically by considering the constraint violations. Table 8.16 provides the solution times as well as the number of nodes solved for each solution. The solution times vary greatly, between 24 minutes up to more than 5 hours, depending on the case solved.

**Table 8.16: Result Nonlinear Approximations**

| Case | Objective | Time (sec) | NLP Nodes |
|---|---|---|---|
| 1 | 11220.3 | 10347 | 35846 |
| 2 | 10517.7 | 7038 | 22586 |
| 3 | 9267.5 | 2060 | 15315 |
| 4 | 11226.2 | 11212 | 77276 |
| 5 | 10473.1 | 33511 | 162398 |
| 6 | 9383.7 | 1990 | 13085 |
| 7 | 11221.4 | 3608 | 17729 |
| 8 | 10860.6 | 5957 | 47124 |
| 9 | 9751.3 | 2855 | 12120 |

All solutions provided by NA are estimated infeasible by the Black-Box algorithm breaking several pressure restrictions $p_{ml}^{M} \leq p_{mj}^{W}$ by up to ~5 bar. This is believed to be caused by the nonlinear approximations for well performance and pipeline pressure drops. The reason is two-fold, the solutions end up in areas where parts of the proxy models lies above the simulator data for oil production at certain wellhead pressures and gas lift allocations. At the same time the solutions end up in areas where that part of the proxy models lie below the simulator data for pressure losses over the pipelines for certain flows.

**Solution Quality Comparisons**

The solutions obtained for all case instances by all approaches are given in Table 8.17. They are given as a percentage relative to the solutions obtained by SmartOpt-B&B version 1 for easy comparison. SmartOpt-B&B and MILP provide the best solution interchangeably for the 9 cases. The remaining solution methods are inferior to a varying degree regarding solution quality. The feasibility pump is also included as a standalone solution method. Interestingly, it provides quite good solutions to all cases.

Table 8.17: Normalized Results of All Solution Methods

| Case | SO-B&B | | MILP | | Bonmin | Black-Box | NA | FP |
|---|---|---|---|---|---|---|---|---|
| | V1 | V2 | 3h | 12h | | | | |
| 1 | 100.00 | 100.00 | 99.82 | 99.93 | 95.10 | 90.46 | 100.79 | 97.78 |
| 2 | 100.00 | 100.06 | 100.03 | 100.19 | 96.56 | 90.36 | 100.17 | 99.08 |
| 3 | 100.00 | 100.01 | 99.94 | 100.32 | 89.32 | 72.26 | 100.53 | 98.44 |
| 4 | 100.00 | 100.09 | 99.66 | 99.78 | 90.50 | 88.91 | 100.27 | 98.09 |
| 5 | 100.00 | 100.03 | 100.08 | 100.11 | 94.41 | 89.39 | 99.58 | 99.52 |
| 6 | 100.00 | 100.00 | 100.39 | 100.39 | 92.08 | 80.77 | 101.79 | 98.40 |
| 7 | 100.00 | 100.03 | 99.90 | 99.90 | 96.62 | 89.22 | 100.49 | 97.40 |
| 8 | 100.00 | 99.86 | 100.37 | 100.50 | 91.02 | 89.31 | 101.67 | 98.42 |
| 9 | 100.00 | 100.00 | 100.29 | 100.32 | 91.61 | 81.70 | 100.65 | 98.30 |
| Average | **100.00** | **100.01** | **100.05** | **100.16** | **93.03** | **85.82** | **100.66** | **98.38** |

# 9. Discussion

In this chapter the findings and results of the computational study of the proposed algorithm will be discussed in the context of solution time and solution quality, and further analyzed in terms of robustness and applicability in the work process. Additionally the other considered solution methods are briefly compared to the proposed algorithm with an emphasis on the same issues.

