NTNU – Trondheim
Norwegian University of
Science and Technology

Mixed-Integer Nonlinear Programming Heuristics Applied to a Shale Gas Production Optimization Problem

Shaurya Sharma

Master of Science in Engineering Cybernetics
Submission date: June 2013
Supervisor: Bjarne Anton Foss, ITK
Co-supervisor: Brage Rugstad Knudsen, ITK
Bjarne Grimstad, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Problem Description

The use of heuristic algorithms can for some complex mixed integer problems be a viable solution alternative when (exact) general purpose algorithms are too computationally demanding. These algorithms generally constitute a compromise between the time spent on quickly finding an integer feasible solution and the quality of the solution found with respect to the primal objective. Heuristics are used both as standalone algorithms and included in enumerative schemes such as branch-and-cut methods to construct feasible solutions or to improve incumbent solutions. Both RTO and MPC applications put requirements on the solution times of the associated optimization problems. When systems include integer decisions such as switchings, routing of mass flows or other types of logics (i.e. hybrid systems) and in addition are described by nonlinear dynamics, the respective optimization problems may become computationally prohibitive. In the worst case, the chosen algorithm may fail to find a feasible solution within the given sampling time. Many RTO problems involve finding set points, control objectives or routing decisions for distributed or network systems, typically leading to some sort of block structure in the optimization problems.

Based on the project report by the same candidate, where a computational study indicated that some heuristics may perform particularly well on problems with block separable structures, the master project should include the following tasks:

Task description:

1. Analyze and perform a computational study of using construction heuristics in the root node of branch and bound tree for convex MINLPs. Solve the problems to optimality. Consider the same for nonconvex MINLPs.
2. Explore and evaluate the implementation of an objective feasibility pump for convex MINLPs. Asses the applicability of such a method on nonconvex MINLPs.
3. Given a complex, possibly nonconvex, dynamic MINLP for an RTO application: Evaluate how the use of heuristics, in particular the standard and the objective feasibility pump, can contribute/assist with respect to solution time and solution quality in solving these types of problems.
4. Derive and evaluate different reformulations and relaxations of the original MINLP.
5. Perform a computational study on a large set of different and perturbed problems instances to gain statistical analysis of using heuristics on these problems. Consider how different known construction and improvement heuristics can be combined and tailored for solving the given network problems, also in the context of an enumerative (branching) scheme.

The thesis report may include a draft paper to a selected conference with the main

results of this work.

The master project should use the project report "Heuristics and general purpose algorithms for mixed-integer nonlinear optimization" by the same candidate as background material.

Starting date: 20.01.2013

End date: 16.06.2013

Co-supervisor: Brage Rugstad Knudsen, PhD student Bjarne Grimstad, PhD student

Trondheim, 18.01.2013

Bjarne Foss Professor/supervisor

Abstract

Mixed-Integer nonlinear programs(MINLPs) are a general class of nonlinear optimization problems that have a wide array of real-world applications. These problems are in general notoriously difficult to solve, and it is therefore of great interest to develop heuristics that can aid the solution process. This thesis contains two major parts.

In the first part, an *objective feasibility pump*, a heuristic for finding high quality feasible solutions of MINLPs is developed and implemented in the open source C++ project `BONMIN`. A computational study of this heuristic revealed that it is generally not more effective compared to other heuristics, but that it can be tailored to specific problems to yield improvements over other heuristics with regards to objective value.

In the second part, extensions of a complex, dynamic shale gas production optimization problem are described and a simple heuristic for this problem is developed. A set of test problems is used to perform a benchmark study of the impact of using heuristics on this problem. The results of this study revealed that the new heuristics outperform other currently available heuristics and can find good feasible solutions in a fraction of the CPU time required by the default branch-and-bound solver in `BONMIN`.

Sammendrag

Blandet kontinuerlig- heltall ulineære optimeringsproblemer(MINLPs) er en generell klasse av ulineære optimeringsproblemer som kan anvendes på et bredt spekter av praktiske problemer. Denne typen problemer er fundamentalt vanskelige å løse, og det er derfor av stor interesse å utvikle *heuristikker* som kan bistå i løsningsprosessen. Denne avhandlingen er delt opp i to større deler.

I den første delen beskrives utviklingen og implementasjonen av en heuristikk kalt *objective feasibility pump*. Dette er en heuristikk utviklet med tanke på å finne gode løsninger p å kort løsningstid, si konstrast til andre heuristikker som gjerne kun søker etter gyldige løsninger. Heuristikken er implementert i programmeringsspråket C++ innenfor open-source prosjektet som heter `BONMIN`. En studie av å bruke denne heuristikken viste at den generelt ikke er effektiv, men at den kan tilpasses til å finne bedre løsninger enn andre tilgjengelige heuristikker.

Den andre delen beskriver formuleringen og videreutviklingen av et optimeringsproblem, innen skifergassproduksjon, sammen med en heuristikk som er utviklet spesielt for dette problemet. Et sett av testproblemer blir brukt for å gjennomføre en studie av å bruke heuristikker på denne typen problemer. Resultatet av denne studien viste at de nye heuristikkene finner svært gode løsninger ved bruk av en brøkdel av CPU-tiden som kreves av branch-and-bound-løseren i `BONMIN`.

Preface

This master thesis is written during the final semester of the 5 year MSc. program in Engineering Cybernetics at the Norwegian University of Science and Technology.

Over the course of this past semester, I have learned a great deal about mixed-integer nonlinear programs and have been able to apply novel optimization techniques to a shale gas production optimization problem. The work has been challenging and equally rewarding.

I feel extremely fortunate to have been given the opportunity to work with PhD. candidate Brage Knudsen, who in this past year has provided invaluable guidance and assistance. Further, I would like to thank Professor Bjarne Foss for supervising this project and providing feedback, and encouraging me to attend a course in Belgium. I am also very grateful to PhD. candidate Bjarne Grimstad for the help he provided with the project work in the fall. Lastly, I would like to thank Xiaohua, without whom, this thesis would have been unreadable.

Shaurya Sharma
June 2013

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter the main topics for this thesis are introduced. Parts of the content of this chapter are from [76].

1.1 Mixed-Integer Nonlinear Programs

Many real-world optimization problems inherently possess a structure which can best be modelled by a combination of discrete and continuous variables. Consider the problem of planning oilfields [79]: the decision of which oilfield to develop can be modelled as a discrete variable, while the physical properties of an oilfield such as pressures or flows are best represented by continuous variables. For this problem, while the objective is to maximize the *net present value*(NPV) which is a linear function of the generated revenues subtracted by the maintenance and production costs, the equations which describe the dynamics of the flows and pressures in the pipelines may be nonlinear. From this simplifed example of the oilfeld planning problem, it is clear that an optimization framework which can encapture nonlinearities and integrality conditions of a problem-a mixed integer nonlinear program, is needed

Mixed-Integer nonlinear programs are a particularly challenging problem as they combine the difficulties introduced by nonlinear, and possibly nonconvex, functions with the combinatorial nature of discrete variables. Initially the approach to solving MINLPs was to approximate the problem by formulating a *mixed integer linear program*(MILP) by linearizing the problem [5]. The main issue with this approach, depending on the linearization technique,

is that an optimal solution to the linearized problem may be suboptimal or even infeasible for the original problem [5].

As a generalization of MILPs, MINLPs belong to the class of $\mathcal{NP}$-hard problems [43]. A consequence of this property which has dire implications on the tractability of solving these problems is that in the worst case scenario, the computation time required to find the optimal solution increases exponentially with the size of the problem. Hence, for large problems it becomes increasingly difficult for MINLP algorithms to converge to the optimal solution or even a feasible solution. Under such circumstances, resorting to solving for a high quality suboptimal solution using a *heuristic* may be the only viable option.

1.2 Heuristics

The word heuristic is derived from the greek word $\varepsilon\upsilon\rho\iota\sigma\kappa\varepsilon\iota\nu$, which means 'to find' or 'to discover'. In the context of computer science, the word is used to describe techniques or methods that find approximate solutions to computationally demanding problems such as the *travelling salesman problem* [60] or the *knapsack problem* [73]. These techniques or methods are typically developed through experiential learning, but some of them may be rigorously proven to have beneficial properties that can aid the solution process. Several heuristics that were originally used for solving difficult problems in computer science have since become well established for MILPs [13].

One of the most well known heuristics which was initially developed for finding feasible solutions of difficult MILPs, is the *feasibility pump*(FP) proposed in [36]. This heuristic has since been extended to convex MINLPs [20, 21]. Since its initial proposal, the FP heuristic has spurred a significant amount of research effort and is a topic in the following articles [3,6,12,17,18,28,32,37]. It has also been integrated into a large number of modern MILP solvers [62] and well known MINLP solvers such as `BONMIN` [19] and `MINOTAUR` [64]. Although the original FP for MILPs has been shown to be able to find feasible solutions in a relatively short amount of CPU time, it often comes at the expense of finding solutions with a poor solution quality with respect to the original objective [12]. A similar result is also observed for the FP when applied to MINLPs [21]. In order to improve the objective value found by the FP heuristic, a modification of the original FP for MILPs called the *objective feasibility pump*(OFP) was proposed in [3]. The OFP was shown to yield significant improvements in the objective value with only a minor increase

in the required CPU time. This result raised the question of whether it was possible to extend this approach to MINLPs and achieve similar results. An objective feasibility pump for MINLPs is one of the original contributions of this thesis.

A recently proposed heuristic for finding feasible solutions to MINLPs called *undercover*(UC) [14] has shown promising results. Although this heuristic does not have the same kind of extensive background as FP, it has garnered attention for the variety of techniques it employs and the originality in its approach. A simplified modification of this heuristic is developed and applied to the shale gas production optimization problem together with the OFP in this thesis.

1.3 Shale Gas Production Optimization

The development of the technology for extracting gas from shale resources through the use of *hydraulic fracturing*, also called *fracking*, has over the past 5 years redefined the energy landscape of the U.S [1], and has recently been considered to hold a significant potential for certain countries in Europe [2]. Due to a steady increase in the demand for natural gas in the North American market, there is a large interest in maximizing the production from operational shale gas wells. However, there are several challenges associated with the production which motivate the use of a *decision support system* to determine both short, mid -and long-term planning of the production.

The problem of maximizing the NPV of the production, which is a function of the volume of gas produced over a given period of time, while taking into account the physical properties related to shale gas wells and gas compression, can be formulated as a MINLP. In a realistic model, this MINLP has a large number of variables and constraints, thus algorithms may fail to converge to an optimal solution and use a significant amount of CPU time to find a feasible solution. If there is a hard requirement on the solution time, then this may reduce the value of using this model within a decision support system and raise doubts on the practicality of such a system. The inefficacy of these algorithms motivates the use of heuristic methods for finding high quality feasible solutions. Therefore, in this thesis the modified undercover heuristic together with the OFP for MINLPs will be applied to a large number of perturbed test problems of the shale gas production optimization problem to study the impact of using heuristics as an alternative to currently available methods.

1.4 Scope

The scope of this thesis is to develop an objective feasibility pump for MINLPs, present modifications to the shale gas production optimization problem and to develop heuristics for this problem. The effectiveness of using these heuristics will be assessed by performing a computational study on a set of test problems. The thesis is in a sense separated into two parts: The first part deals with general MINLPs and the objective feasibility pump and the second part is devoted to the shale gas production optimization problem.

1.5 Outline

This thesis is presented as follows:

- Chapter 2 reviews fundamental concepts in optimization theory and MINLPs
- Chapter 3 presents the FP heuristic for MINLPs and OFP for MILPs, and then describes the new OFP for MINLPs and includes the results and discussion on the impact of using this method
- Chapter 4 describes the shale gas production optimization problem and introduces additional modifications to the model
- Chapter 5 presents the undercover heuristic and its modification tailored to the shale gas problem from Chapter 4
- Chapter 6 summarizes the implementations of the production optimization problems which serve as a basis for the computational study reported in Chapter 7
- Chapter 7 presents the results of the computational study
- Chapter 8 consists of a discussion of the results from the computational study in Chapter 7
- Chapter 9 presents the conclusion of the results of the thesis
- Chapter 10 summarizes suggestions for future work

Chapter 2

Preliminaries

In this chapter the relevant definitions and background materials are presented.

2.1 Convex Optimization

2.1.1 Convex Sets and Convex Functions

A set $\mathcal{C}$ is convex if all the points on a line segment between any two points x_1, x_2 in $\mathcal{C}$ are contained in the set $\mathcal{C}$, i.e.

$$\theta x_1 + (1-\theta)x_2 \in \mathcal{C}, \quad \text{for any } \theta \in [0,1].$$

A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if the domain of f, denoted as $\mathbf{dom}\, f$ is a convex set and if for all $x_1, x_2 \in \mathbf{dom}\, f$ and $\theta \in [0,1]$, the following inequality is valid

$$f(\theta x_1 + (1-\theta)x_2) \leq \theta f(x_1) + (1-\theta) f(x_2).$$

2.1.2 Constrained Optimization

The standard form of a constrained optimization problem can be expressed as

$$\begin{aligned} z = \min_x \quad & f(x) \\ \text{s.t.} \quad & h_i(x) = 0, \quad i = 1, \ldots, m \\ & g_i(x) \leq 0, \quad i = 1, \ldots, p, \end{aligned} \tag{2.1.1}$$

where $x \in \mathbb{R}^n$ and $f, h_i, g_i : \mathbb{R}^n \to \mathbb{R}$. The goal of solving an optimization problem is to minimize (or maximize) the *objective function* f over the *decision variables* x, such that the *equality constraints* $h_i(x) = 0$ and *inequality constraints* $g_i(x) \leq 0$ are satisfied. An inequality constraint is said to be *active* at a point x^* if $g_i(x^*) = 0$, and *inactive* at the point x^* if $g_i(x^*) < 0$.

The domain $\mathcal{C}$ which contains the set of points where the objective function and all the constraint functions are satisfied is defined as

$$\mathcal{C} = \bigcap_{i=1}^{m} \mathbf{dom}\ h_i \cap \bigcap_{i=1}^{p} \mathbf{dom}\ g_i \cap \mathbf{dom} f.$$

A point $x \in \mathcal{C}$ is *feasible* if the constraints are satisfied at this point. The problem (2.1.1) is said to be feasible if there exists at least one feasible point, and *infeasible* otherwise. The set of all feasible points is called the *feasible set*.

The *optimal value* to the problem (2.1.1) is defined as

$$z = f(x^*) = \inf \{ f(x) \mid h_i(x) = 0,\ i = 1, \ldots, m,\ g_i(x) \leq 0,\ i = 1, \ldots, p \},$$

where x^* denotes the *optimal solution* corresponding to the optimal value of the function f. If the problem is infeasible then the optimal value is by convention defined to be $f(x^*) = \infty$. A necessary condition to prove that the solution x^* of a constrained optimization problem is a local minimizer, is that the *KKT-conditions* are satisfied [23].

For the problem (2.1.1) to be a *convex optimization problem* it is required that these three conditions are true:

1. The objective function is convex
2. The inequality constraint functions are convex
3. The equality constraint functions are affine

If one or more of these conditions are not true then the problem (2.1.1) is *nonconvex*. To summarize, in a convex optimization problem a convex function is minimized over a convex set.

2.1.3 Global and Local Optimal Solutions

A solution x is a local optimal solution to an optimization problem if x is feasible and

$$f(x) = \inf \{ f(u) \mid u \text{ feasible}, \|u - x\|_2 \leq R \},$$

for some $R > 0$.

A solution x^* is a global optimal solution to an optimization problem if x^* is feasible and

$$f(x^*) \leq f(x), \text{ for all feasible } x.$$

An important property of a convex optimization problem is that any locally optimal solution is also a globally optimal solution to the problem. Nonconvex problems may have several locally optimal solutions.

2.2 Mixed-Integer Nonlinear Programs

A mixed-integer nonlinear program is a general form of a nonlinear optimization problem in which some of the variables are restricted to take on integer values, and can be defined as

$$\begin{aligned} z = \min_{x,y} \quad & f(x,y) \\ \text{s.t.} \quad & g(x,y) \leq 0, \\ & x \in X, \\ & y \in Y \cap \mathbb{Z}^q, \end{aligned}$$

where $f : \mathbb{R}^n \times \mathbb{R}^q \to \mathbb{R}$ and $g : \mathbb{R}^n \times \mathbb{R}^q \to \mathbb{R}^p$ are smooth functions and the sets X and Y define bounded polyhedrons. An integrality constraint $y \in \mathbb{Z} \cap [y^L, y^U]$ can be expressed through a vector $w \in \{0,1\}^N$ of binary variables by the following formula:

$$y = y^L + w_1 + 2w_2 + \ldots + 2^{N-1} w_N,$$

where N is the minimum number of binary variables that is required to represent the upper and lower bounds. The value of N is given by

$$N = 1 + \text{trunc}\left(\frac{y^U - y^L}{\log 2}\right),$$

where the trunc function truncates its real argument to an integer value. This transformation is only efficient if N is not too large. Without loss of generality, the following definition of a MINLP will be used

$$\begin{aligned} z = \min_{x,y} \quad & f(x,y) \\ \text{s.t.} \quad & g(x,y) \leq 0, \\ & x \in X, \\ & y \in \{0,1\}^q . \end{aligned} \qquad (\mathrm{P}_{\text{MINLP}})$$

With a slight abuse of terminology, a MINLP is said to be convex if the *nonlinear program*(NLP) corresponding to the continuous relaxation

$$\begin{aligned} z = \min_{x,y} \quad & f(x,y) \\ \text{s.t.} \quad & g(x,y) \leq 0, \\ & x \in X, \\ & y \in [0,1]^q, \end{aligned} \tag{$\mathrm{P}_{\mathrm{NLP}}$}$$

is convex. An important property of convex MINLPs is that algorithms can efficiently solve for the globally optimal solution of their continuous relaxations [23]. For a comprehensive treatment of the theoretical aspects and algorithms for NLPs the reader is referred to [15, 63].

A MINLP is said to be nonconvex if the NLP obtained from the continuous relaxation of ($\mathrm{P}_{\mathrm{MINLP}}$) is nonconvex. One of the difficulties associated with nonconvex MINLPs is that finding the globally optimal solution of the relaxed problem may be demanding.

If the functions f and g are linear, then the problem ($\mathrm{P}_{\mathrm{MINLP}}$) is a *mixed-integer linear program*(MILP)

$$\begin{aligned} z = \min_{x,y} \quad & c^T x + d^T y \\ \text{s.t.} \quad & Ax + By \leq b, \\ & x \in \mathbb{R}^n,\ y \in \{0,1\}^q, \end{aligned} \tag{$\mathrm{P}_{\mathrm{MILP}}$}$$

where $c \in \mathbb{R}^n$, $d \in \mathbb{R}^q$, $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{m \times q}$ and $b \in \mathbb{R}^q$. The relaxation of the integrality constraint in ($\mathrm{P}_{\mathrm{MILP}}$) yields a *linear program*, which is a convex optimization problem.

Definition 1. *A point* $(\bar{x}, \bar{y})$ *is said to be constraint feasible for the problem* ($\mathrm{P}_{\mathrm{MINLP}}$) *if* $g(\bar{x}, \bar{y}) \leq 0$, $\bar{x} \in X$ *and* $\bar{y} \in [0,1]$

Definition 2. *A point* $(\hat{x}, \hat{y})$ *is said to be integer feasible for the problem* ($\mathrm{P}_{\mathrm{MINLP}}$) *if* $\hat{y} \in \{0,1\}^q$

2.2.1 Algorithms

There are in general two different types of *deterministic* algorithms that can solve convex MINLPs: *multi-tree methods* and *single-tree methods* [11]. These two classes of methods are distinguished by the approach they use

to find the optimal solution. The *Nonlinear Branch-and-Bound*(BB) [46] and the *LP/NLP Based Branch-and-Bound*(LP/NLP-BB) [71] algorithms are single-tree methods, while the Outer-Approximation(OA) [38, 65] and Extended Cutting Plane(ECP) [82] algorithms are multi-tree methods. For the remainder of this thesis BB will be used as the MINLP solver. More information on algorithms and theory for convex MINLPs and non-convex MINLPs can be found in [11].

Nonlinear Branch-and-Bound

The nonlinear branch-and-bound algorithm is a divide-and-conquer algorithm which finds the optimal solution to a convex MINLP by iteratively dividing the search space and solving for the optimal solutions to smaller subproblems. Initially, the MINLP (P_{MINLP}) is relaxed to an NLP by discarding the integrality constraint. If the solution to the NLP problem is feasible, then the algorithm terminates. Otherwise, a fractional integer variable $y_j \notin \{0, 1\}$ is selected and two new subproblems(*branches*) are created, in which the first problem has the bound $y_j \leq \lfloor y_j \rfloor$ and the second problem has the bound $y_j \geq \lfloor y_j \rfloor$. The new subproblems are solved by an NLP solver which may be using an Interior Point Method [67] or a Sequential Quadratic Programming algorithm [69], and the process of branching and bounding is repeated. In order to improve efficiency, an upper bound is stored so that subproblems with an optimal value that is larger than the upper bound can be removed(*pruned*). When there are no more remaining subproblems left to solve the branch-and-bound algorithm will terminate. A statement of the nonlinear branch-and-bound algorithm is given in Algorithm 1.

2.3 Software for Convex MINLPs

There are several software packages that implement algorithms to solve convex MINLPs. A recent survey on this topic can be found in [30]. The software that is used for solving MINLPs in this thesis is `BONMIN`(Basic Open-source Non-linear Mixed INteger programming) [19], which is an open-source C++ code and is distributed within COIN-OR(http://www.coin-or.org) under the Common Public License(CPL). To solve NLPs and MILPs, `BONMIN` uses `IPOPT` [52] and `CBC` [24], respectively. The problems can be run with `BONMIN` from modelling languages such as `AMPL` [40] or `GAMS` [72] or from the command line with `.nl`-files.

There is a variety of convex MINLP algorithms implemented in BONMIN: Nonlinear Branch-and-Bound, Extended Cutting Plane,
Outer-Approximation, LP/NLP based Branch-and-Bound and Hybrid Outer-Approximation [19]. In addition, several heuristics which are designed to aid in the solution process may be invoked through user options. A summary on the impact of using these heuristics can be found in [22].

2.4 Heuristics

There are two categories of heuristics for mixed-integer programs: *construction heuristics* and *improvement heuristics*. This thesis is focused on construction heuristics.

Construction Heuristics

A construction heuristic is a method or algorithm intended to be used at the start of an optimization routine. For a mixed-integer program, a construction heuristic finds a feasible solution. Both FP and undercover are construction heuristics.

Improvement Heuristics

Once a feasible solution has been found, for instance in a branch-and-bound search, the goal becomes to either find a solution which is an improvement upon the current best solution or to prove that no such solution exists. An improvement heuristic uses the information provided by the best known solution to search for a better solution. The only improvement heuristic that has been applied to MINLPs, is the RINS heuristic [31].

Algorithm 1 The Branch-and-Bound algorithm for Mixed-Integer Nonlinear Programs

Initialization

1: Create a list L containing formulations of the subproblems to be solved. Initially the list contains only the problem at the root node, i.e. $L = \{S\}$
2: Initialize the upper bound, $z^U := +\infty$.
3: Set $i := 0$
4: **while** (the list L is nonempty) **do**

Node Selection and Solution

5: Select a problem S_i from the list L and let $L := L \setminus \{S_i\}$. Let the relaxed feasible set of S_i be denoted by S_R
6: Compute the optimal NLP-value z_{NLP} and its solution (x^*, y^*)
7: **if** (the problem is infeasible, i.e. $z_{\text{NLP}} = +\infty$) **then**
8: Fathom the node and go to step 26
9: **else**
10: Set the lower bound $z^L := z_{\text{NLP}}$

Pruning

11: **if** $(z_{\text{NLP}} > z^U)$ **then**
12: Remove the problem S_i from the list L and go to step 26
13: **end if**
14: **if** (the solution is feasible, i.e. $y^* \in \{0,1\}^q$) **then**
15: **if** $(f(x^*, y^*) < z^U)$ **then**
16: Update the upper bound: $z^U := f(x^*, y^*)$
17: Store the optimal solution: (x^*, y^*)

Fathoming

18: Remove the items in the list L which have a lower bound larger or equal to z^U. Go to step 26
19: **end if**
20: **end if**
21: **end if**

Branching

22: **if** (the solution is not integer feasible, i.e.. $y^* \notin \{0,1\}^q$) **then**
23: Select some branching variable y_j and apply branching constraints
24: Update the list L with new subproblems
25: **end if**
26: Set $i := i + 1$
27: **end while**

Chapter 3

An Objective Feasibility Pump for Convex MINLPs

In this chapter a new feasibility pump heuristic for convex MINLPs which is inspired from the objective feasibility pump for MILPs, is presented. The goal of this heuristic is to compute feasible solutions with better objective values than the rounding based FP, which is reviewed in Section 3.1.2, without sacrificing too much computation time. The new heuristic is implemented in `BONMIN` in order to obtain good comparability with the original FP for MINLPs.

This chapter is presented as follows. Initially the background of the FP is reviewed in Section 3.1, after which a FP for convex MINLPs is presented. In Section 3.2 the OFP for MILPs is reviewed. The new OFP for MINLPs requires some background information on *multiobjective optimization*, therefore a brief summary of relevant concepts from this topic are given in Section 3.3.1. Finally, the proposed OFP for MINLPs is presented in Section 3.3.2.

3.1 The Feasibility Pump

The feasibility pump was originally introduced as a heuristic for finding feasible solutions of difficult MILPs [36]. The main idea of the FP heuristic is to solve for two sequences of points, in which one sequence consists of points which are constraint feasible, while the other sequence consists of points which are integer feasible. If the two sequences converge to the same point, then a feasible solution is found and the FP terminates.

Since its inception, a considerable amount of research effort has been devoted to making developments and assessing the theoretical properties of this heuristic. An improved FP for MILPs which not only finds feasible solutions but also aims to simultaneously improve the objective value was proposed in [3]. Further improvements to the original FP by using preprocessing techniques such as *constraint propagation* [75] and *bound strengthening* [74] are presented in [37]. Other variants of a FP heuristic for MILPs by using a *line search* algorithm [17] and nondifferentiable concave *penalty functions* [32] have recently been proposed. A theoretical study on the use of penalty functions and the implementation of *cutting planes* [26] within a FP framework are presented in [18]. The FP heuristic for MILPs has gained mainstream attention and is integrated in commercial solvers for MILPs such as `CPLEX` [50], `Mosek` [70], `Gurobi` [47], `Xpress-MP` [27] and `LINDO` [51].

Two different extensions of the feasibility pump have been made for convex MINLPs. The first variant presented in [20] uses an outer-approximation approach and finds feasible solutions by solving a sequence of MILPs and NLPs. The second variant presented in [22] bears a closer resemblance to the original FP for MILPs and only solves a sequence of NLPs. An experimental FP for non-convex MINLPs which only uses outer-approximations of convex constraints and random perturbation methods to avoid local minima is proposed in [6]. Software implementations of a FP heuristic for MINLPs can be found in convex MINLP solvers such as `BONMIN` and `MINOTAUR` [64].

3.1.1 The Feasibility Pump as a Successive Projection Method

It is noted in [28] that FP algorithms belong to a general class of algorithms, called *Successive Projection Methods*(SPM) [48], which determine whether or not the intersection of sets is empty. In the case when the sets are convex, this problem is referred to as the *convex feasibility problem* [8] and a mathematical formulation of this problem can be stated as follows.

> *Given the closed convex sets $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_N$ with nonempty intersection $\mathcal{C}$:*
>
> $$\mathcal{C} = \mathcal{C}_1 \cap \mathcal{C}_2 \cap \ldots \cap \mathcal{C}_N \neq \emptyset.$$
>
> *Find some point x in $\mathcal{C}$.*

In the case when the problem is to find a point in the intersection of the two convex sets $\mathcal{C}_1$ and $\mathcal{C}_2$, the successive projection problem can be stated as

$$\min \{||t - s|| \mid t \in \mathcal{C}_1, s \in \mathcal{C}_2\}, \tag{3.1.1}$$

where the norm is assumed to be the Euclidean norm for now. The problem in (3.1.1) can be decomposed into two separate problems that can be solved sequentially

$$t^i \in \arg\min \left\{ \left|\left| t - s^{i-1} \right|\right| \mid t \in \mathcal{C}_1 \right\} \tag{3.1.2a}$$

$$s^i \in \arg\min \left\{ \left|\left| t^i - s \right|\right| \mid s \in \mathcal{C}_2 \right\}, \tag{3.1.2b}$$

where $i \geq 1$ and an initial point $s^0 \in \mathcal{C}_2$ is given. Since the sets $\mathcal{C}_1$ and $\mathcal{C}_2$ are convex and the function $||\cdot||$ is convex [23] it follows that the sequence (t^i, s^i) converge to a globally optimal solution.

3.1.2 A Feasibility Pump for Convex MINLPs

For the MINLP problem ($\mathrm{P}_{\mathrm{MINLP}}$), the feasibility pump described in [22] finds a feasible solution by solving a sequence of NLPs. Initially, the NLP relaxation ($\mathrm{P}_{\mathrm{NLP}}$) is solved to yield a constraint feasible point $(\bar{x}^0, \bar{y}^0)$. The integer variables are then rounded to the nearest integer point, $(\tilde{x}, \tilde{y})$, and then a projection problem of minimizing the distance to the integer feasible point is solved. By applying an SPM approach to a convex MINLP, the problems in (3.1.2) can be stated as

$$\tilde{y}^i \in \arg\min \left\{ \left|\left| y - \bar{y}^{i-1} \right|\right| \mid y \{0,1\}^q \right\}, \tag{3.1.3a}$$

$$(\bar{x}^i, \bar{y}^i) \in \arg\min \left\{ \left|\left| y - \tilde{y}^i \right|\right| \mid g(x,y) \leq 0, x \in X, y \in [0,1]^q \right\}, \tag{3.1.3b}$$

where $i \geq 1$. It should be noted that the problem of projecting the point $\bar{y}^{i-1}$ onto the integer feasible set (3.1.3a), is solved by simply rounding the point $\bar{y}^{i-1}$ to the nearest integer. The second problem (3.1.3b) requires the solution of a convex NLP which can be stated as

$$\begin{aligned} \min_{x,y} \quad & ||y - \tilde{y}||_1 \\ \text{s.t} \quad & g(x,y) \leq 0, \\ & x \in X, \\ & y \in [0,1]^q, \end{aligned} \tag{$\mathrm{FP}_{\mathrm{MINLP}}$}$$

where an ℓ_1-norm is used since it was shown to have slightly better results than the ℓ_2-norm [22]. Let the solution to ($\mathrm{FP}_{\mathrm{MINLP}}$) be denoted by $(\bar{x}, \bar{y})$.

