NTNU

Norwegian University of
Science and Technology

Data Driven Real-Time Petroleum Production Planning Using Optimization and Neural Networks

Arnt Gunnar Malvik

Bendik Witzøe

Industrial Economics and Technology Management
Submission date: June 2018
Supervisor: Henrik Andersson, IØT
Co-supervisor: Bjarne Grimstad, Solution Seeker

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

Problem Description

Real-time optimization of petroleum production is a challenging problem. Rapidly changing conditions demand short solution times, while complex reservoir dynamics lead to complicated models. Uncertainty in the problem requires a decision maker to consider the trade-off between maximizing oil production and remaining feasible with respect to gas capacity constraints.

The objective of this thesis is to develop real-time optimization models able to provide decision support in production planning of a petroleum field. The risk aversion of the decision maker varies with field conditions. Consequently, this thesis aims to provide decision support enabling the identification of an optimal solution given variable appetite for risk. An important goal is to keep solution times sufficiently low in a real-time setting. To this end, the models are based on historical data from the field. The models developed in this thesis are tested on a simplified representation of a real oil field from the Norwegian Continental Shelf.

Well characteristics are modeled with neural networks. By combining modern techniques in machine learning, estimates of uncertainty both in the data and in the models are obtained. The neural networks are incorporated into the optimization models. The value of planning with uncertainty is demonstrated in several optimization settings, and for variable levels of risk aversion.

Preface

This thesis constitutes the completion of our Master of Science degrees in Industrial Economics and Technology Management at the Norwegian University of Science and Technology. The thesis is a continuation of our project report, written as a part of the subject TIØ4500 - Managerial Economics and Operations Research in the fall of 2017.

Our motivation for choosing to continue the work on this problem is in large part due to our interest in techniques from machine learning, with successful applications to a wide variety of optimization problems. We find the prospect of combining elements from machine learning and optimization an attractive opportunity to improve our knowledge and skill in both fields.

We would like to express our gratitude to our industry partner Solution Seeker AS, for their guidance and insight on petroleum optimization. Their accommodating and generous demeanor, providing us with the opportunity to work out of their offices, is highly appreciated. We would especially like to thank the main representative from Solution Seeker, Bjarne Grimstad, for his swift follow-up and helpful comments. We would also like to thank Professor Henrik Andersson for assisting and providing excellent and valuable advice throughout the work on this thesis.

Summary

Real-time optimization (RTO) of petroleum production is concerned with maximizing daily oil production by suggesting minor adjustments to the system and frequently re-solving the optimization model. A central part of the problem consists of modeling well production outputs as a function of decision variable inputs.

Most petroleum production fields have an upper limit on the rate of gas they are able to handle under normal conditions. Providing decision support for adhering to this constraint is complicated by uncertainty in the gas output rates of wells, stemming from uncertain measurement data and the well model approximations to the data. Accounting for both sources is necessary in order to avoid underestimating the total uncertainty.

A maximally risk-averse system operator seeks to avoid gas capacity constraint breaches at all times. However, if the utility of the operator is unknown, there exists a set of Pareto optimal solutions that describe the trade-off between petroleum production and uncertainty in the gas output. In this thesis, we formulate a robust stochastic mixed integer linear program (MILP) for the case of maximum risk aversion, and a multi-objective optimization problem (MOP) which models the oil production and gas uncertainty trade-off.

We propose a novel model of well outputs as a function of inputs, combining state-of-the-art neural network (NN) techniques and thereby obtaining a statistically grounded distribution of outputs, capturing both data and model uncertainty. The NN well models are reformulated as MILPs and included in the optimization problems. Two scenario generation procedures based on the NN well models are presented, producing Factor and Markov Weighted scenarios. A recourse algorithm (RA) is developed, modeling the discovery of new information as operational changes are implemented. We test the models developed in this thesis on a simplified representation of a real oil field.

Scenarios are represented by NN MILPs or special ordered sets. Results show NN MILPs offer poor scalability as the number of scenarios in the problem increases. Identified weaknesses in the Factor scenarios cause excessively conservative solutions and make them incompatible with cases where initial production levels are known. The value of planning with uncertainty in the robust problem depends on the choice of a suitable penalty metric for infeasible solutions, but is found to be significant. The RA offers significant performance gains over the standard robust approach in certain cases. However, performance estimates are slightly optimistic due to simplifications in the RA implementation.

MOP results prove to be suited for identifying the optimal solution for a given level of risk aversion. In particular, we obtain confidence intervals for the probability of a gas capacity constraint breach given a specified minimum oil output. Convergence of the RA and the MOP results indicate that Markov Weighted scenarios are a sensible representation of the distribution of the uncertainty in the problem.

Sammendrag

Sanntidsoptimering (RTO) av petroleumsproduksjon omhandler maksimering av daglig oljeproduksjon ved å foreslå små justeringer av produksjonssystemet, og å hyppig oppdatere og løse optimeringsproblemet. Modellering av sammenhengen som beskriver brønnproduksjon som en funksjon av beslutningsvariable er en sentral del av problemet.

De fleste oljefelt har øvre begrensninger på raten av gass systemet er i stand til å håndtere under normale forhold. Beslutningsstøtte som søker å overholde disse begrensningene må forholde seg til usikkerhet i brønnproduksjon. Denne usikkerheten har opphav i usikre brønnmålingsdata og usikkerhet som innføres under modelleringen av brønnene. Optimal beslutningsstøtte bør ta hensyn til begge kilder.

En maksimalt risikoavers systemoperatør ønsker å unngå å bryte øvre begrensninger på gassproduksjon til enhver tid. For en operatør med variabel risikoaversjon må risikoen for å bryte begrensningene avveies mot maksimeringen av oljeproduksjon. For denne operatøren er optimale beslutninger beskrevet av et sett bestående av Paretooptimale løsninger. I denne oppgaven formulerer vi et robust stokastisk blandet heltallsproblem (MILP) med hensyn på maksimal risikoaversjon. Et multiobjektiv-problem (MOP) modellerer avveiingen mellom oljeproduksjon og usikkerhet i gassproduksjon.

Vi foreslår en ny modell for brønnproduksjon basert på å kombinere moderne teknikker innen nevrale nettverk (NN). Brønnmodellene estimerer distribusjoner for produksjonsrater, og fanger opp usikkerhet i både data og modeller. Arkitekturen i de nevrale nettene er strukturert slik at en reformulering til et NN-MILP er mulig. Følgelig inkluderes brønnmodellene i optimeringsproblemene som formuleres i denne oppgaven. Basert på den estimerte distribusjonen av brønnproduksjon, utvikles prosedyrer for å generere to distinkte typer scenarier: Faktorscenarier og Vektet Markov-scenarier. En regressalgoritme (RA) er utviklet for å modellere utnyttelsen av ny informasjon som blir tilgjengelig ettersom operasjonelle endringer i feltet utføres. Modellene i denne oppgaven er testet på en forenklet fremstilling av et oljefelt på norsk sokkel.

Et scenario er enten formulert som et NN-MILP eller som et ordnet sett (SOS2). Resultater i det tekniske studiet i denne oppgaven tyder på at NN-MILP er lite skalerbart dersom antall scenarier i optimeringsproblemet øker. Flere svakheter med Faktorscenarier påvises. Sterk bias i kurven for brønnproduksjon fører til overdrevent konservative løsninger av det robuste stokastiske problemet når denne typen scenarier benyttes. Videre er Faktorscenarier inkompatible med spesifisering av kjente produksjonspunkt.

Vi viser at verdien av å planlegge med usikkerhet i det robuste problemet er signifikant, men sterkt avhengig av en veldefinert straff av brudd på begrensninger i problemet. Ved bruk av RA-en øker verdien av den optimale løsningen, og i enkelte tilfeller er økningen

svært lovende. Estimatene på økningen er imidlertid optimistiske, ettersom forenklinger er gjort i implementeringen av RA-en.

Resultater fra MOP-et synliggjør optimale løsninger for alle nivåer av risikoaversjon, og lar oss beregne sannsynligheten for at en gitt løsning overholder begrensninger på gassproduksjon. Sammenstillinger av resultater fra RA-en med Vektet Markov-scenarier og MOP-et, viser at begge modeller konvergerer mot tilnærmet samme løsninger. MOP-et er basert direkte på distribusjonen av usikkerhet slik den er estimert av brønnmodellene. Den tilnærmede konvergensen er derfor en indikasjon på at Vektet Markov-scenarier innehar de samme statistiske kvalitetene som denne distribusjonen, noe som er en ønskelig egenskap ved prosedyrer for å generere scenarier.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Petroleum is one of the main sources of energy in the world as well as the main ingredient in a wide range of products used in day-to-day life. Among the most important applications of petroleum are energy for transportation, industrial power, heating and lighting, and production of fertilizers. The use of petroleum resources for energy production causes emissions damaging the environment. As a consequence, the use of alternative technologies to harvest energy from renewable sources is becoming increasingly widespread. Nevertheless, oil and gas still constitute around 56 % of the total energy consumption in the world according to estimates from the International Energy Agency (2018).

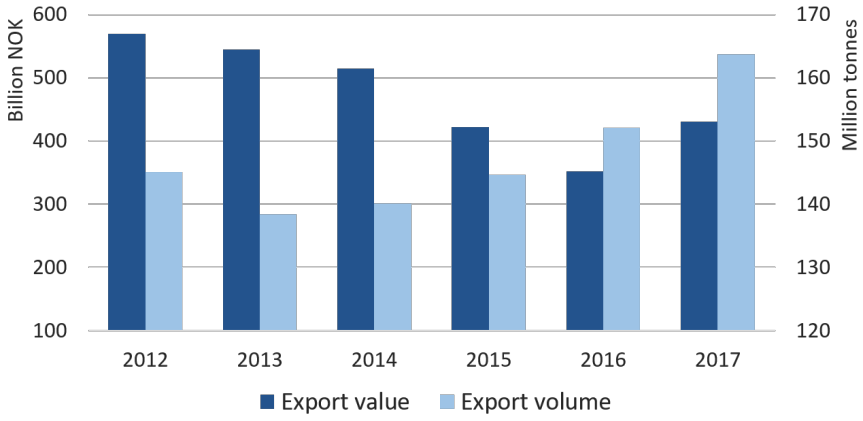

Figure 1.1: Volume of Norwegian export of crude oil and natural gas in gas form (SSB Statistikkbanken, 2018).

The oil industry is a vital part of the Norwegian economy. However, in the last five years, the total value of the Norwegian petroleum export sees a decline of almost 25 % (see Fig-

ure 1.1). This downturn is largely a consequence of increased shale oil activity in the USA and increased production in the Middle East, resulting in an oversupply in the market and a drop in oil price over the recent years. Despite a drop in the price of oil, an increase is seen in the volume of oil and gas exported from Norway over the last four years. More than half of the total valued export from the country in 2017 stems from oil and gas products (SSB Statistikkbanken, 2018).

With lower oil prices, margins of oil companies decline. Development of new oil fields requires large capital expenditures, which companies face with reluctance in times with uncertain or low oil prices (PwC, 2017). Thus, increasing the profitability and production efficiency of existing fields is an increasingly important task for players in the petroleum industry.

As production efficiency is a top priority for petroleum production entities, short-term operational planning problems receive much attention. While operational planning is concerned with maximizing production outputs on a daily basis, rapidly shifting conditions during production means even a daily plan may fail to adapt properly. Thus, operators may wish to update and re-solve the planning problem several times each day. Increasing computational power coupled with modern sensor equipment makes such an approach possible, giving rise to the field of *real-time optimization* (RTO).

Machine learning techniques are widely used in the RTO of oil fields. In the existing literature, this often involves the use of heuristics to produce field wide solutions. Commonly used techniques include neural networks (NNs), genetic algorithms and ant colony optimization. NNs are also used in pre-processing steps, typically modeling well production functions. Recent advances in the literature regarding NNs focus on the prediction of a target distribution rather than point estimates, since the latter potentially is a poor representation of reality in cases where uncertainty is present. Predicting a distribution facilitates the inclusion of uncertainty into the problem setting in which the NN is implemented. However, uncertainty estimates are rarely included in the application of NNs in the RTO of oil fields. Consequently, improvements to the RTO of oil fields are achievable by adopting modern NN techniques for estimating uncertainty.

This thesis explores the possibilities of such modern NN techniques, with the purpose of providing decision support to a production engineer aiming to maximize oil production while adhering to constraints on the gas production. Constraint breaches are potentially costly, and avoiding breaches is complicated by uncertainty regarding the output rates of wells. As a result, the engineer is faced with a trade-off between risking infeasibility and maximizing output rates. Since the risk aversion of the decision maker varies with field conditions, this thesis aims to provide decision support accommodating the preferences of the field engineer at any given time.

We formulate a robust stochastic mixed integer linear program (MILP) which models a risk-averse decision maker, and a multi-objective program (MOP) which models the varying risk aversiveness of the field engineer. Well outputs as a function of inputs are modeled

using NNs trained on historical measurement data from the field. The data exhibits significant uncertainty in the form of noise, and in estimating the well production rates, we introduce model uncertainty to the problem. Combining methods for estimating uncertainty in both the data and models, we obtain statistically grounded approximations to the distributions of the well output functions. The NN well models are constructed such that a reformulation of the networks as MILPs is possible, enabling the inclusion of these models into the optimization programs. Distribution estimates from the NNs facilitate the development of scenario generation procedures in the robust stochastic MILP.

To our knowledge, this thesis represents the first work to utilize NNs to incorporate estimates of the uncertainty in both measurement data and well models in the RTO of petroleum fields. Furthermore, the representation of NNs as MILPs is a novel approach to the use of machine learning techniques in optimization problems providing decision support to a petroleum field engineer.

This thesis is developed in cooperation with Solution Seeker, a Norwegian start-up company specializing in artificial intelligence and optimization software for RTO of upstream petroleum and gas production. The models developed in this report are applied to a simplified representation of an oil field on the Norwegian Continental Shelf.

Following the introduction, relevant background is presented in Chapter 2. A literature study follows in Chapter 3, and an overview of relevant theoretical topics is given in Chapter 4. Chapter 5 presents a description of the problem considered in this thesis. The optimization models are formulated in Chapter 6, and the NN well models are presented in Chapter 7. The implementation and solution method of these models are described in Chapter 8. In Chapter 9, case specific data is presented along with a brief analysis, before a computational study follows in Chapter 10. Finally, concluding remarks are offered and future research suggested in chapters 11 and 12, respectively.

Chapter 2

Background

In this chapter, relevant background to the topics presented in this thesis is introduced. The sections are kept relatively brief and high-level as a thorough discussion is beyond the scope of this thesis. A more comprehensive introduction to background topics in petroleum production optimization can be found in Grimstad (2015).

The chapter is organized as follows. Section 2.1 introduces general nomenclature in the petroleum production industry and presents a typical oil field structure. Next, Section 2.2 introduces some important aspects regarding pressure in the production system. Section 2.3 discusses optimization over different time horizons.

2.1 The Petroleum Production System

An offshore production system consists of several interconnected modules with corresponding interdependencies. The general structure of an oil field is shown in Figure 2.1. The platform, unless unmanned, is the workplace of the well operator. Each platform has one or several wells, pumping up oil, gas and water from a reservoir. The well operator controls the well by adjusting its available parameters, e.g., the current rate of gas lift injection and the choke opening. The choke level is a measure of the opening size of the valve in the well head through which oil, gas and water flow. When the choke of a wellhead is shut, there is no output from the given well. Gas lift is the general term for the use of pressurized gas to supplement natural formation gas in the reservoir with the aim of lifting the fluids contained in the reservoir by increasing the pressure. The reservoir is the subsurface structure where the oil and gas are located prior to extraction and is thereby the target for well drilling. It is defined by Schlumberger Limited (2017) as *"a subsurface body of rock having sufficient porosity and permeability to store and transmit fluids"*.

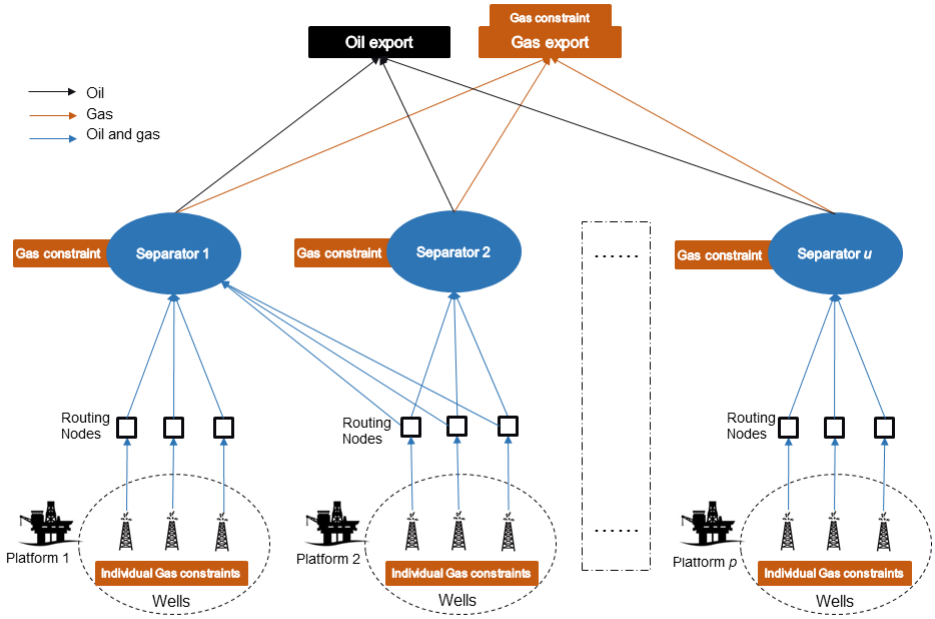

Figure 2.1: General structure of a petroleum production field. Wells belonging to certain platforms may route their output to several separators. Gas rate capacity constraints may be present at several points in the production flow.

The flow coming from the well is a mixture of water, oil and gas, which is separated into its respective parts (or simply into fluids and gas) in a separator before any other post-processing is performed. The flows from a given set of wells are either merged in a manifold and then fed into one of the connected separators or routed directly from the well to a separator. Separators and wells usually have upper flow limits, which the onshore team of production and reservoir engineers needs to take into account. This upper flow limit may be a physical limitation of the separator or well itself, or be a result of limitations downstream of the separator.

When upper flow limits on gas are exceeded, equipment is worn down at a higher rate than under normal conditions. There is also a risk of expensive equipment being destroyed, such as the multiphase flow meters (MPFM). The MPFMs are used to gauge the production of a single well. Even though the data from an MPFM is not always accurate, it is often used as a basis for operational decisions. It is therefore essential to keep the device functional. However, when damages to the equipment do occur, it can be expensive or for all practical purposes impossible to repair or replace it (Dahl et al., 2005). Such accidents also cause a greater workload for the operators and engineers, and exceeding system flow restrictions is therefore in general avoided.

An onshore team receives production data from the wells in the reservoir and analyzes it, before communicating to the well operator a set of adjustments to some input parameters to be executed. This set may be a sequential set of actions, or actions in no specific order. In the process of analyzing the data, many petroleum engineers use optimization software in addition to their experience and knowledge. This software ranges from commercial spreadsheets to sophisticated optimization models and machine learning applications. The flow of information described in this paragraph is visualized graphically in Figure 2.2.

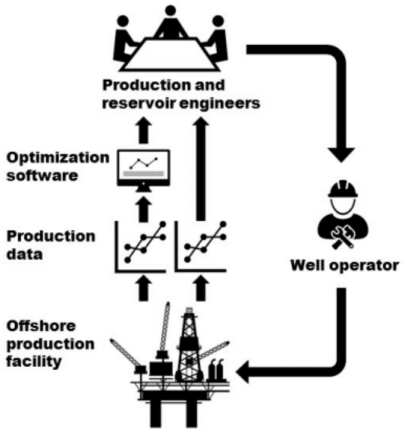

Figure 2.2: The flow of information in a production system, Morken and Sandberg (2016).

2.2 Well Characteristics

As oil flows out of a reservoir through a well, it is expected that some amount of natural gas flows out alongside it. A common metric used to describe a petroleum well is its gas-oil ratio (GOR), which denotes the volume of gas flowing out relative to the volume of oil flowing out under standard conditions, as defined by the Norwegian Ministry of Petroleum and Energy and the Norwegian Petroleum Directorate (2018).

At early stages of their life cycle, wells are typically categorized as *naturally flowing*. A naturally flowing well is one capable of sustaining high production levels without requiring the assistance of enhanced oil recovery (EOR) techniques, due to high pressure in the reservoir to which the well is connected. As the well and reservoir mature, reservoir pressure may drop to a level at which the well is no longer naturally flowing due to the extraction of liquids and gas. At this point, in order to continue producing the well must either be turned into a *pumping well* or some other means of artificial lift must be introduced. Brown (1982) provides an overview of the different artificial lift methods available.

Operating a well is the process of tuning available decision variables in order to control its production rates of various phases. We make a distinction between *explicit* and *implicit* decision variables. Explicit decision variables denote variables over which the well operators have direct control. Examples of explicit decision variables include the choke opening, gas lift injection rates and the routing decision for each well. Implicit decision variables refer to variables over which the production engineer has no direct control. Such parameters include a variety of physical parameters, such as the bottom-hole pressure, the temperature in the wellhead and back-pressure effects. Implicit variables may not be observable through measurements or sensors. Tuning explicit variables affect the implicit variables; increasing the gas lift injection rate generally increases pressure in the well-head.

2.3 Supply Chain Planning Horizons

The petroleum production industry is the backdrop of a diverse supply chain, from the extraction of crude oil from subsea reservoirs to the delivery of a finished product to customers. An example of a petroleum sector supply chain is shown in Figure 2.3. Although the details of a supply chain may vary, all petroleum producing entities are invariably faced with a spectrum of planning problems. These planning problems span from the long-term planning and development of new fields to operational well control problems which may have to be tackled several times each day.

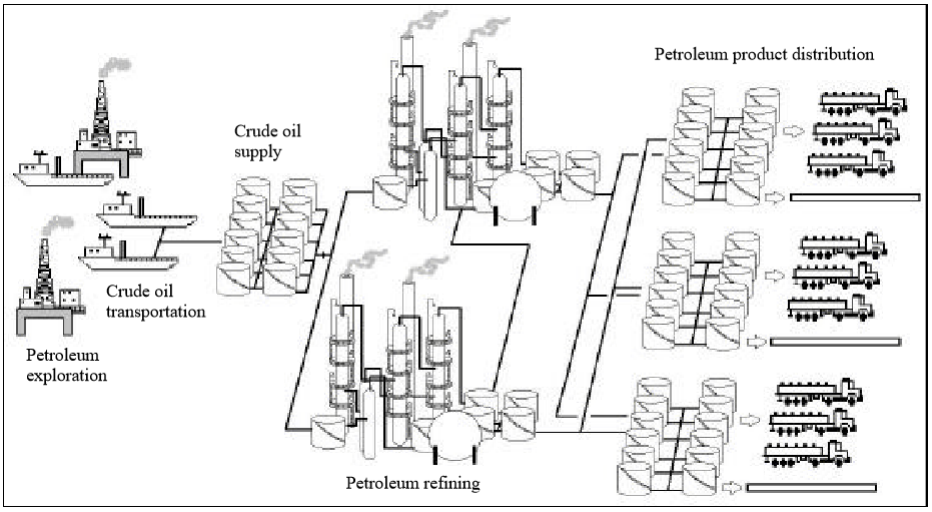

Figure 2.3: Supply chain in the petroleum production industry, Neiro and Pinto (2003).

Thus, planning problems in the petroleum sector vary significantly in scope and time horizon. Not only does this necessitate different solution approaches for different problem types, but it also makes integrated planning difficult. Ensuring that the daily operations of

a platform are in accordance with a 10-year field exploitation plan is both challenging to model and solve. The planning problems are further complicated by factors of uncertainty, a canonical example being the fluctuating oil price.