### 9.1.1 Modeling and Implementation

The production allocation problem is a nonlinear non-convex problem. The emphasis has been on formulating the mathematical model in a compact and intuitive manner. In other words, the model is kept nonlinear and non-convex, and the solution method is tailored for the difficulties posed by such problems. In order to get a more convex problem, it is common to apply linearization techniques to remove local optima. If such techniques are applied to the mathematical model used here, it could potentially lead to a more well-behaved problem, resulting in less local solutions to get caught in. However, convexification of the problem is substantially more relevant when it is possible to obtain a completely convex problem. The problem addressed is non-convex mainly due to the well and pipeline data, which is hard to convexify. It is important to emphasize that because of the non-convexities of the problem, the proposed algorithm can only be taken as a heuristic method. This is to say that the solutions obtained are not guaranteed to be globally optimal.

Aspects of the model implementation with potential for improvement are the interpolation scheme in the simulator approximations, and general scaling of problem data. If interfacing is possible, one option is to call the relevant simulators directly in the optimization. These must be supplied by the oil company. Calling the simulator directly might be quite time consuming because such simulators often compute information regarding a whole network production system. Additionally, there is a risk that the simulator might fail for miscellaneous reasons and obstruct the optimization search.

If the simulators are approximated, filtering can be devised to avoid outlier data to impede the optimization. This option might be more robust and significantly quicker than calling the real simulator if attention is paid to these issues when deciding on the approximation. In this thesis linear interpolation with finite differencing is used, which is quick and sufficiently accurate compared to the underlying data. However, this method might cause convergence problems for the algorithm if the simulator data are not smooth (Shamlou and Ursin-Holm, 2012). More sophisticated interpolation and differentiating techniques can improve the overall algorithm performance and make the method more robust.

### 9.1.2 Technical Aspects

The feasibility pump proposed in this thesis for non-convex MINLPs differs in several ways from the general feasibility pump. The linear problem (FP-MILP) is not only approximated by linear outer approximations. In fact the original problem structure is kept, meaning that equations for mass and pressure balances are included and wells and pipelines are each represented by separate linear approximations (tangent planes). Still, the FP-MILP is a small problem with only 50 binary variables. In addition, several features are included in the FP for handling non-convexities. Due to all this, as a standalone solution method the feasibility pump heuristic proves surprisingly good. It is highly efficient for the non-convex problem at hand and provides good integer solutions for all starting points. Since the parametric choices have little effect on the solution quality, extensive analysis is not necessary to set these values. Because of interfacing issues, the recorded FP solution times are quite high, but since the FP-MILP problem is relatively small the time can be brought down considerably with appropriate solvers and code. If it is brought down to a matter of seconds, the FP can be applied in more nodes during the tree search, as opposed to only the root node as it is here.

To sum up, the feasibility pump is quite robust in addition to being generic and not case dependent. It is usable in its current form on most offshore petroleum production fields, and any non-convex MINLP with slight modifications, which increases the applicability of the heuristic and can encourage implementation. In terms of the branch and bound algorithm the FP is only incorporated in the root node. However, the results show that if implementation is possible including it in more nodes throughout the tree could be highly interesting and should be tested.

Only someone with both knowledge about properties and characteristics of a particular production system, typically the production engineer, as well as understanding for optimization will successfully manage to implement and maintain the two problem specific heuristics. Well characteristics change throughout the lifetime of a field, which will necessitate regular updating of the bounding rules of these heuristics. Both are diving heuristics and tend to produce similar solutions after locking some binary variables, so in reality implementing only one is enough. Since H1 only solves one reduced NLP it can be taken as the more efficient one. Although the solutions are all inferior to the FP, H1 and H2 are quicker, which makes them suitable for being tested in multiple nodes throughout the branch and bound algorithm.

The technical testing conducted in regards to the B&B of the algorithm on the base case show some interesting results. The usual tradeoff between solution time and solution quality can be evidenced by the results, although in this particular case large increases in solution time only lead to marginal improvements of the objective function value.