If the solution satisfies $||\bar{y} - \tilde{y}||_1 = 0$, then the solution $(\bar{x}, \bar{y})$ is feasible for ($\mathrm{P}_{\mathrm{MINLP}}$). On the other hand, if $\tilde{y} \neq \bar{y}$, then the integer feasible solution which is closest to the current solution can be found by rounding the current solution to the nearest integer, i.e. $\tilde{y} := [\bar{y}]$ where $[\cdot]$ denotes rounding to the nearest integer. The problem ($\mathrm{FP}_{\mathrm{MINLP}}$) is solved with the new integer feasible point $\tilde{y}$ and the process outlined above is repeated until a feasible solution is found or a termination criterion, such as an upper limit on the number of iterations, is met.

If the solution to ($\mathrm{FP}_{\mathrm{MINLP}}$) yields a point $(\bar{x}, \bar{y})$ such that the rounding of this point is equal to the previous integer feasible point, i.e. $[\bar{y}] = \tilde{y}$, then the algorithm is said to be *stalling*. The proposed method to deal with stalling is to flip the binary variable which corresponds to the largest value of $|\bar{y}_j - \tilde{y}_j|,\ j = 1 \ldots q$. This choice is shown experimentally to yield the best results [22].

Another difficulty which can occur in the feasibility pump is the issue of *cycling*. After a certain amount of iterations it is possible that the algorithm enters a loop in which the same sequence of points $(\bar{x}, \bar{y})$ and $(\tilde{x}, \tilde{y})$ are continuously revisited without ever converging. To overcome this issue, a random perturbation mechanism is used when a cycle is detected. If a new solution to the problem ($\mathrm{FP}_{\mathrm{MINLP}}$) was also found in one of the previous three iterations, then a uniformly distributed random number $\rho_j \in [-0.3, 0.7],\ j = 1, \ldots, q$ is generated and the value of $\tilde{y}_j$ is flipped if the condition $|\bar{y}_j - \tilde{y}_j| + \max\{\rho_j, 0\} > 0.5$ is satisfied. Note that due to the random perturbations introduced to tackle stalling and cycling, the FP heuristic is a *nondeterministic* algorithm [39].

A specialized scheme to round variables which belong to a *special ordered set* of type 1(SOS1) [9] is implemented in `BONMIN`. A set of variables $y_j, j = 1, \ldots k$ which belong to a SOS1, can be explicitly written as a constraint of the form

$$\sum_{j=1}^{k} y_j = 1,$$

where $y_j \in \{0, 1\}, \mathrm{J} = 1, \ldots, k$. To see why special care must be taken in the rounding procedure when dealing with SOS-constraints, consider the following simple example. Let a SOS-constraint be defined by

$$y_1 + y_2 + y_3 = 1, \tag{3.1.4}$$

where $y_1, y_2, y_3 \in \{0, 1\}$, and let the solution to a continuous relaxation yield

the point

$$\bar{y}_1 = \frac{1}{3},\ \bar{y}_2 = \frac{1}{4},\ \bar{y}_3 = \frac{1}{4}.$$

When the rounding of this point is computed, it will result in the following values

$$\tilde{y}_1 = 0,\ \tilde{y}_2 = 0,\ \tilde{y}_3 = 0,$$

which clearly violates the SOS-constraint in (3.1.4). The approach taken in BONMIN for problems with such constraints is to make sure that one variable is rounded to 1, and the rest are rounded to 0. This is achieved by setting $\tilde{y}_j = 1$ for $j = [s]$ and $\tilde{y}_j = 0$ otherwise, where

$$s = \sum_{j=1}^{k} j\bar{y}_j.$$

Algorithm 2 A feasibility pump for Convex MINLPs.

Initialize
Choose an iteration limit I_L
Solve the NLP-relaxation of ($\mathrm{P}_{\mathrm{NLP}}$) to obtain the solution $(\bar{x}^0, \bar{y}^0)$
if ($\bar{y}^0$ is integer feasible) **then**
    Return the current solution $(\bar{x}^0, \bar{y}^0)$. **Terminate**
end if
Set $\tilde{y} := [\bar{y}^0]$
Set $i := 1$
while $(i < I_L)$ **do**
    Solve ($\mathrm{FP}_{\mathrm{MINLP}}$) to obtain the solution $(\bar{x}^i, \bar{y}^i)$
    if ($\bar{y}^i$ is integer feasible) **then**
        Return the current solution $(\bar{x}^i, \bar{y}^i)$. **Terminate**
    end if
    if $[\bar{y}] \neq \tilde{y}$ **then**
        $\tilde{y} := [\bar{y}^i]$
        if (cycle detected) **then**
            for $(j = 1 \dots q)$ **do**
                Generate a uniformly distributed random number $\rho_j \in [-0.3, 0.7]$
                if $(|\bar{y}_j - \tilde{y}_j| + \max\{\rho_j, 0\} > 0.5)$ **then**
                    Flip $\tilde{y}_j$
                end if
            end for
        end if
    else
        Flip the entry of $\tilde{y}$ with largest $|\bar{y}^i_j - \tilde{y}_j|$
    end if
    $i := i + 1$
end while

3.1.3 Comparison of the First Solution Found by FP and BB

A computational study was performed on a set of 86 convex MINLP test problems from `http://egon.cheme.cmu.edu/ibm/page.htm`. Each problem was downloaded in a `.nl`-format and solved with `BONMIN`. An upper limit on the solution time was set to 1 hour and the problems were solved by BB with and without the FP as a root node heuristic. The other user options were left to their default values. Table 3.1 summarizes the results which can be found in Appendix B.

Table 3.1: Computational results of using BB with and without FP on convex MINLPs.

Comparison	**BB**	**BB w. FP**
Sum of final solutions	6849705.7	6849698.1
Mean of final solutions	80584.8	80584.7
Sum of times to final solutions	147077.4	147469.2
GM time to final solution[a]	198.1	204.2
Sum of first solutions	7078897.3	12401856.3
Mean of first solutions	83281.1	145904.2
Sum of times to first solutions	4501.2	298.5
GM time to first solution	9.2	0.2
Sum of number of nodes to final solutions	3017272	3006036
Mean of number of nodes to final solutions	35497.3	35365.1

[a] : geometric mean.

The problems were run on a personal computer with Intel Core i7-2600 3.40 GHz CPU and 16GB of RAM.

Similar tests were also performed on a set of 38 nonconvex MINLPs from `http://wiki.mcs.anl.gov/leyffer/index.php/MacMINLP` and `http://www.gamsworld.org/minlp/minlplib/minlpstat.htm`, but for 22 of these problems FP fails to find a solution because the maximum iteration limit is exceeded or the NLP solver fails to converge.

It can be seen from Table 3.1 that the final solution value, number of nodes and time to final solution are similar with and without FP. The main difference lies in the time to first solution and objective value at the first solution. Although the geometric mean of the time required by BB to find its first solution is 46 times larger than the FP, on average it finds a solution with an objective value which yields a 57.1% reduction of the average solution value found by the FP.

3.2 The Objective Feasibility Pump for MILPs

The feasibility pump for MILPs briefly described in Section 3.1 generates a sequence of constraint feasible and integer feasible points which hopefully converge to a feasible point. Initially, the constraint feasible point is found by computing the LP-relaxation of the orginal MILP. For the remainder of the solution process the objective value is not considered. Therefore the solution quality with respect to the objective value is usually poor [12].

The objective feasibility pump [3] does not discard the original MILP objective after the initialization, instead it gradually reduces the influence of the objective value and increases the influence of an objective cost which corresponds to integer infeasibility as the iteration number increases. This approach is similar to the use of a penalty function for nonlinear optimization described in [69](chapter 15.4).

Let the ℓ_1-norm distance function be denoted by $\Delta(y, \tilde{y}) := ||y - \tilde{y}||_1$. The objective function used in the OFP consists of a convex combination of the original objective of the MILP and the distance function $\Delta(y, \tilde{y})$.

$$\begin{aligned} \min_{x,y} \quad & \frac{(1-\alpha_i)}{||\Delta||_2}\Delta(y,\tilde{y}) + \frac{\alpha_i}{||[c,d]||_2}(c^T x + d^T y) \\ \text{s.t.} \quad & Ax + By \leq b, \\ & x \in \mathbb{R}^n,\ y \in \{0,1\}^q, \end{aligned} \tag{OFP$_{\text{MILP}}$}$$

where $\alpha_i \in [0, 1]$ and $[c, d]$ denotes the concatenation of the two objective function vectors. In the binary case the norm $||\Delta||_2$ is simply the square root of the number of binary variables. At each iteration i, the coefficient α_i is geometrically decreased by a factor $\phi \in (0, 1)$, i.e $\alpha_{i+1} = \phi\alpha_i$ with $\alpha_0 \in (0, 1]$.

The algorithm for the OFP bears a close resemblance to the original FP, the difference lies in the objective function which is replaced by (OFP$_{\text{MILP}}$) and a modified scheme for cycle detection.

3.3 An Objective Feasibility Pump for Convex MINLPs

Computational results of the FP for convex MINLPs show that it is able to find a feasible solution significantly faster than the default nonlinear branch-and-bound algorithm in BONMIN, but that the solution quality is poor, as

summarized in Table 3.1. This result raises the question of whether it is possible to modify the FP in order to improve the solution quality without sacrificing too much computation time. An OFP heuristic for convex MINLPs which finds a feasible solution while attempting to simultaneously improve the original objective is presented in this section.

The objective function (OFP_{MILP}) is a scaled and weighted combination of two functions which have a conflicting goal. Satisfying the integrality condition is the goal in the first term, while in the second term it is the minimization of the original objective. In this function each term is scaled by the Euclidean norm of their gradient. A straightforward approach to extending this method to MINLPs is to simply use the same scaling, but replace the MILP objective with the MINLP objective and constraints in (P_{NLP}), i.e.

$$\begin{aligned} \min_{x,y} \quad & \frac{(1-\alpha_i)}{||\Delta||_2}\Delta(y,\tilde{y}) + \frac{\alpha_i}{||\nabla f^{i-1}||_2} f(x,y) \\ \text{s.t.} \quad & g(x,y) \leq 0, \\ & x \in X, \\ & y \in [0,1]^q, \end{aligned} \tag{3.3.1}$$

where ∇f^{i-1} is the gradient of the objective function at the previous iteration and is ensured to satisfy $||\nabla f^{i-1}||_2 > 0$. As shown in Table 3.2, this approach did not yield good results. Therefore an alternative approach to compute a scaling of the two objectives was developed.

Without any *a priori* knowledge of how the objective function varies on the relaxed feasible set it may not always be possible to compute an appropriate scaling value. Hence, a methodology which uses multiobjective optimization to solve the objective feasibility pump problem was considered.

3.3.1 Multiobjective Optimization

A multiobjective optimization problem can be written in the following form

$$\begin{aligned} z = \min_{x} \quad & \{f_1(x), f_2(x), \ldots, f_k(x)\} \\ \text{s.t.} \quad & x \in \Omega, \end{aligned} \tag{3.3.2}$$

where $z \in \mathbb{R}^k$, $f_i : \mathbb{R}^n \to \mathbb{R}$ are (possibly) conflicting objectives and $\Omega \subseteq \mathbb{R}^n$ is the feasible region. If the objective functions are competing with each other, then there is most likely not a unique optimal solution which minimizes

all the functions simultaneously. Therefore an optimal solution in a multi-objective optimization context is a solution for which there does not exist any other feasible solution that produces an improvement of at least one objective without deteriorating any other objective. The notion of an optimal solution in multi-objective optimization is used by verifying whether or not it is *Pareto optimal* [33]. A solution $x^* \in \Omega$ is said to be Pareto optimal if there exists no other $x \in \Omega$ for which $f_i(x) \leq f_i(x^*)$ for $i = 1, \ldots, k$ and $f_j(x) < f_j(x^*)$ for at least one index j. The Pareto optimal solutions form a *Pareto optimal set*.

A multiobjective optimization problem can be cast into a single-objective optimization problem by using a *weighted sum method*. In a weighted sum method the problem in (3.3.2) is reformulated as:

$$\begin{aligned} \min_x \quad & \sum_{i=1}^{k} w_i f_i(x) \\ \text{s.t.} \quad & x \in \Omega, \end{aligned} \tag{3.3.3}$$

where $w_i \geq 0$ for $i = 1, \ldots, k$ and $\sum_{i=1}^{k} w_i = 1$. The weights are chosen by a user and are determined according to the specific problem structure. Since different objectives may have different magnitudes, it is required that the objectives are normalized in order to get a Pareto optimal solution that is consistent with the selected weights [34, 54]. Normalized weights can be expressed as $w_i = u_i \eta_i$, where u_i are weights and η_i are normalization factors. The normalization factors can for instance be chosen as the magnitude of the initial point

$$\eta_i = \frac{1}{f_i(x_0)}$$

or the minimum of the objective function

$$\eta_i = \frac{1}{f_i(x^{[i]})}$$

where $x^{[i]} = \arg\min_x \{f_i(x) : \ x \in \Omega\}$. Both of these choices have proved to be ineffective [34]. Another possible normalization method is to compute normalization factors by the differences of optimal function values in the *Nadir* and *Utopia* points. The Utopia point $z^U \in \mathbb{R}^k$ is defined as the ideal objective vector at which each individual objective function attains its minimum value when it is minimized on the feasible set, i.e.

$$z_i^U = f_i(x^{[i]}).$$

Although the Utopia point is unlikely to be feasible due to the conflicting objectives, it does provide lower bounds of the Pareto optimal set. Similarly, the upper bounds of the Pareto optimal set can be obtained by computing the Nadir point $z^N \in \mathbb{R}^k$, which is defined as

$$z_i^N = \max_{1 \le j \le k} (f_i(x^{[j]})), \ \forall i = 1, \ldots, k.$$

An illustration of how the Nadir and Utopia points may look in a 2-dimensional multiobjective optimization problem is given in Figure 3.1.

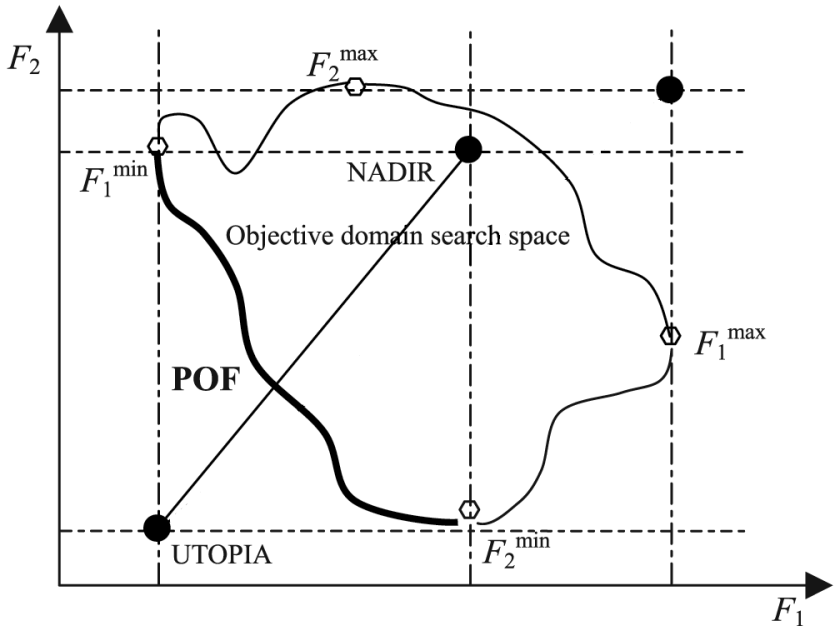

Figure 3.1: Pareto front with Nadir and Utopia points[1]

The normalization factors η_i can then be computed as follows

$$\eta_i = \frac{1}{z_i^N - z_i^U}. \tag{3.3.4}$$

One of the benefits of using this method is that all of the objective functions after normalization are bounded by

$$0 \le \frac{f_i(x) - z_i^U}{z_i^N - z_i^U} \le 1,$$

[1]source: `http://www.emeraldinsight.com/journals.htm?articleid=877827&show=html`

however, in order to compute these normalization factors a total number of k optimization problems need to be solved. There are several methods proposed to avoid using excessive computation time on solving for the normalization factors [34], among them are:

1. Relaxing one or more of the termination criteria in the solver, e.g. allowing for a larger duality gap
2. Removing one or more constraints
3. Avoid solving optimization problems altogether by sampling random points in the feasible set and using estimates of the Nadir and Utopia points

The OFP for MINLPs developed in this thesis will use the first method from the list above.

3.3.2 A Multiobjective Optimization Approach to the Feasibility Pump Problem

The feasibility pump sub-problem ($\mathrm{FP}_{\mathrm{MINLP}}$)

$$\begin{aligned} \min_{x,y} \quad & ||y - \tilde{y}||_1 \\ \text{s.t.} \quad & g(x,y) \leq 0, \\ & x \in X, \\ & y \in [0,1]^q, \end{aligned} \tag{$\mathrm{FP}_{\mathrm{MINLP}}$}$$

and the relaxed NLP problem ($\mathrm{P}_{\mathrm{NLP}}$)

$$\begin{aligned} z = \min_{x,y} \quad & f(x,y) \\ \text{s.t.} \quad & g(x,y) \leq 0, \\ & x \in X, \\ & y \in [0,1]^q, \end{aligned} \tag{$\mathrm{P}_{\mathrm{NLP}}$}$$

can be regarded as two optimization problems with conflicting objectives. A multi-objective optimization problem can be formulated by expressing these two problems as

$$\begin{aligned} z = \min_{x,y} \quad & \{f(x,y), ||y - \tilde{y}||_1\} \\ \text{s.t.} \quad & g(x,y) \leq 0, \\ & x \in X, \\ & y \in [0,1]^q. \end{aligned} \tag{3.3.5}$$

Furthermore, the problem in (3.3.5) can be cast into a single objective optimization problem by the weighted sum method

$$\begin{aligned} z = \min_{x,y} \quad & w_1 \left|\left|y - \tilde{y}\right|\right|_1 + w_2 f(x,y) \\ \text{s.t.} \quad & g(x,y) \leq 0, \\ & x \in X, \\ & y \in [0,1]^q. \end{aligned} \tag{3.3.6}$$

The two terms in the objective function in (3.3.6) may differ by orders of magnitude. Therefore, the method of using Nadir and Utopia points to achieve a normalization of the objective terms presented in the previous section will be used.

Initally, the FP heuristic computes the solution to the NLP relaxation of ($\mathrm{P}_{\mathrm{MINLP}}$) to obtain a constraint feasible point. For the sub-problem ($\mathrm{FP}_{\mathrm{MINLP}}$), the value of the distance function $\left|\left|y - \tilde{y}\right|\right|_1$ at the NLP solution is its component in the Nadir point, while the value at a feasible point is its component in the Utopia point. Since the distance function has the value zero at a feasible point, the objective value of ($\mathrm{FP}_{\mathrm{MINLP}}$) at the Utopia point is known *a priori*.

The component of the Utopia point of the NLP problem ($\mathrm{P}_{\mathrm{NLP}}$) is attained at the NLP solution, while the Nadir point component is found at a feasible solution of the original MINLP problem. A feasible solution is not known *a priori*, which means that a feasible solution must be found in order for the chosen normalization method to be used.

In order to reduce the computational effort that is spent on solving for the normalization factors, a relaxation of the problem ($\mathrm{FP}_{\mathrm{MINLP}}$) is used. The relaxation strategy that is considered here is: Use FP to compute a feasible point, but increase the integer tolerance from its default value of 10^{-7} in `BONMIN` to 10^{-3}. By using this approach it is guaranteed that a feasible solution up to a certain tolerance is used when the normalization factors are computed.

Since the OFP algorithm uses a feasible solution to find a solution with an improved objective value, it can be considered to be both a construction heuristic and an improvement heuristic.

Let the point (x', y') denote a feasible or a relaxed feasible point, the solution to ($\mathrm{P}_{\mathrm{NLP}}$) be denoted by $(\bar{x}^0, \bar{y}^0)$ and let the two functions $f_1(x,y)$ and $f_2(x,y)$ be defined as

$$f_1(x,y) := \left|\left|y - \tilde{y}\right|\right|_1, \; f_2(x,y) := f(x,y).$$

The Nadir and Utopia point components of the first function $f_1(x, y)$ are

$$z_1^N = \left|\left|\bar{y}^0 - \tilde{y}\right|\right|_1, \;\; z_1^U = 0, \tag{3.3.7}$$

and the Nadir and Utopia point components of the second function $f_2(x, y)$ are

$$z_2^N = f(x', y'), \; z_2^U = f(\bar{x}^0, \bar{y}^0). \tag{3.3.8}$$

By substituting these values into Equation (3.3.4), the normalizations are obtained as

$$\eta_1 = \frac{1}{||\bar{y}^0 - \tilde{y}||_1} \tag{3.3.9}$$

$$\eta_2 = \frac{1}{f(x', y') - f(\bar{x}^0, \bar{y}^0)} \tag{3.3.10}$$

The normalized weighted sum problem can now be expressed as

$$\begin{aligned} z = \min_{x,y} \quad & u_1\eta_1 \left|\left|y - \tilde{y}\right|\right|_1 + u_2\eta_2 f(x, y) \\ \text{s.t.} \quad & g(x, y) \leq 0, \\ & x \in X, \\ & y \in [0, 1]^q, \end{aligned} \tag{3.3.11}$$

where the weights are assigned the values $u_1 := 1$ and $u_2 := 10$. It is observed that the values of these weights have a significant impact on the outcome of the algorithm and it is a topic of discussion in Section 3.3.5.

Now that a normalized single objective optimization problem for an OFP has been obtained, the procedure of the algorithm can be described. Similar to the OFP for MILPs, the initial goal is to initially minimize the original objective $f(x, y)$ and after a while shift the focus to finding a feasible solution. This is achieved by introducing a weighting factor α_i which is reduced geometrically at each iteration, i.e. $\alpha_{i+1} = \phi\alpha_i$ for $\alpha_0 \in (0, 1]$ where $\phi \in (0, 1)$. The objective feasibility pump problem at an iteration i is expressed as

$$\begin{aligned} z = \min_{x,y} \quad & (1 - \alpha_i)u_1\eta_1 \left|\left|y - \tilde{y}\right|\right|_1 + \alpha_i u_2\eta_2 f(x, y) \\ \text{s.t.} \quad & g(x, y) \leq 0, \\ & x \in X, \\ & y \in [0, 1]^q. \end{aligned} \tag{$\text{OFP}_{\text{MINLP}}$}$$

Since the continuous relaxation (P_{NLP}) is assumed to be convex, it follows that the problem ($\text{OFP}_{\text{MINLP}}$) is also convex. This comes from the fact that the

ℓ_1-norm function is convex and that the nonnegative sum of convex functions is also convex [23].

The issue of how to properly detect if the algorithm is cycling has not yet been addressed. Cycling occurs when the algorithm revisits the previously computed solutions without converging. Due to the similarity between the objective function ($\text{OFP}_{\text{MINLP}}$) from one iteration to the next, there is a high likelihood that the computed solution will be the same. Hence, the cycle detection method described in Section 3.1.2 is no longer valid. To increase the probability that a cycle is detected correctly, the method for cycle detection used in the OFP for MILPs is applied [3]. If a solution $(\bar{x}^i, \bar{y}^i)$ is equal to a solution that was found in a previous iteration $(\bar{x}^k, \bar{y}^k)$, for some $0 < k < i$, then the algorithm is considered to be cycling if and only if $\alpha_k - \alpha_i \leq \delta_\alpha$. When a cycle is detected, a random perturbation is introduced by generating uniformly random numbers $\rho_j \in [-0.3, 0.7]$ for $j = 1, \ldots, q$, and flipping the values of $\tilde{y}_j$ if the condition $|\bar{y}_j - \tilde{y}_j| + \max\{\rho_j, 0\} > 0$ holds. The method for dealing with stalling is left unchanged from the original FP.

3.3.3 Nonconvex MINLPs

There are a few issues that arise if the continuous relaxation of the MINLP problem is nonconvex. When the normalization factors are computed, the utopia point component found at the solution of the relaxation (P_{NLP}) may not be the global optimum of this problem. This may, however, not have any serious consequences since the OFP will search for a feasible solution by solving NLPs to local optimality. It is therefore likely that the normalization is valid for the part of the feasible domain in which the OFP searches for a feasible solution. Since there is no requirement that the problem ($\text{OFP}_{\text{MINLP}}$) has to be solved a global optimum, it does not have any immediate consequences if a locally optimal solution is found at each iteration. If the OFP does not converge, then a random perturbation will be introduced due to stalling or cycling. Hence, the proposed OFP heuristic is likely to be robust for nonconvex MINLPs. In Chapter 7, the OFP will be applied to a large number of nonconvex MINLPs.

3.3.4 Algorithm

Algorithm 3 An Objective Feasibility Pump for Convex MINLPs.

Initialize
Set upper iteration limit $I_L := 200$ and initial weight $\alpha_0 := 1$
Solve the NLP-relaxation of ($\mathrm{P}_{\mathrm{MINLP}}$) to obtain the solution $(\bar{x}^0, \bar{y}^0)$
Set $\tilde{y} := [\bar{y}^0]$
Solve the Feasibility Problem with low tolerance
Compute the Nadir and Utopia points as described in (3.3.7) and (3.3.8)
Compute the normalization factors as described in (3.3.9) and (3.3.10)
Set $i := 1$
while (i $\leq I_L$) **do**
    Set $\alpha_i = \phi\alpha_{i-1}$
    Solve ($\mathrm{OFP}_{\mathrm{MINLP}}$) to obtain the solution $(\bar{x}^i, \bar{y}^i)$
    if ($\bar{y}^i$ is integer feasible) **then**
        Return the current solution $(\bar{x}^i, \bar{y}^i)$. **Terminate**
    end if
    if $[\bar{y}] \neq \tilde{y}$ **then**
        $\tilde{y} := [\bar{y}^i]$
    else
        Flip the value of $\tilde{y}_j$ corresponding to the largest value of $|\bar{y}^i_j - \tilde{y}_j|$
    end if
    if $((\bar{x}^i, \bar{y}^i) = (\bar{x}^k, \bar{y}^k)$ for some $0 < k < i$ **and** $\alpha_k - \alpha_i \leq \delta_\alpha$) **then**
        Generate uniform random numbers $\rho_j \in [-0.3, 0.7]$ for $j = 1, \ldots, q$
        if $(|\bar{y}_j - \tilde{y}_j| + \max\{\rho_j, 0\} > 0.5)$ **then**
            Flip $\tilde{y}_j$
        end if
    end if
    Set $i := i + 1$
end while

The algorithm was implemented into `BONMIN` in order to obtain good comparability with the original FP. The files that were changed are `BonHeuristicFpump.hpp`, `BonHeuristicFpump.cpp`, `BonTNLP2FPNLP.hpp`, `BonTNLP2FPNLP.hpp`, `BonOsiTMINLPInterface.hpp` and `BonOsiTMINLPInterface.cpp`.

3.3.5 Results and Discussion

The results of a computational study of using the OFP on 84 out of 86 of the convex MINLP test problems that are used in Appendix B are summarized in Table 3.2. Different values of the geometric reduction parameter ϕ are used to gauge the impact this parameter has on the results.

The problems were run on a personal computer with Intel Core i7-2600 3.40 GHz CPU and 16GB of RAM.

Table 3.2: Computational results of OFP on convex MINLP test problems.

FP type	Sum of solutions	Compared to FP(better:draw:worse)	GM time[a]
FP	12401856	-	0.14
OFP07[b]	11792582	27:33:24	0.46
OFP08	11700298	32:31:21	0.58
OFP09	11533453	37:17:30	0.94
OFP095	11061178	32:17:35	1.50
OFPG[c]	11818720	44:04:36	1.88

[a] : geometric mean.
[b] : number indicates value of ϕ, e.g. for OFP07 the value of $\phi := 0.7$.
[c] : Equation (3.3.1).

It is observed that for an increasing value of ϕ, the geometric mean of the time required by the OFPs to find a solution increases while the value of the sum of the solutions found by the OFPs decreases. The best result with regards to the solution value is achieved by OFP095, which yields a 10.81% improvement for the sum of solutions over the regular FP. However, the geometric mean of the time required by OFP095 to find the first solution is 11 times larger than the original FP. This result is expected since the OFP has to use additional time in solving a sequence of NLPs to find a new feasible solution after the FP has found an initial feasible solution.

Compared to the results of the OFP for MILPs reported in [3], the new OFP heuristic proposed for MINLPs performs quite poorly. The OFP for MILPs finds an improved objective value for 89 out of 121 MILP test problems with an average reduction of 53.1% in the objective value over the regular FP for MILPs. For OFP095 an improvement is only found in 32 of the 84 problems, with an average reduction in the objective value of 10% when compared to the FP for MINLPs. A possible explanation is that for the original FP for MINLPs a feasible solution is found in a single iteration in 61 out of the 84 problems. An implication of this result is that a feasible solution exists in the vicinity of the rounded solution to the NLP relaxation. It is therefore

likely that the OFP will eventually converge to this point, but not in a single iteration since the integer infeasibility objective has a low weight in the initial iterations.