Combining planning problems of all time horizons in a single model leads to a problem so complex it is by all practical measures impossible to solve. A more useful approach is to organize problems according to the time horizon with which they are concerned; problems are then tackled in a hierarchical manner. Long-term plans place restrictions on and provide guidelines for shorter horizon problems as one moves down the hierarchy.

Several propositions for organizing planning problems in a hierarchy according to time have been presented, with the perhaps most common being the notion of the strategic, tactical and operational planning levels. More recently, the RTO level has been introduced as the planning level with the shortest time horizon (Bieker et al., 2007). An overview of the four levels and associated time horizons can be seen in Figure 2.4.

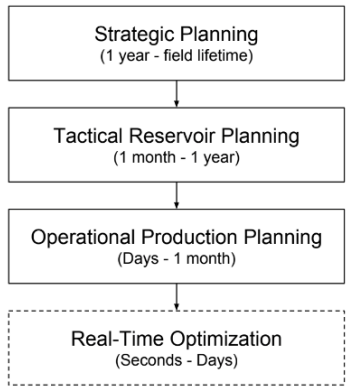

Figure 2.4: Four levels of planning in petroleum production, Morken and Sandberg (2016).

The advent of the real-time optimization level is driven by the increased availability and quality of measurement data being recorded in the oil field. At the core of RTO lies an optimization model that is continuously being updated with recorded data from the production field. The goal of RTO, as coined by Sequeira et al. (2002), *"is to operate a plant, at every instant of time, as near to its optimum operating conditions as possible"*.

Solving problems in a real-time optimization setting entails both opportunities and restrictions. For instance, the capabilities of modern data collection in production fields may synergize well with models that benefit from frequently being updated with new data. However, the goal of staying as close to optimality at every instant of time means solution times need to be kept low and frequent re-runs of the model are typically necessary.

The focus of this section is real-time optimization, and a full discussion of the integrated planning of an entire petroleum supply chain is beyond the scope of this thesis. The

interested reader is referred to Leiras et al. (2011) for a thorough literature review, or Neiro and Pinto (2004) for an example of a supply chain optimization framework.

Chapter 3

Literature Study

This chapter presents a literature study of relevant topics within optimization in the petroleum industry. For more comprehensive literature studies, two recommended surveys of optimization in the petroleum production industry are Rashid et al. (2012) and Khor and Elkamel (2017).

This chapter is organized as follows. First, a review of early optimization applications to maximizing production in oil fields is given in Section 3.1.1. Then, non-linear approaches are presented in Section 3.1.2, after which piecewise linear approaches are discussed in Section 3.1.3. Modern machine learning approaches are presented in Section 3.1.5, before economic aspects of petroleum production optimization are discussed in Section 3.1.4. A brief overview of the use of neural networks in optimization is given in Section 3.2, and Bayesian neural networks are reviewed in Section 3.3. A table of selected references is presented in Section 3.4. Finally, comments on the literature study are given in Section 3.5.

3.1 Optimization Methods for Petroleum Production Wells

A crucial step in the optimization process of any petroleum production system is modelling the well output as a function of its input decision variables. In general, obtaining good approximations to the true well output curves increases model complexity, solution times and computational requirements. Conversely, approximations of a less complex nature may result in inaccurate solutions due to lacking in flexibility. This section presents different approaches as they appear in the literature, as well as selected machine learning and economic perspectives.

3.1.1 Early Optimization in the Petroleum Industry

An early focus of optimization in the petroleum industry on an operational level is the performance of a single producing well. Nodal analysis, first developed by Schlumberger and later regarded as a general approach to well analysis, divides a single well into distinct nodes and models the interactions between the nodes. Mach et al. (1979) present an example of nodal analysis. Bahadori et al. (2001) incorporate nodal analysis in the optimization of gas lift injection rates.

Authors considering optimization over an entire production field, rather than a single well, develop heuristics that improve on single well nodal analysis. Perhaps most well known is the method of equal slopes. The method is based on the concept of incremental gas-oil ratio (IGOR), which describes the volume of gas lift required to produce an incremental barrel of oil. Kanu (1981) applies equal slopes to a system of wells while considering economic factors. The main idea is to step-wise allocate additional volumes of gas lift according to the economic slopes until all gas lift is spent. Weiss et al. (1994) show mathematically that applying the equal slopes method based on IGOR yields optimal production levels.

Several issues with the method of equal slopes are identified in later works. The proof by Weiss et al. (1994) assumes it is always optimal to utilize all gas lift available. In reality, physical phenomena such as back pressure (see Section 2.2) lead to sub-optimal oil production if the gas lift injection rate is excessively high. Nishikiori et al. (1995) point out that the method of equal slopes suffers when wells do not respond immediately to operational adjustments, as it may converge slowly and get stuck in local optima.

3.1.2 Non-Linear Modelling

With increasing computational power, many complex models are later developed and solved in the literature. Several early non-linear models address the problems with the method of equal slopes.

Lang and Horne (1983) compare step-wise linear programming with dynamic programming for a gas lifted well problem. Nishikiori et al. (1995) present a non-linear quasi-Newton solution method in which gas lift injection rates act as decision variables. The convergence rate depends heavily on an initial solution guess but is potentially faster than a gradient approach. Buitrago et al. (1996) address the case where wells respond slowly to gas lift injection with a non-linear model. Dutta-Roy and Kattapuram (1997) consider well interactions and use sequential quadratic programming to solve the resulting optimization problem. They contend that any model ignoring well pressure interactions is overly optimistic due to the significance of back pressure when wells share pipelines.

Zhang et al. (1985) devise the Penalty Successive Linear Programming algorithm and provide a convergence proof for non-linearly constrained problems of general form. The algorithm is suited for highly constrained problems and is in certain cases quadratically

convergent. The approach is proven to be robust enough to be applied in modern non-linear approaches such as Kosmidis et al. (2004, 2005).

Alarcón et al. (2002) optimize production output in an oil field by modeling oil production as a function solely dependent on gas lift injection rates. They present a novel mathematical fit to the non-linear gas output curve of each well, and subsequently solve the non-linear problem using sequential quadratic programming (SQP). A concluding remark is that minimizing the uncertainty regarding the estimated well output curves is critical to the success of the model.

Wang (2003) uses a two-level programming approach to determine the optimal short-term production rates and well input variable settings. In this structure, the top problem is the optimization of well connections and the bottom problem is a rate allocation problem, with gas lift injection rates and choke openings as decision variables. The top problem is solved by partial enumeration (PE) while the bottom problem is solved using an SQP method. Well interaction effects are captured by non-linear relationships.

Grimstad et al. (2016) use B-splines to approximate pressure drop correlations in a mixed integer non-linear program (MINLP) formulation. B-splines are applied in a regular interpolation grid, as determining irregular knot positions is noted to be a difficult problem. The optimization formulation is well tailored to the CENSO (Convex Envelopes for Spline Optimization) solution framework. The model is tested on three cases of increasing complexity which are solved to both global and local optimality. Obtaining a solution certificate of global optimality is deemed too computationally expensive for RTO for larger cases, but the small gap between global and local optimality indicates that the formulation produces a largely convex NLP relaxation with stable and promising local results.

3.1.3 Piecewise Linearization

Although non-linear programming has the ability to capture complex relationships between wells, solution times may be excessively long. As a compromise between model complexity and efficient solution methods, several works turn to piecewise linear models.

An early work by Fang and Lo (1996) applies separable programming principles with piecewise linear well output curves to solve a gas lift allocation problem with the simplex method. Although showing promising results, the model is limited to concave well output curves. This limitation may lead to poor approximations of the true curve in real case applications since assuming concavity limits the generalization power of the model.

Kosmidis et al. (2005) present a MINLP formulation of a complete production field with the intention of providing a more holistic model than previous approaches. Non-linear well output curves are approximated by discretized, piecewise linear functions which are stored as look-up tables for each well. The authors remark that the accuracy of the model depends on the granularity of the piecewise linear functions, and claim that once the look-up table is constructed, it is valid for the lifetime of each well. As a well matures and physical conditions in the reservoir change, such a claim may not hold over a longer time

horizon.

A comparative study of piecewise linear formulations for well operations is conducted by Misener et al. (2009). Four approaches are considered, and the study concludes that the classic approach by Nemhauser and Wolsey (1988) is inefficient. The classic approach involves using binary variables to activate linear segments which are subsequently weighted by a combination of the corresponding endpoints. Efficiency gains can be achieved by a formulation without binary variables using Special Ordered Sets of type 2 (SOS2).

Bieker (2007) applies piecewise linearization to approximate non-linearities in the pressure drop in well pipelines. Gunnerud and Foss (2009) further build on this approach to construct a model suitable for decomposition techniques. The resulting mixed integer linear program (MILP) is based on a piecewise linearization in three dimensions by using a regular grid of breakpoints and SOS2.

Silva and Camponogara (2014) apply multidimensional piecewise linear models to a petroleum optimization problem, and show a tight formulation of routing constraints within the oil field. To describe the non-linear nature of the output function, several approaches to piecewise linearization are suggested, and a thorough computational analysis of the options is given. While the authors suggest expanding the model to include additional physical factors, the model is compatible with a formulation only encompassing explicit decision variables such as gas lift injection rates and choke openings.

3.1.4 Economic Perspectives in Petroleum Production Optimization

Several authors contend that the goal of any optimization approach in the petroleum production industry should be concerned with maximizing profits and not the total volume of oil produced per day. Such models incorporate the economic aspects of production in an oil field, such as lift gas compression costs and fluctuating oil prices.

Early approaches to economic optimization typically rely on the method of equal slopes, adjusted for costs and profits. Some relevant examples are presented in Simmons (1972), Redden et al. (1974) and Kanu (1981). Mora et al. (2005) construct a model which maximizes the net present value (NPV) of an oil field. The NPV of the asset is evaluated over a 10-year horizon with present-time well parameters estimated by a simulator. A notable result is that the gas lift injection rate that maximizes oil production is found to be inadequate in maximizing the NPV of the asset. Khishvand et al. (2015) use daily cash flow as the objective to be maximized, and find that gas lift compression costs have a significant impact on the solution quality.

3.1.5 Machine Learning Applications to Petroleum Production Optimization

With a surge in popularity over the recent years, machine learning (ML) applications to petroleum production optimization are numerous. ML techniques are applied to parts of the optimization approach such as estimating parameters in a pre-optimization step, or to produce a complete solution of the optimization problem. In the latter case, a guarantee of optimality is rarely given for the obtained solution, effectively producing heuristic methods.

A notable early example of applying ML to a pre-optimization step of petroleum production optimization is given by Stoisits et al. (1992, 1994). NNs with back-propagation are used to obtain a model of the well output curve, which subsequently is integrated into a simulator and optimizer. The performance and computation speed of the total field optimization are later improved upon by the application of a genetic algorithm (GA) in Stoisits et al. (1999). Gharbi et al. (1997) use NNs to predict petroleum-fluid properties such as the GOR and bubble-point pressure based on real data from the Middle East. The NN achieves higher accuracy than conventional regression techniques. In a more recent work, Shokir et al. (2017) apply NNs to produce a well output model and estimate bottom-hole pressure levels. Their implementation is able to predict oil well parameters outside of the training set with satisfying accuracy, proving particularly accurate for the well output curve as a function of gas lift injection rates. Again, the NN is integrated into a complete optimization system. Saputelli et al. (2002) take a hierarchical approach to the development of data driven models with a focus on NNs. They note that artificial NNs are powerful tools when used to assist in frequent optimization in petroleum production.

For the approach of using ML techniques to producing a full solution to an optimization problem, Martinez et al. (1994) show an early application of a genetic algorithm to a field-wide problem formulation. Results show a significant increase in oil production in comparison to single-well optimization. Ray and Sarker (2006) apply an evolutionary algorithm (EA) to a multi-objective formulation of the production maximization problem. The conflicting objectives are maximizing oil production while minimizing lift gas injection rates. In a computational study, the EA compares favorably to the results obtained by Buitrago et al. (1996). The algorithm produces satisfying solutions to a problem with 56 wells in a few seconds.

Zerafat et al. (2009) apply ant colony optimization (ACO) to a large scale petroleum production field and compare results to approaches such as SQP (Dutta-Roy and Kattapuram, 1997) and a GA (Ray and Sarker, 2006). They find that ACO, although slightly outperformed by the GA, is able to produce good results that outperform the classical approaches such as SQP.

Hamedi et al. (2011) apply particle swarm optimization (PSO), which they find to outperform the SQP approach as presented by Alarcón et al. (2002). Ruz-Hernandez et al. (2010) construct a neural network which is able to produce solutions for single- and two well systems. Their results compare favorably to a commercial optimization software, but

no discussion of the ability of the neural network to generalize to larger problems is given.

As noted, bio-inspired algorithms such as EA, ACO and PSO are commonly characterized as meta-heuristics which give no guarantee of optimality. Although some convergence proofs exist (see Villalobos-Arias et al. (2006) or Stutzle and Dorigo (2002)), in practice the algorithms are not used as exact optimization procedures.

3.2 Neural Networks in Optimization

Early stages of applying NNs to optimization problems involve the design of circuit systems able to solve complex tasks in a rapid manner. An influential early model is presented by Tank and Hopfield (1986), who propose a highly interconnected circuit NN of analog processors. Several researchers build on these ideas. Kennedy and Chua (1988) design circuit based NNs for non-linear programming, while Simon et al. (1988) focus on systems tackling integer linear programs for the job-shop scheduling problem (Manne, 1960). Refinements to the design of the NNs are found in Rodriguez-Vazquez et al. (1990) and Xia (1996), while Zak et al. (1995) provide a comparative study of circuit based NNs for LP.

Zhang (2013) thoroughly examines the application of NNs in optimization problems. A wide range of problem classes is discussed, including linear programming, quadratic programming and general non-linear programming.

Szegedy et al. (2013) solve an optimization program in order to generate adversarial input data to trained NNs. The formulation is able to cause high rates of misclassification in state-of-the-art networks, and represents a novel take on the combination of an optimization model and NNs. Cheng et al. (2017) examine the resilience of neural networks to input noise or maliciously constructed input examples. The problem of determining resilience is solved as an optimization model, in which the entire NN is reformulated as a mixed integer program (MIP).

Fischetti and Jo (2017) examine the reformulation of a deep NN as a MILP. They present a method for tightening the bounds on the resulting constraints and provide applications to both hand-written digit recognition and adversarial examples.

3.3 Bayesian Neural Networks

A distinct interpretation of NNs stems from Bayesian probability theory, giving rise to the term Bayesian neural network (BNN). Early ideas are found in Buntine (1991) and MacKay (1992) where probabilistic frameworks for back-propagation during learning are presented, denoted as Bayesian back-propagation. Contrary to the standard approach for NN learning, the Bayesian back-propagation places prior beliefs on the distributions of weights in the network, and considers their corresponding posterior probabilities as well as the likelihood of the network. Hinton and Van Camp (1993) consider the distributions

over NN weights in line with the Minimum Description Length principle (Rissanen, 1986), and add Gaussian noise in weight parametrization.

Following the aforementioned ideas, Nix and Weigend (1994) propose a novel NN architecture in order to predict target mean and variance simultaneously. The architecture consists of two output units, each with their separate hidden layer. Optimization is performed by maximizing the log-likelihood of the network outputs. Once trained, the network is able to provide a measure of the uncertainty in its predictions as described by its estimated variance. This approach is often referred to as Mean-Variance Estimation (MVE), and although it is capable of predicting distributions rather than point estimates, it is not fully Bayesian since it does not consider the posterior probability of its weights. Bishop (1994) develops a more advanced Mixture Density Network which is able to approximate a mixture of distributions and is particularly well suited for inverse regression problems.

As noted by Neal (1995), an inherent challenge with the Bayesian interpretation of NNs is the generally intractable problem of integrating posterior probabilities when calculating predictions. In what may be considered the first comprehensive treatment of BNNs, Bishop (1995) discusses strategies for tackling this integral. Of special interest in this thesis are the simplifying Gaussian approximations used in e.g., the MVE network of Nix and Weigend (1994), and Monte Carlo methods. Neal (1995) provides an example of the latter by presenting a hybrid Monte Carlo algorithm. In a later work, de Freitas (2003) uses a reversible jump Markov Chain Monte Carlo simulation algorithm to handle the intractable calculations and proves its convergence.

Although the NN presented by Nix and Weigend (1994) is to some extent able to capture uncertainty in the predictions of the network, Dybowski and Roberts (2001) note that confidence intervals based on this uncertainty alone underestimate the true uncertainty. This is due to the inability of the network to capture model uncertainty, making variance estimates biased. The underestimation of uncertainty in predictions by an MVE network is confirmed in comparative studies of prediction intervals by Papadopoulos et al. (2001), Ding and He (2003) and Khosravi et al. (2011a). However, approximations to the model uncertainty can be obtained by methods such as bootstrapping, as Heskes (1997) shows.

Lampinen and Vehtari (2001) note the sensitivity of a BNN to the selection of prior distributions. They show in 3 case applications that poor assumptions during the modeling phase lead to wrong models with high probabilities as specified by the likelihood function.

Modern work on BNNs turns to what is known as Variational Inference (VI) to handle the difficulties of intractable integrals over posterior probabilities. According to Graves (2011), the main idea behind VI is the replacement of a problematic distribution by an approximate distribution which is more amenable to numerical methods. The divergence between the true distribution and the approximation is minimized, a concept which relates to optimization of the Minimum Description Length as presented by Hinton and Van Camp (1993).

Graves (2011) presents a tractable VI scheme for neural networks and shows the resulting calculations for various choices of posterior distributions over the weights in a neural network. The scheme is applied to phoneme recognition and achieves reductions in the number of weights required in the network as well as improved generalization over traditional neural networks. Blundell et al. (2015) introduce a new back-propagation scheme for learning distributions over weights, based on a VI approximation to the true posterior distribution. BNNs trained with the scheme display improvements in generalization performance over traditional NNs. Other recent works of interest on VI include Kingma and Welling (2013) and Rezende et al. (2014).

Following the reparametrization trick of Kingma et al. (2015), Gal and Ghahramani (2015) show how a multilayer perceptron with arbitrary depth, non-linearities and dropout (Srivastava et al., 2014) applied after every weight-layer is mathematically equivalent to an approximation of a probabilistic deep Gaussian process (Rasmussen, 2004). Notably, Gal (2016) finds that optimizing a neural network with dropout is equivalent to VI in a BNN under certain conditions. This allows for the estimation of model uncertainty for any NN with dropout. A connection between Gaussian processes and BNNs is also shown in Louizos and Welling (2016), where a matrix-variate Gaussian distribution is used in the BNN.

3.4 Table of Selected References

Table 3.1: Selected Works Presented in the Literature Study

				Decision Variables			
Article	**Objective**	**Problem Type**	**Solution Method**	**Gas Lift**	**Choke**	**Routing**	**Physical Factors**
Lang and Horne (1983)	Oil production	Non-linear	DP	Yes	No	No	Yes
Stoisits et al. (1992, 1994)	Oil production	Non-linear	NN, Simulator	Yes	No	No	No
Nishikiori et al. (1995)	Oil production	Non-linear	Quasi-Newton	Yes	No	No	No
Buitrago et al. (1996)	Oil production	Non-linear	Stochastic-heuristic hybrid	Yes	No	No	No
Fang and Lo (1996)	Oil production	PWL MILP	Separable programming	Yes	No	No	No
Dutta-Roy and Kattapuram (1997)	Oil production	Non-linear	SQP	Yes	No	No	No
Stoisits et al. (1999)	Oil production	Non-linear	NN, GA	Yes	No	Yes	Yes
Alarcón et al. (2002)	Oil production	Non-linear	SQP	Yes	No	No	No
Wang (2003)	Oil production	Non-linear	Partial enumeration, GA	Yes	Yes	Yes	No
Kosmidis et al. (2005)	Profits	MINLP	Sequential MILP	Yes	Yes	Yes	Yes
Mora et al. (2005)	NPV of asset	Non-linear	Simulation	Yes	No	No	Yes
Ray and Sarker (2006)	Oil production	PWL MILP	NSGA-II	Yes	No	No	Yes
Bieker (2007)	Oil production	PWL MILP	B&B	Yes	Yes	No	Yes
Gunnerud and Foss (2009)	Oil production	PWL MILP	DWD, LD	Yes	No	Yes	Yes
Zerafat et al. (2009)	Oil production	Non-linear	ACO, GA	Yes	No	No	No
Ruz-Hernandez et al. (2010)	Oil production	Non-linear	NN	Yes	No	No	No
Silva and Camponogara (2014)	Oil production	PWL MILP	B&B	Yes	No	Yes	Yes
Khishvand et al. (2015)	Daily cash flow	Non-linear	GRG, SLP	Yes	No	No	No
Grimstad et al. (2016)	Oil production	Non-linear	CENSO, BONMIN	Yes	No	Yes	Yes
Shokir et al. (2017)	Oil production	Non-linear	NN	Yes	No	Yes	No

Table 3.2: Nomenclature used throughout the literature study and in the table of selected references.

Literature Study Nomenclature	
Abbreviation	**Item**
PWL	Piecewise Linear
B&B	Branch & Bound
LD	Lagrange Decomposition
DWD	Danzig-Wolfe Decomposition
SLP	Successive Linear Programming
MILP	Mixed Integer Linear Program
MINLP	Mixed Integer Non-Linear Program
DP	Dynamic Programming
SQP	Sequential Quadratic Program
NN	Neural Networks
GA	Genetic Algorithm
NSGA-II	Non-Dominated Sorting Genetic Algorithm II
ACO	Ant Colony Optimization
GRG	Generalized Reduced Gradient
CENSO	Convex Envelopes for Spline Optimization
BONMIN	Basic Open-source Nonlinear Mixed Integer programming

In Table 3.1, a selection of works is presented. For each article, the objective being maximized is listed, along with the problem type and solution method. In addition, the modeling choice regarding which decision variables the authors include is shown in the four rightmost columns. Table 3.2 gives an overview of the nomenclature used in the table of selected references.

The majority of work focuses on maximizing the total oil production in the oil field. This is achieved by a variety of problem formulations that are either linear, piecewise linear or non-linear, with complexity generally increasing in the same order. Solution methods are more varied still, with applications ranging from decomposition techniques to the popular NSGA-II algorithm. As is evident from the decision variable columns, all of the presented work incorporates explicit decision variables in the model formulation, a consequence of focusing on precisely such models. However, the modeling choices with respect to physical factors vary.

3.5 Comments on the Literature Study

The literature study is introduced with the conflicting priorities of obtaining a good approximation to well output curves and the need to keep model complexity low enough to allow for fast and efficient solving. As there is no single best prioritization, the resulting implementations are many and diverse.

Several of the early approaches to optimization in the petroleum industry described in Section 3.1.1, such as the method of equal slopes, are outperformed by more recent applications. Such approaches are considered outdated by modern standards. An example of the contrary is the nodal analysis, which is still in use in modern simulation software, e.g., PROSPER (2015). However, nodal analysis is concerned with describing the fluid mechanics of a producing well while this thesis seeks to avoid optimizing over physical parameters such as bottom-hole pressure.

A strong argument for non-linear optimization is the fact that modelling pressure interactions between wells is difficult in a linear fashion, a point which is emphasized by Dutta-Roy and Kattapuram (1997). This consideration is a returning motivation in several of the works presented in Section 3.1.2. A range of different solution strategies is presented to alleviate the long solution times often associated with non-linear programming. Nonetheless, reported solution times are often in the order of hours or days. If the application of the optimization model requires re-solving several times per day, the computational requirements of non-linear approaches may make them ill-suited for the relevant application of this thesis. Even so, non-linear formulations with acceptable solve times are presented, e.g., by Grimstad et al. (2016). However, solving such models often requires the use or development of specialized solution strategies of which not all guarantee optimality.

Piecewise linear mixed integer models seek to achieve a satisfying complexity while keeping computational requirements low. Numerous ways of modelling oil field production systems are presented in Section 3.1.3, both including and excluding physical parameters such as pressure drop and pipeline characteristics. Models which exclude physical parameters are closely related to the scope of this thesis, and typically approaches including such parameters can be adapted to optimize solely over explicit decision variable settings.