Heuristics are usually incorporated into B&B algorithms to provide lower bounds. When using a best-first strategy, lower bounds only affect the tree size when the gap termination tolerance is above 0, making the inclusion of heuristics relevant. When this tolerance is set to 0.04 % inclusion of heuristic 1 reduces the total number of solved nodes by 2400 when applied to all nodes as shown in test number 8. But, heuristic 1 solves a problem each time it is used, so the total number of solved problems is higher than when no heuristics are applied. Unfortunately this leads to an inferior solution time compared to the best-first strategy alone. Results also show that obtaining a lower bound early in the optimization leads to fewer created nodes as many nodes are not branched on as a consequence.

Obviously the frequency with which heuristics are applied during the tree search is important, and ideally it will lead to fewer nodes solved without increasing the total number of problems solved. So, it seems promising to include H1 in some nodes of the tree to potentially obtain good lower bounds such that the algorithm solves fewer problems. Another point is that having several sources of lower bounds makes the B&B more robust.

The performance of SmartOpt-B&B is dependent on starting points used, just like most other nonlinear (and non-convex) solution methods. 1000 random points were generated and solved with Ipopt for further investigation. These are subsequently clustered, and the starting points leading to the 8 most promising solutions are extracted, 4 coming from the solver reaching maximum iterations and 4 coming from the solver converging. The starting points leading to NLP infeasible root node solutions are included because they might make the algorithm enter areas of the solution space which are not visited if only feasible root node solutions and corresponding starting points were considered.

The graph in Figure 8.2 is quite flat, considering that the y-axis only ranges from 11500 to 12300. Other runs of the code with 1000 generated solutions showed similar characteristics. Most solutions, about 90 %, obtain objective function values that are located in the central band, ranging between 12000 and 12100 for the base case. Note that all points to the left of the first NLP feasible solution (12069.7) have reached maximum iterations in Ipopt and are NLP infeasible.

Because of the high chance of obtaining a good NLP feasible solution when generating a random starting point, it seems unnecessary to produce 1000 such points. Only producing e.g. 20 and subsequently solving these will produce at least one NLP feasible solution in the central band with a very high probability. Subsequently choosing the starting point among the 20 that lead to the best root node solution seems promising.

The inclusion of several starting points helps search a bigger area of the feasible region and increases the chance of finding the global optimal solution, but this comes at the cost of solution time. Results show very large trees and consequently long solution times, with minimal increase in oil productions. The effect of including several starting points for the base case is very small, which seems natural considering the low spread in resulting root node objective function values. But, the measure is well suited for parallelization. The results also show that using starting points that obtain NLP feasible root node solutions might be preferable. This leads to smaller trees and lower solution times.

The effect of the two measures targeting a negative gap seems to be low in the tests conducted, at least compared to the large increase in solution time that they require. To sum up, all non-convexity measures only give marginal improvement of the objective function values, with a relatively high computational cost. When analyzing different optimal solutions and their characteristics, the results show that the objective functions have similar values regardless of routing. This shows that many local optima exist within a tight interval of objective function values. This might be the reason why the non-convexity measures have so little effect on solution quality.

The two versions of the SmartOpt-B&B algorithm perform quite well considering objective function values and solution quality on the additional cases. The difference in produced oil is marginal between the two versions. Even though version 2 is much more sophisticated in including several non-convexity measures, the effect on the end result has been minimal in this particular oil production network.

### 9.1.3 Comparing Solution Methods

In the subsequent discussions, the advantages and disadvantages of solving the production allocation problem with the SmartOpt-B&B version 1 in comparison to the methods developed in preparatory work are covered (Shamlou and Ursin-Holm, 2012). Attention should be paid to the issue of comparing models and methods that are fundamentally different. The SmartOpt and Black-Box methods are based on including simulators directly and use the same data and interpolation code, while the NA technique and the MILP are based on approximations.

SmartOpt-B&B version 1 and the MILP (3h) approach prove superior, when taking both solution quality and time into consideration. In half of the cases the best solution originates from the MILP, while the other half's optimal solutions are provided by SmarOpt-B&B. However, the difference is never more than 0.52%. The results show that the MILP always provides good quality solutions given its linear approximations in the designated time of 3 hours, making it a robust solution method. SmartOpt-B&B version 1 also provides good solutions within 3 hours, although it had problems with one of the case instances. This is evidenced by case 8 for version 1 of SmartOpt-B&B which converges after more than 30 hours of runtime.