The CPU time required to find a feasible solution by all of the OFPs for MINLPs is significantly longer than what is reported for the OFP for MILPs. On average the OFP for MILPs required only 1.17 times as much CPU time as the regular FP for MILPs to find a solution, while the fastest OFP for MINLPs reported in Table 3.2, OFP07, has a geometric mean of the CPU time that is 3.29 times larger than the original FP for MINLPs. It is expected that the OFP for MINLPs should require more time to find solutions than the OFP for MILPs since solving NLPs is in general more computationally demanding than solving LPs [63], this is, however, dependent on the problem. Moreover, it is not a completely fair comparison since MILPs and MINLPs are fundamentally different, but the trend in the differences in the results indicate that the OFP for MINLPs has an overall poor performance.

Stalling is an issue which causes a notable increase in the time required by OFP. It was observed that the results of varying the number of binary variables that are flipped when stalling occurs were ambiguous. For some problems it is beneficial to flip many variables while for others it is best to flip only one variable. To demonstrate the impact the number of binary variables that are flipped has on the outcome, consider the problems BatchS101006M and RSyn0830M04M. If the number of binary variables that are flipped are varied between the values 1 and 4, then the results obtained by the FP and OFP09 given in Table 3.3 show that both the objective value and the CPU time are affected.

Table 3.3: Results of varying the number of binary variables flipped when stalling occurs on problems BatchS101006M and RSyn0830M04M.

Problem	**Solution**	**Time[s]**	**Flip value**[a]	**FP type**
BatchS101006M	821554.2	0.07	1	FP
BatchS101006M	804632.1	0.65	1	OFP09[b]
BatchS101006M	783453.0	1.08	4	OFP09
RSyn0830M04M	-706.1	0.36	1	FP
RSyn0830M04M	-718.3	6.40	1	OFP09
RSyn0830M04M	-519.3	12.13	4	OFP09

[a] : number of binary variables flipped when stalling occurs.
[b] : OFP09 corresponds to the OFP with a value $\phi := 0.9$.

While only two problems are shown in Table 3.3, their results are similar to

what is observed in other problem instances and they are included here to demonstrate the impact that the flipping parameter has on the outcome.

Another parameter which has a significant impact on the result of OFP is the weight of the original objective, u_2, in the problem ($\text{OFP}_{\text{MINLP}}$). Consider the problems CLay0204M and Syn20M04M, the results of varying the value of u_2 on OFP09 are shown in Table 3.4.

It can be seen that increasing the weight of the original objective u_2 does not necessarily lead to a better solution value.

Table 3.4: Results of varying objective weight parameter u_2 on problems CLay0204M and Syn20M04M.

Problem	**Solution**	**Time[s]**	u_2	**FP type**
CLay0204M	11045.0	0.18	-	FP
CLay0204M	10205.0	1.56	1	OFP09[a]
CLay0204M	11405.0	0.76	10	OFP09
Syn20M04M	-1393.9	0.06	-	FP
Syn20M04M	-1332.8	0.68	1	OFP09
Syn20M04M	-3143.0	1.49	10	OFP09

[a] OFP09 corresponds to the OFP with a value $\phi := 0.9$.

An illustration of how the integer infeasibility $||\bar{y} - \tilde{y}||_1$ evolves on the problem Syn20M04M when OFP09 is applied, is shown in Figure 3.2. It can be seen that the random perturbation introduced when a cycle is detected increases the magnitude of the integer infeasibility, but assists with the convergence of the algorithm.

A plot of the objective values $f(x, y)$ that are found by OFP09 on the problem Syn20M04M is presented in Figure 3.3. It can be seen that the random perturbations yield a decrease in the objective value.

The Pareto front of the solutions found by OFP09 is given in Figure 3.4. It can be seen that 12 out of 48 solutions computed on the problem Syn20M04M are not pareto optimal. None 3 of the solutions that were found due to the random perturbation are in the Pareto optimal set.

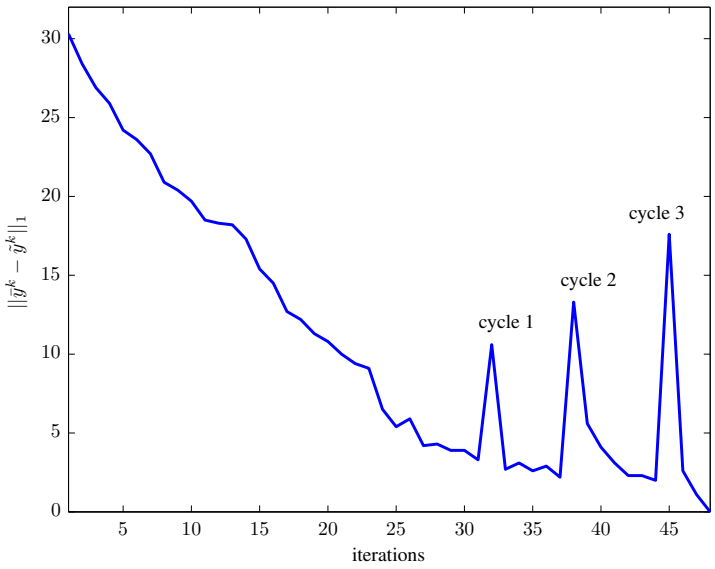

Figure 3.2: Integer infeasibility vs. iterations on problem Syn20M04M.

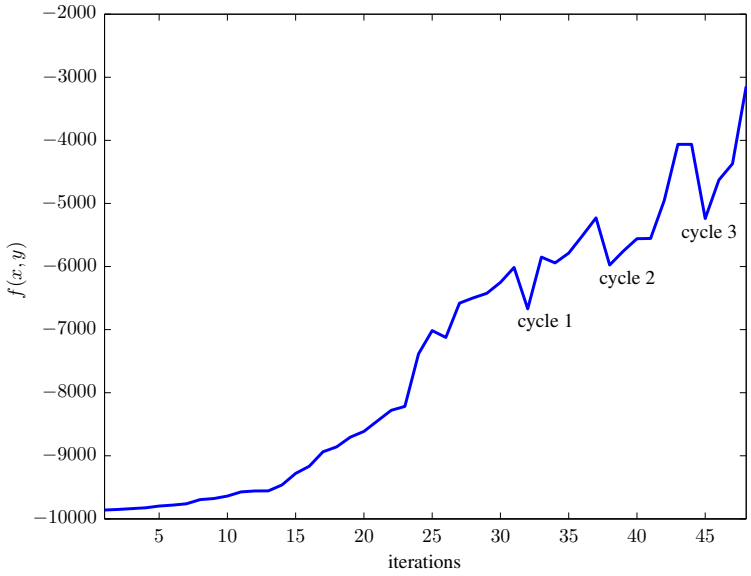

Figure 3.3: Objective value vs. iterations on problem Syn20M04M.

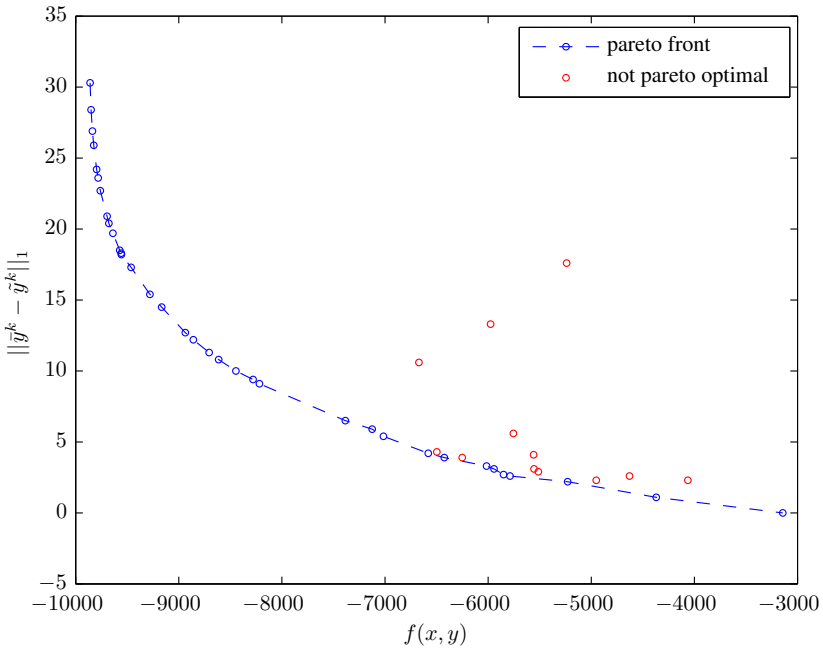

Figure 3.4: Pareto front of integer infeasibility vs. objective value on problem Syn20M04M.

Chapter 4

Shale Gas Production Optimization

In this chapter a shale gas production optimization is model is presented. Initially, a summary on the background of the model is given in Section 4.1. New suggestions for modelling *shut-in* and handling issues such as *liquid-loading* are given in Section 4.2. In Section 4.3, reformulations of the original model are described. A discussion on a convex relaxation of the model is presented in Section 4.4.

Nomenclature

Sets

$\mathcal{J} = \{1, \dots J\}$	set of wells
$\mathcal{I} = \{1, \dots N_r\}$	set of gridblocks from spatial discretization
$\mathcal{K} = \{1, \dots K\}$	set of timesteps
$\mathcal{K}' = \{2, \dots K-1\}$	set of timesteps excluding first and last time period

Indices

j	well
k	time

Parameters

G_{p}	unit gas price [\$/m^3]
β	regression parameter[]
$A_j \in \mathbb{R}^{N_r \times N_r}$	spatial discretization state-space matrix for well j[]
$B_j \in \mathbb{R}^{N_r}$	spatial discretization state-space control input matrix for well j[]
$m_{j,\mathrm{init}} \in \mathbb{R}^{N_r}$	initial pseudopressure for well j[Pa/s]
C_{t}	tubing constant[]
S	skin friction factor[]
α_1	regression parameter[]
α_2	regression parameter[]
p^{Line1}	line pressure on low-pressure line[bar]
p^{Line2}	line pressure on high-pressure line[bar]
$q_{\mathrm{tot}}^{\mathrm{low}}$	lower bound of flow routed to compressor[m^3/s]
$q_{\mathrm{tot}}^{\mathrm{up}}$	upper bound of flow routed to compressor[m^3/s]
$q_j^{\max}$	upper limit on flow from each well j[m^3/s]
M_1	big-M parameter for relaxing well inflow equation[]
M_1'	big-M parameter for relaxing well inflow equation[]
q_{gc}^1	critical gas rate for flow on low-pressure line[m^3/s]
q_{gc}^2	critical gas rate for flow on high-pressure line[m^3/s]

Variables

q_{jk}	flow rate from well j at time k[m^3/s]
$m_{jk} \in \mathbb{R}^{N_r}$	pseudopressures of well j at time k[Pa/s]
$m_{\mathrm{wf},jk}$	flowing bottomhole linearized pseudopressure of well j at time k[Pa/s]
$p_{\mathrm{t},jk}$	tubing pressure of well j at time k [bar]
$\bar{p}_{jk}$	bottomhole flowing pressure of well j at time k squared[bar^2]
$y_{jk}^1 \in \{0,1\}$	flow line routing for well j at time k[0 low-pressure , 1 high-pressure]
$y_{jk}^2 \in \{0,1\}$	shut-in for well j at time k[0 open, 1 shut]
v_{jk}	reformulation of $q_{jk}y_{jk}^1$ for well j at time k[m^3/s]
$w_{jk}^1 \in [0,1]$	reformulation of $(1-y_{jk}^2)y_{jk}^1$ for well j at time k[0 false, 1 true]
$w_{jk}^2 \in [0,1]$	reformulation of $(1-y_{jk}^2)(1-y_{jk}^1)$ for well j at time k[0 false, 1 true]

4.1 Problem description

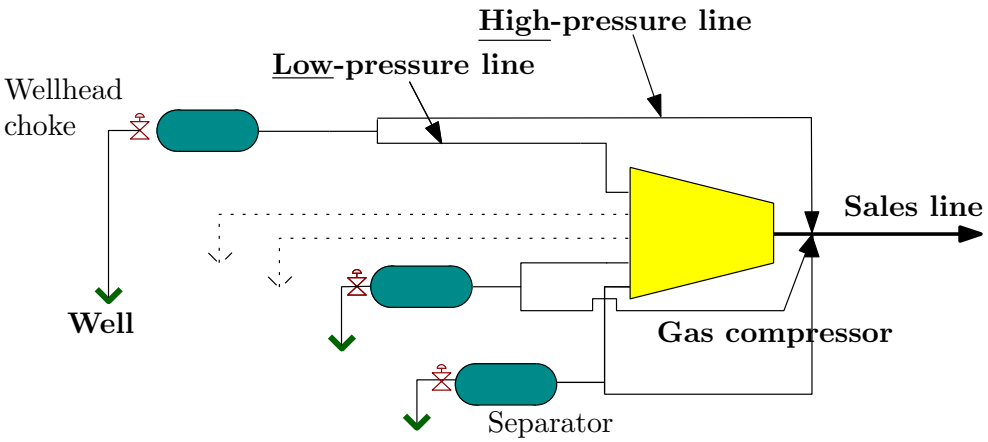

Figure 4.1: Two line system model [1]

Consider the model in Figure 4.1. The surface gathering system is constructed such that each well can operate on two surface pressure lines: One line is a low pressure line, typically around 10 bar, which is connected upstream of the compressor. The other line is a high pressure line with pressures typically around 25-40 bar, which is connected downstream of the compressor. That is, directly on the sales line. The routing of which line the gas is to be transmitted on can be done instantaneously. When the gas is sent on the line upstream of the compressor, the gas must be compressed to a pressure equal the high-pressure line in order to have high enough pressure to transport the gas to the customers. This compression is often performed by midstream companies, which require a certain percentage of the gas sales price to perform the compression, typically 5-10%. On the other hand, if the operators choose to use the high-pressure line, then the resulting wellhead pressure will be higher, hence lowering the gas rate. The routing decision clearly gives rise to an optimization problem.

An integral feature which greatly increases the complexity of the shale gas production optimization problem is the shut-in of the wells. The description of shut-ins is currently left out, but will be included in subsequent sections.

[1] source: [56]

4.1.1 The Model

The problem presented above can be formulated as a mixed-integer nonlinear program. A brief summary of the mathematical model of this problem is given in this section. The reader is referred to [55,58,59] for more information on the derivation and background of this model

Let $j \in \mathcal{J}$ be the well index, and $k \in \mathcal{K}$ be the time index, and let $y^1_{jk} \in \{0, 1\}$ be binary routing variables, equal to one if the low-pressure line is used, and equal to zero if the high-pressure downstream line is used.

The Objective

The objective is to maximize the profit generated by the gas production, and the objective function can be defined as

$$Z = \max\ G_{\mathrm{p}} \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K} \setminus K} \left(1 - 0.05 y^1_{jk}\right) q_{jk} \Delta k, \tag{4.1.1}$$

where q_{jk} is the per-timestep gas rate from each well. A discount factor may be added if preferred.

State-Space Model

The *reservoir proxy model* is formulated as the linear discrete time ODE [59]

$$A_j m_{jk+1} = m_{jk} + B_j q_{jk+1}, \quad \forall j \in \mathcal{J}, k \in \mathcal{K} \tag{4.1.2}$$

$$m_{j0} = m_{j,\mathrm{init}}, \quad \forall j \in \mathcal{J} \tag{4.1.3}$$

where m_{jk} is the so-called *pseudopressure* [4] and A and B are constant matrices. The matrix A is tridiagonal as a result of performing a *spatial discretization* of a parabolic PDE.

Well Inflow Model

The well inflow from the reservoir is given by

$$q_{jk} = \beta \left(m_{jk1} - m_{\mathrm{wf},jk}\right), \quad \forall j \in \mathcal{J}, k \in \mathcal{K}, \tag{4.1.4}$$

where m_{jk1} is the pseudopresure of the innermost gridblock and $m_{\mathrm{wf},jk}$ is the *flowing bottomhole pseudopressure*.

Wellbore Model

The wellbore model is given by the quadratic constraint

$$p_{\mathrm{wf},jk}^2 = e^{\mathrm{S}} \left(p_{\mathrm{t},jk}^2 + \frac{1}{C_{\mathrm{t}}^2} q_{jk}^2 \right), \quad \forall j \in \mathcal{J}, k \in \mathcal{K}, \tag{4.1.5}$$

where $p_{\mathrm{t},jk}$ is the tubing pressure, $p_{\mathrm{wf},jk}$ is the *flowing bottomhole pressure* and C_{t} and S are tubing constants [53]. The constant C_{t} is related to the friction dependent pressure loss, while $\frac{S}{2}$ corresponds to the hydrostatic pressure loss (i.e. the weight of the gas).

The pressure squared can be transformed to pseudopressure for $m_{\mathrm{wf},jk}$ in (4.1.4) by using the linear map [58]

$$\bar{p}_{jk} := p_{\mathrm{wf},jk}^2 \tag{4.1.6}$$

$$m_{\mathrm{wf},jk} = \alpha_1 \bar{p}_{jk} + \alpha_2, \tag{4.1.7}$$

where α_1 and α_2 are regression constants. Note that $\bar{p}_{j,k}$ can be used as a substitution for $p_{\mathrm{wf},jk}^2$ in Equation (4.1.5), hence eliminating this polynomial term.

Routing -and Flow Constraints

The tubing pressure is restricted by the constraint

$$p_{\mathrm{t},jk} \geq p^{\mathrm{Line1}} y_{jk}^1 + p^{\mathrm{Line2}} (1 - y_{jk}^1), \quad \forall j \in \mathcal{J}, k \in \mathcal{K} \tag{4.1.8}$$

The compressor requires a minimum gas rate for stable operation(to prevent *surge* and *shut downs*), i.e.

$$\sum_{j \in \mathcal{J}} q_{jk} y_{jk}^1 \geq q_{\mathrm{tot}}^{\mathrm{low}}, \quad \forall k \in \mathcal{K}. \tag{4.1.9}$$

Normally, there is also an upper bound on the compressor capacity,

$$\sum_{j \in \mathcal{J}} q_{jk} y_{jk}^1 \leq q_{\mathrm{tot}}^{\mathrm{up}}, \quad \forall k \in \mathcal{K}, \tag{4.1.10}$$

as well as a maximum rate $q_j^{\max}$ each well is designed to handle,

$$q_{jk} = \min(\beta \left(m_{jk1} - m_{\mathrm{wf},jk} \right), q_j^{\max}) \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}. \tag{4.1.11}$$

This constraint can be rewritten as

$$q_{jk} \leq q_j^{\max} \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}, \tag{4.1.12a}$$

$$q_{jk} = \beta \left(m_{jk1} - m_{\mathrm{wf},jk} \right) \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}. \tag{4.1.12b}$$

4.1.2 Computing Model Parameters[2]

The model parameters for the shale gas production optimization problem are computed by performing a weighted least-square fit on a high-fidelity model implemented in the state-of-the-art software package, SENSOR [25], for simulating petroleum reservoirs. The model in SENSOR is implemented as a dense gridded multi-fracture horizontal well with a complex fracture network together with a static wellbore model similar to Equation (4.1.5). The tuning procedure is based on solving an NLP, as is described in detail in [57]. However, the model tuning used in this thesis differs from the one described in [57] in the way that the model described in the previous section also estimates the bottomhole pressure. Hence, the residuals (i.e. model discrepancy) are minimized for the flow rate during production, and bottomhole pressure during shut-ins. Note that as the bottomhole pressure is a static term in Equation (4.1.5), the build-up in bottomhole pressure is estimated during shut-ins as the pressure in the innermost grid block, m_1. The same procedure is observed to be applied in the simulator model in SENSOR. The fit of the model is shown in Figures 4.2 and 4.3. Figure 4.2 shows that a good match of the bottomhole pressure is obtained during build-up, while there is a prominent discrepancy in p_{wf} in the transients from the highest peak rates. However, during these transients, the match in flow rate is observed to best as seen in Figure 4.3. For more information on the tuning procedure, see [59] and [57].

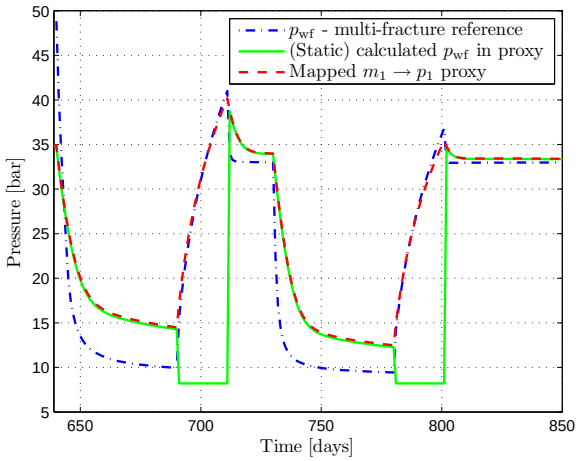

Figure 4.2: Tuning of proxy models using bottomhole pressure with predefined shut-in and wellhead pressure.

[2]Figures and content from this section are from [56].

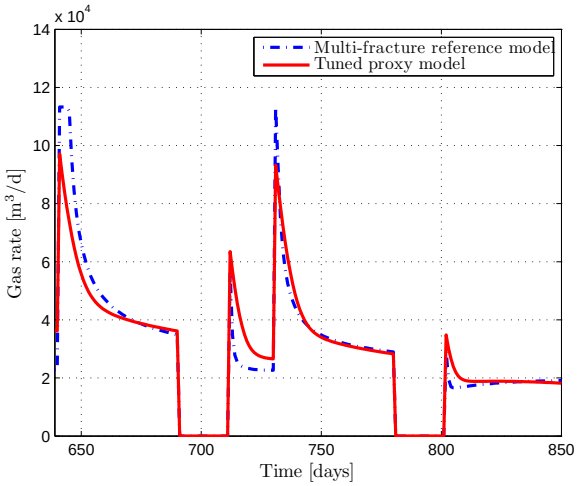

Figure 4.3: Tuning of proxy models using flow rate with predefined shut-in and wellhead pressure.

4.1.3 Optimization Problem

Putting together the objective (4.1.1) with the constraints equations (4.1.3–4.1.5) and (4.1.7–4.1.11) a complete nonconvex MINLP formulation of the production optimization problem can be expressed as

$$Z = \max\ G_{\mathrm{p}} \sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{K}\setminus K} \left(1 - 0.05 y^1_{jk}\right) q_{jk}\, \Delta k, \tag{4.1.13a}$$

s.t. (4.1.13b)

$$\sum_{j\in\mathcal{J}} q_{jk} y^1_{jk} \le q^{\mathrm{up}}_{\mathrm{tot}}, \qquad \forall k \in \mathcal{K} \tag{4.1.13c}$$

$$\sum_{j\in\mathcal{J}} q_{jk} y^1_{jk} \ge q^{\mathrm{low}}_{\mathrm{tot}}, \qquad \forall k \in \mathcal{K} \tag{4.1.13d}$$

$$A_j m_{jk+1} = m_{jk} + B_j q_{jk+1}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K} \setminus K \tag{4.1.13e}$$

$$m_{j0} = m_{j,\mathrm{init}}, \qquad \forall j \in \mathcal{J} \tag{4.1.13f}$$

$$q_{jk} = \beta \left(m_{jk1} - m_{\mathrm{wf},jk}\right), \qquad \forall j \in \mathcal{J}, k \in \mathcal{K} \tag{4.1.13g}$$

$$m_{\mathrm{wf},jk} = \alpha_1 \bar{p}_{jk} + \alpha_2, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K} \tag{4.1.13h}$$

$$\bar{p}_{jk} = e^{\mathrm{S}} \left(p^2_{\mathrm{t},jk} + \frac{1}{C^2_{\mathrm{t}}} q^2_{jk}\right), \qquad \forall j \in \mathcal{J}, k \in \mathcal{K} \tag{4.1.13i}$$

$$q_{jk} \le q^{\max}_j, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K} \tag{4.1.13j}$$

$$p_{\mathrm{t},jk} \ge p^{\mathrm{Line1}} y^1_{jk} + p^{\mathrm{Line2}} (1 - y^1_{jk}), \qquad \forall j \in \mathcal{J}, k \in \mathcal{K} \tag{4.1.13k}$$

The degrees of freedom, i.e. the control variables, are the tubing pressures $p_{\mathrm{t},jk}$ and the routing variables y^1_{jk}.

4.2 Modelling Shut-in

In a shale gas production optimization scheme, it is sometimes beneficial to allow for a well to be shut in certain situations. When a well is shut, there is no production of gas or any other fluid. There are two scenarios discussed in this thesis in which it is desirable to shut a well:

- If the routing for the flow from a well changes from the low-pressure line to the high-pressure in Figure 4.1, shutting that well for a given time period allows for an increase in the pressure which in turn allows for the flow to remain on the high-pressure line for an extended period of time. This is a necessary shut-in due to pressure restrictions in the system. A formulation of the problem (4.1.13) with a time constrained shut-in procedure is described in Section 4.2.1.
- When the flow from a well drops below a certain *critical rate*, there is a possibility that *liquid loading* [77] may occur. This phenomenon introduces several difficulties and in the most severe case it may cause a complete cease of production. This situation may occur if the well is routed from the low to the high pressure line, in which the well must be shut-in until the wellhead pressure $p_{\mathrm{t},jk}$ is larger than the(high pressure) line pressure, p^{Line2}. Once the rate q_{jk} once again is positive, it may be unacceptably low in the sense that liquid loading may occur. By restricting the flow to remain above a critical rate and by enforcing a shut-in of the well when the flow rate is less than the critical rate, it may be possible to eliminate or reduce the impact of liquid loading. A formulation of the problem (4.1.13) with a time constrained shut-in procedure and constraints to avoid liquid-loading is described in Section 4.2.2.

4.2.1 Shut-in by Switching

Logical Constraints

In this section, constraints for the routing of the flow and shut-in of the wells are derived. The modeling of these constraints is done by using logic

propositions, similar to models used in *generalized disjunctive programming* [45].

It is desirable to avoid frequent switching of the flow between the two lines, as this may not be practical in a realistic scenario. A restriction can be enforced by requiring that when the flow switches to a line it has to remain on that line for at least τ_1 time steps. The boolean variable $Y^1_{jk} \in \{\text{true}, \text{false}\}$ will be used to state the logical propositions for the routing y^1_{jk}. A condition on the routing time can be stated by the following propositions

$$Y^1_{jk-1} \wedge \neg Y^1_{jk} \Rightarrow \neg Y^1_{jk+1} \wedge \neg Y^1_{jk+2} \wedge \ldots \wedge \neg Y^1_{jk+\tau_1}, \tag{4.2.1a}$$
$$\neg Y^1_{jk-1} \wedge Y^1_{jk} \Rightarrow Y^1_{jk+1} \wedge Y^1_{jk+2} \wedge \ldots \wedge Y^1_{jk+\tau_1}, \tag{4.2.1b}$$

The first relation states that when the routing switches from the high pressure line to the low pressure line the routing in the next τ_1 time steps must be set to the low pressure line, and the second relation states the opposite. By noting that the two relations can be rewritten as

$$Y^1_{jk-1} \wedge \neg Y^1_{jk} \Rightarrow \neg Y^1_{jk+1}, \tag{4.2.2a}$$
$$Y^1_{jk-1} \wedge \neg Y^1_{jk} \Rightarrow \neg Y^1_{jk+2}, \tag{4.2.2b}$$
$$\vdots \tag{4.2.2c}$$
$$Y^1_{jk-1} \wedge \neg Y^1_{jk} \Rightarrow \neg Y^1_{jk+\tau_1}, \tag{4.2.2d}$$
$$\tag{4.2.2e}$$

and

$$\neg Y^1_{jk-1} \wedge Y^1_{jk} \Rightarrow Y^1_{jk+1}, \tag{4.2.3a}$$
$$\neg Y^1_{jk-1} \wedge Y^1_{jk} \Rightarrow Y^1_{jk+2}, \tag{4.2.3b}$$
$$\vdots \tag{4.2.3c}$$
$$\neg Y^1_{jk-1} \wedge Y^1_{jk} \Rightarrow Y^1_{jk+\tau_1}. \tag{4.2.3d}$$
$$\tag{4.2.3e}$$

These relations can be expressed as linear constraint functions by using the binary variable y^1_{jk},

$$1 - y^1_{jk+1} - y^1_{jk-1} + y^1_{jk} \geq 0, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}_{\tau_1}, \tag{4.2.4a}$$
$$1 - y^1_{jk+2} - y^1_{jk-1} + y^1_{jk} \geq 0, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}_{\tau_1}, \tag{4.2.4b}$$
$$\vdots \tag{4.2.4c}$$
$$1 - y^1_{jk+\tau_1} - y^1_{jk-1} + y^1_{jk} \geq 0, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}_{\tau_1}, \tag{4.2.4d}$$
$$\tag{4.2.4e}$$

and

$$
\begin{aligned}
y^1_{jk+1} + y^1_{jk-1} - y^1_{jk} \geq 0, \qquad & \forall j \in \mathcal{J}, k \in \mathcal{K}_{\tau_1}, && \text{(4.2.5a)}\\
y^1_{jk+2} + y^1_{jk-1} - y^1_{jk} \geq 0, \qquad & \forall j \in \mathcal{J}, k \in \mathcal{K}_{\tau_1}, && \text{(4.2.5b)}\\
\vdots & && \text{(4.2.5c)}\\
y^1_{jk+\tau_1} + y^1_{jk-1} - y^1_{jk} \geq 0, \qquad & \forall j \in \mathcal{J}, k \in \mathcal{K}_{\tau_1}, && \text{(4.2.5d)}\\
& && \text{(4.2.5e)}
\end{aligned}
$$

where $\mathcal{K}_{\tau_1} := \{k \in \mathcal{K} : k + \tau_1 \leq K, k > 1\}$.