The successful application of machine learning techniques to optimization problems in the RTO of oil fields is a widely examined topic in the recent years and is exemplified by e.g., Sra et al. (2011). Several such applications are presented in Section 3.1.5, with the main distinction being whether or not machine learning is applied to produce a complete solution, or as a pre-optimization step to e.g., approximate a well output curve. The latter case is compatible with virtually any solution approach. This approach benefits from effective function approximation with machine learning while still guaranteeing a global optimum using traditional optimization methods. Such a combination of machine learning and optimization is applied in this thesis.

In Section 3.1.4, some economic perspectives to petroleum production optimization are presented. Several authors argue that any agent in the petroleum production industry is

mainly concerned with maximizing profits, measured by, e.g., net present value or cash flow, rather than actual barrels of oil. Although this argument is sound, virtually any model maximizing oil production is adaptable to incorporate costs and profits with relative ease. Thus, the main area of interest is tackling the difficulties in modeling well output curves, and keeping solve times down.

Using NNs to tackle optimization problems is far from a new concept. Early examples in Section 3.2 are based on circuit modelling. However, the reformulation of a trained NN as a MILP is a relatively new concept which is yet to be explored fully. Several interesting topics are discussed in more recent works, such as adversarial input distortion and network resilience.

Finally, in Section 3.3 we present a brief study of BNNs and related ideas. Proponents of BNNs typically contend that the single point estimate of a regular NN is an oversimplification of the real target distribution and its uncertainty. BNNs provide powerful statistical tools which give access to uncertainty estimates. However, the corresponding computational cost is often prohibitively large. Recent results linking BNNs and Gaussian processes show that such uncertainty estimates in certain cases can be obtained almost for free. The most relevant result to this thesis is the use of dropout to estimate model uncertainty.

Chapter 4

Theory

This chapter describes relevant theory supporting the implementations and models in this report. The chapter is organized as follows. First, Section 4.1 introduces topics that are relevant in the setting of supervised learning. Then, neural networks (NN) and their most important features are presented in Section 4.2, before modern topics on estimating uncertainty in NNs are provided. Finally, an introduction to concepts of optimization is given in Section 4.4.

4.1 Supervised Learning and Model Selection

The task of using some set of input variables to predict the value of some set of output variables is what is characterized as *supervised learning*. There are countless approaches to supervised learning; examples include linear regression, kernel methods, neural networks and splines. Ideally, which approach to use is carefully selected in accordance with the problem characteristics at hand. This section describes some general concepts related to selecting the optimal model fit within some chosen approach to supervised learning.

In the following discussion, we assume that our observed data is generated by some system described by

$$y = f(x_1, \dots, x_n) + \epsilon \tag{4.1}$$

where the function f captures the joint predicative relationship of y on some domain $X = x_1, \dots, x_n \in \mathcal{D} \subset \mathcal{R}^n$ which in this setting contains our data. The ϵ term describes the dependence of y on terms not in $\mathcal{D}$ and is thus not observable given our data. We make the assumption that the expected value of ϵ is 0. The problem of interest in supervised learning is to find a model $\hat{f}$ which approximates f.

4.1.1 The Bias-Variance Trade-off

Given our definition of the system generating our observed data described in Equation 4.1, we can examine the squared-error loss (see Equation 4.6) of some model fit $\hat{f}$ to f. According to Hastie et al. (2009), the expected prediction error at some input point $X = x_0$ is decomposed according to the following set of equations:

$$\begin{aligned} Err(x_0) =& E[(y - \hat{f}(x_0))^2 \mid X = x_0] \\ =& \sigma_\epsilon^2 + [E[\hat{f}(x_0)] - f(x_0)]^2 + E[\hat{f}(x_0) - E[\hat{f}(x_0)]]^2 \\ =& \sigma_\epsilon^2 + \text{Bias}^2(\hat{f}(x_0)) + \text{Var}(\hat{f}(x_0)) \\ =& \text{Irreducible Error + Bias}^2 \text{ + Variance} \end{aligned} \tag{4.2}$$

The first term in Equation 4.2 represents the variance of the target function around its true mean, and is irreducible in that we cannot avoid it unless $\sigma_\epsilon^2 = 0$. Bias is the amount the mean of our model deviates from the true mean. Variance is the expected squared deviation of our model around its mean. Both the squared bias term and the variance term are elements of the error that in theory are reducible to 0 given the true model and infinite data with which to calibrate it. However, in real applications, access to near infinite is probably unrealistic and there is a trade-off between minimizing bias and minimizing variance.

In general, models with high bias and low variance tend to be less complex in their structure. This implies that the model has limited flexibility and is unable to fit the data points well. An example of this occurs when we fit a linear regression line to a noisy linear data set with certain extreme outliers - the single line is not flexible enough to capture points that deviate far from the main trend. However, the line generalizes well outside the training set if the data mostly follows the linear trend.

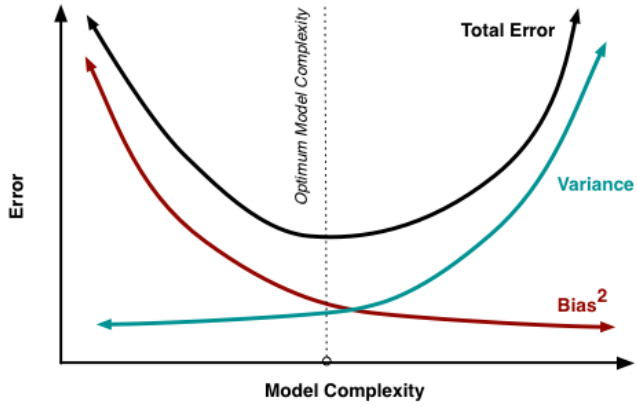

Figure 4.1: Illustration of the bias-variance trade-off for a theoretical regression model, Roe-Fortmann (2012)

On the other hand, models with high variance and low bias in general tend to be more

complex, and are able to capture such outliers with their increased flexibility. An example of this is fitting a high order polynomial to a data set with similar characteristics as previously described. The bias-variance trade-off occurs because models with high bias may be *underfit* to the data and unable to capture the right signal, while models with high variance may be *overfit* to the data and capture noise instead of the true signal - such models generalize poorly outside the training set.

Ideally, the point at which we cannot achieve a reduction in either bias or variance without incurring a greater increase in the other is where we find our optimal model. This is illustrated in Figure 4.1. In practice, it is not straightforward to select a model with the optimal hyperparameters even if we are aware of the bias-variance trade-off. Estimating the prediction error of a model requires the use of certain applicable methods, such as cross-validation (Section 4.1.4).

4.1.2 Aleatoric and Epistemic Uncertainty

According to Kendall and Gal (2017), there are two distinct types of uncertainty which need to be accounted for when we model a function to some observed data: aleatoric uncertainty and epistemic uncertainty. This relates to the error decomposition from Equation 4.2 in that the aleatoric uncertainty represents the irreducible error, while epistemic uncertainty incorporates both model bias and variance. The prediction uncertainty is expressed as

$$Var_{Prediction} = Var_{Aleatoric} + Var_{Epistemic} \tag{4.3}$$

In the following, we assume that we are given input data $\mathbf{x}$ with corresponding output data $\mathbf{y}$. We refer to one arbitrary pair of input and output data as $\mathbf{x}^*$ and $\mathbf{y}^*$, respectively.
Aleatoric uncertainty captures inherent noise in the observations. If we assume homoscedastic variance in the data, the aleatoric uncertainty is constant and can be described by the model precision τ, or equivalently by the variance σ^2. Thus we obtain

$$Var_{Aleatoric}[\mathbf{y}^*] := \tau^{-1}\mathbf{I} = \sigma^2 \tag{4.4}$$

If we assume that certain regions of the data contain more aleatoric noise than others, the variance is heteroscedastic. In such cases, aleatoric uncertainty is a function of $\mathbf{x}^*$, given by

$$Var_{Aleatoric}[\mathbf{y}^*] := \tau^{-1}[\mathbf{x}^*]\mathbf{I} = \sigma^2[\mathbf{x}^*] \tag{4.5}$$

Epistemic uncertainty captures the uncertainty in our model parameters and is therefore often referred to as *model uncertainty*. Given infinite data, we can explain away such uncertainty. Since this rarely is possible in real applications with limited data available, we need to account for the epistemic uncertainty present in our model by some applicable method.

4.1.3 Error Metrics

Several approaches to measuring the error of regression fit $\hat{f}$ to some function f exist. No single best metric quantifies the fitting error, but the following metrics provide different insights and are sensitive to different aspects of the data.

Mean Squared Error (MSE) is a measure of the distance between a regression line and a set of points. It is calculated by taking the mean of the squared difference between the points and the regression line. Thus, MSE is given by

$$\frac{1}{\mathcal{N}} \sum_{n=1}^{\mathcal{N}} (\hat{f}(x_n) - f(x_n))^2 \tag{4.6}$$

where $\hat{f}(x_n)$ is the value estimated by the regression line in the point x_n, and $f(x_n)$ is the true value of the data point. Due to the squaring, large deviations are weighted more heavily than small ones. Thus it is sensitive to any outliers in the data. Taking the root of MSE leads to the also commonly used Root Mean Squared Error (RMSE).

The standard deviation (SD) is a measure of the dispersion of a set of values. A low SD suggests that the data values are close to the mean of the data values, while a higher SD points towards a larger deviation from the mean. The SD is often used to express confidence intervals and distributions and is expressed in the same units as the data. The formula for calculating the SD from sample data (meaning *not* the entire population) is given by

$$SD = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \overline{x})^2}{N - 1}} \tag{4.7}$$

where x_i is the sample point i, and $\overline{x}$ is the average of x_i for all sampled points $i = 1, ..., N$.

In-Sample and Out-of-Sample Error

We distinguish between *in-sample* and *out-of-sample* errors. The former refers to how our regression fit performs with respect to the data set to which it was fit. The latter refers to how our regression fit performs on data it has never seen before. When fitting a regression model, a common way of tracking in-sample and out-of-sample error is by dividing the data set into three separate subsets: a training set, a validation set and a test set, with the training set typically comprising a larger part of the original set, e.g., 80%-10%-10%, respectively.

The training set is the subset on which we perform the regression fit. Thus, the training error is a measure of in-sample error. The validation set is used to measure how our fit performs on unseen data and is a measure of out-of-sample error. After fitting a model

to the training data, we use the validation set and validation error to assist in tuning the parameters of the model to reduce out-of-sample error. Finally, after deciding the model parameters, we evaluate the model on the test set to obtain the test error. The test error gives us an improved estimate of the out-of-sample error of the model.

It is essential to keep the test set completely separate from the training and model selection procedure. If the model performance on the test set is used to select model parameters, the test set essentially becomes a second validation set. This is referred to as *peeking*, and it invalidates the improvement in the estimate of out-of-sample error provided by the test error.

Normally Distributed Errors

We are able to calculate the probability of a function value given an input value if we make certain assumptions regarding the distribution of the data, or equivalently the distribution of the error of the function that generated the data. We assume that the observed data $\mathbf{y}$ is sampled from some function $f(\mathbf{x})$ with normally distributed errors, where a pair of arbitrary input and output data are denoted by $\mathbf{x}_i$ and $\mathbf{y}_i$. According to Nix and Weigend (1994), the probability density function (PDF) of $f(\mathbf{x})$ is given by

$$P(\mathbf{y}_i|\mathbf{x}_i,\mathcal{N}) = \frac{1}{\sqrt{2\pi\sigma^2(\mathbf{x}_i)}} exp\left(\frac{-[\mathbf{y}_i - \mu(\mathbf{x}_i)]^2}{2\sigma^2(\mathbf{x}_i)}\right). \tag{4.8}$$

The PDF denotes the probability that the real output data $\mathbf{y}_i$ is generated by $f(\mathbf{x})$ given the input $\mathbf{x}_i$ and the distribution $\mathcal{N}(\mu(\mathbf{x}), \sigma^2(\mathbf{x}))$. Here, $\mu(\mathbf{x})$ and $\sigma^2(\mathbf{x})$ are the mean and the variance of $f(\mathbf{x})$, respectively. Taking the natural logarithm of both sides, we get an expression for the log-likelihood:

$$lnP(\mathbf{y}_i|\mathbf{x}_i,\mathcal{N}) = -\frac{1}{2}ln(2\pi) - \frac{1}{2}ln[\sigma^2(\mathbf{x}_i)] - \frac{[\mathbf{y_i} - \mu(\mathbf{x}_i)]^2}{2\sigma^2(\mathbf{x}_i)}. \tag{4.9}$$

4.1.4 Cross-Validation

Cross validation is a method of estimating the prediction error of a model fit $\hat{f}$. More precisely, the method estimates the out-of-sample error; that is the average generalization error that occurs when we apply our method to an independent test sample that was not present in the training step (Hastie et al., 2009). For a data set with N observations, *K-fold cross validation* consists of splitting the dataset into K parts of (roughly) equal size. Then, we leave each part out of the training process and record the error of our trained model when predicting over the data in the part that was left out. The prediction error from each of the $k = 1, 2, \ldots, K$ iterations of model training and subsequent testing are averaged to give an estimate of the total prediction error. Let $\hat{f}^{-k}(x)$ be the fitted model with the kth part of the data omitted from the training procedure. We define a mapping function $g(i)$ which for a data point $i = 1, 2, \ldots, N$ returns the part k in which i was omitted from the

training set. The cross-validation error estimate is given by

$$CV(\hat{f}) = \frac{1}{N}\sum_{i=1}^{N} L(y_i,\ \hat{f}^{-g(i)}(x_i)) \tag{4.10}$$

where $L(y_i,\ \hat{f}^{-g(i)}(x_i))$ represents the loss function. We apply this principle to assist in model selection by comparing cross-validation estimates of errors between models trained with different hyperparameters. Let $\hat{f}(x,\ \alpha)$ be a model fitted to data x using hyperparameters α and A be a user-specified set of hyperparameters of interest. We then choose optimal hyperparameters α^* according to

$$\alpha^* = argmin_{\alpha \in A} \quad (CV(\hat{f}(\alpha,\ x)). \tag{4.11}$$

After selecting optimal hyperparameters, the model is trained on the entire training set without leaving any parts out.

The parameter K determines the size of the left out testing set for each round. In the extreme, setting $K = N$ yields what is referred to as *leave-one-out* cross-validation. In this case, a single data point is left out in each step, and the prediction error for this data point is recorded. The value chosen for K affects both computational efforts expended and the behavior of the prediction error estimate.

For large values of K, we observe that we must refit and test our model numerous times to obtain the prediction error estimate. This procedure is potentially computationally taxing. We also observe that for K close to or equal to N, the estimate of the prediction error is virtually bias-free since the K different training sets differ only in a few (or none) points $K - N$. The variance of this estimate is however large. For small K, there potentially is considerable bias since large portions of the data set are left out during training, but the variance is low.

What constitutes a proper value for K is subject to discussion, and K should be selected after inspecting the specific problem at hand. Nonetheless, certain values are recommended as a compromise in the bias-variance trade-off, by for instance Breiman and Spector (1992). Two popular such values are $K = 5$ and $K = 10$.

4.2 Neural Networks

Extensive theory exists regarding NNs. In the following, we give brief introductions to the most important concepts.

4.2.1 Introduction to Neural Networks

A neural network is a machine learning approach where a network of densely connected neurons learns to perform a task by analyzing training examples. More precisely, a neural network is a two-stage non-linear statistical regression or classification model, with the

two stages being called *forward passing* and *backward passing*. The network is loosely based on how biological nervous systems, such as the brain, handle information with electrical impulses sent between neurons. Structurally, neural networks are separated into layers, as shown in Figure 4.2. The first layer (input layer) and the last (output layer) are the only ones that are observed by the user. The layers in between are therefore called hidden layers. Neurons in a given layer are usually connected to neurons in their neighboring layers, but not to other neurons in the same layer. One of the most common types of neural networks is a feed-forward network, meaning that information flows in one direction from the input layer, through the hidden layers, and ends up in the output layer.

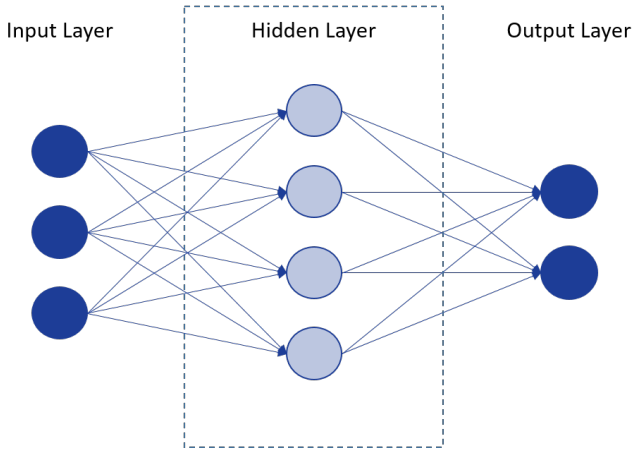

Figure 4.2: The basic structure of a fully connected feed-forward neural network with one input layer, one hidden layer and one output layer.

Each neuron is initially given a bias, and each connection between neurons is given a weight. These weights and biases are tuned during training. The neurons also have a defined activation function, dictating which operations each neuron performs on its input before passing it as output to neurons in the next layer. A standard operation, often referred to as the linear activation function, is multiplying the outputs of the previous layer by their corresponding weights leading to the given neuron and adding the bias term of the given neuron. Some examples of additional activation functions are sigmoid, softmax, maxout and the Rectified Linear Unit (ReLU), which is discussed in more depth in Section 4.2.2. The network as a whole calculates the network loss according to a specified function. That is, the loss is some measure of its output error, i.e., the output compared to the correct case label or value. Common loss functions are MSE and cross-entropy. The goal of the training of the neural network is to minimize this loss, while at the same time avoiding overfitting, a process often referred to as generalizing. This process is explained in Section 4.2.3.

The generic way of minimizing the loss of the network is to train the neurons using a method called back-propagation. Back-propagation can be split into two parts; the forward

pass and the backward pass. In the forward pass, the weights and biases are locked, and the output of the network is calculated from a training case input. Next, the loss is calculated and passed back through the network to adjust the weights and biases. In this algorithm, parameters for a neuron which contributes a lot to the loss of the network are adjusted more heavily in the backward pass than for neurons which contribute less.

4.2.2 Rectified Linear Units

The rectified linear unit is a popular choice of activation function in the neurons of a neural network. The ReLU function returns the maximum of its input and 0, that is

$$ReLU(x_j) = \max\{0, x_j\} \tag{4.12}$$

where x_j is the output of the linear activation function of the neuron. This output is given by

$$\sum_{i=1}^{N_{k-1}} W_{ij}^k x_i^{k-1} + B_j^k \quad = \quad x_j^k \tag{4.13}$$

where x_j^k is the output of the linear activation function in neuron j in layer k, N_k is the number of neurons in layer k, W_{ij}^k is the weight from neuron i in layer $k-1$ to neuron j in layer k and B_j^k is the bias of neuron j in layer k.

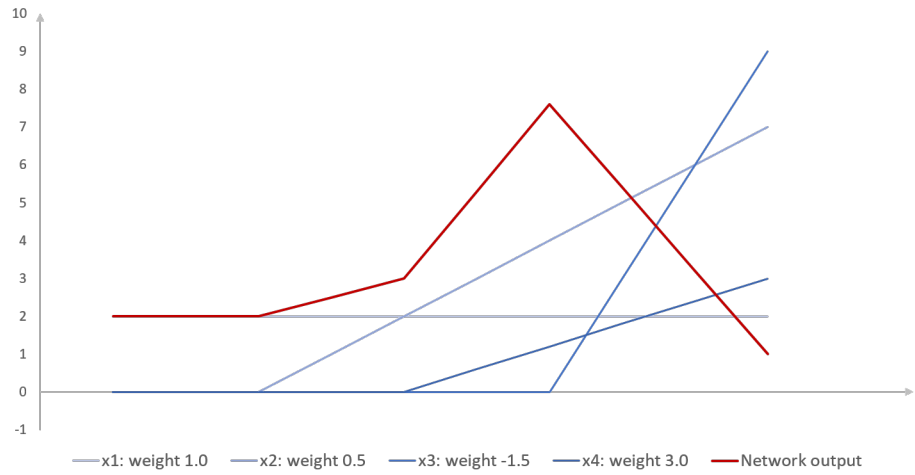

Figure 4.3: A piecewise linear graph, computed by multiplying the output of four ReLU-functions with their corresponding weight and summing.

The output of a ReLU is always non-negative, so to be able to form non-convex (or non-concave) piecewise linear functions, the outputs of several ReLU-functions are multiplied

with a corresponding (possibly negative) weight and summed. An example of this is shown in Figure 4.3, where four ReLU-functions make up a non-convex piecewise linear function. Note that if all weights have the same sign, the resulting function is a convex (or concave) piecewise linear function. When combining multiple ReLUs, the resulting piecewise linear function is segmented according to the points at which each of the ReLU-functions it is constructed of takes values above 0.

4.2.3 Training a Neural Network

In this section, the back-propagation method, introduced in Section 4.2.1, is elaborated on. Assume we want to train a neural network, $\hat{f}(x)$, to approximate some function, $f(x)$, given a set of true x and y values ($y = f(x)$). Each training run starts with a forward pass, meaning that the network is given some input x^* and returns some output, $\hat{y}^*$. By comparing the output to the true label value, y^*, the loss function calculates the loss, L. This is followed by a backward pass, which is where the training happens. Training a neural network is an iterative process, where each iteration involves one forward pass and one backward pass.

As mentioned in Section 4.2.1, training the network means adjusting its weights and biases. The value by which each parameter is adjusted is found by calculating the gradient of the loss function with respect to the parameter, i.e., by calculating the partial derivative of the loss with respect to the parameter. This is achieved by applying the chain rule, yielding the gradient

$$\nabla L(w_{ij}) = \frac{\partial L}{\partial w_{ij}} = \frac{\partial L}{\partial out_j} \cdot \frac{\partial out_j}{\partial in_j} \cdot \frac{\partial in_j}{\partial w_{ij}} \tag{4.14}$$

for two given neurons, indexed by i and j, where w_{ij} is the weight from neuron i to neuron j, in_j is the input to neuron j including the bias of the neuron itself, and out_j is the output of neuron j after the activation function, respectively. When updating the biases, the weight variable in Equation 4.14 is replaced by the bias variable, b_j, which yields

$$\frac{\partial in_j}{\partial b_j} = 1 \implies \nabla L(b_j) = \frac{\partial L}{\partial b_j} = \frac{\partial L}{\partial out_j} \cdot \frac{\partial out_j}{\partial in_j}. \tag{4.15}$$

The weight updates are calculated recursively, starting at the output layer. As a result of the backward pass, the degree to which each weight is adjusted depends on how much it contributes to the network loss in the latest forward pass. The gradients are then used by the implemented optimization algorithm to update the parameters of the neural network.

4.2.4 Optimization Algorithms in Neural Networks

In NNs, optimization algorithms, or optimizers, are used to train the network, usually to minimize some loss function. The most commonly used optimizer is the *Gradient Descent* (GD), which updates parameters according to

$$\theta = \theta - \mu \cdot \nabla L(\theta) \tag{4.16}$$

where $\nabla L(\theta)$ is the gradient (Equation 4.14) of the loss function, $L(\theta)$, with respect to parameters θ, and μ is the learning rate. *Stochastic Gradient Descent* (SGD) is a variant of GD, where the parameters are updated after each training example. This is known to cause the parameters to fluctuate, which helps to escape local optima but may impede convergence. The fluctuation makes the method sensitive to the learning rate parameter, which is constant throughout the optimization and equal for all model parameters. The learning rate can be thought of as how quickly the network adapts to new data. Setting a learning rate that both leads to convergence and allows the escape from local optima is difficult. This difficulty is noted as a weakness of the SGD (Walia, 2017).