An important aspect in assessing these methods is their response to higher resolution well and pipeline approximations, which can increase the accuracy of the model and obtained solutions. The MILP is very sensitive to an increase in number of data points, because of piecewise linearization. Its solution time and solution accuracy is correlated to the number of data points included. SmartOpt-B&B and the other methods are not particularly sensitive to higher resolution tables.

The MILP is a very straightforward solution method, requiring minimal knowledge and analysis of the problem specifics and data. The method has few choices and as such is easy to implement. The proposed SmartOpt-B&B algorithm is a special purpose solution method tailored for non-convex petroleum production problems. It can be applied to most offshore production systems with the expectation of attaining similar results as obtained for the problem instances in this work.

Smartopt-Bonmin finds solutions which are on average 93% of the SmartOpt-B&B V1 results. Surprisingly, Bonmin does not even branch when solving many of the cases but outputs results obtained from its diving heuristic. This seems to have a negative effect on solution quality, but is very quick. Generally, Bonmin cannot be taken as a particularly robust method for the non-convex problem at hand, because of the unpredictability as to when it will branch versus only using its heuristic. The results might be more uniform if linearization and convexification techniques are applied to the modeling. Since there are a large number of options and choices available when using Bonmin, the user must know both the solver and the production problem properly in order to apply it to similar petroleum production problems.

The two remaining solution methods, namely the Nonlinear Approximation approach and the Black-Box method, provide inferior solutions in terms of quality. Previous work indicates that the solutions obtained from this method may be infeasible, therefore the NA solutions are subsequently run through the Black-Box model. This reveals clearly infeasible solutions, and the results given in Table 8.16 on average exceed the SmartOpt-B&B. Solutions are mainly provided within 3 hours, although case 5 showed a large increase in solution time. This method is asset specific, and as such all nonlinear approximations must be created for each problem specifically, and updated in the same way when well performances change during the life of the producing field. This makes the initial preparations and the updating of the optimization models relatively time consuming, and requires the user to have detailed problem knowledge.

The Black-Box approach is different from other mentioned solution methods, in that the optimization algorithm is derivative-free and the whole production network is modeled as one black box. It is relatively fast, although it provides significantly lower oil production rates compared to the SmartOpt-B&B. Despite the low quality solutions, the Black-Box method is extensively used in the industry because it is user friendly and easy to understand.

### 9.1.4 Work Process

All algorithms covered, except the Black-Box and FP as a standalone solution method have solution times that are unsuitable for active use during the production planning meetings. They can be used as a decision support tool by using a previously obtained solution for discussions, but re-optimization and testing will not be possible.

Algorithms that provide solutions within a few minutes are desirable if optimization is to be used in the meetings between various on- and offshore operators and engineers. This will inevitably be at the cost of solution quality. Figure 8.3 shows that the FP and the problem specific heuristics included in the B&B tree provide very good feasible solutions quickly. Additionally, the problem specific heuristics provide even better solutions than the FP after about 200 nodes. Still, the FP developed is considered superior to H1 and H2 because of its generic nature. It is the best alternative among the standalone methods with relatively low solution times. The solutions were on average about 98% of the ones obtained by the SmartOpt-B&B, encouraging the use of the FP as a standalone method and not only as a part of a larger algorithm.

The offshore operator prefers not to implement changes suggested by the production engineer, if the suggestion diverges greatly from the current operation. This is mainly to avoid big changes in the production system, which can lead to transient behavior and impair the quality of the separation process. Analysis of the solution characteristics indicate that different routing variable values can result in similar oil production rates. In such cases it can be preferable to choose the solution that is closer to the current operating point, than one with a slightly higher oil rate but which requires extensive re-routing. This will encourage the offshore operator to apply the suggested solution. It can be implemented in the model by including appropriate constraints. Additionally, the algorithm should have the current operating plan as starting point to realistically portray the problem at hand.