When the routing switches from the low pressure line to the high pressure line, as stated in (4.2.1a), the well should be shut for a certain amount of time τ_2 in order to allow the pressure to build up. Let the boolean variable $Y^2_{jk} \in \{\text{true}, \text{false}\}$ indicate whether a well j is shut at time k and let the shut-in time be specified as τ_2 time steps. The shut-in time as a result of switching is given by the following proposition

$$Y^1_{jk-1} \wedge \neg Y^1_{jk} \Rightarrow Y^2_{jk} \wedge Y^2_{jk+1} \wedge \ldots \wedge Y^2_{jk+\tau_2}. \tag{4.2.6}$$

By using the binary variable y^2_{jk} to express the logical proposition 4.2.6 as constraints, the following constraint formulation is obtained

$$
\begin{aligned}
y^2_{jk} - y^1_{jk-1} + y^1_{jk} \geq 0, \qquad & \forall j \in \mathcal{J}, k \in \mathcal{K}_{\tau_2}, && \text{(4.2.7a)}\\
y^2_{jk+1} - y^1_{jk-1} + y^1_{jk} \geq 0, \qquad & \forall j \in \mathcal{J}, k \in \mathcal{K}_{\tau_2}, && \text{(4.2.7b)}\\
\vdots & && \text{(4.2.7c)}\\
y^2_{jk+\tau_2} - y^1_{jk-1} + y^1_{jk} \geq 0, \qquad & \forall j \in \mathcal{J}, k \in \mathcal{K}_{\tau_2}, && \text{(4.2.7d)}
\end{aligned}
$$

where $\mathcal{K}_{\tau_2}$ is defined similarly to $\mathcal{K}_{\tau_1}$.

Experimental results on test problems showed that even without a switching from the low pressure line to the high pressure line, the shut-in variable y^2_{jk} was occasionally set to 1. The implication of this observation is that a larger cumulative gas production can be obtained by shutting down the wells at other points in time at which switching does not occur. However, if there are no restrictions on how long a well should be shut when no switching has occurred, the wells may stay shut for only a single time step. In order to avoid this, an additional constraint is added so that once a well is shut, it has to remain shut for at least τ_3 time steps. This can be stated by the proposition

$$\neg Y^2_{jk-1} \wedge Y^2_{jk} \Rightarrow Y^2_{jk+1} \wedge Y^2_{jk+2} \wedge \ldots \wedge Y^2_{jk+\tau_3}, \tag{4.2.8}$$

which can be written as the following constraints

$$y^2_{jk+1} + y^2_{jk-1} - y^2_{jk} \geq 0, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}_{\tau_3}, \tag{4.2.9a}$$
$$y^2_{jk+2} + y^2_{jk-1} - y^2_{jk} \geq 0, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}_{\tau_3}, \tag{4.2.9b}$$
$$\vdots \tag{4.2.9c}$$
$$y^2_{jk+\tau_3} + y^2_{jk-1} - y^2_{jk} \geq 0, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}_{\tau_3}, \tag{4.2.9d}$$

where $\mathcal{K}_{\tau_3}$ is defined similarly to $\mathcal{K}_{\tau_1}$. Note that due to the overlap between Equations (4.2.7) and (4.2.9), depending on the value of τ_1 and τ_3, it may not be necessary to include the full set of constraints.

It was also observed in the same experiments that frequent shut-ins occur if there is no limit on how long a well should stay open once it is producing, therefore a constraint which enforces a minimum producing time of τ_4 time steps is included in the formulation. This limitation can be expressed by the proposition

$$\neg Y^2_{jk-1} \wedge Y^2_{jk} \Rightarrow \neg Y^2_{jk+1} \wedge \neg Y^2_{jk+2} \wedge \ldots \wedge Y^2_{jk+\tau_4}, \tag{4.2.10}$$

and written in as the following set of constraints

$$1 - y^2_{jk+1} - y^2_{jk-1} + y^2_{jk} \geq 0, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}_{\tau_4}, \tag{4.2.11a}$$
$$1 - y^2_{jk+2} - y^2_{jk-1} + y^2_{jk} \geq 0, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}_{\tau_4}, \tag{4.2.11b}$$
$$\vdots \tag{4.2.11c}$$
$$1 - y^2_{jk+\tau_4} - y^2_{jk-1} + y^2_{jk} \geq 0, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}_{\tau_4}, \tag{4.2.11d}$$

where $\mathcal{K}_{\tau_4}$ is defined similarly to $\mathcal{K}_{\tau_1}$.

For the remainder of this thesis, the various time-step specifications are set to 1 time-step, i.e. $\tau_1 = \tau_2 = \tau_3 = \tau_4 = 1$. This choice is solely for the purpose of maintaining simple expressions, and can easily be augmented.

The constraints (4.2.4) and (4.2.5) which limit the amount of switching between the low/high-pressure lines are then given as

$$1 - y^1_{jk+1} - y^1_{jk-1} + y^1_{jk} \geq 0, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}', \tag{4.2.12a}$$
$$y^1_{jk+1} + y^1_{jk-1} - y^1_{jk} \geq 0, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}', \tag{4.2.12b}$$

where $\mathcal{K}' := \{k \in \mathcal{K} : k < K, k > 1\}$. Which means that if the flow for a well j at time k is routed to a certain line, then it will stay on that line for at least one more time step. Similarly the constraints (4.2.7) will in this case be reduced to

$$y^2_{jk} - y^1_{jk-1} + y^1_{jk} \geq 0, \forall j \in \mathcal{J}, k \in \mathcal{K}', \tag{4.2.13}$$

which means that at time k well j will be shut for a single time step when the flow routing is switched. In conjunction with the constraints (4.2.9), when a well j is shut at time k, it will remain shut at least until time $k+1$

$$y^2_{jk+1} + y^2_{jk-1} - y^2_{jk} \geq 0, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}'. \tag{4.2.14}$$

Lastly, the constraints (4.2.11) will for this case be given as

$$1 - y^2_{jk+1} - y^2_{jk-1} + y^2_{jk} \geq 0, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}', \tag{4.2.15}$$

which means that if well j is open at time k, then it will remain open for at least one more time step. Let the set of constraints (4.2.12), (4.2.13), (4.2.14) and (4.2.15)

$$\begin{aligned}
1 - y^1_{jk+1} - y^1_{jk-1} + y^1_{jk} &\geq 0, && \forall j \in \mathcal{J}, k \in \mathcal{K}' \\
y^1_{jk+1} + y^1_{jk-1} - y^1_{jk} &\geq 0, && \forall j \in \mathcal{J}, k \in \mathcal{K}' \\
y^2_{jk} - y^1_{jk-1} + y^1_{jk} &\geq 0, && \forall j \in \mathcal{J}, k \in \mathcal{K}' \\
y^2_{jk+1} + y^2_{jk-1} - y^2_{jk} &\geq 0, && \forall j \in \mathcal{J}, k \in \mathcal{K}' \\
1 - y^2_{jk+1} - y^2_{jk-1} + y^2_{jk} &\geq 0, && \forall j \in \mathcal{J}, k \in \mathcal{K}',
\end{aligned}$$

be denoted by the function

$$g_{\mathrm{s},jk}(y^1, y^2) \geq 0, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}'. \tag{4.2.17}$$

Physical Constraints

In the previous section, logical propositions for the routing and shut-in were formulated as linear constraints involving only binary variables. The relationship between the shut-in variable y^2_{jk} and the flow constraints (4.1.12) are described in this section.

When a well is shut, the gas rate q_{jk} must be stopped, i.e. $q_{jk} = 0$. This condition can be modelled as

$$q_{jk} \leq q^{\max}_j (1 - y^2_{jk}), \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}. \tag{4.2.18}$$

One of the goals of the new approach presented in this section is to allow for the pseuodpressures m_{jk1} and $m_{\mathrm{wf},jk}$ to vary independent of each other when the well is shut. Consider the following formulation of constraint (4.1.4)

$$q_{jk} \leq \beta \left(m_{jk1} - m_{\mathrm{wf},jk} \right), \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}, \tag{4.2.19a}$$

$$q_{jk} \geq \beta \left(m_{jk1} - m_{\mathrm{wf},jk} \right), \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}. \tag{4.2.19b}$$

The constraints (4.2.19a) and (4.2.19b) are to be relaxed when $y_{jk}^2 = 1$. The inverse relation can be stated as

$$\neg Y_{jk}^2 \Rightarrow q_{jk} = \beta \left(m_{jk1} - m_{\mathrm{wf},jk} \right), \tag{4.2.20}$$

which can be expressed as the following constraints

$$q_{jk} \leq \beta \left(m_{jk1} - m_{\mathrm{wf},jk} \right) + M_1 y_{jk}^2, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}, \tag{4.2.21a}$$

$$q_{jk} \geq \beta \left(m_{jk1} - m_{\mathrm{wf},jk} \right) - M_1' y_{jk}^2, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}, \tag{4.2.21b}$$

where the *big-M parameters* [81] $M_1, M_1' > 0$ are chosen to be sufficiently large so that the constraint (4.1.4) is relaxed when a well is shut. Numerical values for the big-M parameters which will be used in the computational study of the shale gas production optimization problems are given in Chapter 6.

By including the constraints (4.2.17),(4.2.18) and(4.2.21) into the formulation of the problem (4.1.13), the optimization problem ($\mathrm{P}_{\mathrm{switch,bncvx}}$) is obtained

$$Z = \max\ G_{\mathrm{p}} \sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{K}\setminus K} \left(1 - 0.05 y^1_{jk}\right) q_{jk}\, \Delta k, \qquad (\mathrm{P}_{\mathrm{switch,bncvx}})$$

s.t.

$$\sum_{j\in\mathcal{J}} q_{jk} y^1_{jk} \le q^{\mathrm{up}}_{\mathrm{tot}}, \qquad \forall k \in \mathcal{K}$$

$$\sum_{j\in\mathcal{J}} q_{jk} y^1_{jk} \ge q^{\mathrm{low}}_{\mathrm{tot}}, \qquad \forall k \in \mathcal{K}$$

$$A_j m_{jk+1} = m_{jk} + B_j q_{jk+1}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K} \setminus K$$

$$m_{j0} = m_{j,\mathrm{init}}, \qquad \forall j \in \mathcal{J}$$

$$q_{jk} \le \beta\left(m_{jk1} - m_{\mathrm{wf},jk}\right) + M_1 y^2_{jk}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$q_{jk} \ge \beta\left(m_{jk1} - m_{\mathrm{wf},jk}\right) - M'_1 y^2_{jk}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$m_{\mathrm{wf},jk} = \alpha_1 \bar{p}_{jk} + \alpha_2, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$\bar{p}_{jk} = e^{\mathrm{S}}\left(p^2_{\mathrm{t},jk} + \frac{1}{C^2_{\mathrm{t}}} q^2_{jk}\right), \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$q_{jk} \le (1 - y^2_{jk}) q^{\max}_j, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$p_{\mathrm{t},jk} \ge p^{\mathrm{Line1}} y^1_{jk} + p^{\mathrm{Line2}}(1 - y^1_{jk}), \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$g_{\mathrm{s},jk}(y^1, y^2) \ge 0, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}'.$$

4.2.2 Shut-in by Switching with a Critical Gas Rate

When the flow rate drops below a certain critical value, there is a possibility that liquid may accumulate in the bottom of the well [77]. If the liquid accumulates over an extended period of time, the well may encounter *liquid-loading* problems. These problems include: water and/or condensate build up in the bottom of the well and an overall production decrease which may eventually lead to a cease in production. By ensuring that the gas flow rate is larger than the *critical gas rate*, the accumulation of liquid in the bottom of the well can be avoided. It is therefore important to impose a constraint which specifies a lower bound on the flow rate to reduce the likelihood that liquid-loading occurs. The critical gas rate, q_{gc}, is a nonlinear function of the tubing pressure and the *pressure dependent critical velocity*, $v_{\mathrm{gc}}(p_{\mathrm{t}})$,

$$q_{\mathrm{gc}}(p_{\mathrm{t}}) = C_{\mathrm{gc}} p_{\mathrm{t}} v_{\mathrm{gc}}(p_{\mathrm{t}}), \qquad (4.2.23)$$

where C_{gc} is a constant and v_{gc} is a nonlinear function of p_{t}, see [78]. To avoid the inclusion of the nonlinear expression (4.2.23) in the optimization

problem formulation, the variable q_{gc} is specified conservatively to only take on two values, q_{gc}^1 and q_{gc}^2, depending on which line the flow is routed. When the well is shut, the flow is stopped and consequently the critical gas rate in this case is set to zero. The condition that the flow rate must be greater than or equal to the critical gas rate q_{gc}^1 when the flow is routed to the low-pressure line can be stated by the following relation

$$\neg Y_{jk}^2 \wedge Y_{jk}^1 \Rightarrow q_{jk} \geq q_{\text{gc}}^1, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}. \tag{4.2.24}$$

Similarly, the lower bound condition on the flow rate when the flow is routed to the high-pressure line can be written as

$$\neg Y_{jk}^2 \wedge \neg Y_{jk}^1 \Rightarrow q_{jk} \geq q_{\text{gc}}^2, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}. \tag{4.2.25}$$

A straightforward constraint formulation of conditions (4.2.24) and (4.2.24) can be written as

$$q_{jk} \geq (1 - y_{jk}^2) q_{\text{gc},jk}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}, \tag{4.2.26}$$

where $q_{\text{gc},jk}$ is given by

$$q_{\text{gc},jk} = q_{\text{gc}}^1 y_{jk}^1 + q_{\text{gc}}^2 (1 - y_{jk}^1), \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}. \tag{4.2.27}$$

By substituting (4.2.27) into (4.2.26) the constraint can be expressed as

$$q_{jk} \geq (1 - y_{jk}^2) y_{jk}^1 q_{\text{gc}}^1 + (1 - y_{jk}^2)(1 - y_{jk}^1) q_{\text{gc}}^2, \tag{4.2.28}$$

The resulting optimization problem ($\text{P}_{\text{gc,ncvx}}$) is given as

$$Z = \max\ G_{\mathrm{p}} \sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{K}\setminus K} \left(1 - 0.05 y^1_{jk}\right) q_{jk}\, \Delta k, \qquad (\mathrm{P}_{\mathrm{gc,bncvx}})$$

s.t.

$$\sum_{j\in\mathcal{J}} q_{jk} y^1_{jk} \le q^{\mathrm{up}}_{\mathrm{tot}}, \qquad \forall k \in \mathcal{K}$$

$$\sum_{j\in\mathcal{J}} q_{jk} y^1_{jk} \ge q^{\mathrm{low}}_{\mathrm{tot}}, \qquad \forall k \in \mathcal{K}$$

$$A_j m_{jk+1} = m_{jk} + B_j q_{jk+1}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K} \setminus K$$

$$m_{j0} = m_{j,\mathrm{init}}, \qquad \forall j \in \mathcal{J}$$

$$q_{jk} \le \beta\left(m_{jk1} - m_{\mathrm{wf},jk}\right) + M_1 y^2_{jk}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$q_{jk} \ge \beta\left(m_{jk1} - m_{\mathrm{wf},jk}\right) - M'_1 y^2_{jk}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$m_{\mathrm{wf},jk} = \alpha_1 \bar{p}_{jk} + \alpha_2, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$\bar{p}_{jk} = e^{\mathrm{S}}\left(p^2_{\mathrm{t},jk} + \frac{1}{C^2_{\mathrm{t}}} q^2_{jk}\right), \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$q_{jk} \le (1 - y^2_{jk}) q^{\max}_j, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$q_{jk} \ge (1 - y^2_{jk})(1 - y^1_{jk}) q^2_{\mathrm{gc}} + (1 - y^2_{jk}) y^1_{jk} q^1_{\mathrm{gc}}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$p_{\mathrm{t},jk} \ge p^{\mathrm{Line1}} y^1_{jk} + p^{\mathrm{Line2}}(1 - y^1_{jk}), \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$g_{\mathrm{s},jk}(y^1, y^2) \ge 0, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}'.$$

4.3 Alternative Nonconvex Formulations

The problem ($\mathrm{P}_{\mathrm{switch,bncvx}}$) is nonconvex due to the nonlinear equality constraint (4.1.5) and the bilinear term $q_{jk}y^1_{jk}$ which appears in the objective (4.1.1) and constraints (4.1.9) and (4.1.10). Additionally, the problem ($\mathrm{P}_{\mathrm{gc,bncvx}}$) has another bilinear term of the binary variables $y^1_{jk}y^2_{jk}$ from the critical gas rate constraint (4.2.28). In this section reformulations of the nonconvex bilinear terms are presented to yield a reformulated nonconvex MINLP in which the only source of nonconvexity stems from the nonlinear equality constraint (4.1.5).

The bilinear terms in the objective function (4.1.1) and the constraints (4.1.9),(4.1.10) are nonconvex terms since they involve the product of a continuous variable with a binary variable. These terms can be reformulated by using a Glover reformulation [44]. Let $v_{jk} \in \mathbb{R}_+$, the bilinear product in

(4.1.9) and (4.1.10) together with the maximum rate constraint (4.1.11) can be reformulated as

$$v_{jk} \leq y^1_{jk} q^{\max}_j, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K} \tag{4.3.1a}$$
$$v_{jk} \geq q_{jk} + q^{\max}_j (y^1_{jk} - 1), \qquad \forall j \in \mathcal{J}, k \in \mathcal{K} \tag{4.3.1b}$$
$$v_{jk} \leq q_{jk}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}. \tag{4.3.1c}$$

The reformulation of the bilinear product in the objective described above introduces $|\mathcal{J}||\mathcal{K}|$ additional variables and $3|\mathcal{J}||\mathcal{K}|$ additional constraints.

The first bilinear term in the constraint (4.2.28) can be reformulated by introducing the continuous variable $w^1_{jk} \in \mathbb{R}_+$ and the following constraints

$$w^1_{jk} \leq 1, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}, \tag{4.3.2a}$$
$$w^1_{jk} \leq y^1_{jk}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}, \tag{4.3.2b}$$
$$w^1_{jk} \leq 1 - y^2_{jk}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}, \tag{4.3.2c}$$
$$w^1_{jk} \geq y^1_{jk} - y^2_{jk}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}. \tag{4.3.2d}$$

Although the variable w^1_{jk} is defined to be continuous, it will only take on the values 1 and 0 due to the constraints in (4.3.2). The second bilinear term in the constraint (4.2.28) can be reformulated in an equivalent manner by introducing another continuous variable $w^2_{jk} \in \mathbb{R}_+$ and the following constraints

$$w^2_{jk} \leq 1, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}, \tag{4.3.3a}$$
$$w^2_{jk} \leq 1 - y^1_{jk}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}, \tag{4.3.3b}$$
$$w^2_{jk} \leq 1 - y^2_{jk}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}, \tag{4.3.3c}$$
$$w^2_{jk} \geq 1 - y^1_{jk} - y^2_{jk}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}. \tag{4.3.3d}$$

Finally, the non-convex constraint (4.2.28) can be reformulated as

$$q_{jk} \geq q^1_{\mathrm{gc}} w^1_{jk} + q^2_{\mathrm{gc}} w^2_{jk}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}. \tag{4.3.4}$$

The reformulation of the bilinear product in the critical gas rate constraint (4.2.28) requires $2|\mathcal{J}||\mathcal{K}|$ extra variables and $8|\mathcal{J}||\mathcal{K}|$ new constraints.

4.3.1 Nonconvex Problem Statements

Nonconvex Shut-in by Switching Formulation

$$Z = \max\ G_{\mathrm{p}} \sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{K}\setminus K} (q_{jk} - 0.05 v_{jk})\ \Delta k, \qquad (\mathrm{P}_{\mathrm{switch,ncvx}})$$

s.t.

$$\sum_{j\in\mathcal{J}} v_{jk} \le q_{\mathrm{tot}}^{\mathrm{up}}, \qquad \forall k \in \mathcal{K}$$

$$\sum_{j\in\mathcal{J}} v_{jk} \ge q_{\mathrm{tot}}^{\mathrm{low}}, \qquad \forall k \in \mathcal{K}$$

$$A_j m_{jk+1} = m_{jk} + B_j q_{jk+1}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K} \setminus K$$

$$m_{j0} = m_{j,\mathrm{init}}, \qquad \forall j \in \mathcal{J}$$

$$q_{jk} \le \beta\,(m_{jk1} - m_{\mathrm{wf},jk}) + M_1 y_{jk}^2, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$q_{jk} \ge \beta\,(m_{jk1} - m_{\mathrm{wf},jk}) - M_1' y_{jk}^2, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$m_{\mathrm{wf},jk} = \alpha_1 \bar{p}_{jk} + \alpha_2, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$\bar{p}_{jk} = e^{\mathrm{S}} \left(p_{\mathrm{t},jk}^2 + \frac{1}{C_{\mathrm{t}}^2} q_{jk}^2 \right), \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$q_{jk} \le (1 - y_{jk}^2) q_j^{\max}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$p_{\mathrm{t},jk} \ge p^{\mathrm{Line1}} y_{jk}^1 + p^{\mathrm{Line2}} (1 - y_{jk}^1), \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$v_{jk} \le y_{jk}^1 q_j^{\max}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$v_{jk} \ge q_{jk} + q_j^{\max} (y_{jk}^1 - 1), \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$v_{jk} \le q_{jk}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$g_{\mathrm{s},jk}(y^1, y^2) \ge 0, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}'.$$

Nonconvex Shut-in by Switching with a Critical Gas Rate Formulation

$$Z = \max\ G_{\mathrm{p}} \sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{K}\setminus K} \left(q_{jk} - 0.05 v_{jk}\right) \Delta k, \qquad (\mathrm{P}_{\mathrm{gc,ncvx}})$$

s.t.

$$\sum_{j\in\mathcal{J}} v_{jk} \le q_{\mathrm{tot}}^{\mathrm{up}}, \qquad \forall k \in \mathcal{K}$$

$$\sum_{j\in\mathcal{J}} v_{jk} \ge q_{\mathrm{tot}}^{\mathrm{low}}, \qquad \forall k \in \mathcal{K}$$

$$A_j m_{jk+1} = m_{jk} + B_j q_{jk+1}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K} \setminus K$$

$$m_{j0} = m_{j,\mathrm{init}}, \qquad \forall j \in \mathcal{J}$$

$$q_{jk} \le \beta\left(m_{jk1} - m_{\mathrm{wf},jk}\right) + M_1 y_{jk}^2, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$q_{jk} \ge \beta\left(m_{jk1} - m_{\mathrm{wf},jk}\right) - M_1' y_{jk}^2, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$m_{\mathrm{wf},jk} = \alpha_1 \bar{p}_{jk} + \alpha_2, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$\bar{p}_{jk} = e^{\mathrm{S}}\left(p_{\mathrm{t},jk}^2 + \frac{1}{C_{\mathrm{t}}^2} q_{jk}^2\right), \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$q_{jk} \le (1 - y_{jk}^2) q_j^{\max}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$q_{jk} \ge q_{\mathrm{gc}}^1 w_{jk}^1 + q_{\mathrm{gc}}^2 w_{jk}^2, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$p_{\mathrm{t},jk} \ge p^{\mathrm{Line1}} y_{jk}^1 + p^{\mathrm{Line2}}(1 - y_{jk}^1), \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$v_{jk} \le y_{jk}^1 q_j^{\max}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$v_{jk} \ge q_{jk} + q_j^{\max}(y_{jk}^1 - 1), \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$v_{jk} \le q_{jk}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$g_{\mathrm{s},jk}(y^1, y^2) \ge 0, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}'$$

$$w_{jk}^1 \le 1, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$w_{jk}^1 \le y_{jk}^1, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$w_{jk}^1 \le 1 - y_{jk}^2, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$w_{jk}^1 \ge y_{jk}^1 - y_{jk}^2, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$w_{jk}^2 \le 1, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$w_{jk}^2 \le 1 - y_{jk}^1, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$w_{jk}^2 \le 1 - y_{jk}^2, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$w_{jk}^2 \ge 1 - y_{jk}^1 - y_{jk}^2, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}.$$

4.4 Convex MINLP Relaxation

The quadratic equality constraint (4.1.5) renders the optimization problems ($P_{switch,ncvx}$) and ($P_{gc,ncvx}$) to be nonconvex MINLPs. In [55] it was shown that a MILP formulation for a similar problem in which this constraint is not included was more robust with regards to solution time and convergence than the MINLP formulation with the constraint included. While the linearized model has the benefit of being able to take advantage of sophisticated MILP solvers, it does so at the cost of a diminished accuracy. The aim of including the nonlinear constraint in the current formulation is to obtain a better approximation of the physical process. In this section, convex MINLP relaxations of the problems ($P_{switch,ncvx}$) and ($P_{gc,ncvx}$) which will be used in later chapters are presented. Additional convex relaxations are given in Appendix D.

4.4.1 Constraint Relaxation[3]

Under certain conditions it is possible to relax an equality constraint involving a convex function if certain criteria hold. Consider the following nonconvex optimization problem

$$\begin{aligned} z = \min_x \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \leq 0, \quad i = 1, \ldots, p, \\ & h(x) = 0, \end{aligned} \tag{4.4.1}$$

where $f, g_i : \mathbb{R}^n \to \mathbb{R}, i = 1, \ldots p$ are convex functions and the equality constraint function $h : \mathbb{R}^n \to \mathbb{R}$ is also a convex function, but not affine. A convex relaxation to the problem (4.4.1) can be obtained by relaxing the equality constraint to an inequality constraint

$$\begin{aligned} z = \min_x \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \leq 0, \quad i = 1, \ldots, p, \\ & h(x) \leq 0. \end{aligned} \tag{4.4.2}$$

If the relaxed constraint in (4.4.2) is active at an optimal solution x^* of the problem (4.4.2), i.e. $h(x^*) = 0$, then the optimal solution of (4.4.1) is equivalent to (4.4.2). This will be the case if for an index r, the following criteria hold:

[3]The problem considered in this section is posed in exercise problem 4.6 in [23]

1. f is strictly decreasing in x_r
2. $g_1 \dots g_p$ are nonincreasing in x_r
3. h is strictly increasing in x_r.

Proof. Suppose at an optimal solution x^* of (4.4.2), the equality constraint is inactive, i.e. $h(x^*) < 0$. Since the constraints $g_1 \dots g_p$ are nonincreasing in x_r, the value of x_r^* can be increased without causing a constraint violation. Furthermore, the objective value can be reduced if x_r^* is increased, this follows from the strictly decreasing property. Therefore a solution x^* at which the constraint $h(x) \leq 0$ is inactive, is not an optimal solution. □

It should be noted that this property does not hold for feasible solutions which are not optimal.

In the case when there are other equality constraints than the constraint $h(x) = 0$, the condition stated above will hold if these equality constraint functions do not contain the variable x_r . This follows from the observation that an equality constraint $\tilde{h}(x) = 0$ can be expressed as two inequality constraints

$$\begin{aligned} \tilde{h}(x) &\leq 0 \\ -\tilde{h}(x) &\leq 0, \end{aligned}$$

and that these inequality constraints are only nonincreasing in x_r if $\frac{\partial \tilde{h}(x)}{\partial x_r} = 0$.

If the equality constraint (4.1.5) is relaxed to an inequality constraint, the resulting optimization problem is a convex MINLP. This follows from the fact that the inequality is the sum of a positive definite quadratic form subtracted by a linear term

$$e^S \left(p_{\mathrm{t},jk}^2 + \frac{1}{C_{\mathrm{t}}^2} \tilde{q}_{jk}^2 \right) - \bar{p}_{jk} \leq 0, \ \forall j \in \mathcal{J}, k \in \mathcal{K}. \tag{4.4.3}$$

The constraint function in (4.4.3) is strictly increasing in q_{jk}, and the objective function (4.1.13a) in a minimization form is strictly decreasing in q_{jk} except for the final time step $k = K$. In order to strengthen the convex relaxation the final time step is included in the optimization formulation, yielding the objective

$$\min \ -G_{\mathrm{p}} \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} \left(q_{jk} - 0.05 v_{jk} \right) \Delta k, \tag{4.4.4}$$

and then the objective value at the final time step K over all wells $\mathcal{J}$ can be subtracted after the optimization routine terminates.

Although the objective function and the relaxed equality constraint satisfy the criteria stated above for q_{jk}, the inequality constraint (4.1.12a) and (4.1.10) are nondecreasing in q_{jk}. Additionally, the q_{jk} variables appear in the equality constraint (4.1.4). Consequently, the optimization problem resulting from relaxing the equality constraint (4.1.5) can not be certified to share the same optimal solution as the nonconvex MINLP using these criteria. However, if the parameters $q_j^{\max}$ and $q_{\text{tot}}^{\text{up}}$ are sufficiently large such that the inequalities will never be active, then the nondecreasing property of the inequality constraints is not a necessary condition.

For this problem the criteria under which a convex relaxation has the same optimal value as the nonconvex problem, are not satisfied by the system of constraints in (4.1.13). There may, however, be other conditions which can establish an equivalence between the original nonconvex MINLP and the relaxed convex MINLP.