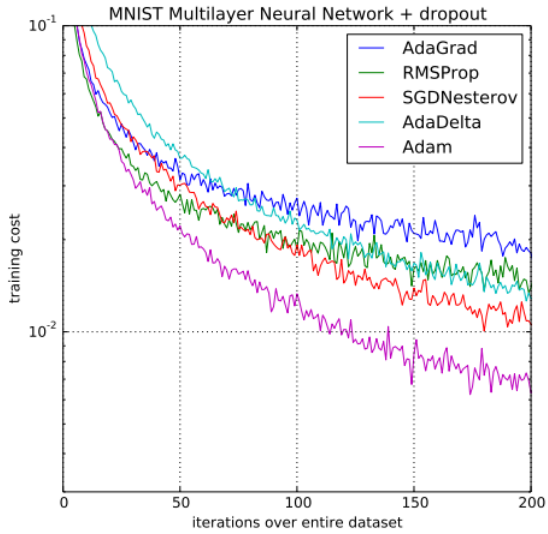

Figure 4.4: Loss of a NN trained with Adam compared to other GD-based optimizers (Kingma and Ba, 2014)

Momentum is introduced to reduce the fluctuations. Momentum works by remembering the gradient of the last iteration and thereby slowing the possibility of the algorithm to alter direction and speed when traversing the solution space. The issue of constant learning rate is handled by adaptive optimizers, such as the Adaptive Moment Estimation (Adam, Kingma and Ba (2014)). Adam uses exponentially decaying averages of past squared gradients and past gradients to scale the learning rate for each parameter in each training iteration. The effect of this is similar to that of momentum. In Figure 4.4, the loss of NNs when training on the famous machine learning data set MNIST with Adam is compared to other optimizers. The plot shows a significantly lower loss for Adam compared to the other GD-based optimizers.

4.2.5 L2 Regularization

L2 regularization, also called weight decay, is a technique used to avoid overfitting in NNs. The idea is to add a regularization term given by

$$\frac{\beta}{2n}\sum_{\gamma}\gamma^2 \tag{4.17}$$

to the loss function of the NN. Here, n is the size of the training set, γ are the parameters, i.e., weights or bias terms, and β is a user determined parameter deciding the intensity of the regularization. The regularized loss function punishes having weights and biases that are large in absolute value. Not all parameters end up approaching zero since the original term in the loss function is still in play. If the reduction in error from increasing the absolute value of a parameter is larger than the punishment of the regularization term, the parameter is allowed to increase in absolute value. However, regularization can drive some neurons to a weight and bias of zero, thereby effectively neutralizing the neuron and decreasing the complexity of the network. The β parameter decides the relative importance of the two terms in the loss function. A low β encourages minimization of the original cost function, and a high β encourages low weights.

4.2.6 Dropout

Dropout is a technique for avoiding overfitting in NNs. E. Hinton et al. (2012) introduce the technique and achieve a considerably lower validation set error compared to networks trained without dropout (and without other regularization techniques). When applying dropout, a binary variable is assigned to each neuron in the specified layers, which typically includes all hidden layers or all hidden layers and the input layer. The binary variable takes the value 0 with probability p, in which case the output value of the corresponding neuron is set to 0. The random dropout makes the network less likely to produce complex co-adaptions, meaning it to a lesser extent overfits the training data and to a greater extent generalizes. Dropout thus typically makes the network perform with a greater loss on the training data, but with a lower loss on previously unseen data.

During the test or validation phase of the network, often referred to as making predictions, forward passes without dropout are performed. Since all neurons in this phase are active, each layer receives a greater total value from the previous layer than when some neurons are dropped out. For this reason, the outputs of the neurons must either be scaled up during training (inverted dropout) or scaled down during testing (dropout). Inverted dropout scales the neuron output by $\frac{1}{1-p}$ in the training phase, while regular dropout scales outputs by $1-p$ in the testing phase.

4.3 Estimating Uncertainty with Neural Networks

In many applications, NNs are used to obtain point estimates of the target variable. Such estimates contain no information of how confident the NN is in its prediction. In this section, we show how NNs are able to estimate a target distribution rather than point estimates. Following ideas from BNNs, both epistemic and aleatoric uncertainty is included in the estimated distribution.

4.3.1 Bayesian Neural Networks and Dropout as a Variational Distribution

A Bayesian neural network (BNN) (MacKay, 1992) is a probabilistic interpretation of a NN in which we place a prior belief over the distribution of the weights and bias terms. For a BNN with $i = 1, 2, \ldots, I$ layers, an often used prior for the weight matrix $\mathbf{W}_i$ of layer i is $p(\mathbf{W}_i) = \mathcal{N}(0, 1)$. Given input data $\mathbf{x}$ and corresponding output data $\mathbf{y}$, both of size N, we can estimate the posterior probability of our parameters given our data. That is, if we define $\boldsymbol{\omega} = \{\mathbf{W}_1, \mathbf{b}_1, \ldots, \mathbf{W}_I, \mathbf{b}_I\}$ to be the weight and bias parameters of our BNN, the posterior probability is given by

$$p(\boldsymbol{\omega}|\mathbf{x}, \mathbf{y}). \tag{4.18}$$

We then perform inference, i.e., the prediction of an output value $\mathbf{y}^*$, for a new input value $\mathbf{x}^*$, by

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{x}, \mathbf{y}) = \int p(\mathbf{y}^*|\mathbf{x}^*, \boldsymbol{\omega}) p(\boldsymbol{\omega}|\mathbf{x}, \mathbf{y}) d\boldsymbol{\omega}. \tag{4.19}$$

Equations (4.18) and (4.19) are generally not tractable. We therefore define a *variational* distribution, $q_\theta(\boldsymbol{\omega})$ with its own set of parameters θ, which approximates the posterior probability. The idea is to select a variational distribution that is easier to handle. To ensure sufficient accuracy, we attempt to minimize the distance between our approximation and the true posterior using the Kullback-Leibler (KL) divergence (Kullback and Leibler, 1951), given by

$$KL(q_\theta(\boldsymbol{\omega})||p(\boldsymbol{\omega})) = \int \frac{q_\theta(\boldsymbol{\omega})}{p(\boldsymbol{\omega}|\mathbf{x}, \mathbf{y})} d\boldsymbol{\omega}. \tag{4.20}$$

According to Gal (2016), minimizing the KL divergence is equivalent to minimizing the negative evidence lower bound loss function, given by

$$\mathcal{L}_{\mathcal{VI}}(\theta) = -\sum_{i=0}^{N} \int q_\theta(\boldsymbol{\omega}) \log p(\mathbf{y}_i|\mathbf{f}^{\boldsymbol{\omega}}(\mathbf{x}_i)) d\boldsymbol{\omega} + KL(q_\theta(\boldsymbol{\omega})||p(\boldsymbol{\omega})). \tag{4.21}$$

Here, $\mathbf{f}^{\boldsymbol{\omega}}(\mathbf{x}_i)$ denotes the BNN output in point i. Using Monte Carlo integration and optimizing over a subset $\mathcal{S}$ of size M of the data, we approximate this loss function as

$$\mathcal{L}_{\mathcal{VI}}(\theta) \approx \widehat{\mathcal{L}}_{MC}(\theta) = -\frac{N}{M} \sum_{i \in \mathcal{S}} \log p(\mathbf{y}_i|\mathbf{f}^{\boldsymbol{\omega}}(\mathbf{x}_i)) + KL(q_\theta(\boldsymbol{\omega})||p(\boldsymbol{\omega})). \tag{4.22}$$

We now consider the loss function of a NN with dropout applied to its weight parameters. Treating this NN as the variational distribution $q_\theta(\boldsymbol{\omega})$, we let the weights and biases of the NN with dropout be denoted by θ. We let the loss function of such a NN, using, e.g., MSE as the error metric, be denoted by $\mathcal{L}_{dropout}(\theta)$, and stress that this NN is not assumed to be a BNN. In a recent result, Gal (2016) shows that minimizing $\mathcal{L}_{dropout}(\theta)$ is equivalent to minimizing $\widehat{\mathcal{L}}_{MC}(\theta)$, since under certain assumptions about the BNN prior, the following property holds

$$\frac{\partial}{\partial\theta}\mathcal{L}_{dropout}(\theta) = \frac{1}{N\tau}\frac{\partial}{\partial\theta}\widehat{\mathcal{L}}_{MC}(\theta). \tag{4.23}$$

Here, τ denotes the model precision, i.e., the reciprocal of the prior variance. This result has important implications. Any NN trained with dropout is equivalent to a BNN, and the task of performing inference is tractable. This also lets us utilize BNN methodology to assess the epistemic model uncertainty.

4.3.2 Epistemic Uncertainty Using Dropout

In Section 4.3.1 it is shown how a NN with dropout amounts to performing variational inference in a BNN. An estimate of model uncertainty can then be obtained. To this end, we define a dropout probability, p_i, for each layer i in the network, excluding the input and output layers. During forward passes, the output of the model varies as a result of the dropped neurons. We consider a NN with an arbitrary number of output neurons $d \in \mathcal{D}$. Given N forward passes, with $\mathbf{f}_d^{\hat{\mathbf{W}}_n}(\mathbf{x}^*)$ being the network output of output dimension d in forward pass n, and $\hat{\mathbf{w}}_n$ being the realization of the weight matrix in pass $n = 1, 2, \ldots, N$ as a result of the realized dropout, we approximate the expected value, y_d^*, of output dimension d by

$$\tilde{\mathbb{E}}[y_d^*] := \frac{1}{N}\sum_{n=1}^{N}\mathbf{f}_d^{\hat{\mathbf{W}}_n}(\mathbf{x}^*) \xrightarrow[N\to\infty]{} \mathbb{E}[y_d^*]. \tag{4.24}$$

This process of averaging several forward passes with dropout is referred to as Monte Carlo (MC) dropout. We capture the model uncertainty in a similar way. The epistemic variance is approximated by

$$\widetilde{Var}_{Epistemic}[y_d^*] := \frac{1}{N}\sum_{n=1}^{N}\mathbf{f}_d^{\hat{\mathbf{W}}_n}(\mathbf{x}^*)^T\mathbf{f}_d^{\hat{\mathbf{W}}_n}(\mathbf{x}^*) - \widetilde{\mathbb{E}}[y_d^*]^T\widetilde{\mathbb{E}}[y_d^*] \xrightarrow[N\to\infty]{} Var_{Epistemic}[y_d^*]. \tag{4.25}$$

4.3.3 Aleatoric Uncertainty

If we assume that the aleatoric variance of our data is homoscedastic, aleatoric uncertainty is described by the parameter τ^{-1} for the entire input domain. However, in the case of heteroscedasticity, the variance is a function of $\mathbf{x}$. In the following, we assume the data points, $\mathbf{y}$, are sampled from some function $f(\mathbf{x})$, with normally distributed errors. To capture the

heteroscedastic uncertainty, it is possible to construct a neural network with two outputs: $\mu(\mathbf{x})$ and $\ln \sigma^2(\mathbf{x})$. The former is an approximation of the mean of the mentioned distribution, while the latter is an approximation of the natural logarithm of the aleatoric variance. The reason for re-parameterizing, i.e., using the natural logarithm of the aleatoric variance instead of the aleatoric variance itself, is to ensure that the variance output returned from the network always produces non-negative values when we transform it to the aleatoric variance. This is achieved by taking the natural exponential function of the $\ln \sigma^2$-term.

The loss function of the network fits the $\mu(\mathbf{x})$-output to $\mathbf{y}$, while at the same time adjusting the $\ln \sigma^2(\mathbf{x})$-output to an approximation to the natural logarithm of the aleatoric uncertainty of $f(\mathbf{x})$. The following loss function is applied in the NN, as described by Nix and Weigend (1994),

$$L = \sum_{i \in I} \frac{1}{2} \left(\frac{[\mathbf{y}_i - \mu(\mathbf{x}_i)]^2}{\sigma^2(\mathbf{x}_i)} + ln[\sigma^2(\mathbf{x}_i)] \right). \tag{4.26}$$

This loss function is based on the expression for the log-likelihood defined in Equation 4.9. The first term of the log-likelihood is removed since it is constant. The loss-function is then obtained by taking the negative of the trimmed log-likelihood and summing over the data points, $i \in I$.

4.3.4 Prediction Uncertainty

The ideas behind aleatoric and epistemic uncertainty leads to the NN with dropout and the two output neurons as described in the previous section. Using the network architecture described in Section 4.3.3, we let output dimension $d = 1$ correspond to the output neuron predicting $\mu(\mathbf{x})$ and output dimension $d = 2$ correspond to the output neuron predicting $\ln \sigma^2(\mathbf{x})$. The output neurons vary with the dropout realization in each forward pass. Letting $\mu_n(\mathbf{x}^*)$ and $\ln \sigma_n^2(\mathbf{x})$ be the outputs of the network during forward pass $n = 1, 2, \ldots N$, we have

$$\mu_n(\mathbf{x}^*) = \mathbf{f}_1^{\hat{\mathbf{W}}_n}(\mathbf{x}^*) \tag{4.27}$$

$$\ln \sigma_n^2(\mathbf{x}^*) = \mathbf{f}_2^{\hat{\mathbf{W}}_n}(\mathbf{x}^*) \tag{4.28}$$

Inserting the re-parameterized network output for aleatoric uncertainty and Equation (4.25) for epistemic uncertainty into the expression for prediction uncertainty (4.3), we obtain

$$\widetilde{Var}[\mathbf{y}^*] = \frac{1}{N} \sum_{n=1}^{N} \exp(\ln \sigma_n^2) + \frac{1}{N} \sum_{n=1}^{N} \mu_n^2 - \left(\frac{1}{N} \sum_{n=1}^{N} \mu_n \right)^2. \tag{4.29}$$

The first term represents the heteroscedastic aleatoric uncertainty and the second and third terms represent the epistemic uncertainty. Here, we have dropped the dependence on $\mathbf{x}^*$ for readability.

4.4 Optimization

In this section, a selection of relevant concepts in stochastic and multi-objective optimization are presented.

4.4.1 Stochastic Optimization

We present a brief introduction to topics in stochastic optimization that are relevant to this thesis. For more thorough treatments, the interested reader is referred to King and Wallace (2012) or Prékopa (2013).

A canonical example of a stochastic optimization problem is the two-stage recourse problem. We define decision variables $\mathbf{x}$ and $\mathbf{y}$, and a random vector to be observed $\boldsymbol{\xi}$. Furthermore, we define the order in which decisions and observations are made as (1) decisions $\mathbf{x}$, (2) observation of $\boldsymbol{\xi}$ and (3) decisions $\mathbf{y}$. The problem is denoted as a two-stage problem since decisions are made in two distinct *stages*, i.e., variables $\mathbf{x}$ correspond to the first stage decisions while variables $\mathbf{y}$ correspond to the second stage decisions. We define the *second stage problem* as

$$\begin{aligned} \min \quad & \mathbf{q}^T\mathbf{y} && && (4.30)\\ \text{s.t.} \quad & && && \\ T\mathbf{x} + W\mathbf{y} \quad &= \quad && \boldsymbol{\xi} && (4.31)\\ \mathbf{y} \quad &\geq \quad && 0 && (4.32) \end{aligned}$$

where the first stage variables $\mathbf{x}$ are assumed to be fixed, $\mathbf{q}$ denotes the cost vector in the second stage objective, and Equations (4.31) and (4.32) define the feasible region for decision variables $\mathbf{y}$. Clearly, the existence of a solution for the second stage problem for all possible realizations of $\boldsymbol{\xi}$ depends on the fixed value for $\mathbf{x}$. We denote by K the region for which such a solution indeed exists for all $\boldsymbol{\xi}$, and let $q(\mathbf{x}, \boldsymbol{\xi})$ be the optimal objective value for the second stage problem. When we make decisions $\mathbf{x}$ before observing $\boldsymbol{\xi}$, we are interested in the expected value of the second stage problem given our first stage decisions, that is, we are interested in $Q(\mathbf{x}) = \mathbf{E}\left[q(\mathbf{x})\right]$. Given cost vector $\mathbf{c}$ for the first stage decisions, we define the two-stage recourse problem as

$$\begin{aligned} \min \quad & \mathbf{c}^T\mathbf{x} + Q(\mathbf{x}) && && (4.33)\\ \text{s.t.} \quad & && && \\ A\mathbf{x} \quad &= \quad && b && (4.34)\\ \mathbf{x} \quad &\geq \quad && 0 && (4.35) \end{aligned}$$

where constraints (4.34) and (4.35) define the feasible region for $\mathbf{x}$. Since the second stage decisions are made after the observation of $\boldsymbol{\xi}$, they are commonly referred to as the *recourse* decisions. Intuitively, the two stage recourse problem consists of making decisions $\mathbf{x}$ before observing all unknown factors with the intent of minimizing first stage costs and the *expected* cost of the second stage problem, given that we are able to identify optimal recourse decisions $\mathbf{y}$ after observing said random factors.

So far, we have made no assumptions regarding the distribution of $\boldsymbol{\xi}$. A continuous distribution means we integrate over the support of $\boldsymbol{\xi}$ when calculating $Q(\mathbf{x})$, a task that is in general considered intractable for all practical problems. Assuming that we instead are dealing with a discrete random variable vector, we let Ξ denote the finite set of realizations of $\boldsymbol{\xi}$, and write $\boldsymbol{\xi}_s \in \Xi$ for outcome $s = 1, 2, \ldots, |\Xi|$ of the random variables. Commonly, we refer to such discrete realizations of the random variables as *scenarios*. In such cases, if the probability p_s of scenario $\boldsymbol{\xi}_s$ occurring is known, we have

$$Q(\mathbf{x}) = \sum_{s=1}^{|\Xi|} p_s q(\mathbf{x}, \boldsymbol{\xi}_s). \tag{4.36}$$

The two-stage recourse problem is well studied, and extensive theory regarding problems of this type exists in the literature. An important concept in this thesis is $robustness$. Informally, we characterize the robustness of a solution as its ability to withstand random effects. Conversely, we characterize the *flexibility* of a solution as its ability to accommodate random effects. Various sources use the terms differently. In this thesis, we refer to a solution as robust if it is able to withstand random effects regardless of the realized outcome. In the two-stage setting, a first stage solution $\hat{\mathbf{x}}$ is robust if

$$T\hat{\mathbf{x}} + W\mathbf{y} \quad = \quad \boldsymbol{\xi}_s, \qquad s = 1, 2 \ldots |\Xi| \tag{4.37}$$

In other words, robustness involves the guaranteed feasibility of the second stage problem for all possible outcomes of the random variables, i.e., $\hat{\mathbf{x}} \in K$. Note that this notion of robustness is not dependent on the existence of the recourse decisions $\mathbf{y}$. Problems for which no recourse actions are available after observing the random variable have a similar definition of robustness. In general, robust solutions require the sacrifice of flexibility and subsequently obtain conservative objective values.

4.4.2 Scenario Generation and Stability

In the previous section, we note that continuous random variables generally lead to intractability when solving a stochastic problem. Many algorithms for solving stochastic problems assume the randomness is represented by a discrete variable. An important issue thus arises when we model a problem from a real setting in which the randomness is known to have a continuous distribution. In such cases we discretize the randomness using some method, often referred to as *scenario generation*.

Generally, the goal is to discretize in such a way that it cannot be observed in the obtained solution of the model that a discrete approximation is used, i.e., results are identical to results obtained when using the true continuous distribution. Consequently, our discrete scenarios ideally capture all aspects of the continuous distribution that are of importance to the solution of the optimization problem. In practice, it is often not possible to verify whether or not this is the case.

Robust solutions provide an illustrative example of the problem of discrete approximations. Assume we discretize some known continuous distribution of a random variable in

our problem, but fail to capture the behavior of the tail in which the worst case outcomes are situated. Upon solving the problem, we obtain a solution which the model claims is robust to all random variable outcomes since the solution is always feasible for the discrete scenarios we include in the model. However, once we implement the solution in real life, we cannot choose to disregard tail probabilities, and our solution may fail horribly. Clearly, the scenario generation is part of the modelling process and is of crucial importance to ensure we model the problem in a sensible way. We often use in-sample and out-of-sample stability to measure the quality of generated scenarios.

If a scenario generation procedure is not deterministic in nature, it produces different scenario trees from different runs with the same data. According to King and Wallace (2012), *in-sample* stability ensures that we achieve roughly the same solution regardless of which scenario tree we use in our optimization. Let $\mathcal{T}_i$ and $\mathcal{T}_j$ be two different scenario trees generated using the same non-deterministic scenario generation procedure. Solving our optimization problem using each tree, we obtain solutions x_i^* and x_j^*, respectively. We have in-sample stability if

$$F(x_i^*, \mathcal{T}_i) \approx F(x_j^*, \mathcal{T}_j) \tag{4.38}$$

where $F(x_n, \mathcal{T}_n)$ is the objective value of solution x_n for scenario tree $\mathcal{T}_n$. Furthermore, we have out-of-sample stability if

$$F(x_i^*, \boldsymbol{\xi}) \approx F(x_j^*, \boldsymbol{\xi}) \tag{4.39}$$

where $\boldsymbol{\xi}$ denotes the true distribution of the random variables in our problem. That is, we solve our problem with two different sets of generated scenarios and evaluate the solutions in a setting in which the uncertainty is no longer approximated by the same discretization or sub-sample that generated the scenarios. Out-of-sample stability ensures we have not generated some stability in the scenarios that is not present in the real problem. However, evaluating a solution using the true random distribution is often difficult. If the true distribution is discrete but with an immense number of realizations, we are often able to measure out-of-sample stability since evaluating solutions is typically computationally inexpensive compared to solving the problem. Other approaches include the construction of simulations, or the weaker out-of-sample stability condition

$$F(x_i^*, \mathcal{T}_j) \approx F(x_j^*, \mathcal{T}_i). \tag{4.40}$$

This (weaker) condition for out-of-sample stability states that solutions obtained when solving a model for a generated scenario tree yield approximately the same objective values when evaluated in a different scenario tree.

4.4.3 VSS & EVPI

In general, solving a stochastic optimization problem, denoted as the recourse problem (RP), is harder than solving its deterministic counterpart. However, the stochastic formulation captures aspects of the problem that the deterministic model is unable to. We refer to the *value of the stochastic solution* (VSS) as the measure of the difference in the objective value when planning with uncertainty from when planning with expected values

only. That is, the VSS denotes how much a decision maker is willing to pay to incorporate uncertainty into the modelling. For a minimization problem, it is expressed as

$$VSS = EEV - RP \tag{4.41}$$

where EEV is the *expectation of the expected value model*. The EEV is calculated by setting all uncertain parameters to their expected value and taking the expectation in objective value from evaluating the solution in all scenarios. Note that the VSS is never negative if the model solutions are evaluated over the same set of scenarios as are included in the stochastic model, because planning with uncertainty never decreases the value of the solution in such cases.

The *expected value of perfect information* (EVPI) is defined as the difference in objective value between the *wait-and-see* (WS) solution and the RP. To calculate the WS, we first solve the model for each scenario with perfect information regarding the outcome of the uncertain variables. Then, WS is the expectation in objective value of these solutions over all scenarios. As the name suggests, EVPI denotes how much a decision maker is willing to pay for perfect information about the stochastic variables before making a decision. Thus, for a minimization problem, it is given by

$$EVPI = RP - WS. \tag{4.42}$$

In summary, for a conventional stochastic minimization problem with recourse, the following properties hold

$$EEV \geq RP \geq WS \tag{4.43}$$
$$EVPI \geq 0 \tag{4.44}$$
$$VSS \geq 0 \tag{4.45}$$

4.4.4 Multi-Objective Programming

Multi-objective optimization problems (MOPs) include more than one objective function. A MOP with k objectives is typically of the form

$$\begin{aligned} \max \quad & f_1(x) \\ \max \quad & f_2(x) \\ & \ldots \\ \max \quad & f_k(x) \\ \text{s.t.} \quad & x \in S \end{aligned} \tag{4.46}$$

where S is the set of feasible solutions. The objectives are often conflicting in that an improvement in one objective results in a lower value for one or more of the others. Thus there is no single optimal solution to the problem, but a set of solutions which we call *Pareto* optimal. Pareto optimality is based on the notion of dominance. A solution is dominated if there exists a solution that is strictly better for at least one of the objectives, while not being worse for any of the other objectives. A solution is *strictly* dominated if there

exists another solution that is strictly better for *all* objectives.