# 10. Conclusion

In view of the preceding results obtained from the computational study and associated discussions, some concluding remarks will be given here.

The short-term production optimization problem is a simulation based non-convex MINLP problem, necessitating solution methods tailored to handle the difficulties posed by such a problem. One problem is the non-convexity, making it difficult to identify local solutions versus global ones. Another challenging aspect is the existence of nonlinearities and binary variables, which each require special attention. With these adversities in mind, a special purpose B&B algorithm fit for manipulation is developed specifically for optimizing a generic short-term oil production problem. This is done to allow sophisticated operations on each node of the B&B tree, and the method is called SmartOpt-B&B.

In addition to using a standard B&B framework as the basis for the algorithm, two types of heuristics are included to help obtain good feasible solutions and thus lower bounds during the tree search. The generic FP heuristic is modified to fit non-convex production allocation problems, and creates very good feasible solutions when applied to the root node of the B&B tree. Two problem specific heuristics are also created based on problem case knowledge and data analysis. These are very quick in giving feasible results, although they are often inferior to the FP in terms of solution quality.

Specific non-convexity measures are included to avoid eliminating interesting parts of the solution space and help push towards finding the global solution. Pruning poses a challenge when solving non-convex problems with B&B, therefore negative gaps are allowed in order to assess the impact on the solution found. Further, the influence of including several starting points is studied.

Putting all these elements together, the technical testing gives way to two different designs of the algorithm with different emphasis, the first one on solution time, and the second more sophisticated one on solution quality. Several case instances of the problem are subsequently solved to assess solution time, solution quality, robustness and applicability of the algorithm. The results show quite a dramatic increase in solution time for the sophisticated version, which includs multiple starting points and negative gaps. Unfortunately this increase in solution time is not accompanied by a justifiable increase in solution quality, which leads to the conclusion that using the more sophisticated and time consuming version of the algorithm might not be necessary. Of course the solutions are closer to the global optimum, but sometimes having a nearly optimal solution is superior to having the optimal one if it is obtained in a fraction of the time.

SmartOpt-B&B version 1 proves satisfactory compared to other existing methods for optimizing the production allocation problem. It provides good quality solutions within a timeframe of about 3 hours, and can be used if longer solution times are acceptable in order to gain higher oil production rates. Solution time is important if optimization is to be applied in production planning meetings between the engineers and operators. SmartOpt-B&B includes a feasibility pump heuristic, which can also function as a standalone solution method. This approach is recommended if solution time is of great importance, as it can potentially provide near optimal solutions in less than a minute.

# 11. Bibliography


Almklov, P. and Gunnerud, V. (2013) Dailiy Production Optimization, 1st edition, White Paper, Trondheim: Center for Integrated Operations in the Petroleum Industry.

Alpha Thames Ltd (2004) Dictionary of Oil Industry Terminology, [Online], Available: http://www.scribd.com/doc/19552139/Dictionary-of-Oil-Industry-Terminology [22 May 2013].

AMPL Optimization LLC (2012) AMPL Website, [Online], Available: http://www.ampl.com/ [15 Dec 2012].

April, J., Glover, F., Kelly, J.P. and Laguna, M. (2003) 'Practical Introduction to Simulation Optimization', Proceedings of the 2003 Winter Simulation Conference, Boulder, Colorado, pp. 71-78.

Bertaccoa, L., Fischettic, M. and Lodi, A. (2007) 'A feasibility pump heuristic for general mixed-integer problems', Discrete Optimization, vol. 4, no. 1, pp. 63-76.

Berthold, T. (2006) Primal Heuristics for Mixed Integer Programs, Working Paper. Berlin: Zuse Institute Berlin.

Bodington, C.E. and Baker, T.E. (1990) 'A History of Mathematical Programming in the Petroleum Industry', Interfaces, vol. 20, no. 4, pp. 117-127.

Bonami, P., Cornuéjols, G., Lodi, A. and Margot, F. (2009) 'A Feasibility Pump for mixed integer nonlinear programs', Mathematical Programming, vol. 19, no. 2, pp. 331-352.