4.4.2 Formulation of Convex MINLP Relaxations

By replacing the nonlinear equality constraint (4.1.5) with the relaxed nonlinear inequality constraint (4.4.3), the nonconvex MINLPs ($\mathrm{P}_{\mathrm{switch,ncvx}}$) and ($\mathrm{P}_{\mathrm{gc,ncvx}}$) can be expressed as

$$Z = \max\ G_{\mathrm{p}} \sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{K}} \left(q_{jk} - 0.05 v_{jk}\right)\ \Delta k, \qquad (\mathrm{P}_{\mathrm{switch,cvx}})$$

s.t.

$$\sum_{j\in\mathcal{J}} v_{jk} \le q_{\mathrm{tot}}^{\mathrm{up}}, \qquad \forall k \in \mathcal{K}$$

$$\sum_{j\in\mathcal{J}} v_{jk} \ge q_{\mathrm{tot}}^{\mathrm{low}}, \qquad \forall k \in \mathcal{K}$$

$$A_j m_{jk+1} = m_{jk} + B_j q_{jk+1}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K} \setminus K$$

$$m_{j0} = m_{j,\mathrm{init}}, \qquad \forall j \in \mathcal{J}$$

$$q_{jk} \le \beta\left(m_{jk1} - m_{\mathrm{wf},jk}\right) + M_1 y_{jk}^2, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$q_{jk} \ge \beta\left(m_{jk1} - m_{\mathrm{wf},jk}\right) - M_1' y_{jk}^2, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$m_{\mathrm{wf},jk} = \alpha_1 \bar{p}_{jk} + \alpha_2, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$\bar{p}_{jk} \ge e^{\mathrm{S}}\left(p_{\mathrm{t},jk}^2 + \frac{1}{C_{\mathrm{t}}^2} q_{jk}^2\right), \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$q_{jk} \le (1 - y_{jk}^2) q_j^{\max}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$p_{\mathrm{t},jk} \ge p^{\mathrm{Line1}} y_{jk}^1 + p^{\mathrm{Line2}}(1 - y_{jk}^1), \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$v_{jk} \le y_{jk}^1 q_j^{\max}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$v_{jk} \ge q_{jk} + q_j^{\max}(y_{jk}^1 - 1), \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$v_{jk} \le q_{jk}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$g_{\mathrm{s},jk}(y^1, y^2) \ge 0, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}'.$$

$$Z = \max\ G_{\mathrm{p}} \sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{K}} \left(q_{jk} - 0.05 v_{jk}\right)\ \Delta k, \qquad (\mathrm{P}_{\mathrm{gc,cvx}})$$

s.t.

$$\sum_{j\in\mathcal{J}} v_{jk} \le q_{\mathrm{tot}}^{\mathrm{up}}, \quad \sum_{j\in\mathcal{J}} v_{jk} \ge q_{\mathrm{tot}}^{\mathrm{low}}, \qquad \forall k \in \mathcal{K}$$

$$A_j m_{jk+1} = m_{jk} + B_j q_{jk+1}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K} \setminus K$$

$$m_{j0} = m_{j,\mathrm{init}}, \qquad \forall j \in \mathcal{J}$$

$$q_{jk} \le \beta\left(m_{jk1} - m_{\mathrm{wf},jk}\right) + M_1 y_{jk}^2, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$q_{jk} \ge \beta\left(m_{jk1} - m_{\mathrm{wf},jk}\right) - M_1' y_{jk}^2, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$m_{\mathrm{wf},jk} = \alpha_1 \bar{p}_{jk} + \alpha_2, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$\bar{p}_{jk} \ge e^{\mathrm{S}}\left(p_{\mathrm{t},jk}^2 + \frac{1}{C_{\mathrm{t}}^2} q_{jk}^2\right), \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$q_{jk} \le (1 - y_{jk}^2) q_j^{\max}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$q_{jk} \ge q_{\mathrm{gc}}^1 w_{jk}^1 + q_{\mathrm{gc}}^2 w_{jk}^2, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$p_{\mathrm{t},jk} \ge p^{\mathrm{Line1}} y_{jk}^1 + p^{\mathrm{Line2}}(1 - y_{jk}^1), \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$v_{jk} \le y_{jk}^1 q_j^{\max}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$v_{jk} \ge q_{jk} + q_j^{\max}(y_{jk}^1 - 1), \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$v_{jk} \le q_{jk}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$g_{\mathrm{s},jk}(y^1, y^2) \ge 0, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}'$$

$$w_{jk}^1 \le 1, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$w_{jk}^1 \le y_{jk}^1, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$w_{jk}^1 \le 1 - y_{jk}^2, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$w_{jk}^1 \ge y_{jk}^1 - y_{jk}^2, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$w_{jk}^2 \le 1, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$w_{jk}^2 \le 1 - y_{jk}^1, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$w_{jk}^2 \le 1 - y_{jk}^2, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

$$w_{jk}^2 \ge 1 - y_{jk}^1 - y_{jk}^2, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

Chapter 5

Heuristics for the Shale Gas Production Optimization Problem

In this chapter a heuristic for finding feasible solutions of the shale gas production optimization problems given in Chapter 4 is presented. The basic concept of the new heuristic is to use linearizations to solve MILP approximations of the original MINLP problems and then to fix the integer variables to the solution obtained from solving the MILP. After the integer variables are fixed, a NLP relaxation is solved. The idea for the new approach is inspired by the *undercover*(UC) heuristic, which is reviewed in Section 5.1. In Section 5.2 linearization procedures applied to the MINLPs from the previous chapter are presented. The new heuristic method, which for now is called "*reverse-undercover*"(RUC), is described in Section 5.3.

5.1 The Undercover Heuristic

A relatively new heuristic which solves a mixed-integer linear subproblem of a given MINLP is proposed in [14]. The underlying concept is to find a *set-cover*(see chapter 1.1 in [68]) of a MINLP in order to identify the smallest subset of variables that must be fixed such that the (nonlinear) functions $f(x, y)$ and $g_k(x, y),\ k = 1 \dots p$ in ($\mathrm{P}_{\mathrm{MINLP}}$) are linear. A *cover* of a nonlinear function $g_k(x, y)$ is defined as follows

Definition 3. *Let the set* $\mathcal{C} \subseteq \{1, \dots, n + q\}$ *be a set of variable indices of* ($\mathrm{P}_{\mathrm{MINLP}}$). *The set* $\mathcal{C}$ *is called a* cover *of a function* $g_k(x, y),\ k \in \{1 \dots p\}$, *if*

and only if for all $x^* \in X$ *and* $y^* \in Y$ *the set*

$$\left\{((x,y), g_k(x,y)) : x \in X, y \in Y, x_i = x_i^*, y_j = y_j^* \text{ for all } (i,j) \in \mathcal{C}\right\}$$

is affine for all $x^* \in X$ and $y^* \in Y$. If the set $\mathcal{C}$ is a cover of all functions $f, g_1, \ldots, g_p$, then the set $\mathcal{C}$ is said to be a cover of the problem ($\mathrm{P}_{\mathrm{MINLP}}$).

A generic algorithm statement of the undercover heuristic is given in Algorithm 4.

Algorithm 4 Generic undercover heuristic.

begin

1: compute a solution (x^*, y^*) of an approximation or relaxation of ($\mathrm{P}_{\mathrm{MINLP}}$)
2: round y^*
3: determine a cover $\mathcal{C}$ of ($\mathrm{P}_{\mathrm{MINLP}}$)
4: solve the sub-MILP obtained by fixing $x_i = x_i^*, y_j = y_j^*$ for all $(i,j) \in \mathcal{C}$

end

The undercover heuristic was applied to all of the problems ($\mathrm{P}_{\mathrm{switch,bncvx}}$), ($\mathrm{P}_{\mathrm{gc,bncvx}}$), ($\mathrm{P}_{\mathrm{switch,ncvx}}$), ($\mathrm{P}_{\mathrm{gc,ncvx}}$), ($\mathrm{P}_{\mathrm{switch,cvx}}$) and ($\mathrm{P}_{\mathrm{gc,cvx}}$) from Chapter 4, by solving the NLP relaxation and fixing the variables q_{jk} and $p_{\mathrm{t},jk}$ $\forall j \in \mathcal{J}, k \in \mathcal{K}$. In all of the problem formulations, the sub-MILP was reported to be infeasible by the MILP solver `CBC`. The main reason for this failure is that the NLP relaxation computes a solution at which all the binary variables y_{jk}^1 have values in the range $[0.4, 0.8]$, consequently the continuous variables q_{jk} and $p_{\mathrm{t},jk}$ take on values which are not possible to attain if the binary restriction is enforced. Furthermore, fixing the continuous variables q_{jk} and $p_{\mathrm{t},jk}$ prohibits the sub-MILP in looking for solutions where switching and shut-in occurs. This is a major drawback, since switching and shut-in is essential to obtaining good objective values for the problems that are being considered.

5.2 Linearizing the Model

In this section, two methods to linearize the nonlinear function in Equation (4.4.3) are presented. To approximate the nonlinear terms, the first method exploits the convexity property of the quadratic terms in (4.4.3) and uses outer-approximations to obtain a linear approximation, while the second method uses *piecewise linear functions*.

5.2.1 Outer-Approximation

A linear approximation of the quadratic terms in Equation (4.4.3) can be obtained by substituting the first-order term in the *Taylor series* expansion, which for a general continuous function $g(x)$ is defined as

$$g(x) \approx g(\hat{x}) + \nabla g(\hat{x})^T (x - \hat{x}),$$

where $\hat{x} \in \mathbf{dom}\ g$. Since both of the quadratic functions are positive definite, and hence convex, it follows that the first-order Taylor approximations are *global underestimators* of the functions [23]. This implies that these approximations do not remove any part of the feasible space. If an infinite number of these approximations are included to replace the nonlinear functions, then the original MINLP can be reformulated as a *semi-infinite* MILP [41]. Although it is possible to obtain a more accurate representation of the nonlinear functions by adding several outer-approximations, it comes at the cost of an increased complexity due to the additional constraints. More specifically, each outer-approximation of the quadratic terms in (4.4.3) requires an additional $|\mathcal{J}||\mathcal{K}|$ linear constraints. In order to maintain a simple approximation of the original MINLP, the linearization obtained by outer-approximation will for the remainder of this thesis use 3 carefully selected points at which the quadratic terms are linearized.

Experiments on the shale gas production optimization problems revealed that the flow rate q_{jk} varies over its full range $[0, q_j^{\max}]$, therefore a reasonable approximation of the flow rate quadratic term can be found by performing a first order Taylor series expansion at the midpoint of this range, i.e. $\hat{q}_{jk} := \frac{q_j^{\max}}{2}$. Let the function $g_q(q_{jk}) := \frac{e^S}{C_{\mathrm{t}}^2} q_{jk}^2$ be approximated as

$$\hat{g}_q^{\mathrm{OA}}(q_{jk}) = \frac{e^S}{C_{\mathrm{t}}^2}\hat{q}_{jk}^2 + 2\frac{e^S}{C_{\mathrm{t}}^2}\hat{q}_{jk}(q_{jk} - \hat{q}_{jk}). \tag{5.2.1}$$

Since the function $g_q(q_{jk})$ is convex, the outer-approximation $\hat{g}_q^{\mathrm{OA}}(q_{jk})$ is a global underestimator, which means that for the interval $[0, q_j^{\max}]$, the following holds

$$\hat{g}_q(q_{jk}) \geq \hat{g}_q^{\mathrm{OA}}(q_{jk}), \text{ for all } q_{jk} \in [0, q_j^{\max}]. \tag{5.2.2}$$

Similar experiments on the same problems showed that for a majority of the time steps the tubing pressure $p_{\mathrm{t},jk}$ maintains a value of p^{Line1} or p^{Line2}. On the grounds of this observation, the two line pressure values are selected as points at which the quadratic term of the tubing pressure is linearized. Let

$g_{p_t}(p_{t,jk}) := e^S p_{t,jk}^2$, the two outer-approximations $\hat{g}_{p_t}^{OA_1}(p_{t,jk})$ and $\hat{g}_{p_t}^{OA_2}(p_{t,jk})$ are given by

$$\hat{g}_{p_t}^{OA_1}(p_{t,jk}) = e^S p^{\text{Line1}^2} + 2e^S p^{\text{Line1}}(p_{t,jk} - p^{\text{Line1}}), \tag{5.2.3a}$$
$$\hat{g}_{p_t}^{OA_2}(p_{t,jk}) = e^S p^{\text{Line2}^2} + 2e^S p^{\text{Line2}}(p_{t,jk} - p^{\text{Line2}}), \tag{5.2.3b}$$
$$\tag{5.2.3c}$$

where $\hat{g}_{p_t}^{OA_1}(p_{t,jk})$ and $\hat{g}_{p_t}^{OA_2}(p_{t,jk})$ are global underestimators of $g_{p_t}(p_{t,jk})$.

By replacing the nonlinear inequality constraint (4.4.3) from the problems ($P_{\text{switch,cvx}}$) and ($P_{\text{gc,cvx}}$) with the constraints

$$\bar{p}_{jk} \geq \hat{g}_q^{OA}(q_{jk}) + \hat{g}_{p_t}^{OA_1}(p_{t,jk}), \quad \forall j \in \mathcal{J}, k \in \mathcal{K} \tag{5.2.4a}$$
$$\bar{p}_{jk} \geq \hat{g}_q^{OA}(q_{jk}) + \hat{g}_{p_t}^{OA_2}(p_{t,jk}), \quad \forall j \in \mathcal{J}, k \in \mathcal{K} \tag{5.2.4b}$$
$$\tag{5.2.4c}$$

a MILP approximation is obtained. The constraints (5.2.4) are left as inequalities due to the underestimating property.

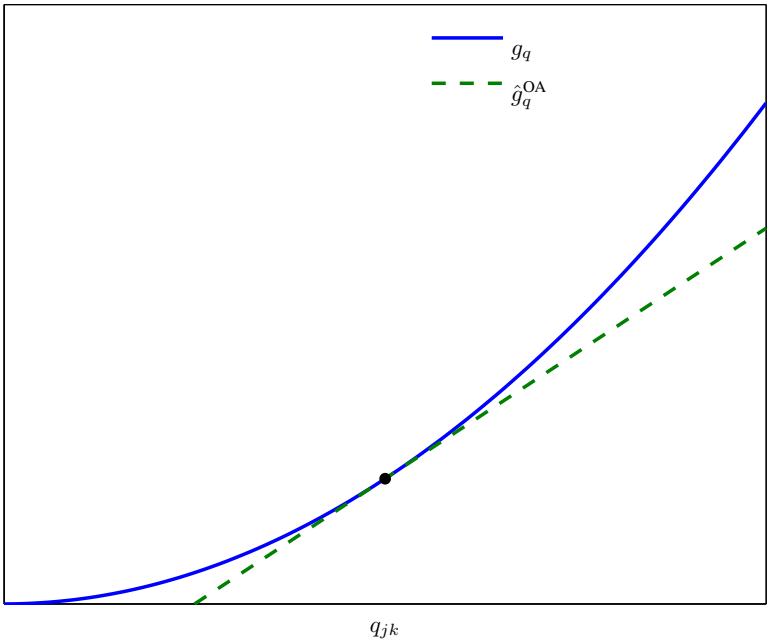

Figure 5.1: Outer-Approximation of $g_q(q_{jk})$ on $q_{jk} \in [0, 0.463]$.

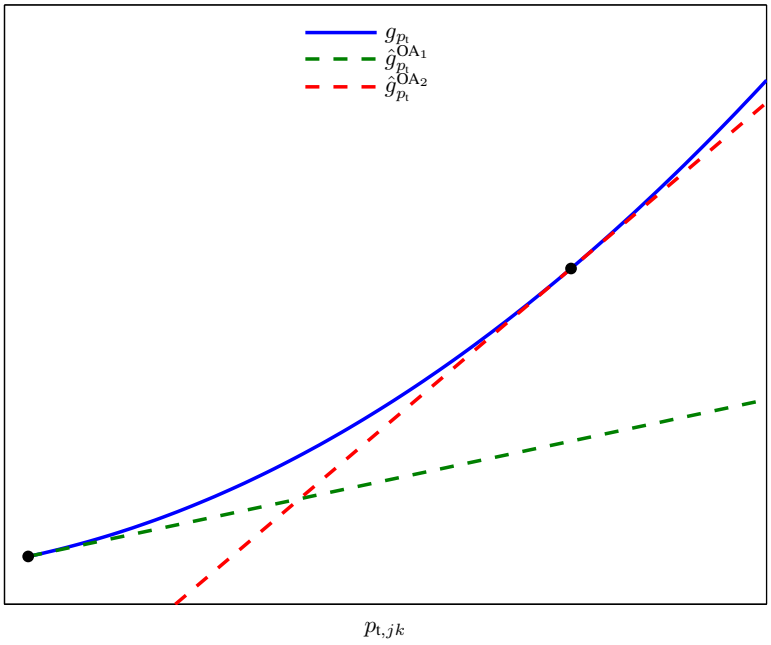

Figure 5.2: Outer-Approximation of $g_{p_t}(p_{t,jk})$ on $p_{t,jk} \in [6.9, 35]$.

5.2.2 Piecewise Linearization

A continuous nonlinear *separable function* is a function that can be written as

$$g(x_1, \ldots, x_k) = \sum_{j=1}^{k} g_j(x_j), \tag{5.2.5}$$

where each $g_j : \mathbb{R} \to \mathbb{R},\ j = 1, \ldots, k$ is a continuous univariate function. An approximation of $g(x)$ in Equation (5.2.5) can be computed by solving for piecewise linear functions of each $g_j(x_j),\ j = 1, \ldots, k$.

A function $g_j(x_j)$ defined on an interval $[a, b] \subseteq \mathbb{R}$ can be approximated by selecting N *breakpoints* on which the function is evaluated, where $a = x_j^1 \leq x_j^2 \leq \ldots \leq x_j^N = b$ with function values $g_j^i = g_j(x_j^i)$ for $i = 1, \ldots, N$. For any given value of x_j, say $\bar{x}_j$, with $x_j^i \leq \bar{x}_j \leq x_j^{i+1}$, the function value of g_j is approximated by a convex combination of $g_j(x_j^i)$ and $g_j(x_j^{i+1})$. Let $\theta \in [0, 1]$ be the unique value such that

$$\bar{x}_j = \theta x_j^i + (1 - \theta) x_j^{i+1},$$

then an approximate value of $g_j(\bar{x}_j)$ is:

$$\hat{g}_j(\bar{x}_j) = \theta g_j(x_j^i) + (1-\theta) g_j(x_j^{i+1}).$$

The method described above can be used within a MILP solver by introducing additional variables and constraints to force the variable x_j to lie within consecutive breakpoints(or at a single breakpoint). For each breakpoint $i \in \{1, \ldots, N\}$, a continuous variable $\theta^i \in [0,1]$ is used to construct convex combinations of the breakpoints. A binary variable b^i is included to make sure that at most two θ^i's are positive and that if θ^j and θ^k are positive, then $k = j-1$ or $k = j+1$. An approximate value of $g_j(x_j)$ can then be found by the following system of constraints:

$$\sum_{i=1}^{N-1} b^i = 1 \tag{5.2.6a}$$

$$\theta^i \leq b^{i-1} + b^i \quad i = 1, \ldots N \tag{5.2.6b}$$

$$\sum_{i=1}^{N} \theta^i = 1 \tag{5.2.6c}$$

$$x_j = \sum_{i=1}^{N} \theta^i x_j^i \tag{5.2.6d}$$

$$\hat{g}_j(x_j) = \sum_{i=1}^{N} \theta^i g_j(x_j^i) \tag{5.2.6e}$$

$$b^0 = b^N = 0 \tag{5.2.6f}$$

$$\tag{5.2.6g}$$

The constraints (5.2.6) can be reduced within modern MILP solvers by defining the variables θ^i to belong to a *special ordered set of type 2*(SOS2) [10]. If a SOS2 formulation is used, then it is not necessary to include the binary variables $b^i,\ i = 0, \ldots, N$, and it suffices to only include the constraints in Equations (5.2.6c–5.2.6e) in the formulation. An added benefit of explicitly defining variables to belong to SOS2, is that modern MILP solvers have special purpose branching rules for variables that are SOS2 which improve the performance of enumerative methods for solving MILPs [9].

The piecewise linearization scheme described above can be directly applied to the two quadratic terms in Equation (4.4.3). A piecewise linear approximation $\hat{g}_q^{\mathrm{PWL}}(q_{jk})$ of $g_q(q_{jk})$ can be found by computing five equidistant breakpoints on the interval $[0, q_j^{\max}]$. An illustration of what this may look like is given in Figure 5.3. Similarly, a piecewise linear approximation $\hat{g}_{p_\mathrm{t}}^{\mathrm{PWL}}(p_{\mathrm{t},jk})$ of

$g_{p_t}(p_{t,jk})$ can be obtained by selecting breakpoints on the interval $[p^{\text{Line1}}, p_t^{\text{ub}}]$, where p^{Line1} and p^{Line2} are included in the set of breakpoints and p_t^{ub} is a specified upper bound for the variables $p_{t,jk}$. Experiments on the MINLP problems from the previous chapter showed that $p_{t,jk}$ does not increase beyond a value of approximately $\frac{5}{2}p^{\text{Line2}}$, hence an upper bound can be specified to be at or above this value. An example of what this approximation looks like for five breakpoints is shown in Figure 5.4.

A MILP approximation of the problems ($P_{\text{switch,cvx}}$) and ($P_{\text{gc,cvx}}$) can be obtained by adding a set of constraints as shown in (5.2.6) for each of the functions $\hat{g}_q^{\text{PWL}}$ and $\hat{g}_{p_t}^{\text{PWL}}$ and replacing the constraint (4.4.3) with

$$\bar{p}_{jk} \geq \hat{g}_q^{\text{PWL}}(q_{jk}) + \hat{g}_{p_t}^{\text{PWL}}(p_{t,jk}), \quad \forall j \in \mathcal{J}, k \in \mathcal{K}. \tag{5.2.7}$$

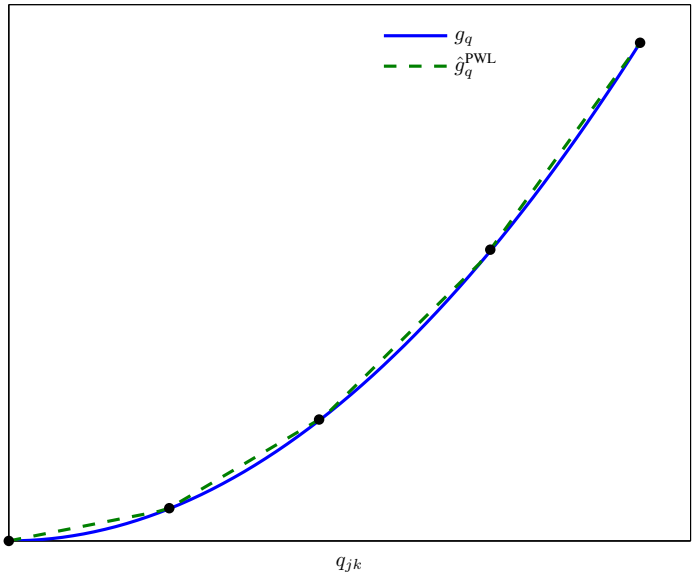

Figure 5.3: Piecewise linearization of $g_q(q_{jk})$ on $q_{jk} \in [0, 0.463]$.

5.3 Reversing the Undercover Heuristic

The failure of the UC heuristic applied to the shale gas production optimization problem raised the question of whether or not it was possible to reverse the order in which the subproblems in the UC algorithm are solved. The

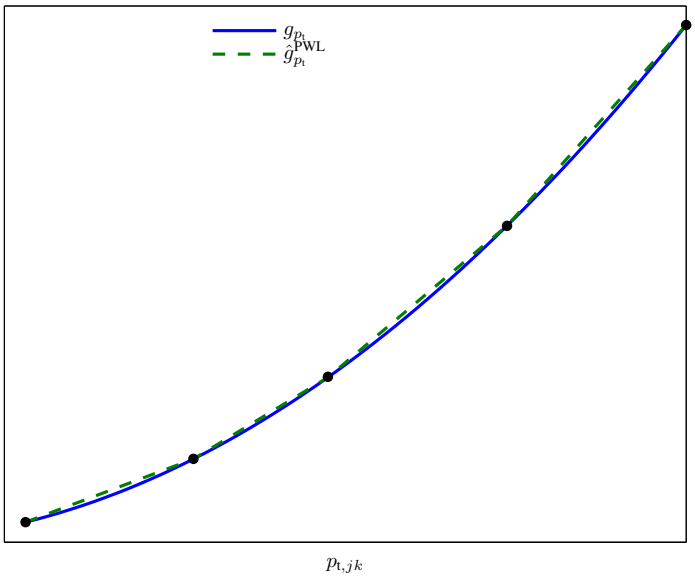

Figure 5.4: Piecewise linearization of $g_{p_t}(p_{t,jk})$ on $p_{t,jk} \in [6.9, 35]$.

result of investigating this topic yielded a method which for now is called the reverse-undercover heuristic. In the RUC heuristic a MILP approximation of an MINLP is created and solved to feasibility. Once a feasible solution to the approximated problem is found, the integer variables are fixed and the NLP-subproblem of the original MINLP is solved. Neither the UC -or RUC heuristic guarantee that the solution obtained from solving the combination of subproblems will return a feasible solution of the original MINLP. However, if the solution from the last subproblem which is solved by either the UC -or RUC heuristic is feasible, then the solution is also feasible for the original MINLP. Although it is possible to solve the MILP approximation to optimality, experimental results on the problems from Chapter 4 revealed that this was in general time consuming and that to maintain a fast solution time it was beneficial to terminate after finding the first feasible solution. A generic algorithm statement of the RUC heuristic is given in Algorithm 5.

Algorithm 5 Generic reverse-undercover heuristic.

begin

1: solve a MILP approximation of the original MINLP to feasibility
2: fix y^*
3: solve the sub-NLP of the MINLP with fixed integer variables

end

The details of how to perform step 1 in Algorithm 5 are left open in order to allow for modifications. For a MINLP where the nonlinear functions are separable, a MILP approximation can be obtained by applying the piecewise linearization method described in the previous section. Even if the nonlinear functions are not separable, it is still possible to obtain piecewise linear approximations by employing techniques described in [5, 61, 66]. In the case of the shale gas production optimization problem, both the MILPs obtained by outer-approximation and piecewise linear functions can be used as a substitute for the approximate MILPs. In Chapter 7 both of these MILP approximations are used within a RUC-heuristic and compared on set of test problems from Chapter 4.

The RUC-heuristic is implemented at the modelling level, which means that the burden of using this heuristic is placed with the modeller. A flowchart of the RUC-heuristic with the corresponding tools to perform each step in Algorithm 5 is shown in Figure 5.5.

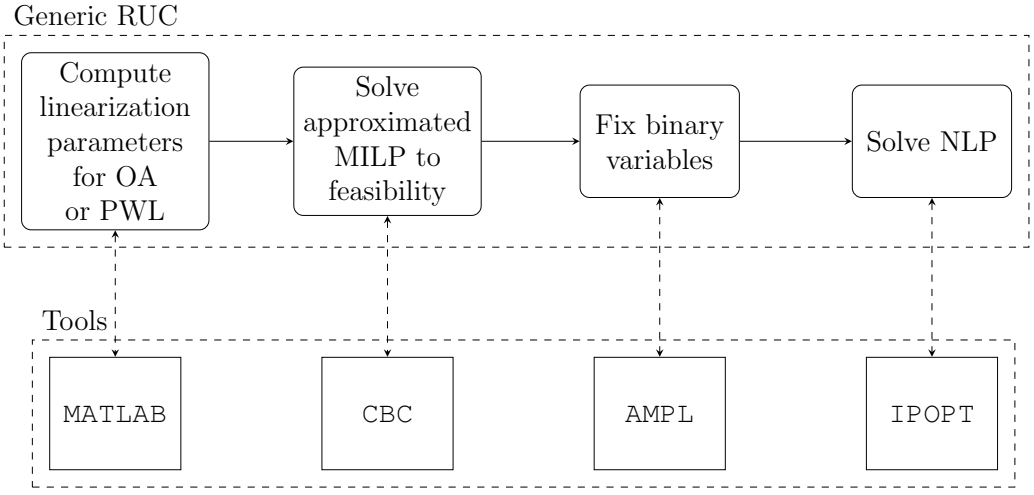

Figure 5.5: Flowchart of the RUC-heuristic.

There are a few similarities between the GOA, LP/NLP-BB and the RUC

heuristic. For instance, they all solve an approximate MILP and then a NLP subproblem with the integer variables fixed. The distinction between these methods lies mainly with the approach in which the MILP approximations are generated. Both GOA and LP/NLP-BB initially solve a NLP and then instantiate a MILP by substituting the first-order Taylor approximation of the nonlinear functions at the NLP solution. The RUC heuristic does not solve an NLP initially, instead it creates the MILP approximation by for instance generating outer-approximations at a set of given points or by using piecewise linear approximations. Moreover, GOA solves the MILPs to optimality whereas RUC only solves the MILPs to feasibility.

Chapter 6

Implementation

In this chapter the parameter values and details of a variety of test problems on which computational studies are performed in Chapter 7 are presented. All of the models are implemented in AMPL. The computational study is divided into two sets of problems, where the first set is used as a benchmark study and the second set is a large production optimization problem bearing a closer resemblance to a real case study.

6.1 First Set of Problems

The first set of test problems serves as a benchmark study to observe the impact of using the different formulations of the problems given in Chapter 4, and to determine the efficacy of using heuristics to solve for feasible solutions of these problems.

6.1.1 Problem Parameters

Parameter values which are used in all of the problems in the first set of test problems are given in Table 6.1. The big-M parameters M_1 and M_1' for relaxing the well-inflow equation (4.1.4) were determined experimentally by evaluating the value of the expression $\beta\left(m_{jk1} - m_{\mathrm{wf},jk}\right)$ on a sample of problems. It was found that values of $M_1, M_1' \in [2, 10]$ gave similar results and were sufficiently large to yield differences between m_{jk1} and $m_{\mathrm{wf},jk}$.

Table 6.1: Universal parameters for first set of test problems.

Parameter	Value	Unit
$\lvert\mathcal{J}\rvert$	6	-
$\lvert\mathcal{K}\rvert$	29	-
Δk	86400	[s]
p^{Line1}	6.9	[bar]
$q_j^{\max}$	40003	[m^3/d]
β	3389.3	-
α_1	6.71244e-04	-
α_2	3.2611e-01	-
C_{t}	4.2627e-02	-
S	0.34867	-
M_1	3	-
M_1'	2	-
G_{p}	0.177	[\$/m^3]
q_{gc}^1	11197	[m^3/d]

Both the switching problem and the critical gas rate problem from the previous chapter will be used in the computational study. For the switching problems ($\text{P}_{\text{switch,bncvx}}$), ($\text{P}_{\text{switch,ncvx}}$) and ($\text{P}_{\text{switch,cvx}}$) the parameter values are given in Table 6.2, while the parameter values for the critical gas rate problems ($\text{P}_{\text{gc,bncvx}}$), ($\text{P}_{\text{gc,ncvx}}$) and ($\text{P}_{\text{gc,cvx}}$) can be found in Table 6.3.