These two classifications of domination serve as the basis for two types of Pareto optimality. A solution is Pareto optimal if it is not dominated by any other solution, and *weakly* Pareto optimal if it is not strictly dominated by another solution. A set of (weakly) Pareto optimal solutions constitutes a (weak) Pareto front. This means that in a weak Pareto front, there might exist solutions that are strictly better for some of the objectives and equally good for the others. Such solutions are called weakly efficient solutions. An example of a weak Pareto front is shown in Figure 4.5.

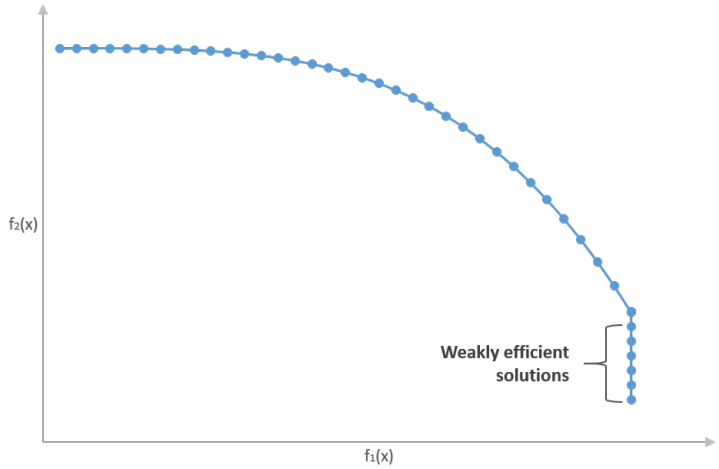

Figure 4.5: A Pareto front for a two objective functions with weakly efficient solutions. The two objectives, $f_1(x)$ and $f_2(x)$ are both maximized.

Chapter 5

Problem Description

In the following, the problem description is divided into two parts. First, the general aspects of RTO in a petroleum production setting are described, followed by an outline of the structural specifics of a petroleum production field.

Real-Time Optimization of Petroleum Production

Modern production of petroleum relies on the use of optimization in many key areas of the value chain. In this thesis, we are concerned with the real-time optimization of petroleum production based on historical measurement data. The real-time optimization provides decision support for a production engineer, with the main goal of maximizing petroleum production over a short time horizon while adhering to capacity constraints on uncertain production rates.

Historical measurement data from the oil field is used to build models of well outputs given input decision variables. Noise in the data and regions for which no measurements exist complicate the well modeling process and introduce considerable uncertainty with respect to the true output rates of each well. We aim to produce well models capable of capturing both the estimated output rates and the associated uncertainty.

Exceeding upper flow limits in the production system is costly. Although the objective value in RTO of petroleum production measures oil output rates, the main concern of the field engineer with respect to uncertainty is avoiding gas capacity constraint breaches. We simplify the problem in this thesis by treating oil production rates as deterministic, meaning any uncertainty in the problem is located in the gas production rates.

The well models are incorporated into the formulation of optimization problems. We aim to formulate problems able to provide insights into the effect of planning with uncertainty in a robust manner, as well as the trade-off between maximizing petroleum production and minimizing uncertainty in gas production. The short time horizon of RTO problems

typically implies they are solved repeatedly, often several times per day. It is vital that decision support arrives in a timely manner since field conditions continuously change, making outdated solutions no longer applicable. Thus, an important consideration is the computational requirements of solving our optimization problems.

Which set of operational adjustments to implement is ultimately decided by the production engineer. Ideally, decision support from the optimization models comes in a form that is directly implementable in the field. Consequently, we include in the well and optimization models explicit decision variables which the engineer can directly control.

In summary, the problem of interest in this thesis consists of modeling the performance and uncertainty of petroleum field wells, and including the well models in the formulation of RTO problems. Finally, the problem involves presenting the uncertainty in the obtained solutions in a manner that provides value to a production engineer with a variable degree of risk aversion.

Petroleum Field Structure

In the following, the general structure of a petroleum production field is characterized. The field is comprised of a number of production units, i.e., wells, each belonging to a platform. The routing hierarchy in the production field is organized as follows. Wells produce a mix of oil and gas which is routed to a separator. Separators split the production phases and route the resulting streams of oil and gas to an export line. The export lines lead production phases off-site.

A well may route its production flow to a subset of the separators in the field, depending on which platform the well belongs to. A well is only able to route to a single separator at any given time. The routing decision is made separately for each well, effectively allowing two wells belonging to the same platform to route to different separators. The function mapping from input variables to the outputs of a well is dependent on which separator it routes to.

There are a number of constraints present in the oil field production environment. Separators and wells have a capacity limit on the amount of gas they are able to process per time unit. This capacity is individual for each well and separator. The main export line is also subject to an upper limit on the gas rate.

Operational constraints are imposed by the preferences and risk aversion of the field operator. The operator typically wishes to limit the number of operational changes, as well as the relative change in decision variable settings with respect to the current operating point. That is, the operator prefers to implement small adjustments to decision variables and may have individual preferences for the upper limit on the relative change in the positive and negative direction for each variable separately.

Chapter 6

Optimization Models

This chapter presents and formulates MILP optimization models. The chapter is structured as follows. In Section 6.1 we formulate a stochastic model for the problem of optimizing petroleum production. A deterministic model follows from the stochastic model if all stochastic variables are locked to an assumed known scenario. We then formulate a neural network as a MILP in Section 6.2, before a simple model using Special Ordered Sets of Type 2 (SOS2) is presented in Section 6.3. Finally, a multi-objective version of the deterministic model is presented in Section 6.4.

6.1 Stochastic Model

We now formulate a stochastic model for maximizing oil production in an oil field. Since all uncertainty in the problem is assumed to be located in the gas production, the objective function is deterministic. For now, we make no assumptions regarding the model representing well production output as functions of decision variable inputs, but treat these models as general functions. A deterministic model follows from the stochastic model if scenario-indexed variables are locked to an assumed known scenario $s \in \mathcal{S}$, where $\mathcal{S}$ is the set of scenarios.

Model

Sets

$\mathcal{S}$	Set of scenarios
$\mathcal{K}$	Set of input dimensions
$\mathcal{P}$	Set of platforms
$\mathcal{I}_p$	Set of wells belonging to platform p
$\mathcal{U}$	Set of separators
$\mathcal{U}_p$	Set of separators connected to platform p
$\mathcal{P}_u$	Set of platforms connected to separator u

Indices

i	Well
k	Input dimension
u	Separator
p	Platform
s	Scenario

Parameters

$\overline{R}_{ik}$	Upper limit relative change for input dimension k of well i
$\underline{R}_{ik}$	Lower limit relative change for input dimension k of well i
$\overline{M}^{O}_{iu}$	Upper limit for oil output for well i when routing to separator u
$\overline{M}^{G}_{ius}$	Upper limit for gas output for well i when routing to separator u in scenario s
$\overline{M}_{iku}$	Upper limit for input dimension k for well i when routing to separator u

Y_i	1 if well i is producing prior to optimization, 0 otherwise
N	Maximum number of operational changes
X_{ik}	Initial setting of input dimension k for well i
Q_u^G	Capacity gas for separator u
Q^T	Capacity for total gas export
Q_i^W	Capacity for gas output locally in well i
G^*	Total amount of gas all platforms retain on-site for energy purposes

Variables

$\widetilde{o}_{iu}$	Piecewise linear approximation of oil production function for well i when routing to separator u
$\widetilde{g}_{ius}$	Piecewise linear approximation of gas production function for well i when routing to separator u in scenario s
x_{ik}	Setting for input dimension k for well i
$\mathbf{x}_i$	Vector of input settings x_{ik} for all input dimensions k for well i.
ϕ_{ik}	1 if input dimension k of well i is adjusted, 0 otherwise
z_{iu}	1 if well i routes its output gas and oil to separator u, 0 otherwise

Functions

$f_{iu}^O(\mathbf{x}_i)$	Function mapping from input vector $\mathbf{x}_i$ to the output of oil when routing from well i to separator u
$f_{ius}^G(\mathbf{x}_i)$	Function mapping from input vector $\mathbf{x}_i$ to the output of gas in scenario s when routing from well i to separator u

Objective Function
The goal of the optimization is to maximize the oil production rate, i.e., the sum of the approximated piecewise linear functions $\widetilde{o}_i$ for each well i summed over connected separators u.

$$\max \quad \psi \quad = \quad \sum_{p \in \mathcal{P}} \sum_{i \in \mathcal{I}_p} \sum_{u \in \mathcal{U}_p} \widetilde{o}_{iu} \tag{6.1}$$

Piecewise Linear Approximations of Oil and Gas Production
For each pair of well and connected separator, we construct a piecewise linear approximation function for each production phase, i.e., oil and gas. We constrain the function so that it equals the production rate when well i is routing to separator u, and 0 otherwise. This modeling step is necessary in order to sum over all production output in the objective function and separator gas constraints. The $\overline{M}^O_{iu}$ and $\overline{M}^G_{ius}$ variables take values according to the maximum output of oil and gas, respectively, for well i when routing to separator u, and scenario s in the case of gas output.

Oil approximation function constraints:

$$\widetilde{o}_{iu} \quad \leq \quad \overline{M}^O_{iu} z_{iu} \qquad \forall \quad p \in \mathcal{P}, i \in \mathcal{I}_p, u \in \mathcal{U}_p \tag{6.2}$$

$$f^O_{iu}(\mathbf{x}_i) - \widetilde{o}_{iu} \quad \leq \quad \overline{M}^O_{iu}(1 - z_{iu}) \qquad \forall \quad p \in \mathcal{P}, i \in \mathcal{I}_p, u \in \mathcal{U}_p \tag{6.3}$$

$$\widetilde{o}_{iu} - f^O_{iu}(\mathbf{x}_i) \quad \leq \quad 0 \qquad \forall \quad p \in \mathcal{P}, i \in \mathcal{I}_p, u \in \mathcal{U}_p \tag{6.4}$$

Gas approximation function constraints:

$$\widetilde{g}_{ius} \quad \leq \quad \overline{M}^G_{ius} z_{iu} \qquad \forall \quad p \in \mathcal{P}, i \in \mathcal{I}_p, u \in \mathcal{U}_p, s \in \mathcal{S} \tag{6.5}$$

$$f^G_{ius}(\mathbf{x}_i) - \widetilde{g}_{ius} \quad \leq \quad \overline{M}^G_{ius}(1 - z_{iu}) \qquad \forall \quad p \in \mathcal{P}, i \in \mathcal{I}_p, u \in \mathcal{U}_p, s \in \mathcal{S} \tag{6.6}$$

$$\widetilde{g}_{ius} - f^G_{ius}(\mathbf{x}_i) \quad \leq \quad 0 \qquad \forall \quad p \in \mathcal{P}, i \in \mathcal{I}_p, u \in \mathcal{U}_p, s \in \mathcal{S} \tag{6.7}$$

Routing to at Most One Separator
Each well can route to a maximum of one separator.

$$\sum_{u \in \mathcal{U}_p} z_{iu} \quad \leq \quad 1 \qquad \forall \quad p \in \mathcal{P}, i \in \mathcal{I}_p \tag{6.8}$$

Forcing Input to Zero When Not Producing
If a well is not producing, we constrain its input variables to take a value of 0 for all input dimensions. The big-M value, $\overline{M}_{iku}$, takes on the maximum recorded value for input *k* when routing from well *i* to separator *u*.

$$x_{ik} \quad \leq \quad \sum_{u \in \mathcal{U}_p} \overline{M}_{iku} z_{iu} \qquad \forall \quad p \in \mathcal{P}, i \in \mathcal{I}_p, k \in \mathcal{K} \tag{6.9}$$

Robust Separator and Well Capacity Constraints
In each separator, there is a constraint on the rate of the gas that flows through it. The sum of the contributions from each platform that routes gas to a given separator must therefore be less than or equal to the separator capacity. The gas production is also constrained locally in each well.

$$\sum_{p\in\mathcal{P}_u}\sum_{i\in\mathcal{I}_p}\widetilde{g}_{ius} \quad\leq\quad Q_u^G \qquad\qquad \forall\quad u\in\mathcal{U}, s\in\mathcal{S} \tag{6.10}$$

$$\widetilde{g}_{ius} \quad\leq\quad Q_i^W \qquad\qquad \forall\quad p\in\mathcal{P}, i\in\mathcal{I}_p, u\in\mathcal{U}_p, s\in\mathcal{S} \tag{6.11}$$

Total Export of Gas
The main export line is subject to a constraint on the total gas rate flowing through it. Gas retained in the field for energy purposes on platforms is subtracted from the total export.

$$\sum_{p\in\mathcal{P}}\sum_{i\in\mathcal{I}_p}\sum_{u\in\mathcal{U}_p}\widetilde{g}_{ius} - G^* \quad\leq\quad Q^T \qquad\qquad \forall\quad s\in\mathcal{S} \tag{6.12}$$

Maximum number of operational changes
The value of ϕ_{ik} is 1 if an operational change is made in the setting of input dimension k for well i, and 0 otherwise. The total number of operational changes is not allowed to exceed the maximum limit.

$$\sum_{p\in\mathcal{P}}\sum_{i\in\mathcal{I}_p}\sum_{k\in\mathcal{K}}\phi_{ik} \quad\leq\quad N \tag{6.13}$$

Tracking Operational Changes
We constrain the ϕ_{ik} variables to take the value of 1 if an operational change is made in input dimension k for well i. By altering the values of $\overline{R}_{ik}$ and $\underline{R}_{ik}$, the maximum relative change in the positive and negative direction in input dimension k for well i is adjusted. A well may be be shut off from any prior production level, or turned on from input setting 0. The rightmost terms of constraints (6.14) and (6.15) make sure this is allowed, while also forcing ϕ_{ik} to take the value 1 in this case, where $\overline{M}_{ik} = \max_{u\in U_p}\{\overline{M}_{iku}\}\ \forall\, p\in\mathcal{P},\ i\in\mathcal{I}_p,\ k\in\mathcal{K}$.

$$X_{ik} - x_{ik} \quad\leq\quad X_{ik}\underline{R}_{ik}\phi_{ik} + X_{ik}(1-\sum_{u\in\mathcal{U}_p}z_{iu})(1-\underline{R}_{ik}) \quad\forall\quad p\in\mathcal{P}, i\in\mathcal{I}_p, k\in\mathcal{K} \tag{6.14}$$

$$x_{ik} - X_{ik} \quad\leq\quad X_{ik}\overline{R}_{ik}\phi_{ik} + (1-Y_i)\overline{M}_{ik}\phi_{ik} \qquad\forall\quad p\in\mathcal{P}, i\in\mathcal{I}_p, k\in\mathcal{K} \tag{6.15}$$

Variable Bounds

$$
\begin{aligned}
\widetilde{o}_{iu} &\geq 0, && \forall \quad p \in \mathcal{P}, i \in \mathcal{I}_p, u \in \mathcal{U}_p && (6.16)\\
z_{iu} &\in \{0, 1\}, && \forall \quad p \in \mathcal{P}, i \in \mathcal{I}_p, u \in \mathcal{U}_p && (6.17)\\
\widetilde{g}_{ius} &\geq 0, && \forall \quad p \in \mathcal{P}, i \in \mathcal{I}_p, u \in \mathcal{U}_p, s \in \mathcal{S} && (6.18)\\
x_{ik} &\geq 0, && \forall \quad p \in \mathcal{P}, i \in \mathcal{I}_p, k \in \mathcal{K} && (6.19)\\
\phi_{ik} &\in \{0, 1\}, && \forall \quad p \in \mathcal{P}, i \in \mathcal{I}_p, k \in \mathcal{K} && (6.20)\\
& && && (6.21)
\end{aligned}
$$

6.2 MILP Formulation of Neural Networks with ReLU Activation Functions

In this section, we formulate a NN as a MILP modeling a possibly non-linear function $g(\mathbf{x})$, mapping from some input vector $\mathbf{x}_0$ to an output vector $\mathbf{x}_L$ defined as in the model below. We assume that the ReLU activation function is used in all layers of the network. The model assumes that the NN may include any number of input neurons. Since our purpose here is to obtain a general formulation of a network mapping from some input to an output, we do not specify a specific optimization objective but note that it is possible to maximize or minimize the modeled function.

Model

Parameters

N_l	Number of neurons in layer l
L	Number of layers
W_{ijl}	Weight between neuron i in layer $l-1$ and neuron j in layer l
B_{jl}	Bias of neuron j in layer l

Indices

j	Neuron
l	Layer

Variables

x_{jl}	Output from neuron j in layer l
x_{j0}	Input j to the neural network
s_{jl}	Negative component of the output from neuron j in layer l
z_{jl}	Indicating whether x_{jl} or s_{jl} are forced to zero in neuron j in layer k

Vectors

$\mathbf{x}_0$	Vector containing inputs $x_{1,0}, ..., x_{N_0,0}$
$\mathbf{x}_L$	Vector containing outputs $x_{1,L}, ..., x_{N_L,L}$

Functions

$g(\mathbf{x})$	The function to model. Takes as input $\mathbf{x}_0$ and gives as output $\mathbf{x}_L$

ReLU Neuron Activation Function
The output of each neuron is the maximum of 0, and the sum of its inputs multiplied with their respective weights plus the bias term. To represent this as a linear program, the output is decoupled into a positive and a negative component, x_{jl} and s_{jl}, respectively.

$$\sum_{i=1}^{n_{l-1}} W_{ijl} x_{i(l-1)} + B_{jl} \quad = \quad x_{jl} - s_{jl} \qquad l = 1, ..., L, j = 1, ..., N_l \tag{6.22}$$

Uniqueness
To ensure uniqueness of the solution to constraint (6.22), at least one of (x_{jl}, s_{jl}) is at any time equal to zero. The binary z_{jl} variables are used as indicator variables for this purpose. The big-M values M_{jl}^{+} and M_{jl}^{-} correspond to the maximum pre-rectified positive and negative output values, respectively, for neuron j in layer l.

$$\left.\begin{array}{lcl} x_{jl} & \leq & M_{jl}^{+}(1 - z_{jl}) \\ s_{jl} & \leq & M_{jl}^{-} z_{jl} \end{array}\right\} l = 1, ..., L, j = 1, ..., N_l \tag{6.23}$$

Variable Bounds

$$\begin{array}{rcll} x_{jl} & \geq & 0, & l = 1, \ldots, L, j = 1, \ldots, N_l \quad (6.24) \\ s_{jl} & \geq & 0, & l = 1, \ldots, L, j = 1, \ldots, N_l \quad (6.25) \\ z_{jl} & \in & \{0, 1\}, & l = 1, \ldots, L, j = 1, \ldots, N_l \quad (6.26) \\ x_{j0} & \geq & 0, & j = 1, \ldots, N_0 \quad (6.27) \end{array}$$

6.3 Special Ordered Sets Formulation

In this section, we formulate a SOS2 MILP, modelling a possibly non-linear function $g(\mathbf{x})$, mapping from some input vector $\mathbf{x}$ to an output vector $\tilde{\mathbf{y}}$ defined as in the model below. Since our purpose here is to obtain a general formulation of a function mapping from some input to an output, we do not specify an optimization objective but note that it is possible to maximize or minimize the modeled function. In the following, we assume that the input space has been divided into a set $\mathcal{R}$ of ordered, discrete breakpoints r, each point representing an input value. This model is formulated with only one input variable, but generalizes to higher dimensional input (see, e.g., Gunnerud and Foss (2009)).

Model

Sets

$\mathcal{K}$	Set of input dimensions
$\mathcal{D}$	Set of output dimensions
$\mathcal{R}$	Set of breakpoints

Indices

k	Input dimension
d	Output dimension
r	Breakpoint

Parameters

Y_{rd}	Value of output dimension d for breakpoint r
X_{rk}	Value of input dimension k for breakpoint r

Variables

λ_r	Weighting variable for breakpoint r
x_k	Input variable for input dimension k
$\tilde{y}_d$	Piecewise linear output for dimension d

Vectors

$\mathbf{x}$	Vector containing inputs $x_1, \ldots, x_{\lvert\mathcal{K}\rvert}$
$\tilde{\mathbf{y}}$	Vector containing outputs $\tilde{y}_1, \ldots, \tilde{y}_{\lvert\mathcal{D}\rvert}$

Functions

$g(\mathbf{x})$	The function to model. Takes as input $\mathbf{x}$ and gives as output $\tilde{\mathbf{y}}$

Linking Input and Output Values

We constrain each $\tilde{y}_d$ and x_k to take the value of our outputs and inputs of the modeled function. The sum of the weighting variables is set equal to 1 to ensure no infeasible values are produced.

$$\tilde{y}_d = \sum_{r \in \mathcal{R}} Y_{rd} \lambda_r \qquad \forall \quad d \in \mathcal{D} \tag{6.28}$$

$$x_k = \sum_{r \in \mathcal{R}} X_{rk} \lambda_r \qquad \forall \quad k \in \mathcal{K} \tag{6.29}$$

$$\sum_{r \in \mathcal{R}} \lambda_r = 1 \tag{6.30}$$

Variable Bounds

$$x_k \geq 0 \qquad \forall \quad k \in \mathcal{K} \tag{6.31}$$

$$\lambda_r \geq 0 \qquad \forall \quad r \in \mathcal{R} \tag{6.32}$$

$$\lambda_r \text{ SOS2} \qquad \forall \quad r \in \mathcal{R} \tag{6.33}$$

$$\tilde{y}_d \text{ free} \qquad \forall \quad d \in \mathcal{D} \tag{6.34}$$

6.4 Multi-Objective Optimization Model

In this section, we formulate a multi-objective optimization model for a petroleum production oil field. The model is based on the deterministic model. That is, it is based on the stochastic model formulated in Section 6.1, with stochastic variables locked to an assumed known scenario $s \in \mathcal{S}$. Using the same sets, indices, variables and functions, we make the following modifications to the model.

New Variables

$\mathbf{X}$	Matrix of dimensions $\|K\| \times \|I\|$ containing input setting vectors for all wells in the field

New Function

$g^G(\mathbf{X})$	Function mapping from field-wide input vector $\mathbf{X}$ to the uncertainty in output of gas in the field

New Multi-Objective Function
The objectives of the optimization are to maximize the total oil production rate and minimize the uncertainty in gas production. We rewrite to obtain a minimization objective for both terms and obtain as the multi-objective

$$\min \left[\psi(\mathbf{X}), \omega(\mathbf{X})\right] \tag{6.35}$$

where

$$\psi(\mathbf{X}) = -\sum_{p\in\mathcal{P}} \sum_{i\in\mathcal{I}_p} \sum_{u\in\mathcal{U}_p} \widetilde{o}_{iu} \tag{6.36}$$

$$\omega(\mathbf{X}) = g^G(\mathbf{X}) \tag{6.37}$$

The function $g^G(\mathbf{X})$ is here treated as a general function mapping from the input settings of all wells in the field to some measure of uncertainty. The output is some measure of the uncertainty in the field pertaining to gas outputs, such as the sum of all uncertainty (given that it is additive), or the uncertainty of the well with the maximum uncertainty, making the $\omega(\mathbf{X})$-objective a minmax function.

Chapter 7

Well Models

The optimization models in this thesis are formulated using a general function that takes as input the choke level of a well and gives as output the oil and gas production of the well with some measure of uncertainty for the gas output. Countless approaches exist to modelling this function. In this chapter, we discuss the motivation behind modelling the well outputs as neural networks. First, Section 7.1 gives a high level explanation of the concept of the well model. Then, Section 7.2 describes the well model we obtain from applying the ideas from Section 4.3. Finally, modelling assumptions made throughout this chapter are discussed in Section 7.3.

7.1 Well Model Concepts

In the modelling process of a well, we make assumptions regarding the relationship that describes output production rates as a function of input decision variables. In our setting, we assume that this relationship can be modelled with a single input variable, the choke setting, to a sufficiently accurate degree. In Section 5, it is assumed that oil output rates are deterministic in nature, while there is uncertainty with respect to gas output rates that needs to be accounted for. Figure 7.1 summarizes what a black-box version of our described well model is able to output given choke setting inputs.