Bonami, P., Kilinç, M. and Linderoth, J. (2012) 'Algorithms and software for convex mixed integer nonlinear programs', in Lee, J. and Leyffer, S. Mixed Integer Nonlinear Programming, 17th edition, New York: Springer.

Bonami, P. and Lee, J. (2007) BONMIN Users' Manual, [Online], Available: https://projects.coin-or.org/Bonmin/browser/stable/0.1/Bonmin/doc/BONMIN_UsersManual.pdf?format=raw [22 May 2013].

Bonami, P., Lee, J., Leyffer, S. and Wächter, A. (2011) Eprints for the Optimization Community, [Online], Available: http://www.optimization-online.org/DB_HTML/2011/09/3191.html [22 May 2012].

Burer, S. and Letchford, A.N. (2012) 'Non-Convex Mixed-Integer Nonlinear Programming: A Survey', Surveys in Operations Research and Management Science, vol. 17, no. 2, pp. 97-106.

Camponogara, E. and Nakashima, P. (2006) 'Optimizing gas-lift production of oil wells: piecewise linear formulation and computational analysis', IIE Transactions, vol. 38, no. 2, pp. 173-182.

COIN-OR, Bonmin (2012) BONMIN Website, [Online], Available: https://projects.coin-or.org/Bonmin [15 Dec 2012].

COIN-OR, Ipopt (2012) Ipopt Website, [Online], Available: https://projects.coin-or.org/Ipopt [15 Dec 2012].

Conn, A.R., Scheinberg, K. and Vicente, L.N. (2009) Introduction to Derivative-Free Optimization. MPS/SIAM Book Series on Optimization., 1st edition, Philadelphia: SIAM.

D'Ambrosio, C., Frangioni, A., Liberti, L. and Lodi, A. (2012) 'A storm of feasibility pumps for nonconvex MINLP', Mathematical Programming, vol. 136, no. 2, pp. 375-402.

Díez, M.D., Brusdal, K., Evensen, G., Barkve, T. and Mjaavatten, A. (2005) 'Opportunities and challenges of using sequential quadratic programming (SQP) for optimization of petroleum production networks', European Symposium on Computer Aided Process Engineering - 15, Barcelona, pp. 169-174.

Digabel, S.L. (2011) 'NOMAD: Nonlinear Optimization with the MADS algorithm', ACM Transactions on Mathematical Software (TOMS), vol. 37, no. 4, pp. 44:1-44:15.

Dzubur, L. and Langvik, A. (2012) Optimization of Oil Production - Applied to the Marlim Field, Working Paper. Trondheim: Norwegian University of Science and Technology.

Foss, B., Gunnerud, V. and Díez, M.D. (2009) 'Lagrangian Decomposition of Oil-Production Optimization Applied to the Troll West Oil Rim', SPE Journal, vol. 14, no. 4, pp. 646-652.

Fu, M.C., Glover, F.W. and April, J. (2005) 'Simulation Optimization: A Review, New Developments, and Applications', Simulation Conference, 2005 Proceedings of the Winter, Boulder, Orlando, pp. 83-95.

Glæserud, P. and Syrdalen, J.A. (2009) Optimization of Petroleum Production under Uncertainty - Applied to the Troll C Field, Working Paper. Trondheim: Norwegian University of Science and Technology.

Grossmann, I.E. (2002) 'Review of nonlinear mixed-integer and disjunctive techniques', Optimization and Engineering, vol. 3, no. 3, pp. 227-252.

Grötschel, M. and Padberg, M.W. (1979) 'On the symmetric travelling salesman problem', Mathematical Programming, vol. 16, no. 1, pp. 265-280.

Gunnerud, V., Conn, A. and Foss, B. (2013) 'Embedding Structural Information in Simulation-based Optimization', To appear in Computer and Chemical Engineering, vol. 53, no. 1, pp. 35-43.