Table 6.2: Parameters for switching problems in first set of test problems.

Case	p^{Line2}[bar]	$q_{\text{tot}}^{\text{low}}$[$m^3$/d]	$q_{\text{tot}}^{\text{up}}$[$m^3$/d]
Case1switch	15	34560	146880
Case2switch	15	51840	120960
Case3switch	15	60480	103680
Case4switch	27.56	34560	146880
Case5switch	27.56	51840	120960
Case6switch	27.56	60480	103680

Table 6.3: Parameters for critical gas rate problems in first set of test problems.

Case	p^{Line2}[bar]	$q_{\text{tot}}^{\text{low}}$[$m^3$/d]	$q_{\text{tot}}^{\text{up}}$[$m^3$/d]	q_{gc}^2[m^3/d]
Case1qgc	15	34560	146880	16494
Case2qgc	15	51840	120960	16494
Case3qgc	15	60480	103680	16494
Case4qgc	27.56	34560	146880	22317
Case5qgc	27.56	51840	120960	22317
Case6qgc	27.56	60480	103680	22317

6.1.2 Problem Information

Information on the number of constraints and variables of individual problems in the first set are given in Table 6.4.

Table 6.4: Problem information of MINLPs in first set of test problems.

Problem	Continuous variables	Binary variables	Constraints
($P_{switch,bncvx}$)	1566	348	2920
($P_{switch,ncvx}$)	1740	348	3280
($P_{switch,cvx}$)	1740	348	3280
($P_{gc,bncvx}$)	1566	348	3094
($P_{gc,ncvx}$)	2088	348	4846
($P_{gc,cvx}$)	2088	348	4846

The number of constraints and variables of the MILP-approximations of ($P_{switch,cvx}$) and ($P_{gc,cvx}$) are given in Table 6.5.

Table 6.5: Problem information of MILP approximations in first set of test problems.

Problem	Continuous variables	Binary variables	Constraints
($P_{switch,cvx}$)-OA[a]	1740	348	3616
($P_{gc,cvx}$)-OA	2088	348	5020
($P_{switch,cvx}$)-PWL[b]	3828	2088	6586
($P_{gc,cvx}$)-PWL	4176	2088	8152

[a] : outer-approximation.
[b] : piecewise linearization.

6.2 Second Set of Problems

The second set of test problems consists of problems with characteristics which make them much harder to solve compared to the first set of problems. Unlike the problems presented in the previous section, only variations of the nonconvex MINLP problem ($P_{gc,ncvx}$) are considered in the second set of problems. The reason for this choice is that the benchmark study in Chapter 7 showed that the heuristics perform quite well on this problem and that the solution satisfies the nonlinear equality constraint (4.1.5).

6.2.1 Problem Parameters

The problem parameters which apply to all of the formulations in the second set of test problems are given in Table 6.6.

Table 6.6: Universal parameters for second set of test problems.

Parameter	Value	Unit
$\|\mathcal{J}\|$	6	-
$\|\mathcal{K}\|$	42	-
Δk	259200	[s]
p^{Line1}	6.9	[bar]
p^{Line2}	27.56	[bar]
$q_j^{\max}$	339810	[m^3/3 days]
β	84.89	-
α_1	7.67957e-04	-
α_2	1.95313e-03	-
C_{t}	4.2627e-02	-
S	0.34867	-
M_1	3	-
M_1'	2	-
G_{p}	0.177	[\$/m^3]
q_{gc}^1	33592	[m^3/3 days]
q_{gc}^2	66951	[m^3/3 days]

The problems used in the second set are separated by the value of the initial pseudopressure m_{j0} and whether or not a minimum time limit on how long a well is to remain shut is included. The purpose of removing the minimum time requirement of how long a well should stay shut is to evaluate whether the optimization routine returns a solution for which a well stays shut over a longer period of time without a constraint which enforces this condition. Parameters for the individual problems in the second set are given in Table 6.7.

6.2.2 Problem Information

The number of variables and constraints of the two problem formulations are presented in Table 6.8.

The number of variables and constraints of the MILP approximations of the two problem formulations are presented in Table 6.8.

Table 6.7: Parameters for individual problems in second set of test problems.

Case	$q_{\text{tot}}^{\text{low}}$[m^3/3 days]	$q_{\text{tot}}^{\text{up}}$[m^3/3 days]
Case1qgcNSlow[a,c]	336960	622080
Case2qgcNSlow	362880	596160
Case3qgcNSlow	362880	570240
Case1qgcSlow[b]	336960	622080
Case2qgcSlow	362880	596160
Case3qgcSlow	362880	570240
Case1qgcNShigh[d]	336960	622080
Case2qgcNShigh	362880	596160
Case3qgcNShigh	362880	570240
Case1qgcShigh	336960	622080
Case2qgcShigh	362880	596160
Case3qgcShigh	362880	570240

[a] : no shut-in time constraint.
[b] : shut-in time constraint.
[c] : low initial pseudopressure.
[d] : high initial pseudopressure.

Table 6.8: Problem information of MINLPs in second set of test problems.

Problem	Continuous variables	Binary variables	Constraints
($\text{P}_{\text{gc,ncvx}}$)-NS[a]	3024	504	6816
($\text{P}_{\text{gc,ncvx}}$)-S[b]	3024	504	7056

[a] : no shut-in time constraint.
[b] : shut-in time constraint.

Table 6.9: Problem information of MILP approximations in second set of test problems.

Problem	Continuous variables	Binary variables	Constraints
($\text{P}_{\text{gc,ncvx}}$)-NS-OA[a,c]	3024	504	7068
($\text{P}_{\text{gc,ncvx}}$)-S-OA[d]	3024	504	7308
($\text{P}_{\text{gc,ncvx}}$)-NS-PWL[b]	6048	3024	11604
($\text{P}_{\text{gc,ncvx}}$)-S-PWL	6048	3024	11844

[a] : outer-approximation.
[b] : piecewise linearization.
[c] : no shut-in time constraint.
[d] : shut-in time constraint.

Chapter 7

Results

In this chapter the results of using heuristics and branch-and-bound on the test problems outlined in Chapter 6 are presented.

All of the tests are performed on a personal computer with Intel Core i7-2600 3.40 GHz CPU and 16GB of RAM.

7.1 Computational Results of Branch-and-Bound on First Set of Problems

All of the algorithms implemented in `BONMIN` were evaluated on a limited number of the problems in the first set. After comparing the different solvers the branch-and-bound solver's performance was deemed to be the best. The only other method which matched BB's performance was the ECP, however, the difference in performance between these two algorithms was insignificant, therefore BB was used as the solver on all of the tests.

Preliminary experiments on a small sample of the problems revealed that none of the problems are solved within 72 hours. It was also observed that after 2 or 3 solutions are found within 1-2 hours, the solver does not find any more solutions and that the lower bound hardly increases for the remainder of the computation time. Therefore an upper limit on the solution time was specified as 2 hours. The other settings in `BONMIN` are left to their default values.

The results of using branch-and-bound on the first set of test problems are presented in Table 7.1.

The optimality gap is computed as follows

$$\text{Gap} = 100 \times \frac{|\text{Best possible solution - Best feasible solution}|}{\text{Best possible solution}}, \qquad (7.1.1)$$

where the best feasible solution corresponds to the best solution found by branch-and-bound when it terminates and best possible solution corresponds to the upper bound. It should be noted that the definition of the optimality gap has been specified for the context of a maximization problem.

The geometric mean of the solution and the gap of the results found by branch-and-bound are reported in Table 7.2.

Table 7.1: Branch-and-bound results on ($P_{switch,bncvx}$), ($P_{gc,bncvx}$), ($P_{switch,ncvx}$), ($P_{gc,ncvx}$),($P_{switch,cvx}$) and ($P_{gc,cvx}$).

Case	**Solution** [$]	**Gap**%[d]
Case1switch-B[a]	5.37e+05	0.45
Case2switch-B	5.36e+05	0.31
Case3switch-B	5.32e+05	0.65
Case4switch-B	5.30e+05	0.24
Case5switch-B	5.26e+05	0.29
Case6switch-B	5.01e+05	3.96
Case1qgc-B	5.37e+05	0.51
Case2qgc-B	5.35e+05	0.37
Case3qgc-B	5.32e+05	0.59
Case4qgc-B	5.29e+05	0.34
Case5qgc-B	5.26e+05	0.39
Case6qgc-B	5.00e+05	4.14
Case1switch-N[b]	5.35e+05	1.61
Case2switch-N	5.33e+05	1.26
Case3switch-N	5.31e+05	1.23
Case4switch-N	5.29e+05	1.21
Case5switch-N	5.26e+05	1.07
Case6switch-N	5.00e+05	4.89
Case1qgc-N	5.35e+05	1.65
Case2qgc-N	5.33e+05	1.27
Case3qgc-N	5.29e+05	1.55
Case4qgc-N	5.29e+05	1.35
Case5qgc-N	5.26e+05	1.03
Case6qgc-N	4.99e+05	5.12
Case1switch-C[c]	5.34e+05	1.63
Case2switch-C	5.32e+05	1.52
Case3switch-C	5.27e+05	1.65
Case4switch-C	5.24e+05	2.09
Case5switch-C	5.20e+05	2.22
Case6switch-C	4.92e+05	6.15
Case1qgc-C	5.34e+05	1.57
Case2qgc-C	5.34e+05	1.06
Case3qgc-C	5.27e+05	2.25
Case4qgc-C	5.24e+05	2.10
Case5qgc-C	5.26e+05	1.24
Case6qgc-C	5.01e+05	4.47

[a] : ($P_{switch,bncvx}$) or ($P_{gc,bncvx}$).
[b] : ($P_{switch,ncvx}$) or ($P_{gc,ncvx}$).
[c] : ($P_{switch,cvx}$) or ($P_{gc,cvx}$).
[d] : Equation (7.1.1).

Table 7.2: Comparison of solution and optimality gap from Equation (7.1.1) after two hours of CPU time.

Problem	**GM solution**[$][a]	**GM gap**%[a,b]
($P_{switch,bncvx}$) and ($P_{gc,bncvx}$)	526700.9	0.58
($P_{switch,ncvx}$) and ($P_{gc,ncvx}$)	525427.7	1.64
($P_{switch,cvx}$) and ($P_{gc,cvx}$)	522811.0	2.03

[a] : geometric mean.
[b] : Equation (7.1.1).

7.1.1 Evaluation of the Convex Relaxation on First Set of Problems

In Chapter 4.4 it was discussed that it can not be guaranteed that the convex MINLPs, ($P_{switch,cvx}$) and ($P_{gc,cvx}$), will yield a solution for which the relaxed constraint (4.4.3) is active. The difference between the linear -and quadratic term of the constraint (4.4.3) serves as a measure of violation of the original equality constraint. The magnitude of this violation is measured by evaluating the constraint over all wells j and time steps k at the solution for each of the problems and counting the number of cases for which there is a violation. For indices at which the well is shut, the violation is ignored. A summary of the constraint violations is given in Table 7.3.

Table 7.3: Measure of constraint violation of convex relaxation problems ($P_{switch,cvx}$) and ($P_{gc,cvx}$).

Case	**Violation > 1e-03**	**Violation > 1e-05**
Case1switch-C	3	103
Case2switch-C	12	26
Case3switch-C	112	158
Case4switch-C	20	26
Case5switch-C	112	153
Case6switch-C	56	79
Case1qgc-C	16	149
Case2qgc-C	3	10
Case3qgc-C	25	27
Case4qgc-C	15	148
Case5qgc-C	27	31
Case6qgc-C	95	133

7.2 Comparisons of First Solution Found on First Set of Problems

The heuristics that were presented in Chapters 3 and 5 were applied to the first set of test problems. An upper limit of 200 iterations was specified for the FP and the OFP. Experiments revealed that the FP has a tendency to stall frequently on these problems. Therefore the number of binary variables that are flipped in step 17 in the OFP algorithm, Algorithm 3, is changed from 1 to 4. The weighting parameter and geometric reduction parameter are chosen as $u_2 := 10$ and $\phi := 0.9$, respectively. These values were chosen because they showed promising results on preliminary experiments on the first set of problems. It should be noted that this change applies also to the original FP which is run within the OFP before a feasible solution is found. The results of using the original FP in which only a single binary variable is flipped when stalling occurs are also included. Computational experiments were also performed with the diving heuristics which are available in `BONMIN`, but their performance was similar or worse to the FP heuristic, and their results are therefore excluded from this report.

Table 7.4: Objective values found by heuristics on ($P_{switch,bncvx}$), ($P_{gc,bncvx}$), ($P_{switch,ncvx}$), ($P_{gc,ncvx}$),($P_{switch,cvx}$) and ($P_{gc,cvx}$).

Case	FP[$]	OFP[$]	RUC-OA[$][a]	RUC-PWL[$][a]	BB[$]
Case1switch-B[b]	1.93e+05	4.96e+05	5.08e+05	5.29e+05	**5.36e+05**
Case2switch-B	3.39e+05	5.24e+05	5.17e+05	5.29e+05	**5.31e+05**
Case3switch-B	3.29e+05	5.26e+05	5.11e+05	4.93e+05	**5.31e+05**
Case4switch-B	1.86e+05	4.96e+05	5.24e+05	5.29e+05	**5.30e+05**
Case5switch-B	3.10e+05	5.18e+05	5.19e+05	5.26e+05	**5.23e+05**
Case6switch-B	3.11e+05	4.81e+05	4.82e+05	4.88e+05	**4.97e+05**
Case1qgc-B	2.10e+05	5.08e+05	5.27e+05	5.28e+05	**5.35e+05**
Case2qgc-B	3.68e+05	5.26e+05	5.26e+05	5.25e+05	**5.32e+05**
Case3qgc-B	3.96e+05	5.27e+05	5.12e+05	5.18e+05	**5.32e+05**
Case4qgc-B	3.18e+05	5.23e+05	NA	5.23e+05	**5.29e+05**
Case5qgc-B	3.26e+05	5.18e+05	NA	5.02e+05	**5.23e+05**
Case6qgc-B	3.19e+05	4.90e+05	NA	4.87e+05	**4.96e+05**
Case1switch-N[c]	NA	5.17e+05	5.08e+05	5.29e+05	**5.32e+05**
Case2switch-N	2.95e+05	5.23e+05	5.17e+05	5.29e+05	**5.31e+05**
Case3switch-N	3.04e+05	5.14e+05	NA	4.93e+05	**5.17e+05**
Case4switch-N	1.89e+05	5.23e+05	5.24e+05	5.29e+05	**5.29e+05**
Case5switch-N	2.84e+05	5.10e+05	5.19e+05	**5.26e+05**	5.25e+05
Case6switch-N	2.97e+05	**4.99e+05**	4.82e+05	4.88e+05	4.91e+05
Case1qgc-N	2.09e+05	5.22e+05	**5.29e+05**	5.28e+05	5.28e+05
Case2qgc-N	2.98e+05	5.18e+05	**5.30e+05**	5.25e+05	5.28e+05
Case3qgc-N	2.97e+05	5.15e+05	5.12e+05	**5.18e+05**	5.17e+05
Case4qgc-N	1.89e+05	5.24e+05	5.11e+05	5.23e+05	**5.29e+05**
Case5qgc-N	2.89e+05	5.13e+05	5.21e+05	5.02e+05	**5.26e+05**
Case6qgc-N	2.99e+05	**4.99e+05**	4.81e+05	4.87e+05	4.91e+05
Case1switch-C[d]	2.15e+05	5.20e+05	5.08e+05	5.26e+05	**5.32e+05**
Case2switch-C	3.03e+05	5.20e+05	5.17e+05	**5.29e+05**	5.27e+05
Case3switch-C	3.05e+05	5.16e+05	5.11e+05	4.93e+05	5.18e+05
Case4switch-C	1.86e+05	5.23e+05	5.24e+05	**5.29e+05**	4.87e+05
Case5switch-C	2.93e+05	5.10e+05	5.19e+05	**5.26e+05**	5.20e+05
Case6switch-C	2.99e+05	4.87e+05	4.82e+05	**4.88e+05**	4.83e+05
Case1qgc-C	NA	5.16e+05	5.29e+05	5.28e+05	**5.32e+05**
Case2qgc-C	2.92e+05	5.18e+05	5.30e+05	5.25e+05	**5.32e+05**
Case3qgc-C	3.02e+05	5.02e+05	5.12e+05	5.18e+05	**5.27e+05**
Case4qgc-C	1.98e+05	5.21e+05	5.11e+05	**5.23e+05**	5.20e+05
Case5qgc-C	2.87e+05	5.07e+05	5.21e+05	5.02e+05	**5.25e+05**
Case6qgc-C	3.01e+05	**4.92e+05**	4.81e+05	4.87e+05	4.90

[a] : RUC-OA and RUC-PWL are the reverse undercover heuristics with outer-approximation and piecewise linearization, respectively.
[b] : ($P_{switch,bncvx}$) or ($P_{gc,bncvx}$).
[c] : ($P_{switch,ncvx}$) or ($P_{gc,ncvx}$)
[d] : ($P_{switch,cvx}$) or ($P_{gc,cvx}$).

Table 7.5: Time used to first solution on ($P_{switch,bncvx}$), ($P_{gc,bncvx}$), ($P_{switch,ncvx}$), ($P_{gc,ncvx}$),($P_{switch,cvx}$) and ($P_{gc,cvx}$).

Case	**FP**[s]	**OFP**[s]	**RUC-OA**[s][a]	**RUC-PWL**[s][a]	**BB**[s]
Case1switch-B[b]	30.9	50.5	**4.1**	5.3	449.4
Case2switch-B	345.3	181.5	3.3	**2.2**	350.5
Case3switch-B	121.5	108.9	**5.1**	10.6	357.4
Case4switch-B	14.1	22.4	**2.3**	11.8	673.5
Case5switch-B	71.7	51.0	**4.2**	11.6	462.4
Case6switch-B	23.6	43.0	**5.1**	10.7	1102.4
Case1qgc-B	34.5	42.1	**7.5**	366.8	531.4
Case2qgc-B	343.6	37.4	**9.4**	311.6	531.5
Case3qgc-B	249.3	127.5	**7.3**	216.7	701.3
Case4qgc-B	38.9	**58.7**	NA	200.0	675.0
Case5qgc-B	93.1	**75.2**	NA	244.2	475.3
Case6qgc-B	**29.5**	86.8	NA	222.5	1118.8
Case1switch-N[c]	NA	14.5	**4.6**	5.6	519.2
Case2switch-N	12.6	30.2	**3.9**	12.7	500.5
Case3switch-N	13.0	23.2	NA	**10.9**	759.8
Case4switch-N	29.0	14.0	**3.5**	19.4	444.7
Case5switch-N	46.6	27.9	**8.5**	11.0	489.8
Case6switch-N	31.4	28.7	**5.6**	11.6	1053.8
Case1qgc-N	25.3	18.7	**7.9**	367.3	584.8
Case2qgc-N	16.8	25.1	**7.7**	311.9	564.2
Case3qgc-N	24.4	32.1	**11.8**	216.8	451.4
Case4qgc-N	39.5	16.1	**3.7**	199.8	352.3
Case5qgc-N	24.8	38.1	**5.9**	245.4	669.4
Case6qgc-N	39.5	37.1	**6.0**	223.6	824.9
Case1switch-C[d]	9.2	15.1	**4.3**	5.6	493.7
Case2switch-C	10.1	17.3	**3.3**	8.1	883.3
Case3switch-C	9.1	23.3	**4.2**	10.9	1206.3
Case4switch-C	9.1	10.6	**2.3**	11.5	1935.8
Case5switch-C	10.9	26.1	**4.0**	11.3	553.0
Case6switch-C	12.0	20.5	**5.8**	10.5	1297.1
Case1qgc-C	NA	18.2	**7.6**	366.3	762.9
Case2qgc-C	13.7	25.0	**8.1**	311.6	432.8
Case3qgc-C	11.8	25.9	**7.4**	216.9	1006.8
Case4qgc-C	9.0	18.4	**3.8**	200.9	1145.0
Case5qgc-C	14.8	27.7	**5.8**	245.2	387.8
Case6qgc-C	19.2	29.9	**6.1**	222.9	1185.2

[a] : RUC-OA and RUC-PWL are the reverse undercover heuristics with outer-approximation and piecewise linearization, respectively.
[b] : ($P_{switch,bncvx}$) or ($P_{gc,bncvx}$).
[c] : ($P_{switch,ncvx}$) or ($P_{gc,ncvx}$)
[d] : ($P_{switch,cvx}$) or ($P_{gc,cvx}$).

7.2.1 Average Values

The results of using the heuristics reported in 7.5 above are used to compute averages in order to make inferences on the results. Problems for which one of the heuristics fail are excluded when average values are computed. This leaves a total of 30 problems for which none of the heuristics failed. A comparison of the geometric mean of the time required by the different methods is given in Table 7.6.

Table 7.6: Average time required to find first feasible solution by heuristics and BB on first set of test problems.

Comparison	FP	OFP	RUC-OA[a]	RUC-PWL[a]	BB
GM[s][b]	26.9	30.7	**5.2**	43.9	647.1
No. failure	2	**0**	4	**0**	**0**

[a] : RUC-OA and RUC-PWL are the reverse undercover heuristics with outer-approximation and piecewise linearization, respectively.
[b] : geometric mean.

To measure the quality of the solution found by the heuristics and the solution returned by BB reported in Table 7.1, a gap measuring the percentage difference between the two solution values is used. For some of the problems, the first feasible solution found by BB remains as the best solution found even after 2 hours of CPU time. Moreover, for "Case4-C" both the RUC-PWL and RUC-OA find a solution which is slightly better than the solution found by BB after two hours of CPU time, i.e. the percentage difference is negative. Consequently, the geometric mean can not be used and a mean which is the average of all the gap values is used instead.

Table 7.7: Average percentage difference between first feasible solutions found by heuristics and BB with results from Table 7.1

Comparison	FP	OFP	RUC-OA[a]	RUC-PWL[a]	BB
Average gap%	47.43	2.42	2.39	1.78	**0.92**

[a] : RUC-OA and RUC-PWL are the reverse undercover heuristics with outer-approximation and piecewise linearization, respectively.

7.2.2 Performance Profiles

Performance profiles [35] are a visual tool for evaluating and comparing the performance of optimization solvers on a given set of problems. The performance profile for a solver is the cumulative distribution function for a *performance metric*, e.g. CPU time. Given a set of problems $\mathcal{P}$ where $|\mathcal{P}| = n_p$ and a set of solvers $\mathcal{S}$ where $|\mathcal{S}| = n_s$, the performance profile is generated by comparing the results of applying all solvers $s \in \mathcal{S}$ on all problems $p \in \mathcal{P}$.

For each problem p and solver s, the performance $t_{p,s}$ is defined as

$$t_{p,s} = \text{CPU time required to solve problem } p \text{ by solver } s.$$

The performance on problem p by solver s is compared with the best performance by any solver s on this problem by defining a *performance ratio*

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} : s \in \mathcal{S}\}}.$$

It is assumed that a parameter $r_M \geq r_{p,s}$ for all p, s is specified by defining $r_M = r_{p,s}$ if and only if solver s does not solve problem p. The cumulative distribution function for the performance ratio is defined as

$$\psi_s(\kappa) = \frac{1}{n_p}\text{size}\{p \in \mathcal{P} : r_{p,s} \leq \kappa\}.$$

Hence, $\psi_s(\kappa)$ is the probability that a solver s yields a performance ratio $r_{p,s}$ that is at most slower by a factor of κ of the best ratio. A performance profile is the distribution function of a performance metric. For the sake of simplicity, it was assumed that the performance metric was the CPU time required by a solver s to solve a problem p, this can however be extended to any performance metric(number of nodes, number of function evaluations, objective value, etc.).

The base-2 logarithm of κ will be used for the performance profile plots in this thesis.[1]

The performance profiles of the CPU time of the heuristics applied to the first set of test problems is displayed in Figure 7.1.

The performance profiles shown in Figure 7.2 uses the difference between the solution found by the branch-and-bound solver after 2 hours of CPU time, shown in Table 7.1, and the heuristics, given in Table 7.10, as a metric for comparison.

[1] `MATLAB` script for creating these figures can be downloaded from:
`http://www.mcs.anl.gov/~akannan/noqs/images/perf_profile.m`

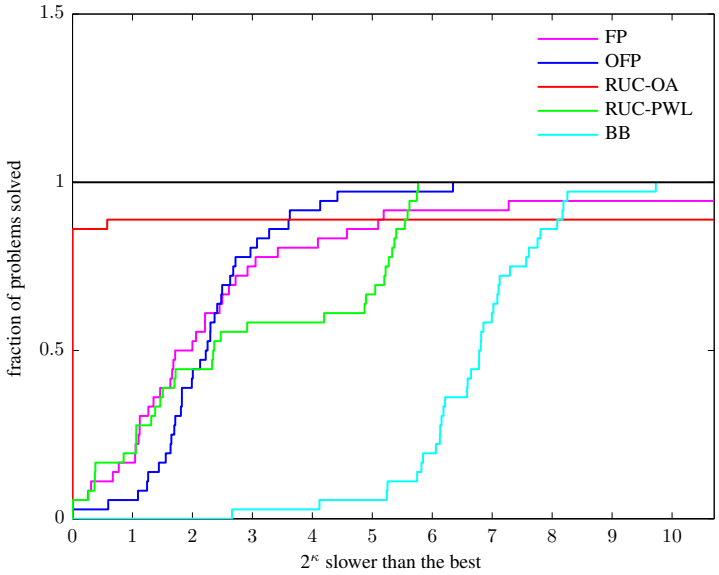

Figure 7.1: Performance of CPU time of heuristics on first set of problems.

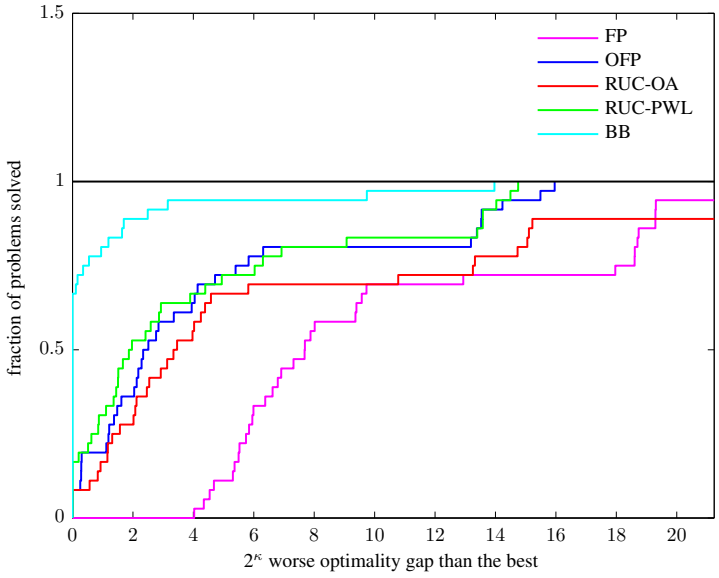

Figure 7.2: Performance of optimality gap with heuristics on first set of problems.

7.3 Computational Results of Branch-and-Bound on Second Set of Problems

The second set of test problems presented in Chapter 6.2.1 are solved with the branch-and-bound solver with an upper time limit of 2 hours. Other user options in BONMIN are left at their default values. The results of these tests are presented in Table 7.8.

Table 7.8: Branch-and-bound on second set of problems.

Case	**Solution**[$]	**Gap**%[a]
CASE1qgcNSlow[b,d]	3.66e+06	3.51
CASE1qgcSlow[c]	3.65e+06	4.00
CASE2qgcNSlow	3.64e+06	3.98
CASE2qgcSlow	3.62e+06	4.74
CASE3qgcNSlow	3.64e+06	3.83
CASE3qgcSlow	3.61e+06	5.07
CASE1qgcNShigh[e]	4.07e+06	4.71
CASE1qgcShigh	4.06e+06	4.89
CASE2qgcNShigh	4.06e+06	4.90
CASE2qgcShigh	4.05e+06	5.00
CASE3qgcNShigh	4.06e+06	4.90
CASE3qgcShigh	4.03e+06	5.36

[a] : Equation (7.1.1).
[b] : no shut-in time constraint.
[c] : shut-in time constraint.
[d] : low initial pseudopressure.
[e] : high initial pseudopressure.

The average values of the results in Table 7.8 are given in Table 7.9.

Table 7.9: Average values of formulations of second set of problems.

Problem type	**GM solution**[$][a]	**GM gap**%[a]
Case#NS[b]	3.85e+06	4.27
Case#S[c]	3.83e+06	4.82

[a] : geometric mean.
[b] : no shut-in time constraint.
[c] : shut-in time constraint.

7.4 Comparisons of First Solution Found on Second Set of Problems

The same heuristics applied to the first set of test problems are also applied to the second set. The results of using these heuristics are presented in Tables 7.11 and 7.10. For this set of problems there were only 3 instances for which all heuristics were able to find a solution. Due to the lack of consistency, no average values of the results are reported.

Table 7.10: Objective value of first solution found by heuristics on second set of problems.