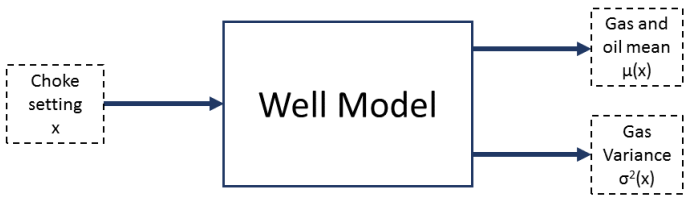

Figure 7.1: Visualization of what a well model takes as input and produces as output.

Given our specification of input variables and outputs from the well model, numerous choices for modelling their relationship are available. No standard approach exists in the literature. We now define two important properties that will guide our choice of well model representation:

1. The difficulty or ease with which we are able to update the well model with new data points.

2. The difficulty or ease with which we are able to incorporate the model in an optimization framework.

The first property is important since RTO typically involves frequent solving of an optimization model which ideally considers the most recent data we have learned from operating the field. Whenever new data is available, we update our mapping from inputs to outputs. If the choice of well models means they need to be created from scratch in order to consider new data, computational requirements may prevent us from performing the updates frequently. On the other hand, if the well models are able to start from their current state and learn to consider new data points, updating is a less taxing task.

The second property is important because the well models form the basis for the representation of wells in our optimization problems. Limiting the loss of accuracy in any approximations we apply when transforming our well models into MILP representations is vital. The scalability of the optimization problems is another consideration that affects our well modelling choices.

Based on these properties, we represent the well model as a NN. This choice both lets us swiftly update the model with new data, and, assuming that the proper activation functions are applied, lets us reformulate the model as a MILP with relative ease.

7.2 Neural Network Well Model

Combining the ideas of Section 4.3, we obtain a dropout NN able to produce the outputs from our conceptual well model from Section 7.1. Considering property 1 from Section 7.1, NNs are ideal models for frequently adding new data since the network is able to resume learning from its current state. An example of this is shown in Figure 7.2. Here, we first train a well model to points sampled from the function $f(x) = \sin(x)$. A gas output model is used in this case, estimating both the mean, plotted as the solid blue line, and total uncertainty represented by the blue shades, each corresponding to a standard deviation from the mean. We then add noise to a part of the input domain and resume training. The network updates its predicted uncertainty in the noisy region, as is seen by the expanded uncertainty around the added noise. The estimated mean curve is only slightly altered. This resumed training is considerably less computationally taxing than training a new model from scratch after noise is added.

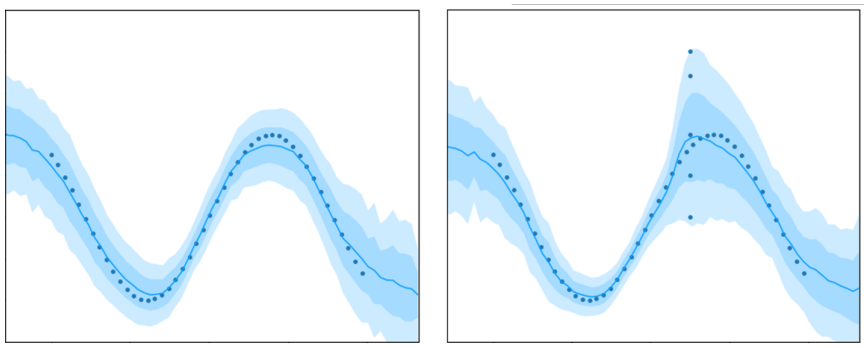

Figure 7.2: A well model trained on a sine curve before and after noise is added. The solid blue line represents the predicted mean, while each blue shade corresponds to a standard deviation from the mean.

Considering property 2 from Section 7.1, NNs are compatible with an optimization framework to a certain extent. If we choose linear or piecewise linear activation functions, NNs are reformulated as MILPs with relative ease. However, non-linear uncertainty components require sampling before we are able to represent them in a MILP model. This is discussed in Section 8.2.2.

An example visualization of a final NN well model is seen in Figure 7.3, where MC dropout is used to approximate a distribution for gas production and the estimated mean for oil production. The blue lines represent the mean, and the blue shades correspond to a standard deviation from the mean. The figures represent the two outputs from the conceptual well model described in Section 7.1.

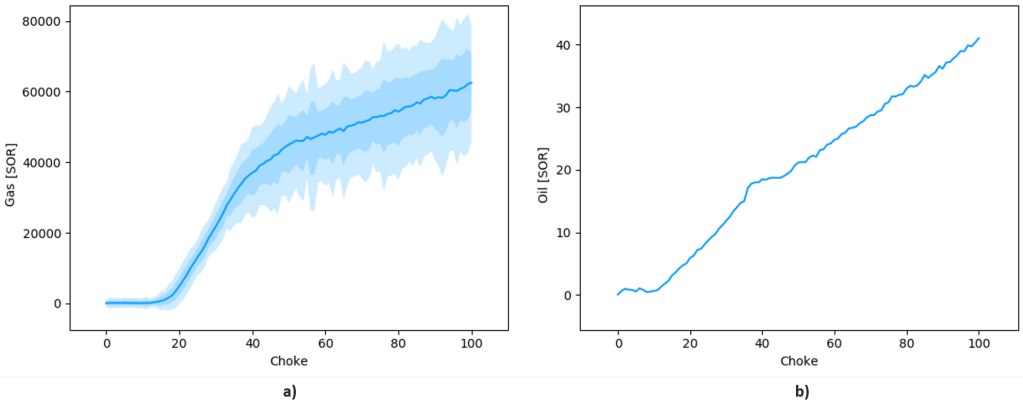

Figure 7.3: a) Example NN approximation of the distribution of the gas output of a well. The line represents the mean, and the blue shades each represent one standard deviation from the mean. **b)** Example NN approximation of the mean of the oil output of a well.

7.3 Well Model Assumptions

Physical factors like pressure, temperature, friction and water in the reservoir are not included in this model. In reality, these factors are explanatory variables for a function that describes the oil and gas output of a producing well, as visualized in Figure 7.4a). When we make the simplifying assumption that a single input dimension explains the output functions properly, we are projecting a high dimensional function down to a single dimension. In doing so, we inevitably lose part of our ability to explain the output function. However, we assume that the choke setting contains sufficient explanatory information such that a reasonable approximation is still possible to obtain. Furthermore, in the problem description of this thesis we state as a goal the inclusion of explicit decision variables in the problem which the decision maker is able to directly control. The choke setting is one such decision variable, with other examples including routing decisions and gas lift injection rates.

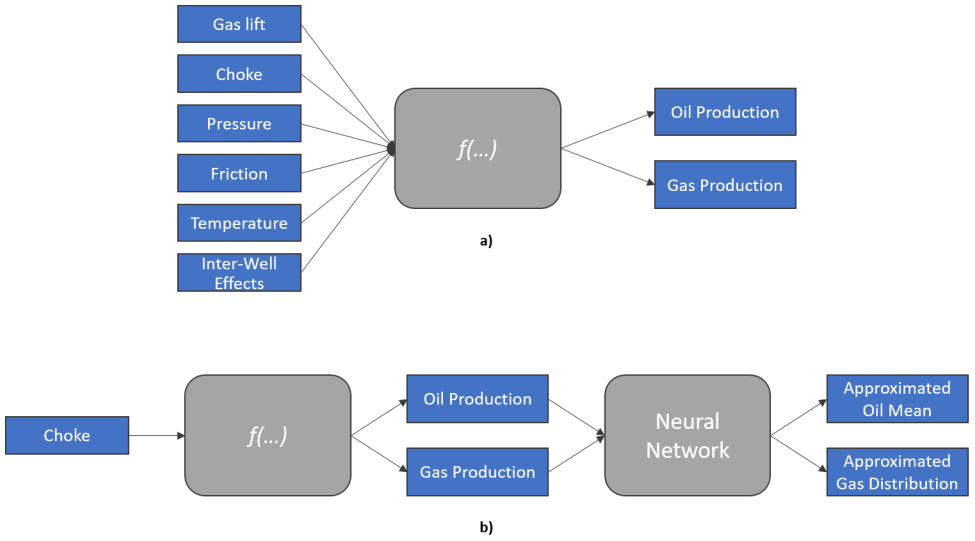

Figure 7.4: **a)** Shows what a real-world black box function takes as input and gives as output. **b)** Shows the approximation used in this report. Choke is the initial input and a neural network approximation is performed on the outputs of the black box function, to estimate distributions for oil and gas production.

Using Equation 4.26 as the loss function in our NN well model means we assume that the generating function of our well measurement data has normally distributed errors. This assumption allows us to predict the distribution of the well gas output. If the normal distribution is a poor approximation to the true generating function, we may end up with a poor estimate. This issue is discussed in Chapter 12.

We further note that wells are assumed to be independent of each other in our well models.

This assumption is not completely accurate in a real setting, since wells may share riser pipelines and are typically connected to the same reservoir.

Figure 7.4b) gives a visual representation of the approximations applied in this thesis. We assume that there exists some function f mapping from the choke level to the oil and gas production. This function is unknown, so we approximate the distribution of f using a neural network, based on observed outputs for varying inputs. In this thesis, these observations are represented by historical well measurements.

Chapter 8

Solution Method

In this chapter, we present the implementations of the models and methods described in this thesis. First, an overview of the components that form a solution framework is presented in Section 8.1. Then, Section 8.2 discusses the implementation of the well models described in Chapter 7. Section 8.3 presents the generation of scenarios for stochastic optimization, and approximations to the EVPI and VSS are described in Section 8.4. A heuristic recourse algorithm is described in Section 8.5, before implementation details of the optimization models are provided in Section 8.6.

8.1 Solution Framework

In this report, we develop optimization models based on historical data from a petroleum production field. As the models we present are unable to incorporate raw measurement data directly, the overall solution structure takes the form of a pipeline of different procedures. An overview of this pipeline is shown in Figure 8.1.

We provide a short summary of the different components in the solution method, while more detailed descriptions follow in subsequent sections.

1: Instruments and sensors in the production field record measurements during well tests or using MPFMs. We assume that any data validation has already been performed.

2: Measurements are stored as well data in a comma separated values (.csv) file. This file format allows for easy reading into, e.g., Python scripts.

3: In the first approximation step, the production mean and variance for oil and gas output are approximated using neural networks, as described in Chapter 7.

4: Datapoints, to be used in stochastic formulations, are sampled from the well models. The data points can be sampled in two manners: 1) from the mean and the variance of the

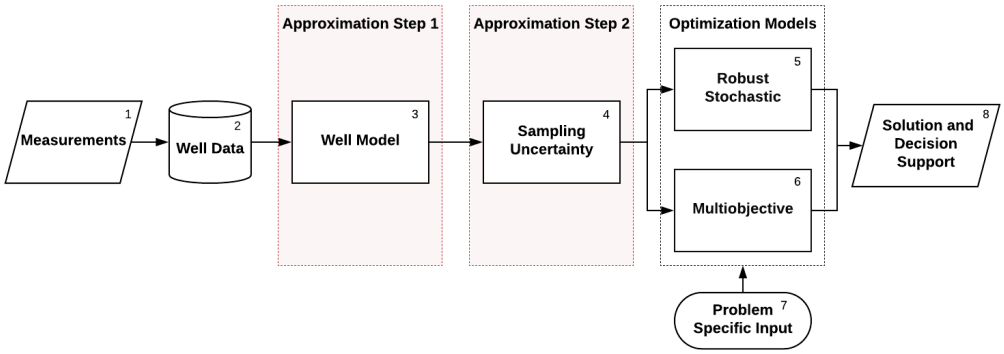

Figure 8.1: Holistic view of the solution framework from measurements to decision support.

outputs of a well separately, or 2) from some distribution capturing both the mean and the variance, and thereby sampling one of many possible realizations or scenarios for a given well. Sampled points are represented as SOS2 or NN MILPs.

5: In this stage, scenarios for the stochastic model are generated based on data sampled from the well models. Then, the robust stochastic optimization model maximizes oil production based on the scenarios and problem specific input.

6: In the multi-objective optimization model, the oil production is maximized and the total variance of the system is minimized, using the same problem specific input as the robust stochastic model.

7: The problem specific input includes constraint right-hand sides and the current operating point.

8: The output from the optimization models provides decision support and suggested system settings.

8.2 Well Model Implementations

We implement the well models as described in Chapter 7. Several steps are required; pre-processing of data, hyperparameter selection, NN training and sampling of non-linear uncertainty estimates. The pre-processing step is implemented by a discounting algorithm. A detailed presentation of this algorithm is found in Appendix A.

8.2.1 Tuning Neural Network Hyperparameters

No method exists for accurately determining the optimal hyperparameters for neural networks. In practice, we perform a grid or randomized search over the parameter space

and choose values which yield the best performance. The mean test score with 3-fold cross-validation is used to measure performance. Grid and randomized search lead to a combinatorial explosion of the search space as we increase the number of hyperparameters to search over. A way to limit this effect is to only search over a subset of all available parameters, and use heuristic rules for setting the remaining ones. We use heuristic rules based on what is considered reasonable values in the literature, from case studies of well-known problems. In Table 8.1, we present an overview of hyperparameters for our neural network models and our approach used to determine their corresponding values.

Reformulating our NNs as MILPs constrains us to linear or piecewise linear activation functions. We thus choose ReLU as activation functions in all hidden layer neurons, and linear activation functions in output neurons. We use the Adam optimizer and its recommended default learning rate *alpha* parameter setting of 0.001. Recommended values for the batch size typically are in the order of $32 - 512$. Keskar et al. (2016) show that larger batch sizes can lead to poor generalization of NN models. We thus choose a batch size of 32. To limit the time expended during grid search, we set the number of grid search training epochs to $5{,}000$. The network performance after this number of epochs gives a good indication of hyperparameter performance. A higher number of epochs is used when training the final networks to be applied in an optimization model.

This leaves a smaller subset of hyperparameters to be determined by a grid and randomized search; regularization term, dropout rate, the number of layers and the number of neurons in each layer. Prior to searching, we determine a search space for each parameter. The number of hidden layers and the number of neurons in each such layer directly affect the number of variables we introduce in our optimization model when reformulating the network as a MILP. Thus there is a trade-off between a sufficiently complex model and the computational cost we incur when introducing more integer variables in our optimization model. Since our regression task is a relatively simple problem compared to complex neural network applications such as image recognition, we find it reasonable to assume that 1 or 2 hidden layers with 20-40 neurons provide sufficient complexity. The regularization and dropout rates are sampled from ranges of what are common values found in the literature (see, e.g., Bergstra and Bengio (2012)).

Table 8.1: The available hyperparameters when training a neural network.

Hyperparameter	Symbol	Determination Method	Search Space
Regularization Term	λ	Grid/Randomized Search	$[1 \times 10^{-6}, 1 \times 10^{-2}]$
Dropout Rate	p	Grid/Randomized Search	$[5 \times 10^{-2}, 4 \times 10^{-1}]$
Number of Hidden Layers	L	Grid/Randomized Search	$1, 2$
Number of Neurons in Hidden Layers	N	Grid/Randomized Search	$20, 40$
Learning Rate	α	Heuristic Rule	N/A
Batch Size	B	Heuristic Rule	N/A
Training Epochs	T	Heuristic Rule	N/A
Activation Function	ϕ	Heuristic Rule	N/A

For each well, the hyperparameters from the grid and randomized search yielding the best performance are used to train the corresponding NNs and obtain the well models.

8.2.2 Sampling the Variance of the Well Models

Once we obtain our trained NN well models, we incorporate the models in a MILP optimization model. The Factor scenario generation procedure (Section 8.3.2) and the multi-objective optimization model (Section 8.6) both require a piecewise linear representation of the well model variance. However, the variance-output from the well model is not of a piecewise linear nature. This is because MC dropout is used to estimate the epistemic uncertainty, and the aleatoric uncertainty involves the natural exponential function of the output of the neural network.

To transform uncertainty estimates to a piecewise linear form, the variance is sampled from the well model for different values of the input variable. We then approximate the sampled points by linear or piecewise linear functions, e.g., another NN or a SOS2 formulation. Assuming that we train the NN to interpolate the sample points, the approaches produce equivalent results when incorporated in an optimization model. There are, however, differences in how these approaches scale with higher dimensional inputs. For a discussion regarding this topic, see Section 10.1 and Chapter 12. Note that contrary to the variance, we are able to directly reformulate the the mean output of the NN well model, assuming it uses linear or piecewise linear activation functions.

The NNs in the well model use MC dropout in order to estimate the mean output. When we reformulate the NNs as MILPs, applying dropout is no longer practically feasible. We defer a discussion and study of the differences between the MC dropout and NN MILP estimated mean to the computational study in Section 10.2.2.

8.3 Scenario Generation

For the optimization model developed in Section 6.1, a scenario s is represented by a function $f^G_{ius}(\mathbf{x})$ which provides a mapping from input vector $\mathbf{x}$ to the gas output of well i when routing to separator u. In this section we develop two different scenario generation procedures for generating $f^G_{ius}(\mathbf{x})$ based on our well models, which we name the *Factor* scenario generation and the *Markov Weighted* scenario generation.

Note that indices for wells i and separators u have been dropped for ease of notation for the remainder of this section.

8.3.1 Desirable Properties of Generated Scenarios

When a given well is producing, we assume that the true input and output values are observable in the oil field. However, the true production function for the remaining input

domain is uncertain. A generated scenario therefore ideally is able to intersect a known operating point, so that a status quo can be maintained by not changing any input variables.

A scenario also ideally is not excessively complex to generate. When setting new operating points, the system may require the generation of a large number of new scenarios within a short window of time.

In this thesis, our belief of the underlying distribution of the uncertainty in the problem is represented by the normal distributions obtained from the well model NNs. Following the discussion in Section 4.4.2, we wish to discretize the uncertainty in the gas output such that our scenarios capture all relevant aspects of this distribution. One straightforward approach is to sample directly from the NN well model distribution $\mathcal{N}(\mu(\mathbf{x}), \sigma^2(\mathbf{x}))$ for some uniformly distributed x in the input domain. As seen in Figure 8.2, the result of this approach is a rapidly fluctuating scenario. In reality, we do not expect that minor changes in the choke settings lead to extreme variations in the gas output. That is, we expect a higher degree of smoothness in a scenario well curve than is observed in Figure 8.2. Consequently, it seems the method of directly sampling from the well model distribution fails to produce meaningful scenarios.

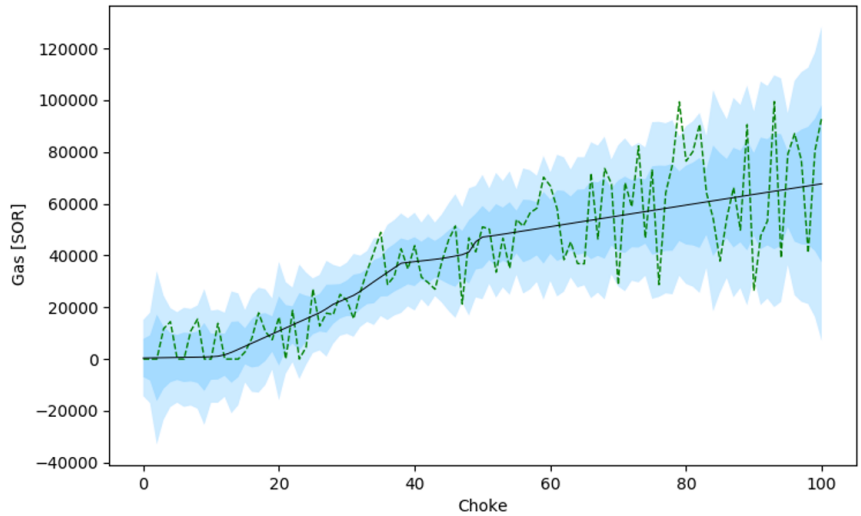

Figure 8.2: Scenario as the dashed green line, sampled from a well model distribution, with the mean as the solid black line and each blue shade representing a standard deviation.

8.3.2 Factor Scenario Generation

The idea behind the Factor scenario generation procedure is to represent the gas output of a well by a linear combination of its mean gas output $\mu(\mathbf{x})$ and the standard deviation of

its gas output $\sigma(\mathbf{x})$, that is

$$g_s^G(\mathbf{x}) = \mu(\mathbf{x}) + A_s\sigma(\mathbf{x}) \tag{8.1}$$

where A_s represents the *factor* for the standard deviation of the gas output for the well in scenario s. Negative factors produce negative output values in certain cases, so we re-parameterize the output as

$$f_s^G(\mathbf{x}) = max(0, g_s^G(\mathbf{x})). \tag{8.2}$$

From our well models described in Chapter 7 we obtain $\mu(\mathbf{x})$ and $\sigma(\mathbf{x})$. Thus generating one scenario s consists of choosing one constant factor A_s for each well when paired with each of its connected separators. We draw these constant factors from some random distribution with mean 0 and variance 1, for example, the triangular or truncated normal distribution. It is desirable to use a truncated distribution as we do not wish to generate scenarios with factors that are excessively large in absolute value. If we choose the (truncated) normal distribution $\mathcal{N}(0, 1)$, approximately 68% of generated scenario curves fall within 2 standard deviations of the mean.

Since we model well output variance as heteroscedastic, the Factor scenario generation procedure is not equivalent to shifting the mean gas output by a constant amount. Regions of the curve are distorted differently along the input axis according to the estimate of $\sigma(\mathbf{x})$. The deviation from the mean relative to the standard deviation is constant for all input values $\mathbf{x}$ in a single scenario.

Examples of different Factor scenarios are visualized in Figure 8.3. Here, 10 scenarios are plotted with the mean of the well represented by the black line, while the two shades of blue represent one standard deviation from the mean each. A variety of factors are selected for visualization purposes. In reality, most scenarios are clustered close to the mean with some outliers when drawing from, e.g., the truncated normal distribution.

In order to incorporate Factor scenarios in a MILP optimization model, we need a piecewise linear representation of $\mu(\mathbf{x})$ and $\sigma(\mathbf{x})$. From our well models we reformulate the NN as a MILP to obtain the piecewise linear $\hat{\mu}(\mathbf{x})$. We obtain the piecewise linear approximation $\hat{\sigma}(\mathbf{x})$ through sampling the variance and either fitting a new NN to the sample or using SOS2 variables, as described in Section 8.2.2. The re-parameterization from Equation (8.2) is implemented by the following equations:

$$\begin{aligned}
f_s^G(\mathbf{x}) &= g_s^+ && (8.3)\\
\hat{\mu}(\mathbf{x}) + A_s\hat{\sigma}(\mathbf{x}) &= g_s^+ - g_s^- && (8.4)\\
g_s^+ &\leq M^+(1 - z_s) && (8.5)\\
g_s^- &\leq M^- z_s && (8.6)\\
g_s^+ &\geq 0 && (8.7)\\
g_s^- &\geq 0 && (8.8)\\
z_s &\in \{0, 1\} && (8.9)
\end{aligned}$$

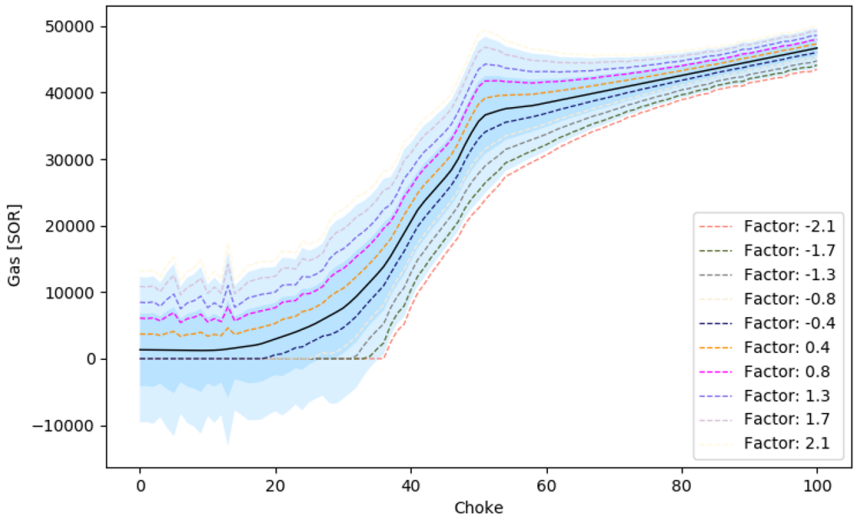

Figure 8.3: Ten Factor scenarios plotted with the mean (black) and the standard deviation from the mean (each shade of blue representing one standard deviation).