Gunnerud, V. and Foss, B. (2010) 'Oil production optimization—A piecewise linear model, solved with two decomposition strategies', Computers and Chemical Engineering, vol. 34, no. 11, pp. 1803-1812.

Gunnerud, V., Foss, B., McKinnon, K. and Nygreen, B. (2011) 'Oil production optimization solved by piecewise linearization in a Branch and Price Framework', Computers and Operations Research, vol. 39, no. 11, pp. 2469-2477.

Kosimidis, V.D., Perkins, J.D. and Pistikopoulos, E.N. (2005) 'A mixed integer optimization formulation for the well scheduling problem on petroleum fields', Computers & Chemical Engineering, vol. 29, no. 7, p. 1523–1541.

Land, A.H. and Doig, A.G. (1960) 'An automatic method for solving discrete programming problems', Econometrica, vol. 28, no. 3, pp. 497-520.

Law, A.M. and Kelton, W.D. (1991) Simulation Modeling and Analysis, 2nd edition, New York: McGraw-Hill Inc.

Leyffer, S., Linderoth, J., Luedtke, J., Miller, A. and Munson, T. (2009) 'Applications and algorithms for mixed integer nonlinear programming', Journal of Physics: Conference Series, vol. 180, no. 1, pp. 012014:1-5.

Lundgren, J., Värbrand, P. and Rönnqvist, M. (2010) Optimization, 1st edition, Lund: Studentlitteratur AB.

Nocedal, J. and Wright, S.J. (2006) Numerical Optimization, 1st edition, New York: Springer.

Offshore Technology (2012) Offshore Technology, [Online], Available: http://www.offshore-technology.com/projects/marlimpetro/ [17 April 2013].

Shamlou, S. (2012) Production Optimization, Working Paper. Trondheim: Center for Integrated Operations (IO-Center).

Shamlou, S., Gunnerud, V. and Conn, A.R. (2012) 'Optimization of petroleum production networks – Through proxy models and structural constraints', 8th IFAC International Symposium on Advanced Control of Chemical Processes, Advanced Control of Chemical Processes, Singapore, pp. 543-548.

Shamlou, S. and Ursin-Holm, S. (2012) Comparative study of optimization solution techniques, Working Paper. Trondheim: Norwegian University of Science and Technology.

Sharma, S. (2012) Heuristics and general purpose algorithms for mixed-integer nonlinear optimization, Working Paper. Trondheim: Norwegian University of Science and Technology.

Storvold, V. (2012) Optimization of Investment decisions and production planning in aging offshore petroleum fields, Working Paper. Trondheim: Norwegian University of Sceince and Technology.

Torgnes, E., Gunnerud, V., Hagem, E., Rönnqvist, M. and Foss, B. (2012) 'Parallel Dantzig–Wolfe decomposition of petroleum production allocation problems', Journal of the Operational Research Society, vol. 63, no. 7, pp. 950-968.

Ursin-Holm, S. (2012) Production Optimization Marlim P-35, Working Paper. Trondheim: Center for Integrated Operations (IO-Center).

Wang, P. (2003) Development and applications of production optimization techniques for petroleum fields, Working Paper. Stanford: Stanford University.

Wikipedia (2012) Bilinear Interpolation, [Online], Available: http://en.wikipedia.org/wiki/Bilinear_interpolation [15 Dec 2012].

Wikipedia (2012) Trilinear Interpolation, [Online], Available: http://en.wikipedia.org/wiki/Trilinear_interpolation [12 Oct 2012].

Wächter, A. (2009) Short Tutorial: Getting Started With Ipopt in 90 Minutes, [Online], Available: http://drops.dagstuhl.de/volltexte/2009/2089/pdf/09061.WaechterAndreas.Paper.2089.pdf [22 May 2013].

# 12. Appendix

**Electronical Attachments**

The electronical folder attached contains code and data related to:

1. SmartOpt-B&B implementation
2. MILP-Xpress model
3. SmartOpt-Bonmin implementation
4. Nonlinear Approximation AMPL model
5. Black-Box framework