Case	**FP**[$]	**OFP**[$]	**RUC-OA**[$][a]	**RUC-PWL**[$][a]	**BB**[$]
CASE1qgcNSlow[b,d]	2.55e+06	3.17e+06	NA	3.55e+06	**3.63e+06**
CASE1qgcSlow [c]	NA	3.20e+06	3.36e+06	3.61e+06	**3.62e+06**
CASE2qgcNSlow	NA	3.24e+06	3.41e+06	3.57e+06	**3.63e+06**
CASE2qgcSlow	NA	3.13e+06	NA	3.57e+06	**3.62e+06**
CASE3qgcNSlow	2.74e+06	3.24e+06	NA	3.57e+06	**3.62e+06**
CASE3qgcSlow	2.76e+06	3.20e+06	NA	3.54e+06	**3.60e+06**
CASE1qgcNShigh[e]	NA	3.70e+06	3.80e+06	NA	**4.04e+06**
CASE1qgcShigh	NA	3.73e+06	3.89e+06	4.01e+06	**4.01e+06**
CASE2qgcNShigh	3.14e+06	3.83e+06	3.76e+06	3.97e+06	**4.01e+06**
CASE2qgcShigh	3.22e+06	3.78e+06	3.72e+06	3.97e+06	**4.00e+06**
CASE3qgcNShigh	3.15e+06	3.68e+06	3.75e+06	3.92e+06	**3.96e+06**
CASE3qgcShigh	3.23e+06	3.23e+06	3.78e+06	NA	**3.96e+06**

[a] : RUC-OA and RUC-PWL are the reverse undercover heuristics with outer-approximation and piecewise linearization, respectively.
[b] : no shut-in time constraint.
[c] : shut-in time constraint.
[d] : low initial pseudopressure.
[e] : high initial pseudopressure.

Table 7.11: Time to first solution found by heuristics on second set of problems.

Case	FP[s]	OFP[s]	RUC-OA[s][a]	RUC-PWL[s][a]	BB[s]
CASE1qgcNSlow[b,d]	76.2	**49.3**	NA	60.9	1159.1
CASE1qgcSlow[c]	NA	62.3	**21.5**	56.9	1308.5
CASE2qgcNSlow	NA	53.9	**11.0**	22.9	1157.1
CASE2qgcSlow	NA	**56.9**	NA	91.7	1698.8
CASE3qgcNSlow	68.5	59.0	NA	**53.9**	1436.2
CASE3qgcSlow	83.8	53.9	NA	**44.9**	1414.7
CASE1qgcNShigh[e]	NA	55.1	**30.9**	NA	1231.0
CASE1qgcShigh	NA	41.4	**18.7**	47.4	1095.1
CASE2qgcNShigh	66.6	100.1	**8.5**	66.9	1649.0
CASE2qgcShigh	63.6	46.7	**16.0**	76.0	1095.9
CASE3qgcNShigh	65.8	61.0	**11.3**	79.5	1399.9
CASE3qgcShigh	71.1	113.4	**12.6**	NA	1128.5

[a] : RUC-OA and RUC-PWL are the reverse undercover heuristics with outer-approximation and piecewise linearization, respectively.
[b] : no shut-in time constraint.
[c] : shut-in time constraint.
[d] : low initial pseudopressure.
[e] : high initial pseudopressure.

7.4.1 Performance Profiles

Performance profiles for the heuristics with CPU time and optimality gap as metrics are given in Figures 7.3 and 7.4, respectively.

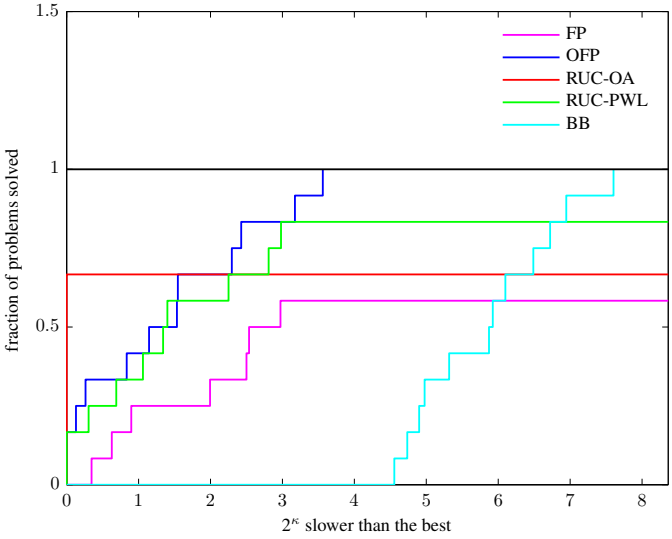

Figure 7.3: Performance profile for CPU time of heuristics on second set of problems.

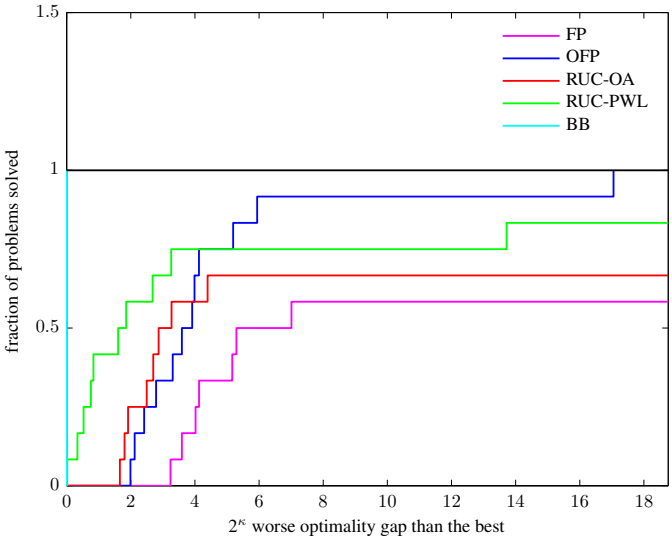

Figure 7.4: Performance profile for optimality gap with heuristics on second set of problems.

Chapter 8

Discussion

In this chapter, the results of the computational study from the previous chapter are discussed. The results are evaluated by comparing the performance of the different methods and problem formulations. In order to assess the results, the tables and performance profiles of the heuristics from Chapter 7 will be used.

8.1 First Set of Test Problems

8.1.1 Comparing the Formulations

The comparison of the results of using BB on the different problem formulations presented in Table 7.2 show that the average optimality gap for the bilinear nonconvex problems is 64% less than for the reformulated nonconvex problems and 71% less than for the relaxed convex problems. Moreover, the optimal solution is on average also slightly better for the bilinear nonconvex problems, with an increase of 0.24% from the average solution obtained by the reformulated nonconvex problems and 0.74% from the average solution obtained by the relaxed convex problems. Since the problems ($P_{switch,bncvx}$),($P_{gc,bncvx}$) and ($P_{switch,ncvx}$),($P_{gc,ncvx}$) differ only by the reformulation of bilinear products, it is reasonable to assume that the reformulations are the cause for the increase in the optimality gap. Moreover, relaxing the nonlinear equality constraint (4.1.5) to an inequality constraint produces a larger search space for the convex relaxation, which subsequently results in a larger value for the best possible solution and the optimality gap. It was

initially expected that the larger search space in the convex MINLP relaxation would also produce better objective values, but this was however not observed. It should be taken into account that BB was terminated after 2 hours of CPU time. It may be possible that both the objective value and the optimality gap improve for the convex relaxation if BB is run for a longer period of time.

8.1.2 Evaluating the Convex Relaxation

The convex relaxations ($P_{switch,cvx}$) and ($P_{gc,cvx}$) have proven to be ineffective in approximating the original nonconvex MINLP. It is shown in Table 7.3 that with a tolerance of 1e-03 between the linear and quadratic terms in Equation (4.4.3), over half of the $|\mathcal{J}||\mathcal{K}| = 174$ constraints are violated in three of the instances. When the tolerance is 1e-05, the constraint violation is more severe - with a total of six instances for which half of the constraints are violated. This result and the fact that for the nonconvex problems both BB and the heuristics perform well, suggests that the convex relaxation does not have beneficial properties which make it worthwhile.

Note that the *dual variable* could have been used as an indicator of whether the constraints are active, but the default tolerance in `AMPL` showed that none of the constraints were active at the solution. Therefore the difference between the two terms was chosen as a measure.

8.1.3 Heuristics

The Feasibility Pump

The results in Table 7.7 show that the feasibility pump performs worst when it comes to the objective value. This result is aligned with the computational study summarized in Chapter 3.1.3, where FP was shown to find solutions with poor objective values. In Table 7.6 it is shown that the FP is the second fastest heuristic to find a feasible solution and that it gives an average reduction of 12% in CPU time compared to OFP, which is the third fastest heuristic. The performance profile in Figure 7.1 shows that the FP is a competitive heuristic with respect to the CPU time for a fraction of 0.61 of the problems, but that it is dominated by OFP and subsequently by RUC-PWL if a stringent requirement of solving all of the problems is used as a criterion for the comparison. Moreover, the performance profiles for the

optimality gap given in Figure 7.2 shows that FP is outperformed by all other heuristics for the majority of the problems. Additionally, FP fails on two problems, a result which in combination with the poor solution quality and mediocre CPU time can be used to substantiate the claim that FP has a poor performance on the first set of problems. The overall poor performance of FP can to some extent be attributed to that the other heuristics are tweaked specifically for these problems, while FP is used with its default settings.

The Objective Feasibility Pump

The summary of the results presented in Tables 7.7 and 7.6 shows that with only an increase of 12% of the average CPU time, the OFP finds solutions that give a 95% reduction in the average optimality gap when compared with the FP. Furthermore, both the performance profiles for CPU time and optimality gap shown in Figures 7.1 and 7.2, respectively, show that OFP has a good overall performance, especially because of its ability to solve all of the problems. Due to the excessive stalling that was observed in experimental studies of these problems, the OFP was implemented with four binary variables being flipped instead of just one, as is used in the FP. This adjustment was able to mitigate the impact of stalling on the overall performance.

The Reverse Undercover-methods

The most unexpected result of the experiments was the overall good performance of RUC-OA. It is the fastest heuristic with an average CPU time that is 5 times faster than the FP, which is the second fastest heuristic. The fast solution time is also prevalent in the comparison of the heuristics shown in the performance profiles in Figure 7.1, where it can be seen that RUC-OA is the fastest heuristic on a fraction of 0.86 of the problems. Furthermore, it finds solutions with an average optimality gap which is the second lowest of all the heuristics, as shown in Table 7.7. The fast solution times are most likely due to the fact RUC-OA uses a simple approach to generating a MILP approximation to the MINLP, as opposed to RUC-PWL which needs to introduce a large number of additional variables and constraints as summarized in Table 6.5, and that `CBC` by default employs several sophisticated heuristics to find high quality solutions. Although RUC-OA yields good solutions overall while simultaneously maintaining the lowest average CPU time, it is arguably the least robust heuristic since it fails on 4 problem instances. The failure is likely to stem from the fact that the MILP approximation is

poor and yields solutions for which the routing and shut-in variables can not satisfy the nonlinearly constrained continuous problem. Furthermore, 3 of the instances on which RUC-OA fails is the bilinear problem ($P_{gc,bncvx}$). This may be an indication that the nonconvex NLP relaxation of this problem is not suited for this method, however more tests would be required to make a conclusive statement on this issue.

Overall, the RUC-PWL heuristic has a good performance on the first set of problems. The performance profiles in Figure 7.2 and 7.1 show that this heuristic has a high probability of being the best heuristic, both with respect to the optimality gap and CPU time, when all the problems being solved is used as a criterion. Moreover, it has the smallest average optimality gap of all the heuristics as shown in Table 7.7, but the worst average CPU time as can be seen in Table 7.6. The average CPU time is worse for the RUC-PWL because it consistently performs poorly on the critical gas rate problems, however, it is faster than FP, the second fastest heuristic, on 14 out of 18 of the switching problems.

An interesting result is the high quality of the solutions that are found by the RUC-methods. The solution values for both RUC-OA and RUC-PWL are superior to the ones found by OFP with a reduction in the average optimality gap of 1% and 26%, respectively. This result can be explained by the intrinsic differences in the approach by which these heuristics search for a feasible solution. The OFP finds feasible solutions by gradually diminishing the influence of the original objective, while the RUC-methods maintain the same objective both when they solve the initial approximate MILP and the relaxed NLP. The-RUC methods are implemented at the modelling level, which means that `CBC` does not necessarily terminate after it computes its first feasible solution. Typically it uses a feasibility pump for MILPs to initially find a feasible solution, which in general has a poor objective value, then it will use improvement heuristics on this solution to improve this solution before terminating and returning the improved solution. This means that the RUC methods do not necessarily terminate after the first feasible solution is found, instead they terminate after improving the first feasible solution.

8.2 Second Set of Test Problems

The computational results of using heuristics on the second set of problems presented in Tables 7.11 and 7.10, show that this set of problems is significantly harder to solve. In the first problem set there were only 6 out of 36 problems for which at least one heuristic failed, whereas at least one heuristic fails on 9 out of 12 problems in the second set. This result was expected since the second set of problems has 936 more continuous variables, 156 additional binary variables and 3210 more constraints(for the most difficult case).

8.2.1 Branch-and-Bound Comparison

The average results of the objective value and the optimality gap given in Table 7.9 show that the formulation with a shut-in time constraint perform worse than the formulation without this constraint both with regards to the solution value, which is increased by 0.51%, and the optimality gap, which is reduced by 11.41%. This result implies that allowing the well to remain shut for a single time-step can yield an improvement in the objective value. However, the physical interpretation of the solutions for which a well is shut for a single time step may lack practical feasibility.

8.2.2 Heuristics

The Feasibility Pump

The original FP heuristic has the worst performance of all the heuristics. It is only able to find a solution for 7 of the 12 problems, and for 6 of these problems it returns the worst solution of all the heuristics. The performance profiles given in Figures 7.3 and 7.4 show with only a fraction of 0.58 of the problems being solved, the FP performs worse than all of the other heuristics both with respect to CPU time and optimality gap.

The Objective Feasibility Pump

Despite of the difficulties introduced by the large problem size, the OFP is shown to be the most robust heuristic. The performance profiles for the CPU time given in Figure 7.3 shows that OFP has the highest probability of being the fastest heuristic which finds a solution to all of the problems. It

is also seen in Table 7.11 that for 2 of the 12 problems, OFP finds a feasible solution faster than any other heuristic. Although OFP yields good results with respect to CPU time, it generally found solutions with a relatively poor objective value. The performance profiles for the optimality gap given in Figure 7.4 show that for a fraction of 0.75 of the problems, the OFP is outperformed by RUC-PWL, and for a fraction of 0.58 of the problems, the OFP has a worse performance than both RUC-OA and RUC-PWL.

The Reverse Undercover-methods

Although RUC-PWL fails to find a solution for 2 problems, it has the highest probability of finding a solution with the lowest value of the optimality gap amongst the heuristics. This result can be seen in the performance profiles in Figure 7.4, where it is shown that the probability that RUC-PWL finds a solution with smallest optimality gap is 0.75. The next best heuristic shown on this performance profile, RUC-OA, has a probability of 0.58 finding the solution with the smallest optimality gap.

The performance profiles for the CPU time given in Figure 7.3 show that RUC-OA has the highest probability of being the optimal solver with regards to CPU time. It can also be seen in the same figure that the performance profile for RUC-PWL shows that it has a higher probability than OFP of being the best solver with respect to CPU time on a fraction of 0.58 of the problems.

8.3 Summary

Although there are no average values reported for the heuristics on the second set of problems due to the large number of failures, it is observed that the overall trend of the performance of the heuristics is similar on both problem sets. For instance, on the problems that RUC-OA is able to find a solution, it is usually the fastest of the heuristics, and RUC-PWL tends to find solutions with good objective values. The FP heuristic is dominated by all of the other heuristics with respect to both performance measures. It is also observed that the largest deviation in performance between the problem sets comes from OFP. In the first problem set, it was seen that OFP had an overall good performance, but for the second problem set it falls behind the RUC-methods with the optimality gap as performance measure. In general BB finds the best solutions, but it is also slower than all of the heuristics.

Chapter 9

Conclusions

In Chapter 3 of this thesis, the OFP heuristic was developed and applied to a set of convex MINLP test problems. When compared with the results for the OFP for MILPs [3] it was found that the proposed OFP for general MINLPs resulted only in a minor improvement in the objective value at the cost of a drastically increased CPU time. However, it was observed that the parameters in the OFP algorithm can be tweaked for specific problems and subsequently find better solutions than the original FP.

A few different formulations of the shale gas production optimization problem have been evaluated, and it has been shown that the nonconvex formulation has several benefits over the convex relaxation. Both of the nonconvex formulations perform in overall well with respect to the performance of a branch-and-bound solver and heuristics. Furthermore, the convex relaxation was shown to yield solutions at which the relaxed inequality constraint was inactive. The implication of this result is that the solutions found with the convex relaxation do not satisfy all of the inherent physical properties of the problem.

The results of this thesis have shown that there are a variety of heuristic techniques that can be tailored to solve the shale gas production optimization problem yielding an improvement in CPU time and solution quality in comparison with the currently available heuristics in `BONMIN`. In particular, experiments revealed that both the OFP and the RUC-methods were able to find high quality solutions in a fraction of the time required by BB, albeit in general not as good solutions as found by BB. The benefit of using heuristics within a decision support system will be most apparent in situations where there is a hard requirement of a quick solution time.

Chapter 10

Future Work

The OFP that has been developed in this thesis and the original FP both use a simple rounding scheme to find the nearest integer feasible point. This approach is quite naive and can be improved by for instance choosing the direction of the rounding such that linear constraints involving the integer variables are never infeasible at the rounded point. If the integer variables also appear in the objective, then it is beneficial to round the variables down rather than up(in a minimization problem). Sophisticated rounding methods have been developed for MILPs, it is likely that the same methods can be applied to MINLPs with a few adjustments.

None of the MINLP formulations of the shale gas production optimization problem are able to terminate within a reasonable amount of CPU time. Developing a tighter formulation of this problem could be advantageous with regards to achieving a faster termination. Semidefinite relaxations of mixed-integer quadratic programs have been shown to yield tighter relaxations [7]. It is possible that these type of relaxations can be applied to the problems studied in this thesis, as discussed in Appendix D.2.

There has recently been proposed a global optimization method for solving nonconvex MINLPs in which the source of nonconvexity stems from separable nonconvex functions [29]. This method uses a sequential convex MINLP framework, in which a sequence of approximating convex MINLPs are solved and refined until a termination criterion is met. Since the nonconvexity in the problems studied in this thesis are due to a nonlinear equality constraint with two separable convex functions, it may be possible to apply a sequential convex MINLP framework to solving these problems to global optimality.

Bibliography

[1] Bonanza or Bane; America's Cheap Gas. *The Economist(US)*, 2013.

[2] Frack to the Future. *The Economist(US)*, 2013.

[3] Tobias Achterberg and Timo Berthold. Improving the feasibility pump. *Discrete Optimization*, 4(1):77–86, March 2007.

[4] Rafi Al-Hussainy, HJ Ramey Jr, and PB Crawford. The flow of real gases through porous media. *Journal of Petroleum Technology*, 18(5):624–636, 1966.

[5] Claudia D Ambrosio. *Application-oriented Mixed Integer Non-Linear Programming*. Phd, University of Bologna, 2009.

[6] Claudia D Ambrosio, Antonio Frangioni, Leo Liberti, and Andrea Lodi. Experiments with a Feasibility Pump Approach for Nonconvex MINLPs. In *Experimental Algorithms*, pages 350–360. Springer Berlin Heidelberg, 2010.

[7] Daniel Axehill, Lieven Vandenberghe, and Anders Hansson. Convex relaxations for mixed integer predictive control. *Automatica*, 46(9):1540–1545, September 2010.

[8] Heinz H Bauschke and Jonathan M Borwein. On projection algorithms for solving convex feasibility problems. *SIAM review*, 38(3):367–426, 1996.

[9] EML Beale and JJH Forrest. Global optimization using special ordered sets. *Mathematical Programming*, 10(1):52–69, 1976.

[10] Evelyn Martin Lansdowne Beale and John A Tomlin. Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables. *OR*, 69:447–454, 1970.

[11] Pietro Belotti, Christian Kirches, Sven Leyffer, Jeff Linderoth, and Jim Luedtke. Mixed-Integer Nonlinear Optimization. Technical report, Argonne National Laboratory, Mathematics and Computer Science Division, 2012.

[12] Livio Bertacco, Matteo Fischetti, and Andrea Lodi. A feasibility pump heuristic for general mixed-integer problems. *Discrete Optimization*, 4(1):63–76, March 2007.

[13] Timo Berthold. Primal Heuristics for Mixed Integer Programs. Msc, TU Berlin, 2006.

[14] Timo Berthold and Ambros M. Gleixner. Undercover: a primal minlp heuristic exploring a largest sub-mip. *Mathematical Programming*, pages 1–32, 2013.

[15] Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2nd edition, 1999.

[16] Alain Billionnet, Sourour Elloumi, and Marie-Christine Plateau. Improving the performance of standard solvers for quadratic 0-1 programs by a tight convex reformulation: The QCR method. *Discrete Applied Mathematics*, 157(6):1185–1197, March 2009.

[17] Natashia L Boland, Andrew C Eberhard, Faramroze G Engineer, Matteo Fischetti, Martin WP Savelsbergh, and Angelos Tsoukalas. Boosting the feasibility pump. 2011.

[18] NL Boland, AC Eberhard, F Engineer, and A Tsoukalas. A new approach to the feasibility pump in mixed integer programming. *SIAM Journal on Optimization*, 22(3):831–861, 2012.

[19] Pierre Bonami, Lorenz T. Biegler, Andrew R. Conn, Gérard Cornuéjols, Ignacio E. Grossmann, Carl D. Laird, Jon Lee, Andrea Lodi, François Margot, Nicolas Sawaya, and Andreas Wächter. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization*, 5(2):186–204, May 2008.

[20] Pierre Bonami, Gérard Cornuéjols, Andrea Lodi, and François Margot. A Feasibility Pump for mixed integer nonlinear programs. *Mathematical Programming*, 119(2):331–352, March 2008.

[21] Pierre Bonami and João P. M. Gonçalves. Heuristics for convex mixed integer nonlinear programs. *Computational Optimization and Applications*, 51(2):729–747, September 2010.

[22] Pierre Bonami and Mustafa Kilinc. Algorithms and Software for Convex MIxed Integer Nonlinear Programs. volume 154 of *The IMA Volumes in Mathematics and its Applications*, chapter 1. Springer New York, New York, NY, 2012.

[23] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[24] CBC. https://projects.coin-or.org/cbc.

[25] Coats Engineering Inc. *SENSOR Manual*, 2009.

[26] Gérard Cornuéjols. Valid inequalities for mixed integer linear programs. *Mathematical Programming*, 112(1):3–44, 2008.

[27] Fair Isaac Corporation. http://www.lindo.com/.

[28] Claudia D'Ambrosio, Antonio Frangioni, Leo Liberti, and Andrea Lodi. A storm of feasibility pumps for nonconvex minlp. *Mathematical programming*, 136(2):375–402, 2012.

[29] Claudia D'Ambrosio, Jon Lee, and Andreas Wächter. An algorithmic framework for minlp with separable non-convexity. In *Mixed Integer Nonlinear Programming*, pages 315–347. Springer, 2012.

[30] Claudia D'Ambrosio and Andrea Lodi. Mixed integer nonlinear programming tools: a practical overview. *4OR*, 9:329–349, 2011.

[31] Emilie Danna, Edward Rothberg, and Claude Le Pape. Exploring relaxation induced neighborhoods to improve mip solutions. *Mathematical Programming*, 102(1):71–90, 2005.

[32] Marianna De Santis, Stefano Lucidi, and Francesco Rinaldi. New concave penalty functions for improving the feasibility pump. Technical Report 10, 2010.

[33] Kalyanmoy Deb. Multi-objective optimization. *Multi-objective optimization using evolutionary algorithms*, pages 13–46, 2001.

[34] Yichuan Ding, Sandra Gregov, Oleg Grodzevich, Itamar Halevy, Zanin Kavazovic, Oleksandr Romanko, Tamar Seeman, Romy Shioda, and Fabien Youbissi. Discussions on normalization and other topics in multiobjective optimization. In *Fields-MITACS, Fields Industrial Problem Solving Workshop*, 2006.

[35] Elizabeth D Dolan and Jorge J Moré. Benchmarking optimization software with performance profiles. *Mathematical programming*, 91(2):201–213, 2002.

[36] Matteo Fischetti, Fred Glover, and Andrea Lodi. The feasibility pump. *Mathematical Programming*, 104(1):91–104, March 2005.

[37] Matteo Fischetti and Domenico Salvagnin. Feasibility pump 2.0. *Mathematical Programming Computation*, 1(2-3):201–222, 2009.

[38] Roger Fletcher and Sven Leyffer. Solving mixed integer nonlinear programs by outer approximation. *Mathematical Programming*, 66(1-3):327–349, August 1994.

[39] Robert W Floyd. Nondeterministic algorithms. *Journal of the ACM (JACM)*, 14(4):636–644, 1967.

[40] Robert Fourer, David M Gay, and Brian W Kernighan. *Ampl*. Scientific Press San Francisco, 1993.

[41] A Frangioni and C Gentile. A computational comparison of reformulations of the perspective relaxation: Socp vs. cutting planes. *Operations Research Letters*, 37(3):206–210, 2009.

[42] Tetsuya Fujie and Masakazu Kojima. Semidefinite programming relaxation for nonconvex quadratic programs. *Journal of Global Optimization*, 10(4):367–380, 1997.

[43] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)*. W. H. Freeman & Co Ltd, first edition edition, January 1979.

[44] Fred Glover. Improved linear integer programming formulations of nonlinear integer problems. *Management Science*, 22(4):455–460, 1975.

[45] Ignacio E Grossmann. Review of nonlinear mixed-integer and disjunctive programming techniques. *Optimization and Engineering*, 3(3):227–252, 2002.

[46] Omprakash K. Gupta and A. Ravindran. Branch and bound experiments in convex nonlinear integer programming. *Management Science*, 31(12):1533–1546, 1985.

[47] Inc. Gurobi Optimization. Gurobi optimizer reference manual, 2012.

[48] Shih-Ping Han. A successive projection method. *Mathematical Programming*, 40(1-3):1–14, 1988.

[49] Christoph Helmberg and Franz Rendl. Solving quadratic (0,1)-problems by semidefinite programs and cutting planes. *Mathematical Programming*, 82(3):291–315, 1998.

[50] IBM. User ' s Manual for CPLEX, 2009.

[51] Lindo Systems Inc. http://www.lindo.com/.

[52] Ipopt. https://projects.coin-or.org/ipopt.

[53] Donald La Verne Katz and Robert L Lee. *Natural gas engineering: production and storage*. McGraw-Hill New York, New York, 1990.

[54] Il Yong Kim and OL De Weck. Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Structural and multidisciplinary optimization*, 29(2):149–158, 2005.

[55] B. R. Knudsen. Production Optimization in Shale Gas Reservoirs. Msc., Norwegian University of Science and Technology, 2010.

[56] B. R. Knudsen. Personal communication, 2013.

[57] B. R. Knudsen and B. Foss. Shut-in based production optimization of shale-gas systems. *Computers and Chemical Engineering*. (to appear).

[58] B. R. Knudsen and B. Foss. Shale-gas Well and Compressor Optimization by Mixed Integer Nonlinear Programming. Technical report, NTNU, 2011.

[59] B. R. Knudsen, B. Foss, C. H. Whitson, and A. R. Conn. Target-rate Tracking for Shale-gas Multi-well Pads by Scheduled Shut-ins. Technical report, Norwegian University of Science and Technology, 2012.

[60] Eugene L Lawler, Jan Karel Lenstra, AHG Rinnooy Kan, and David B Shmoys. *The traveling salesman problem: a guided tour of combinatorial optimization*, volume 3. Wiley Chichester, 1985.

[61] Jon Lee and Dan Wilson. Polyhedral methods for piecewise-linear functions i: the lambda method. *Discrete applied mathematics*, 108(3):269–285, 2001.

[62] Jeffrey T. Linderoth and Andrea Lodi. *MILP Software*. John Wiley & Sons, Inc., 2010.

[63] David Luenberger and Yinyu Ye. *Linear and Nonlinear Programming*. Springer, third edition, 2008.

[64] Ashutosh Mahajan, S. Leyffer, and C. Kirches. Algorithms for solving convex minlps with minotaur. *21st International Symposium on Mathematical Programming*, 08/2012 2012.

[65] Marco A. Duran and Ignacio E. Grossmann. An Outer-Approximation Algorithm for a Class of Mixed-Integer Nonlinear Programs. *Mathematical Programming*, 36:307–339, 1986.

[66] Alexander Martin, Markus Möller, and Susanne Moritz. Mixed integer models for the stationary case of gas network optimization. *Mathematical programming*, 105(2-3):563–582, 2006.

[67] S. Mehrotra. On the implementation of a primal-dual interior point method. *SIAM Journal on Optimization*, 2(4):575–601, 1992.

[68] G. L. Nemhauser and L. A. Wolsey. *Integer and combinatorial optimization*. Wiley-Interscience, New York, NY, USA, 1988.

[69] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer Verlag, 2006.

[70] The MOSEK optimization software. http://www.mosek.com/.

[71] I. Quesada and I.E. Grossmann. An lp/nlp based branch and bound algorithm for convex minlp optimization problems. *Computers and chemical engineering*, pages 937–947, 1992.

[72] Richard E Rosenthal. Gams–a user's guide. 2004.

[73] Harvey M Salkin and Cornelis A De Kluyver. The knapsack problem: a survey. *Naval Research Logistics Quarterly*, 22(1):127–144, 1975.

[74] Martin WP Savelsbergh. Preprocessing and probing techniques for mixed integer programming problems. *ORSA Journal on Computing*, 6(4):445–454, 1994.

[75] Christian Schulte and Peter J Stuckey. Speeding up constraint propagation. In *Principles and Practice of Constraint Programming–CP 2004*, pages 619–633. Springer, 2004.

[76] Shaurya Sharma. Heuristics and general purpose algorithms for MINLPs. Technical report, Norwegian University of Science and Technology, 2012.

[77] Michael Stein, Gilberto Venturini, and SM Avasthi. Optimizing gas field performance to increase gas production rates and reserves. In *Latin American and Caribbean Petroleum Engineering Conference*, 2009.