Here, M^+ and M^- represent the maximum value of the Factor scenario gas output in the positive and negative direction, respectively. The g^+ and g^- variables represent the positive and negative component of the Factor scenario gas output, respectively, while z_s is an indicator variable to force uniqueness of the solution to Equation (8.4). Not forcing uniqueness leads to g_s^+ and g_s^- effectively becoming free variables with infinitely many solutions to Equation (8.4). In such cases we are not able to recover the positive component of the scenario and end up with unpredictable values.

An attractive feature of this scenario generation procedure is that $\mu(\mathbf{x})$ and $\sigma(\mathbf{x})$ are scenario independent. Regardless of the number of scenarios we include in our stochastic optimization model, we require only two piecewise linear models per possible well-separator pair. Furthermore, generating scenarios is simple and efficient.

A drawback of the Factor scenario generation procedure is that the scaling occurs along the entire input axis. Thus an inherent weakness of this scenario generation procedure is the inability of the generated curves to deviate from the general shape of the mean. The parameterization over A_s instead of over $\mathbf{x}$ thus introduces considerable bias in the scenario, and factor generated scenarios never intersect the mean curve (except in the case of $A_s = 0$). Additionally, there exists a unique mapping from an input-output pair $(\mathbf{x}, f_s^G(\mathbf{x}))$ to a scenario factor A_s since no two scenarios intersect the same point. In Figure 8.4 the plotted Factor scenario is the only scenario to intersect the point marked by the red dot. One consequence of this is that specifying an initial input setting and corresponding output gas

value for a well-separator pair forces all scenario constants for this pair to take the same value. To the model, the production curve has in this case become deterministic for the specific well-separator pair. The Factor scenarios are therefore more suited in applications where we either are uncertain about initial output values, or where one or more wells are switched off prior to optimization.

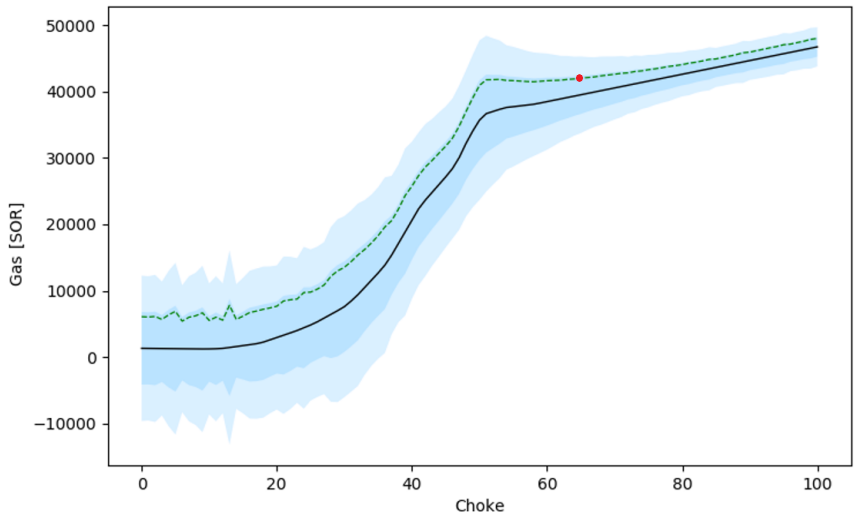

Figure 8.4: A Factor scenario shown as the green dotted line is generated with a pre-specified point marked with a red dot. The plotted scenario is the only Factor scenario to intersect the pre-specified point.

8.3.3 Markov Weighted Scenario Generation

The Markov Weighted scenario generation procedure is based in part on the Markov assumption that the conditional probability of future states is independent of past states, given the present state. In our setting, we consider a state to be a point describing an input setting and a corresponding gas output value. We find the Markov assumption to be reasonable in this setting since we do not expect that minor modifications to the choke settings lead to extreme variations in the corresponding output rates. Such extreme behavior is observed in the approach described in Section 8.3.1, where the output of each state is drawn independently of the previous state.

Under the Markov assumption, the dependence of a previous state has a smoothing effect on the curve. We note that it is possible to vary this smoothing effect by varying the number of previous states on which any state depends. For instance, exponential smoothing has the ability to depend on an arbitrary number of past states. We model the dependence on

the previous state only, based on our assumptions and expectations of the smoothness of each scenario curve. In the following, we limit the discussion to a single input dimension. The procedure can, however, be adapted to higher dimensions with minor modifications.

Assume that the input domain is divided into N discrete points and that input and output values are denoted by x_n and $g_s^G(x_n)$, respectively, for $n = 1, 2, \ldots, N$. Under the Markov assumption, we let $g_s^G(x_n)$ be dependent on a weighted combination of one of its neighbors in the input domain, given by $g_s^G(\widehat{x}_n)$, and a random draw from the distribution of the well models developed in Chapter 7, given by $\alpha \sim \mathcal{N}(\mu(x_n), \sigma^2(x_n))$. That is, we let

$$g_s^G(x_n) = w g_s^G(\widehat{x}_n) + (1 - w)\alpha \tag{8.10}$$

where $w < 1$ represents the weighting variable and is user-specified. Since Equation 8.10 produces negative output values in certain cases, we re-parameterize to obtain

$$f_s^G(x_n) = max\{0, g_s^G(x_n)\}. \tag{8.11}$$

For any point x_n, a single neighbor $\widehat{x}_n$ is selected as the state x_n partially depends on. We select neighbors such that no cross dependencies occur. In other words, if we start the generation in a point $x_n > 0$, all points x_m where $m > n$ depend on points x_{m-1}, while all points x_k where $k < n$ depend on points x_{k+1}. If we start in $x_1 = 0$, all points x_l where $l = 2, 3, \ldots, N$ depend on points x_{l-1}. Thus, in order to start generating scenarios, we specify a starting point somewhere in the input domain and a corresponding output value. This point represents a ground truth through which the Markov Weighted scenario gas curve intersects. We choose either an initial input setting for which the current gas output rate is known, or the 0 input point for which we with certainty have a gas output rate of 0 since the well is shut off. We note that if we specify $w = 0$, the method is equivalent to the random distribution draw approach as discussed in Section 8.3.1 and shown in Figure 8.2.

Starting with the known point, we then iterate and construct a scenario by obtaining values for the remaining $N - 1$ points. Algorithm 1 shows the step-wise procedure. A few comments on the details of the implementation are in order. Lines $5-7$ create the choke values for which we will sample the scenario gas output curve. If an initially known point is specified, the algorithm first iterates in the positive choke direction from the known point up to the maximum choke setting of 100, before iterating downward from the known point to the minimum choke setting of 0. Lines 11 and 17 create a scaling parameter which is 1 (and hence redundant) in all cases except for the neighbors $Y[\hat{i}-1]$ and $Y[\hat{i}+1]$ of the specified known point. This is necessary since the change in choke value is less when moving from the pre-specified point to its neighbors than it is when moving between other neighboring points. Sampling the scenario with less spacing in the input dimension close to the specified point leads to increased volatility around it since the curve is allowed to vary more "often". This behavior is the opposite of what we expect to see in a real scenario since we normally associate lower uncertainty with points close to a known production setting. We thus scale down the variance of the random draw for $Y[\hat{i} - 1]$ and $Y[\hat{i} + 1]$. If no initially known point is specified, the algorithm iterates from the 0 choke and gas output point, and

no scaling occurs.

The algorithm returns a Markov Weighted scenario, represented by a set of choke values and corresponding gas output values. In order to incorporate such scenarios in an optimization model, we need a piecewise linear formulation of the scenario curves. Representing the points with SOS2 variables is a straightforward task, particularly in the case of a single input dimension. Alternatively, a neural network with ReLU activation functions is fit to the scenario curves and reformulated as a MILP in the optimization model.

Algorithm 1: Algorithm for generating a Markov Weighted scenario for a single well-separator pair. Single input dimension implementation.

Input : Weight parameter $w < 1$
Number of sampling points N
Distribution of well-separator gas output given by well models $\mathcal{N}(\mu(x), \sigma(x))$
Optional: An initial choke setting $\hat{x}$ and corresponding gas output value $\hat{y}$

Output: A scenario represented by a set of choke values X and corresponding gas output values

if $(\hat{x}, \hat{y})$ *not specified* **then**
  $(\hat{x}, \hat{y}) \leftarrow (0, 0)$;
$\lambda^* = \frac{100}{N}$;
Let $Y \leftarrow \{0, \ldots, N\}$ be a new array;
$X \leftarrow \{0, \lambda^*, 2\lambda^*, \ldots, N\lambda^*\}$;
$X \leftarrow X \cup \{\hat{x}\}$;
Sort X by increasing choke values;
$\hat{i} \leftarrow$ index of $\hat{x}$ in X;
for $i = \hat{i} + 1$ **to** N **do**
  $x \leftarrow X[i]$;
  $\lambda \leftarrow \frac{X[i]-X[i-1]}{\lambda^*}$;
  $\alpha \sim \mathcal{N}(\mu(x), \sigma(x)^2)$;
  $Y[i] \leftarrow max\{0, w * Y[i-1] + (1-w) * ((\alpha - \mu(x)) * \lambda + \mu(x))\}$
end
for $i = \hat{i} - 1$ **to** N **do**
  $x \leftarrow X[i]$;
  $\lambda \leftarrow \frac{X[i+1]-X[i]}{\lambda^*}$;
  $\alpha \sim \mathcal{N}(\mu(x), \sigma(x)^2)$;
  $Y[i] \leftarrow max\{0, w * Y[i+1] + (1-w) * ((\alpha - \mu(x)) * \lambda + \mu(x))\}$
end
return X, Y;

Figure 8.5: Procedure for generating a Markov Weighted scenario for a single well-separator pair.

Figure 8.6 shows 10 Markov Weighted scenarios generated with no initial point specified, meaning the algorithm starts its iteration from the point of 0 choke and 0 output gas. 10 points are sampled and weight parameter $w = 0.4$ is used. Figure 8.7 shows 10 Markov Weighted scenarios generated with the same hyperparameter settings, but with a known point specified. All scenarios intercept the point, marked by a red dot in the figure.

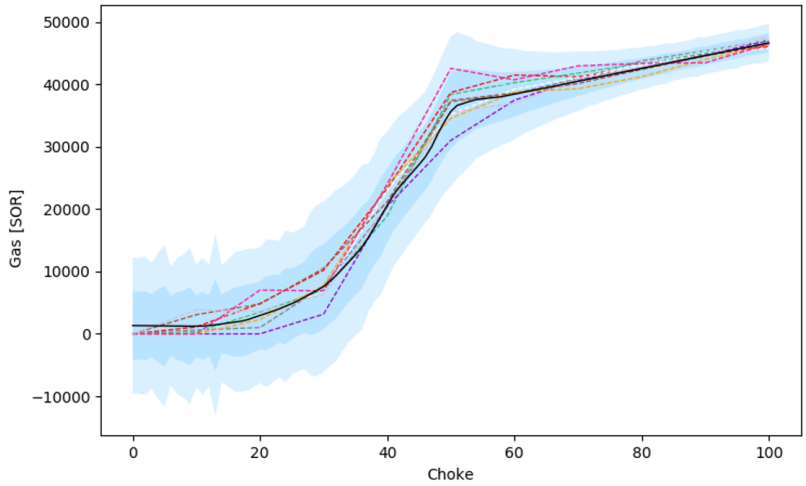

Figure 8.6: Ten Markov Weighted scenarios plotted with the mean as the solid black line and each shade of blue representing one standard deviation from the mean.

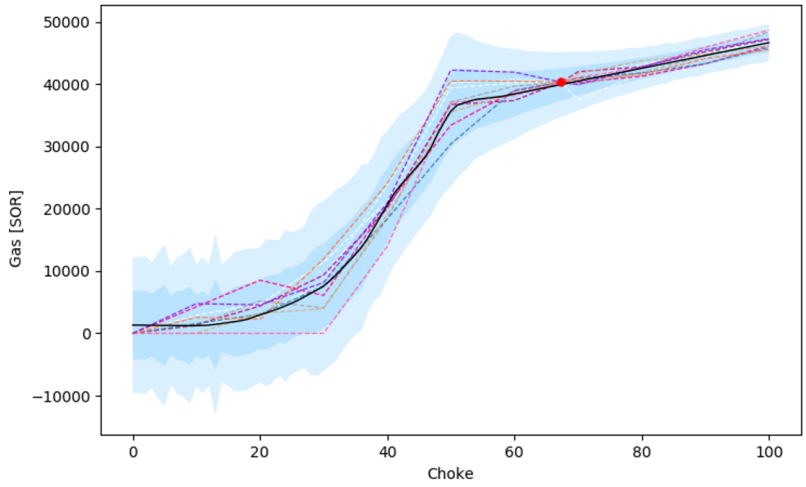

Figure 8.7: Ten Markov Weighted scenarios plotted with a known operating point marked by the red dot. The mean is given by the solid black line and each shade of blue represents one standard deviation from the mean.

Two hyperparameters are user specified when generating Markov Weighted scenarios: The weighting w between neighboring points and the random draw, and the number of points N in which the scenario curve is sampled. In general, lower values for w and higher values for N increase the volatility of the generated scenario since the curve is allowed to vary to a larger extent more often. The number of sampling points N also affects the complexity of the scenario in the optimization problem, i.e. the number of breakpoints needed in a SOS2 representation and the minimum amount of neurons needed in the NN MILP representation. Visualizations of how the smoothness of Markov Weighted scenarios varies with the hyperparameters are found in in Appendix D. Determining the optimal hyperparameters is case dependent. In our case, 10 sampling points (11 if the 0 choke setting is counted) are used, along with a weight $w = 0.4$, producing sufficiently smooth scenarios which are not overly complex to represent as MILPs.

8.4 Value of the Stochastic Solution and Perfect Information

The concepts of VSS, EVPI and EEV we introduce in Section 4.4.3 are in their conventional sense developed for multi-stage recourse problems. The stochastic optimization model we formulate in Section 6.1 consists only of a single decision stage with a robust formulation to ensure that the uncertain gas output does not exceed the gas capacity constraint for any of the scenarios the problem includes. We cannot directly calculate the VSS, EVPI and EEV because solutions, when evaluated in external "true" scenarios, are infeasible if the solution fails to adhere to the gas capacity constraint. We thus must turn to some approximation of the value of solving a stochastic robust optimization model over a deterministic one, as well as the EVPI. We perform this approximation over a set of true scenarios $s \in \mathcal{S}$.

Our approach is to introduce a penalty if the suggested solution of the optimization model is infeasible given the true scenario. A possible penalty is to record an objective value of 0 whenever infeasibility occurs. We call this approach the *Strict* penalty. We observe that this penalty makes no distinction between a constraint breach of 1 SOR and a breach of $100{,}000$ SOR. A less strict alternative is to assume that we are always able to revert the optimization solution to the initial operational settings if these are feasible. Thus, in the case of an infeasible solution, we simply record the objective value of the initial setting prior to optimization if it is feasible. We call this approach the *Reversion* penalty. If the initial setting is infeasible, e.g., due to a constraint right-hand side changing, we cannot revert. In a real application, the production engineer at this point takes action to reach feasibility. We therefore assume that wells are shut off in some prioritized order until feasibility is reached. We call this approach the *Switch-Off* penalty, applicable to cases in which we have no feasible initial setting or where the initial setting corresponds to all wells being shut off. We choose the simple priority of first switching off wells that violate individual gas output constraints, if any. If this does not produce a feasible solution we turn wells off in the order of highest to lowest GOR in the hopes of reducing gas output at minimal cost to oil output.

Note that in the case of the WS solution, infeasibility never occurs since the optimization model reduces to a deterministic robust formulation with perfect information.

Let $\psi^*(\mathbf{x}^*)$ be the objective value of the optimal solution $\mathbf{x}^*$ to an optimization model, which may be a deterministic model based on expected values in the EEV case, a stochastic robust formulation or a WS model with perfect information. Let $\psi^0(\mathbf{x}^0)$ be the objective value of our initial production setting $\mathbf{x}^0$, and $\psi^{OFF}(\mathbf{x}^*)$ the objective value obtained after switching wells off from an infeasible solution according to a priority list. For a true scenario s, the true objective value ψ_s is calculated according to the following equations.

Strict penalty:

$$\psi_s(\mathbf{x}^*) = \begin{cases} 0 & \text{if } \mathbf{x}^* \text{ infeasible} \\ \psi^*(\mathbf{x}^*) & \text{otherwise} \end{cases} \tag{8.12}$$

Reversion penalty:

$$\psi_s(\mathbf{x}^*) = \begin{cases} \psi^0(\mathbf{x}^0) & \text{if } \mathbf{x}^* \text{ infeasible} \\ \psi^*(\mathbf{x}^*) & \text{otherwise} \end{cases} \tag{8.13}$$

Switch-Off penalty:

$$\psi_s(\mathbf{x}^*) = \begin{cases} \psi^{OFF}(\mathbf{x}^*) & \text{if } \mathbf{x}^* \text{ infeasible} \\ \psi^*(\mathbf{x}^*) & \text{otherwise} \end{cases} \tag{8.14}$$

In real applications, the appropriate penalty is case dependent and ideally is determined by field engineers with intimate knowledge of the relevant production system.

8.5 Heuristic Recourse Algorithm

In Section 2.1, the way in which decisions in the petroleum production field are implemented is discussed. When we solve an optimization model, the obtained objective value is a measure of how well we expect our solution to perform if we implement all suggested operational changes. In a real application, operational changes are often implemented in a serial manner with significant time passing between each change to let the production system settle in a steady state. Ideally, an RTO model is updated with new data and resolved between each such change. In this section, we develop an algorithm that attempts to model this sequential implementation strategy.

From Chapter 7 we obtain well models in which the gas output is uncertain, while we formulated the gas output capacity constraint as a hard limit in the optimization models developed in Chapter 6. Thus there is a possibility that implementing an operational change in the field yields an infeasible solution. In such cases, the optimization models do not include recourse decisions and are unable to produce a suggested action that moves

the solution to feasibility.

In a real application, if an implemented operational change leads to infeasibility it is partially or fully reverted. However, we are not back to square one; we gain information about the nature of the gas output curve of the well. We find it reasonable to attempt to update the RTO model with this new information in order to produce more accurate solutions.

Algorithm 2: Algorithm for step-wise optimizing petroleum production with recourse actions.

Input : Set of initial case specific parameters $\mathcal{P}$
Maximum number of allowed changes N
Number of scenarios in the optimization model S
A change prioritization criteria $\varphi(X)$
Markov Weighted scenario $\mathcal{S}^*$ containing true well curves,
i.e., $f^{G*}_{ius}(\cdot)$ for all well-separator pairs i, u.

Output: The total number of infeasible settings that occurred, final true oil and gas output, final input settings.

Model $\leftarrow$ Stochastic model initiated with parameters $\mathcal{P}$;
infeasible_count $\leftarrow 0$;
for $n \leftarrow N$ **to** *1* **do**
    Set maximum allowed changes in Model to n;
    Solution $\leftarrow$ Solve Model to optimality;
    $X^* \leftarrow$ optimal choke settings for Solution;
    $x^*_{change} \leftarrow$ choke setting of well selected by priority criteria $\varphi(X^*)$;
    $(i^*, u^*) \leftarrow$ index of prioritized well and connected separator;
    $G^{x^*_{change}} \leftarrow$ New gas output for x^*_{change}, i.e. $f^{G*}_{(i^*,u^*)s}(x^*_{change})$;
    $G^{TOT} \leftarrow$ Total new true gas output if we implement x^*_{change};
    if $G^{TOT} > Q^T$ **or** $G^{x^*_{change}} > Q^G_i$ **then**
        infeasible_count $\leftarrow$ infeasible_count+1;
    else
        Model $\leftarrow$ update with new choke setting x^*_{change};
    end
    if $x^*_{change} \neq 0$ **then**
        Model $\leftarrow$ replace $f^G_{(i^*,u^*)s}(\cdot)$ with $f^{G*}_{(i^*,u^*)s}(\cdot)$ for all scenarios s
    end
end

Figure 8.8: Algorithm for implementing sequential changes as suggested by a stochastic optimization model.

Algorithm 2 shows a recourse algorithm (RA), which models the process of performing one change at a time and updating the optimization model with true well performance data after each change. One run of the algorithm requires a specification of the ground truth

scenario which the model learns as it implements changes. The for loop initiated in line 4 iterates from the maximum number of changes allowed down to a single allowed change. As we solve the stochastic model in the loop with more than one change allowed, we need to make a choice regarding which suggested change to implement first. The *priority criteria* function $\varphi(X)$ implements this choice according to our preferences. Typically we implement negative changes to choke settings prior to positive ones, as doing the opposite often means we visit an infeasible state. When no negative changes are suggested, we choose which well to implement a positive change in according to a heuristic. One simple heuristic is to sort all wells from lowest GOR to highest, and then select the first well in the list for which the optimization model suggests a change.

If a suggested change does not lead to infeasibility in the ground truth scenario, it is implemented in the model in line 15 of the algorithm. Note that in line 17, the model learns the ground truth for the well it suggests a change for (unless the change is to switch a well off), regardless of whether the suggested change leads to infeasibility or not. This is because in a real application, the infeasible suggestion is implemented and then rolled back after observing infeasibility. Thus we still gain information about the true well performance even when reverting the operational change in the well.

We remark that if the RA is applied to cases where the initial setting is infeasible, the algorithm is altered slightly. Infeasible initial states are possible when the right-hand side of the gas capacity constraint suddenly changes, e.g., due to separator failure. In such cases, we allow implementing operational changes that lead to an infeasible state, until a feasible state is visited. Without this modification, the algorithm causes all changes to be reverted until only 1 permitted change remains. At this point, the optimization model typically is forced to switch a well off in its entirety in order to reach feasibility. With the modification, the model is allowed to gradually approach feasibility while limiting the loss in oil production. In such cases, the RA returns an infeasible_count of 1 if it failed to reach feasibility and 0 otherwise.

When the model learns the ground truth, it does so for the entire well curve. This is slightly unrealistic as we in reality only gain a new data point and other unvisited input settings are still uncertain. Consequently, any performance gains we obtain from the RA over the standard robust model are optimistic estimates of gains that are achievable in a real setting. Possible approaches to a more realistic model of the process of learning points of the true well curves are discussed more thoroughly in Chapter 12.