[78] R.G. Turner, M.G. Hubbard, and A.E. Dukler. Analysis and Prediction of Minimum Flow Rate for the Continuous Removal of Liquids from Gas Wells. *Journal of Petroleum Technology*, 21(11):1475–1482, 1969.

[79] Susara A Van Den Heever and Ignacio E Grossmann. An iterative aggregation/disaggregation approach for the solution of a mixed-integer nonlinear oilfield infrastructure planning model. *Industrial & engineering chemistry research*, 39(6):1955–1971, 2000.

[80] Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM review*, 38(1):49–95, 1996.

[81] Aldo Vecchietti, Sangbum Lee, and Ignacio E Grossmann. Modeling of discrete/continuous optimization problems: characterization and formulation of disjunctions and their relaxations. *Computers & chemical engineering*, 27(3):433–448, 2003.

[82] Tapio Westerlund and Frank Pettersson. An extended cutting plane method for solving convex MINLP problems. *Computers & Chemical Engineering*, 19(95):131–136, June 1995.

Appendix A

Acronyms

LP Linear Program

QP Quadratic Program

NLP Nonlinear Program

MILP Mixed-Integer Linear Program

MINLP Mixed-Integer Nonlinear Program

BB Branch-and-Bound

FP Feasibility Pump

OFP Objective Feasibility Pump

OA Outer-Approximation

PWL Piecewise Linear

RUC-OA Reverse Undercover Outer-Approximation

RUC-PWL Reverse Undercover Piecewise Linearization

CPU Central Processing Unit

UC Undercover

GOA Generalized Outer-Approximation

ECP Extended Cutting Plane

SPM Successive Projection Method

LP/NLP BB LP/NLP Based Branch-and-Bound

SOS1 Special Ordered Set of Type 1

SOS2 Special Ordered Set of Type 2

ODE Ordinary Differential Equation

PDE Partial Differential Equation

Appendix B

Computational Results of Branch-and-Bound

Table B.1: Computational results Branch-and-Bound, Convex.

Problems	Solution	Time	Nodes	First solution	Time first solution
SLay06M	32757	1.06	110	32929.7	0.95
SLay07H	64748.8	5.1	241	65253.9	3.33
SLay07M	64748.8	2.09	241	65253.9	1.42
SLay08H	84960.2	7.9	269	84960.2	5.94
SLay08M	84960.2	3.07	265	84960.2	2.38
SLay09H	107805.8	16.52	438	108438.6	11.08
SLay09M	107805.8	5.82	387	107805.8	4.39
SLay10H	129579.9	277.59	7902	132240.1	16.25
SLay10M	129579.9	70.32	6682	132663.2	5.83
Syn10M02M	-2310.3	2.98	246	-2289.5	0.84
Syn10M03M	-3354.7	14.06	874	-3328.2	1.78
Syn10M04M	-4557.1	40.9	1946	-4498.3	3.25
Syn10M	-1267.4	0.2	32	-1267.4	0.1
Syn15M02M	-2832.7	10.18	466	-2793.7	1.79
Syn15M03M	-3850.2	66.67	1688	-3811.2	3.92
Syn15M04M	-4937.5	292.58	5056	-4736.7	7.22
Syn15M	-853.3	0.52	70	-836.3	0.18
Syn20M02M	-1752.1	354.69	16690	-1674.6	3.19
Syn20M03M	-2647	3600.12	94327	-2576.3	7.77
Syn20M04M	-3519.7	3600.18	54604	-3468.6	15.84
Syn20M	-924.3	3.71	598	-924.3	0.3

Table B.2: Computational results Branch-and-Bound, Convex.

Problems	Solution	Time	Nodes	First solution	Time first solution
BatchS101006M	769440.4	14.88	442	769440.4	6.26
BatchS121208M	1241126	31.42	594	1241125.5	15.2
BatchS151208M	1543472	89.97	1800	1543877.6	18.18
BatchS201210M	2295349	117.05	1594	2295585.3	30.86
CLay0203H	41573.3	5.04	212	54835.8	0.58
CLay0203M	41573.3	1.97	225	41737.5	0.22
CLay0204H	6545	44.17	1487	71390.2	1.76
CLay0204M	6545	11.55	1802	55213.5	0.41
CLay0205M	8092.5	175.29	18986	8708.9	0.78
CLay0303H	26669.1	11.82	272	54835.8	1
CLay0303M	26669.1	4.26	388	26669.1	0.34
CLay0304M	40262.4	44.83	2920	61511.7	0.56
CLay0305M	8092.5	157.57	14679	36651.1	1.04
Fo7-2	17.7	3337.92	174191	30.5	51.06
Fo7	22.4	3600.21	199737	28.5	23.69
Fo8	34.2	3600.47	136894	34.2	643.67
O7-2	125.7	3600.39	159172	162.1	139.44
O7	141.1	3600.36	146144	146.3	612.69
RSyn0805H	-7174.2	3599.98	32987	-7029.8	50.29
RSyn0805M02M	-2238.4	1765.23	44330	-2154.1	13.74
RSyn0805M03M	-3068.8	3599.98	53683	-3013.3	28.88
RSyn0805M04M	-7174.2	3600.03	32986	-7029.8	50.33
RSyn0810H	-6533.1	3600	19557	-6301.3	71.13
RSyn0810M	-1721.4	205.29	14633	-1651.3	3.28
Syn40M02M	-360.4	3600.58	130978	-273	9.79
Syn40M03M	-331.4	3600.29	61599	-254.3	34.04
Syn40M04M	-796.7	3600.24	35729	-678	54.94
Water0202R	97.9	1.63	26	2690.2	0.56
Water0202	125.2	140.8	24	2159.9	97.79
Water0303R	424.5	231.41	292	3903.5	9.25
Water0303	208	250.76	56	3235.1	134.55

Table B.3: Computational results Branch-and-Bound, Convex.

Problems	Solution	Time	Nodes	First solution	Time first solution
RSyn0810M02M	-1709.7	3600.22	81213	-1649.7	15.18
RSyn0810M03M	-2706.7	3600.1	37858	-2654.4	39.51
RSyn0810M04M	-6533.1	3600.04	19543	-6301.3	71.12
RSyn0815H	-3252.2	3599.99	14222	-3105.3	173.89
RSyn0815M02M	-1752.5	3600.15	40916	-1588.7	32.67
RSyn0815M03M	-2778	3600.01	19188	-2778	70.85
RSyn0815M04M	-3252.2	3600.03	14241	-3105.3	173.77
RSyn0815M	-1269.9	1835.73	99434	-1263.9	4.58
RSyn0820H	-2326.1	3600.08	16208	-2291.8	146.83
RSyn0820M02M	-1022.7	3600.35	59271	-900.8	25.07
RSyn0820M03M	-1955.4	3600.2	34216	-1948.3	69.13
RSyn0820M04M	-2326.1	3600.21	16132	-2291.8	147.1
RSyn0820M	-1150.3	3600.17	199600	-1146	4.01
RSyn0830H	-2369.8	3600.14	17272	-2237.4	204.74
RSyn0830M02M	-701.8	3600.25	51633	-549.3	41.15
RSyn0830M03M	-1368.9	3600.16	27900	-1245.9	106.59
RSyn0830M04M	-2369.8	3600.05	17247	-2237.4	204.8
RSyn0830M	-502.5	3600.57	165909	-483.5	6.23
RSyn0840H	-2248.9	3600	16698	-2185.7	256.83
RSyn0840M02M	-651	3600.2	47727	-491.9	51.86
RSyn0840M03M	-2631.2	3600.12	25456	-2488.5	124.49
RSyn0840M04M	-2248.9	3600.16	16728	-2185.7	256.88
RSyn0840M	-318.6	3600.57	150751	-253.3	7.81
SLay04H	9859.7	0.46	35	9859.7	0.46
SLay04M	9859.7	0.25	35	9859.7	0.25
SLay05H	22664.7	0.94	59	22664.7	0.94
SLay05M	22664.7	0.48	59	22664.7	0.48
SLay06H	32757	2.31	110	32929.7	2.08
Syn30M02M	-396.7	3600.34	130158	-319.9	5.86
Syn30M03M	-619.6	3600.4	93597	-512.7	16.41
Syn30M04M	-783.1	3600.28	60827	-572.1	33.56
Syn30M	-134	16.85	1914	-134	0.72
Syn40M	-67.7	593.98	59564	-46.9	1.25

Table B.4: Computational results Branch-and-Bound, Non-convex.

Problems	Solution	Time	Nodes	First solution	Time first solution
batchdes	167427.6	0.06	2	167427.6	0.06
batch	285506.5	0.38	14	333257.2	0.36
cecil13	-115656.5	3600.06	26894	-115606.7	22.32
csched-1	-30639.3	0.81	76	-29279.2	0.47
deb10	209.4	0.34	20	209.4	0.22
deb6	201.7	45.55	60	261.7	0.7
deb7	NA	NA	NA	NA	NA
deb8	NA	NA	NA	NA	NA
deb9	NA	NA	NA	NA	NA
eniplac	-132117.1	388.32	22472	-130773.2	1.3
enpro48pb	187277.3	4.5	215	200059.5	2.4
enpro48	187277.3	4.6	219	200059.5	2.36
enpro56b	263428.3	7.98	488	280209.4	2.41
enpro56	263428.3	7.75	490	280209.4	2.21
ex1252	128893.7	1.13	50	134263.6	0.71
ex1263	19.6	21.84	2246	12171	4.81
ex1264	8.6	48.64	5934	13.3	1.63
ex1265	10.3	22.49	1340	11.3	2.44
ex1266	16.3	5.04	62	16.3	5.04
ex3	68	0.09	12	76.4	0.07
fossolo-iron	181074.9	3596.36	25893	186543.9	324.93
gasnet	NA	NA	NA	NA	NA
gastrans	89.1	0.58	15	89.1	0.58
hanoi	6109621	151.73	5872	6216252	15.72
johnall	-224.7	0.02	0	-224.7	0.02
lop97icx	4104.5	3599.87	25274	4143.7	404.43
mbtd	NA	NA	0	NA	NA
nous1	1.6	0.29	0	1.6	0.29
nous2	0.6	0.33	1	0.6	0.33
oil2	-0.7	0.88	4	-0.7	0.6
oil	-0.9	3603.15	3091	-0.8	706.66
pescara	NA	NA	NA	NA	NA
shamir	419999.9	4.34	96	419999.9	4
trimlon2	5.3	0.53	139	5.3	0.22
trimlon4	8.5	3600.42	634149	10.8	0.72
trimlon5	10.7	3600.69	508248	11.7	3.43
trimlon6	15.5	3600.46	460732	24.2	1.94
trimlon7	NA	NA	NA	NA	NA

Table B.5: Computational results Branch-and-Bound with Feasibility Pump, Convex.

Problems	Solution	Time	Nodes	First solution	Time first solution
SLay06M	32757	1.08	110	142039.9	0.02
SLay07H	64748.8	5.19	241	280057.8	0.06
SLay07M	64748.8	2.1	241	328823.6	0.02
SLay08H	84960.2	8.02	269	653767.7	0.08
SLay08M	84960.2	3.15	265	471470.4	0.03
SLay09H	107805.8	16.71	438	704132.2	0.09
SLay09M	107805.8	5.86	387	576089.2	0.03
SLay10H	129579.9	277.75	7902	762238.2	0.14
SLay10M	129579.9	70.1	6682	600245.5	0.04
Syn10M02M	-2310.3	3.07	246	-965.6	0.02
Syn10M03M	-3354.7	14.22	874	-2301.1	0.04
Syn10M04M	-4557.1	40.8	1946	-2936.2	0.04
Syn10M	-1267.4	0.24	30	-1239.4	0.02
Syn15M02M	-2832.7	10.22	466	-165.9	0.02
Syn15M03M	-3850.2	66.9	1688	-285.4	0.04
Syn15M04M	-4937.5	292.63	5056	-408.2	0.06
Syn15M	-853.3	0.55	70	-811.2	0.01
Syn20M02M	-1752.1	354.37	16702	-636.7	0.04
Syn20M03M	-2647	3600.15	94041	-1006.6	0.06
Syn20M04M	-3519.7	3600.14	54578	-1393.9	0.08
Syn20M	-924.3	3.77	596	-904.3	0.02

Table B.6: Computational results Branch-and-Bound with Feasibility Pump, Convex.

Problems	Solution	Time	Nodes	First solution	Time first solution
BatchS101006M	769440.4	15	442	821554.2	0.08
BatchS121208M	1241125.5	32.12	600	1323739	0.1
BatchS151208M	1543472.3	91.06	1792	1621631.2	0.17
BatchS201210M	2295348.7	119.57	1594	3273666.3	1.46
CLay0203H	41573.3	5.28	212	41709.8	0.33
CLay0203M	41573.3	1.99	225	56141.5	0.03
CLay0204H	6545	44.73	1487	74266.8	0.92
CLay0204M	6545	11.57	1802	72513.8	0.06
CLay0205M	8092.5	176.52	18986	24441.7	0.12
CLay0303H	26669.1	18.16	385	41737.5	1.81
CLay0303M	26669.1	5.38	388	47287.6	1.14
CLay0304M	40262.4	45.52	2920	78542.2	0.81
CLay0305M	8092.5	420.59	20258	11136.9	0.02
Fo7-2	17.7	3404.62	160599	40.4	18.27
Fo7	22.4	3600.22	199390	38.9	1.41
Fo8	32.5	3600.5	138022	45.2	60.24
O7-2	125.7	3600.39	158843	178.7	13.7
O7	140.4	3600.37	146282	168.5	26.97
RSyn0805H	-7174.2	3599.98	32908	-4873.6	0.18
RSyn0805M02M	-2238.4	1766.69	44330	-1414.8	0.07
RSyn0805M03M	-3068.8	3599.97	53625	-2281	0.12
RSyn0805M04M	-7174.2	3599.98	32905	-4873.6	0.18
RSyn0810H	-6533.1	3600.21	19430	-4612	0.26
RSyn0810M	-1721.4	205.21	14633	-1423.9	0.03
Syn40M02M	-360.4	3600.58	130671	-319.6	0.06
Syn40M03M	-336	3600.3	61511	-267.2	0.1
Syn40M04M	-794.7	3600.18	35647	-769.2	0.15
Water0202R	97.9	1.83	26	5076.5	0.22
Water0202	125.2	147.03	24	3190.5	73.13
Water0303R	424.5	243.94	292	8311	11.7
Water0303	208	257.99	56	3235.1	76.32

Table B.7: Computational results Branch-and-Bound with Feasibility Pump, Convex.

Problems	Solution	Time	Nodes	First solution	Time first solution
RSyn0810M02M	-1709.7	3600.22	81206	-716.6	0.1
RSyn0810M03M	-2706.7	3600.14	37807	-1989.3	0.17
RSyn0810M04M	-6533.1	3600.09	19472	-4612	0.26
RSyn0815H	-3252.2	3599.98	14182	-600.4	0.38
RSyn0815M02M	-1752.5	3600.18	40844	4.7	0.11
RSyn0815M03M	-2778	3600.24	19079	-726.6	0.17
RSyn0815M04M	-3252.2	3600.08	14190	-600.4	0.38
RSyn0815M	-1269.9	1849.69	99434	-983.3	0.06
RSyn0820H	-2326.1	3600.23	16158	-946.6	0.44
RSyn0820M02M	-1022.7	3600.25	59136	-139.7	0.16
RSyn0820M03M	-1955.4	3600.21	34124	-967.8	0.2
RSyn0820M04M	-2326.1	3600.6	16162	-946.6	0.44
RSyn0820M	-1150.3	3600.17	199305	-829.7	0.05
RSyn0830H	-2369.8	3600.04	17258	-706.1	0.4
RSyn0830M02M	-701.8	3600.24	51132	-69.6	0.19
RSyn0830M03M	-1368.9	3600.16	27563	-308.1	0.34
RSyn0830M04M	-2369.8	3600.05	17228	-706.1	0.39
RSyn0830M	-502.5	3600.61	165715	-205.2	0.07
RSyn0840H	-2248.9	3600.07	16677	-696.8	0.77
RSyn0840M02M	-651	3600.24	47661	-38.9	0.17
RSyn0840M03M	-2631.2	3600.18	25397	-1718.8	0.87
RSyn0840M04M	-2248.9	3600.11	16706	-696.8	0.77
RSyn0840M	-318.6	3600.56	150674	58.4	0.07
SLay04H	9859.7	0.45	35	45331.6	0.02
SLay04M	9859.7	0.24	35	54457.1	0.01
SLay05H	22664.7	0.97	59	91395.5	0.03
SLay05M	22664.7	0.5	59	109670.1	0.02
SLay06H	32757	2.36	110	128394.4	0.04
Syn30M02M	-395.3	3600.33	129982	-346.8	0.05
Syn30M03M	-619.6	3600.4	93574	-605.1	0.08
Syn30M04M	-787.1	3600.28	60481	-787.1	0.12
Syn30M	-134	16.93	1914	-116.3	0.02
Syn40M	-67.7	597.91	59564	-23.2	0.03

Table B.8: Computational results Branch-and-Bound with Feasibility Pump, Non-convex.

Problems	Solution	Time	Nodes	First solution	Time first solution
batchdes	167427.6	0.05	0	167427.6	0.01
batch	285506.5	0.29	2	285506.5	0.02
cecil13	-115656.5	3600.96	26717	NA	NA
csched-1	-30639.3	0.83	76	NA	NA
deb10	209.4	0.32	10	209.4	0.02
deb6	201.7	45.53	60	NA	NA
deb7	NA	NA	NA	NA	NA
deb8	NA	NA	NA	NA	NA
deb9	NA	NA	NA	NA	NA
eniplac	-132117.1	389.1	22472	-130323.2	0.04
enpro48pb	187277.3	3.42	128	194392.4	0.04
enpro48	187277.3	3.47	128	194392.4	0.04
enpro56b	263428.3	8.05	490	284142	0.05
enpro56	263428.3	7.91	492	284142	0.04
ex1252	128893.7	1.19	50	204321.6	0.02
ex1263	19.6	23.53	2246	NA	NA
ex1264	8.6	51.28	5934	NA	NA
ex1265	10.3	24.55	1340	NA	NA
ex1266	16.3	7.6	62	NA	NA
ex3	68	0.1	12	113.4	0.01
fossolo-iron	180599.4	3596.02	24119	955497.5	61.56
gasnet	NA	NA	NA	NA	NA
gastrans	89.1	0.03	0	89.1	0.03
hanoi	6109621	143.65	6459	6871124	1.41
johnall	-224.7	0.02	0	NA	NA
lop97icx	4104.5	3599.88	25356	NA	NA
mbtd	NA	NA	NA	NA	NA
nous1	1.6	0.38	0	NA	NA
nous2	0.6	0.35	1	NA	NA
oil2	-0.7	0.72	0	-0.7	0.32
oil	-0.9	3600.08	3421	-0.9	0.39
pescara	NA	NA	NA	NA	NA
shamir	419999.9	6.19	128	923999.9	1.05
trimlon2	5.3	0.52	139	NA	NA
trimlon4	8.5	3600.41	637208	NA	NA
trimlon5	10.7	3600.69	507349	NA	NA
trimlon6	15.5	3600.46	460517	NA	NA
trimlon7	NA	NA	NA	NA	NA

Appendix C

Plots of Example Solutions From The Second Set of Test Problems

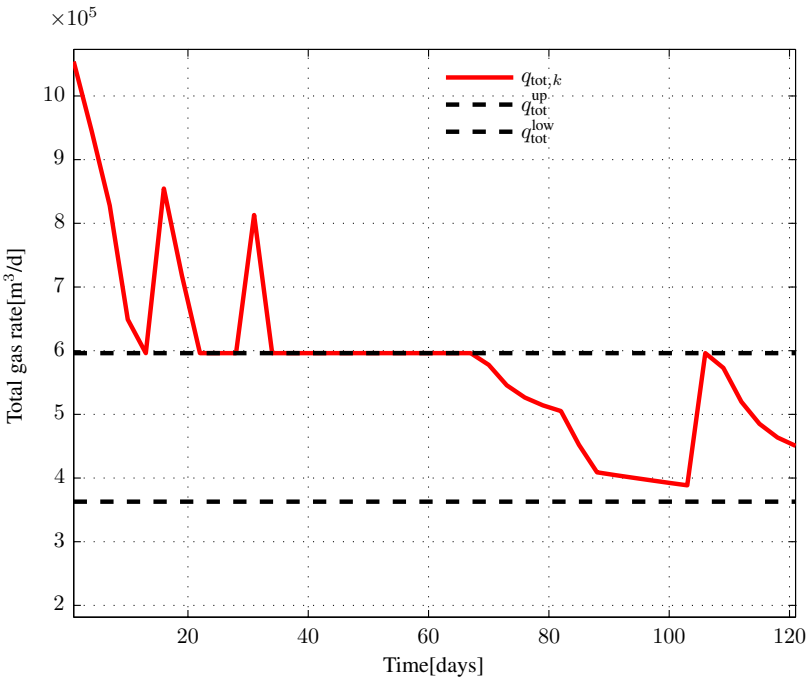

Figure C.1: Total gas rate on problem Case2qgcShigh.

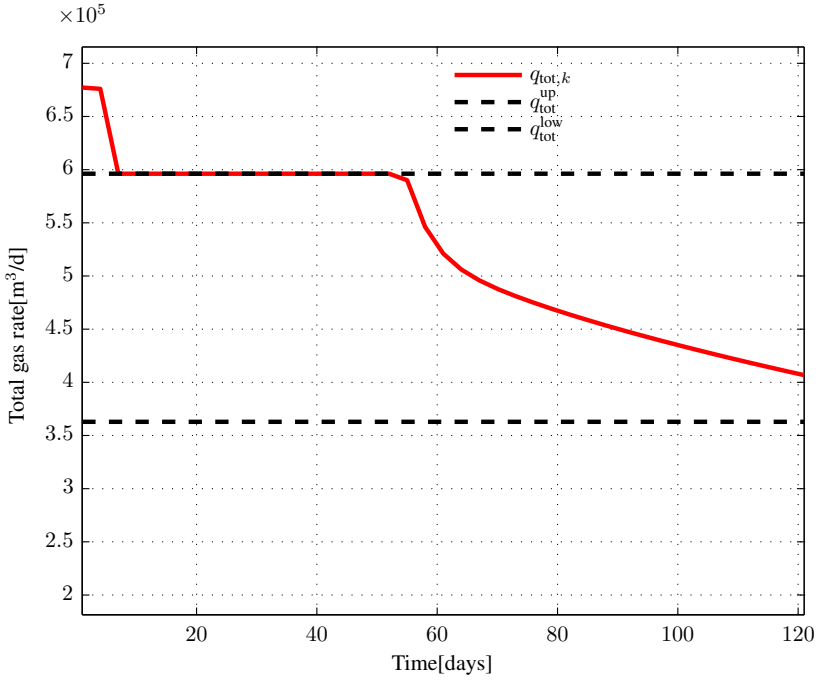

Figure C.2: Total gas rate on problem Case2qgcSlow.

Appendix D

Additional Convex Relaxations of the Shale gas Production Optimization Problem

In this chapter, two additional convex relaxations of the shale gas production optimization problem from Chapter 4 are presented.

D.1 Penalty Function

By moving the equality constraint (4.1.5) into the objective function and by squaring and summing over all indices, a penalty function [69] relaxation of the problems ($\mathrm{P}_{\mathrm{switch,ncvx}}$) and ($\mathrm{P}_{\mathrm{gc,ncvx}}$) can be obtained. In this case, the objective function can be expressed as:

$$\min\ -G_p \sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{K}\setminus K} (q_{jk} - 0.05 v_{jk}) + \mu \sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{K}} \left(e^{\mathrm{S}} \left(p_{\mathrm{t},jk}^2 + \frac{1}{C_{\mathrm{t}}^2} q_{jk}^2 \right) - \bar{p}_{jk} \right)^2,$$

where $\mu > 0$ is a *penalty parameter*. The problem has been cast into a minimization problem to allow for the penalty formulation. Alternatively, a nonsmooth penalty function can be formulated by using the ℓ_1 penalty function

$$\min\ -G_p \sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{K}\setminus K} (q_{jk} - 0.05 v_{jk}) + \mu \sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{K}} \left| e^{\mathrm{S}} \left(p_{\mathrm{t},jk}^2 + \frac{1}{C_{\mathrm{t}}^2} q_{jk}^2 \right) - \bar{p}_{jk} \right|.$$

Both of these formulations were written as an AMPL model, however, BONMIN was unsuccessful in solving these models. The error message generated by IPOPT cited a lack of convergence as the cause.

A penalty formulation is not an ideal formulation since it alters the intrinsic properties of the objective function, and consequently the optimization problem as a whole.

D.2 Semidefinite Relaxation

A special class of mixed-integer optimization problems which have a quadratic objective function, quadratic inequality constraints and possibly affine equality constraints are called *mixed-integer quadratically constrained quadratic programs*(QCQP)

$$\begin{aligned} z = \min_x \quad & \frac{1}{2}x^T P_0 x + c_0^T x \\ \text{s.t} \quad & \frac{1}{2}x^T P_i x + c_i^T x + r_i \leq 0, && i = 1, \ldots, p \\ & x_i \in \{0, 1\}, && i = 1, \ldots, q \\ & x_i \in \mathbb{R}, && i = q+1, \ldots, n \\ & Ax = b. \end{aligned} \tag{$\mathrm{P}_{\mathrm{QCQP}}$}$$

If the matrices $P_0, \ldots, P_p$ are positive semidefinite, i.e. $u^T P_i u \geq 0$, for $i = 0, \ldots p$ for all $u \in \mathbb{R}^n$, then the continuous relaxation of ($\mathrm{P}_{\mathrm{QCQP}}$) obtained by removing the integrality constraint is a convex optimization problem.

The problem ($\mathrm{P}_{\mathrm{QCQP}}$) can be expressed as a semidefinite program(SDP) [80] in which a linear objective is minimized subject to a matrix inequality

$$\begin{aligned} z = \min_x \quad & c^T x \\ \text{s.t} \quad & F(x) \geq 0, \end{aligned} \tag{D.2.1}$$

where $F(x) := F_0 + \sum_{i=1}^{p} F_i(x)$ and the matrices $F_0, \ldots, F_p \in \mathbb{R}^{n \times n}$ are symmetric. It has been shown that semidefinite programming can provide strong convex relaxations of hard optimization problems such as nonconvex QCQPs [42] and *zero-one quadratic programs*(0-1 QP) [49]. In a recent paper [16] it was shown that a method which relies on SDP relaxations of 0-1 QPs outperformed CPLEX on a set of difficult combinatorial optimization problems. Furthermore, a comparison of the QP-relaxation and two different

SDP relaxations of a mixed-integer predictive control problem showed that the SDP relaxation provides the best lower bound at the cost of an increased computation time [7].

The main principle behind SDP formulations of quadratic problems is to introduce a symmetric matrix variable $X \in \mathbb{R}^{n\times n}$ which is defined as the outer product of the variables, i.e. $X = xx^T$. By using this variable in ($\mathrm{P}_{\mathrm{QCQP}}$), the problem can be reformulated as

$$\begin{aligned} z = \min_x \quad & \frac{1}{2}\mathbf{Tr}P_0X + c_0^Tx \\ \text{s.t} \quad & \frac{1}{2}\mathbf{Tr}P_iX + c_i^Tx + r_i \leq 0, \qquad i = 1,\dots,p \\ & x_i \in \{0,1\}, \qquad i = 1,\dots,q \\ & x_i \in \mathbb{R}, \qquad i = q+1,\dots,n \\ & Ax = b. \\ & X = xx^T, \end{aligned} \tag{D.2.2}$$

where **Tr** denotes the *trace*-operator. The outer product constraint is a nonconvex constraint, and is therefore relaxed to an inequality $X \geq xx^T$ which can be written as

$$\begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \geq 0, \tag{D.2.3}$$

by using the *Schur complement*(see Appendix C.4 in [23]). A binary variable can be enforced by the nonlinear equality constraint

$$x_i(1-x_i) = 0$$

with a continuous variable x_i, which is equivalent to the linear constraint

$$X_{ii} = x_i. \tag{D.2.4}$$

Combining the relaxation (D.2.3) and the constraint (D.2.4) into (D.2.5) yields the following SDP relaxation of ($\mathrm{P}_{\mathrm{QCQP}}$)

$$\begin{aligned} z = \min_x \quad & \frac{1}{2}\mathbf{Tr}P_0X + c_0^Tx \\ \text{s.t} \quad & \frac{1}{2}\mathbf{Tr}P_iX + c_i^Tx + r_i \leq 0, \qquad i = 1,\dots,p \\ & X_{ii} = x_i, \qquad i = 1,\dots,q \\ & x_i \in \mathbb{R}, \qquad i = q+1,\dots,n \\ & Ax = b. \\ & \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \geq 0, \end{aligned} \tag{D.2.5}$$

A SDP relaxation of (4.1.13) can be obtained by rewriting the constraint (4.1.5) through introducing a symmetric matrix variable $\tilde{P}_{jk} := \tilde{p}_{jk}\tilde{p}_{jk}^T$ and adding its relaxation as shown in (D.2.3) to the problem formulation, where

$$\tilde{p}_{jk} := \begin{pmatrix} p_{\mathrm{t},jk} \\ q_{jk} \end{pmatrix}.$$

The constraint (4.1.5) can then be written as

$$\mathbf{Tr}(C\tilde{P}jk) - \bar{p}_{jk} = 0 \qquad \forall j \in \mathcal{J}, k \in \mathcal{K},$$

where

$$C := \begin{pmatrix} e^S & 0 \\ 0 & \frac{e^S}{C_{\mathrm{t}}^2} \end{pmatrix}$$

Similarly, a symmetric matrix variable $Y^1 \geq y^1 y^{1^T}$ for the binary variables can be included in the problem formulation with the constraints (D.2.3) and (D.2.4).