8.6 Multi-Objective Program Implementation

The multi-objective model formulated in Section 6.4 minimizes total negative oil output $\psi(\mathbf{X})$ and some measure of total uncertainty $\omega(\mathbf{X})$. Here, $\mathbf{X}$ represents the set of choke settings for all wells in the problem. Our well models yield an estimate of the variance $\sigma_i^2(x_i)$ of the gas output of each well i. Note that the indices for separators are dropped for ease of notation. Since we assume in Section 7.3 that wells are independent of each other, a measure of total uncertainty is given by the sum of the variances of all wells. That

is, we let

$$\omega(\mathbf{X}) \quad = \quad \sum_{i \in \mathcal{I}} \sigma_i^2(x_i). \tag{8.15}$$

This optimization objective requires a piecewise linear representation of $\sigma_i^2(x_i)$, obtained using the sampling methods described in Section 8.2.2. A few notes on the choice of objective are in order. First, since we have estimates of the mean and variance of the sum of gas outputs of all wells, we are able to calculate the probability of breaching the total gas capacity constraint. The calculations depend on the assumption of normally distributed errors in the true function generating our data, as stated in Section 7.3. Furthermore, we stress that we in the following calculations do not consider the probability of breaching individual well capacity constraints since the resulting calculations require estimation of the conditional cumulative probability distribution. Let the probability of breaching the constraint given (field wide) solution $\mathbf{X}^*$ be denoted by $P(\text{Infeasible}|\mathbf{X}^*)$, the summed variance in gas output be denoted by $\omega(\mathbf{X}^*)$ and the summed mean gas output be denoted by $\mu(\mathbf{X}^*)$. We have

$$Z^* = \frac{\mathbf{X}^* - \mu(\mathbf{X}^*)}{\sqrt{\omega(\mathbf{X}^*)}} \tag{8.16}$$

$$P(\text{Infeasible}|\mathbf{X}^*) = \Phi\left(Z^*\right) \tag{8.17}$$

where $\Phi(\cdot)$ denotes the standard cumulative normal distribution and Z^* is the standardized Z-score for our solution. An attractive objective function for the total uncertainty is then to minimize $P(\text{Infeasible})$. However, in our case we assume that variance is heteroscedastic. Thus the denominator in the Z-score is a function of $\mathbf{X}$, and the problem is no longer linear. Nonetheless, extracting $P(\text{Infeasible}|\mathbf{X}^*)$ from a solution obtained with $\omega(\mathbf{X}^*)$ as the objective is possible. We permute the objective by adding a small factor multiplied with $\mu(\mathbf{X}^*)$ in order to favor solutions with lower gas output in the case of ties, and denote the probability obtained from this approximation by $\widehat{P}(\text{Infeasible}|\mathbf{X}^*)$. We note that our approximation is a conservative estimate of the optimal probability obtained through non-linear optimization, that is,

$$\widehat{P}(\text{Infeasible}|\mathbf{X}^*) \geq P(\text{Infeasible}|\mathbf{X}^*). \tag{8.18}$$

Since no single optimum exists for a multi-objective optimization program, we generate the Pareto front of non-dominated solutions. Several approaches to generating the front exist in the literature. We choose the ε-constrained method (Changkong and Haimes, 1983) due to its simple implementation and ability to generate non-convex and non-concave parts of the Pareto front. In the ε-constrained approach for a minimization problem, one of the multiple objective functions is chosen to be the goal of the optimization. The remaining functions are constrained to be less than or equal to specified target values. The model is repeatedly solved with varying target values in order to generate the Pareto front. We let the sum of gas output variances $\omega(\mathbf{X})$ remain as the objective to be minimized and constrain the negative total oil output $\psi(\mathbf{X})$. This yields the following single objective

program:

$$\begin{aligned} & \min \quad \omega(\mathbf{X}) && (8.19) \\ \text{s.t.} & \\ & \psi(\mathbf{X}) \quad \leq \quad \varepsilon_j && (8.20) \\ & \mathbf{X} \in A && (8.21) \end{aligned}$$

where A is the feasible domain specified by the multi-objective model in Section 6.4 and ε_j is the target value in iteration j. In order to obtain sensible values of ε_j, we first solve

$$\begin{aligned} & \min \quad \psi(\mathbf{X}) && (8.22) \\ \text{s.t.} & \\ & \mathbf{X} \in A && (8.23) \end{aligned}$$

to obtain the minimum value of $\psi(\mathbf{X})$, denoted by Q^*. We then let

$$\varepsilon_j \quad = \frac{j}{N} Q^*, \quad j = 1, 2, \ldots, N \tag{8.24}$$

where N denotes the number of points from the Pareto front we calculate. The ε-constrained method produces a weak Pareto front (see Section 4.4.4).

Chapter 9

Case Study

In this chapter, we introduce the case that is used as a test case for the models and frameworks developed in this report. Section 9.1 introduces the simplified representation of the case-specific field and its structure. Section 9.2 discusses the data sets which are based on well test measurements from the field. Finally, in Section 9.3 constructed initial cases to be used in the computational study are presented.

9.1 Case Specific Oil Field

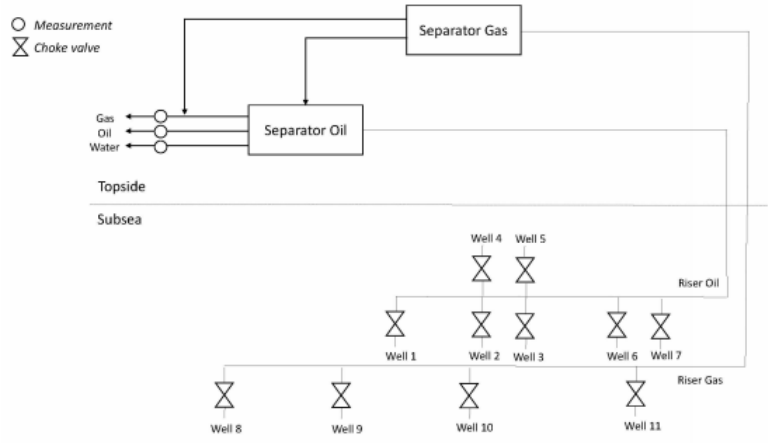

Figure 9.1: Graphic description of the case structure. The case involves wells 1-7, connected to the oil separator. (Morken and Sandberg, 2016)

The case is based on a large oil field on the Norwegian Continental Shelf. It involves seven anonymized wells from the field, connected to one common separator. The wells are

dubbed W1, W2, W3, W4, W5, W6 and W7, and are represented by the wells connected to the oil separator in Figure 9.1. The objective of the case is to maximize oil production, and the only case-specific constraints are explained in the following bullet points:

- Total gas export - the total gas export from the oil separator must not exceed a given value. Under normal conditions this value is 250,000 SOR.
- Gas flow from each well - the amount of gas being routed from each well to the separator must not exceed a given value. Under normal conditions this value is 54,166 SOR.

The decision variable for each well is the choke valve opening level.

9.2 Data Analysis

The data source is comprised of a set of measurements from the wells presented in Section 9.1. These measurements include values for gas output, oil output and certain physical parameters such as well-head pressure. In addition, each measurement includes a time stamp. The decision variable (choke) at the moment when the measurements are made is stored alongside the corresponding production output rates, and together they make up what we refer to as a data point.

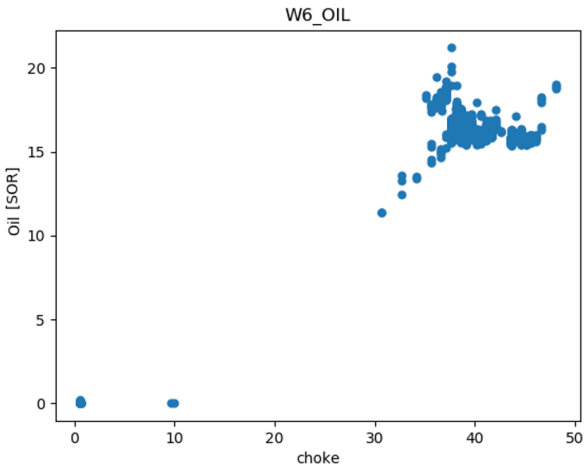

Figure 9.2: The oil output of well W6 plotted with the choke level of the data points.

There are 1,311 measurements for each well, resulting in a total number of data points of 9,177. The data for oil and gas output for different choke settings for well W6 can be seen in Figure 9.2 and 9.3, respectively. The majority of the data points are clustered around the input domain range from about 35 to 45.

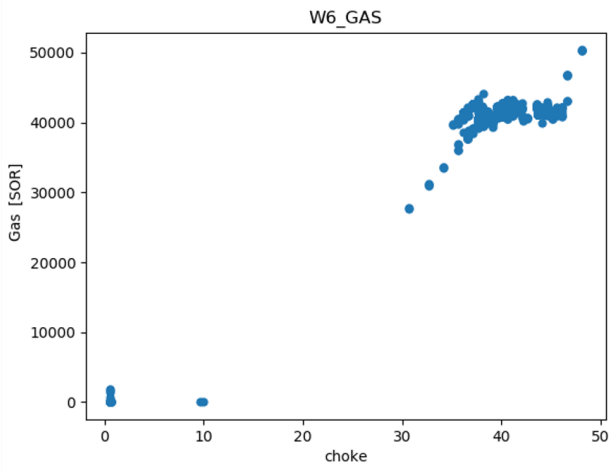

Figure 9.3: The gas output of well W6 plotted with the choke level of the data points.

Several of the wells contain data with significant gaps in the domain range, where most measurements are gathered in a few concentrated clusters. Figure 9.4 shows another trend in the data of multiple production output measurements which are present for similar decision variable values. This Figure is a zoomed in view of the plot for gas output for W1, focused on the area where the data points are the most clustered. For minuscule changes in the choke x-value, there are relative differences of up to almost 100% on the gas output y-axis. Data exhibiting this behavior may be poorly suited for regression, and the discounting algorithm presented in Section A.1 is applied to such wells.

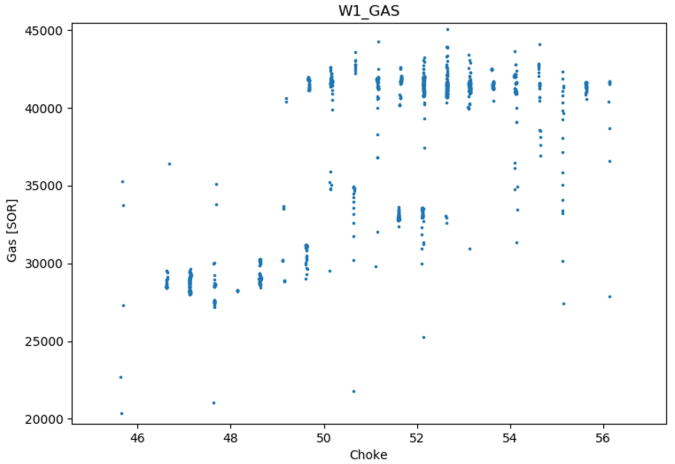

Figure 9.4: Well W1 with differences in gas output of up to 20,000 SOR for similar choke levels.

9.3 Initial Scenarios

In this section, three initial scenarios are introduced. The scenarios are dubbed case A, B and C for simplicity, and are summarized in Table 9.1, Table 9.2 and Table 9.3, respectively.

Initial case A starts in an operational setting that is feasible and where there is slack on the restriction on total gas capacity. This allows for increasing production of oil and gas, and changing the operating point by, e.g., switching wells on or off.

Table 9.1: Well settings and outputs for the unused total gas capacity case.

Initial Case A: Under Total Gas Capacity			
Well	**Choke Setting**	**Oil Output**	**Gas Output**
W1	72.98	28.75	39942.14
W2	47.07	12.60	53520.27
W3	0.00	0.00	0.00
W4	0.00	0.00	0.00
W5	47.84	21.56	50477.35
W6	62.60	24.75	47852.05
W7	50.34	18.54	50547.29
Sum	**-**	**105.66**	**242339.10**

Total Gas Capacity	250000.00
Individual Gas Capacity	54166.00
Maximum Amount of Changes	3

Table 9.2: Well settings and outputs for the breached total gas capacity case.

Initial Case B: Over Gas Capacity			
Well	**Choke Setting**	**Oil Output**	**Gas Output**
W1	72.98	28.75	40023.99
W2	0.00	0.00	0.00
W3	40.00	18.38	42823.11
W4	22.52	8.31	9828.87
W5	53.23	24.40	51929.03
W6	62.61	24.76	51915.69
W7	53.43	19.62	53866.87
Sum	**-**	**124.23**	**250389.60**

Total Gas Capacity	225000.00
Individual Gas Capacity	54166.00
Maximum Amount of Changes	3

Table 9.3: Well settings and outputs for the zero production case.

Initial Case C: Zero Production			
Well	**Choke Setting**	**Oil Output**	**Gas Output**
W1	0.00	0.00	0.00
W2	0.00	0.00	0.00
W3	0.00	0.00	0.00
W4	0.00	0.00	0.00
W5	0.00	0.00	0.00
W6	0.00	0.00	0.00
W6	0.00	0.00	0.00
Sum	**-**	**0.00**	**0.00**

Total Gas Capacity	250000.00
Individual Gas Capacity	54166.00
Maximum Amount of Changes	6

Initial case B starts in an infeasible operational point, where the total gas capacity constraint is breached. This scenario is typically the result of the right hand side of the total gas capacity constraint shifting downwards, e.g. as a result of system failure or maintenance. We also add additional restrictions on the turning off and on of wells. We assume that the production engineers are unwilling to perform drastic changes to the operating point, but instead wish to move to feasibility by performing smaller changes and tweaking the choke levels. Thus, we restrict the optimization model so that no wells that are switched off prior to optimization may be switched on, and no wells that are producing prior to optimization may be switched off.

Initial case C starts with all wells switched off. Consequently, no information about the well gas output curves is known prior to optimization. This allows for the solution methods described in Chapter 8 to unfold in a different way than when in more active starting points.

Chapter 10

Computational Study

In this chapter, we present the results of applying the implemented models to the test case from the Norwegian Continental Shelf. Since the test case is a simplified representation of a real oil field, real data with which solutions can be compared does not exist. The focus of this study is therefore on comparing the different presented approaches to each other and discussing corresponding similarities and differences.

All computations are performed on a 3.40GHz Intel Core i7-6700 CPU with 32 GB RAM. Python version 3.6 is used as the programming language. The Keras (Chollet et al., 2015) library is used for all NN related tasks, and the Gurobi (2014) solver is used for the optimization modeling.

This chapter is organized as follows. In Section 10.1, the results from a technical study investigating solution times and scenario stability are presented. Results from the well model implementations are discussed in Section 10.2, before an optimization study of the initial cases defined in 9.3 is provided in Section 10.3. Finally, in Section 10.4 the results from the implementation of the multi-objective program are presented, and a relation between RA and MOP results is studied.

10.1 Technical Study

In this section, we examine the technical aspects of our well models and our stochastic optimization models.

10.1.1 Problem Size

Solution times are of vital importance in an RTO setting. The model of interest in this study is the stochastic model formulated in Section 6.1. Parameters affecting the solution time of our models include the number of wells in the problem, the number of scenarios we include in our model and, when NN MILP reformulation is used, the NN well model

architectures. In Section 10.2.1, we identify architectures yielding the best performance and apply these to the well model NNs. Thus the architectures of our NN MILPs are not variable in the problem. Our emphasis in this section is on the effect of varying the number of scenarios.

Two procedures for generating scenario well curves $f^G_{ius}(\cdot)$ are presented; The Factor procedure in Section 8.3.2 and the Markov Weighted procedure in Section 8.3.3. A Markov Weighted scenario is generated from a set of samples. In this technical study, we use 11 samples per scenario. The performance of the Markov Weighted models thus depends on the efficiency of the representation of this sample set as a piecewise linear function. The NN MILP formulation consists of a single hidden layer with 20 neurons, while the SOS2 representation places a breakpoint for each sample in a scenario. Finally, the Factor scenario model requires 3 NN MILPs per well (gas and oil mean and gas variance), regardless of the number of scenarios. Each NN in the Factor model consists of a single hidden layer with 40 neurons.

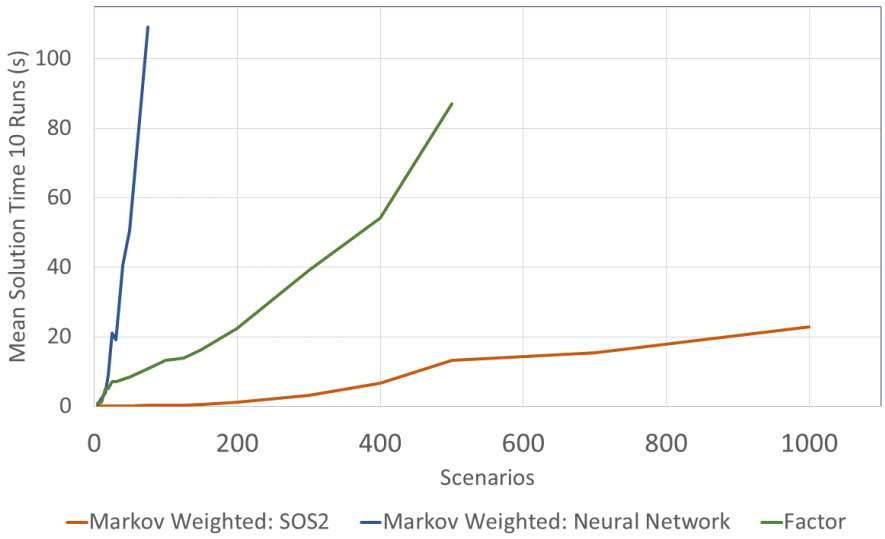

Figure 10.1: Solution times using the Factor and Markov Weighted scenario generation procedures applied to case C.

Results for the Factor model and the Markov Weighted model using both SOS2 and NN MILPs are shown in Figure 10.1 for case C. The results show that the optimization model solution time is lowest when representing well curve scenarios using the SOS2 Markov Weighted type formulation. The Factor scenario model is more computationally expensive to solve, while the Markov Weighted model with NNs reformulated as MILPs clearly is the most computationally taxing. When solving the problem using Markov Weighted NNs for 100 scenarios, the average solution time is 825 seconds (13.75 minutes). This measurement is not included in the figure for readability purposes. Based on expert knowl-

edge from our industry partner, solution times of an RTO model ideally do not exceed ten minutes on the system used in this computational study. From the problem size results, it is clear that the Markov Weighted NN model quickly becomes too computationally expensive to solve within a reasonable time in an RTO setting.

Intuitively, we expect the Factor model to scale well since increasing the number of scenarios in the model does not involve the addition of more NN MILPs. However, the scalability of this model is poor compared to the SOS2 Markov Weighted model. Furthermore, the two models using Markov Weighted scenarios, in essence, are two distinct representations of the same piecewise linear functions. Again we see that the NN MILPs offer poor scalability. These results imply that our formulation of NN MILPs leads to excessively long solution times in the Gurobi solver used in the experiments. Since SOS2 is a well-known modeling technique which many commercial solvers are able to exploit efficiently, alternative NN MILP formulations or improved branching strategies for our formulation may help to close the gap. The case used in this study models inputs as one dimensional. We note that SOS2 representations suffer from the curse of dimensionality when applied to higher dimensional inputs, a result confirmed in Malvik and Witzøe (2017). Subsequently, NN MILP formulations may perform better in comparison to SOS2 in case applications of higher dimensions. A discussion of further research topics based on the technical study is found in Chapter 12.

10.1.2 Stability Testing

To measure the in-sample stability of our scenario generation procedures, we solve case A with Markov Weighted and Factor scenarios. For each number of scenarios, we generate 100 samples and solve the model with each sample. The objective values are then recorded. As a quantitative measure of in-sample stability, we use the standard deviation of the objective value over the 100 samples. We repeat the procedure for both scenario generation methods.

Figure 10.2 shows the standard deviation of the objective value as the number of scenarios in the stochastic optimization model increases for Markov Weighted and Factor scenarios. The results are similar for the two scenario generation procedures, with the Factor scenarios leading to a slightly lower standard deviation. One possible explanation for this is that a Factor scenario with a large positive factor leads to a high GOR across the entire input domain for the given well. In a robust stochastic formulation, this causes the Factor scenarios to produce more conservative solutions which are slightly more stable, thus leading to the lower standard deviation. In Figure 10.3 we plot the mean objective value obtained with the two scenario generation procedures, again for case A. This confirms that the Factor model is more conservative in its solutions.

The standard deviation obtained for 500 Markov Weighted and Factor scenarios comprises 0.45% and 0.57%, respectively, of the average objective value for the same number of scenarios. When increasing the number of scenarios above 400, we observe diminishing returns in the reduction of the standard deviation of the Markov Weighted scenarios. We

conclude that both scenario generation procedures offer sufficient stability for 400 scenarios or more.

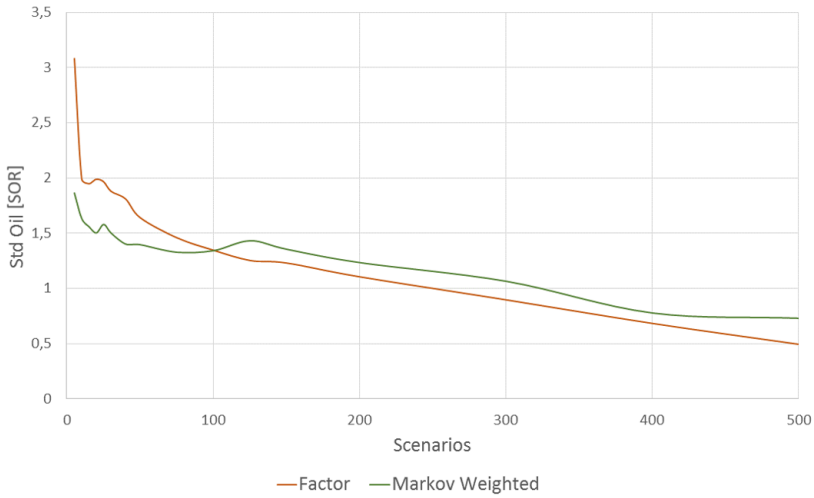

Figure 10.2: The standard deviation of the objective value as a function of the number of scenarios in the optimization model for the Factor and Markov Weighted scenario generation procedures.

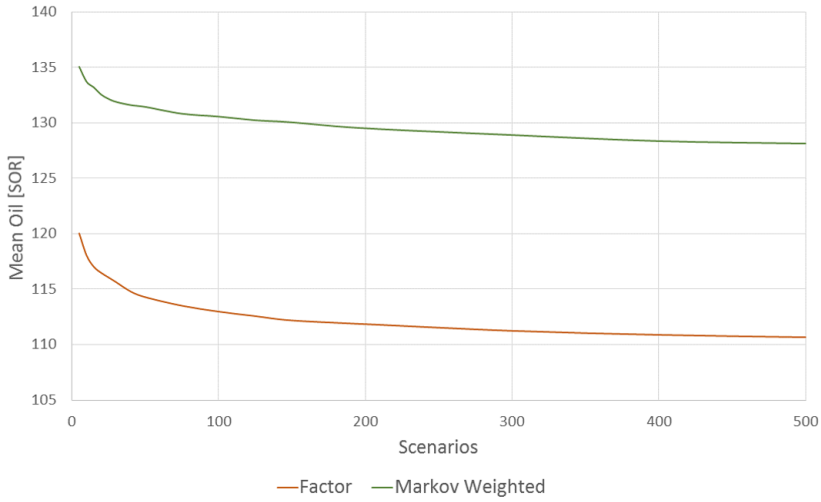

Figure 10.3: Mean objective value obtained from stability tests for case A, using Factor and Markov Weighted scenario generation procedures.

In Section 4.4.2, it is shown that measuring out-of-sample stability requires the comparison of the objective values obtained using different scenario trees. However, in the relevant problem of this thesis, it is assumed that all uncertainty is located in the gas output of wells. Since gas outputs are present in constraints in the optimization problem only, we always have perfect out-of-sample stability in the theoretical sense. That is, a solution, represented by a set of choke values, always produces the same oil output regardless of which scenario tree it is evaluated over since oil outputs are deterministic. The standard approach to measuring out-of-sample stability makes little sense in our application. Nonetheless, we are interested in measuring if our scenario generation procedures capture the relevant aspects of the underlying distribution and if the scenarios produce stable results in an "out-of-sample" setting. An approximation to out-of-sample stability by comparison of the MOP and a model based on Markov Weighted scenarios is discussed in Section 10.4.4.

10.2 Well Model Results

We now present the results from implementing well models using the methodology described in Chapter 7.

10.2.1 Neural Network Hyperparameters

Grid and randomized search are used to identify the NN hyperparameters that yield the best performance when approximating the well curves. Performance is measured by mean test scores using 3-fold cross-validation, as presented in Section 4.1.4. Since we do not, in general, know how the well performance curves compare across wells, we search for parameters for each well for gas and oil separately. The hyperparameters yielding the best test scores for each well are found in Table 10.1 and Table 10.2 for the gas and oil output, respectively. Higher test scores indicate a better result. The test scores take negative values due to the NN loss function, presented in Section 4.3.3. The table includes results for the regularization term λ, the dropout rate p, the number of hidden layers L and the number of neurons in each hidden layer N. A comprehensive list of the top 3 test score results for each well, phase and search method is found in Appendix A.

In Table 10.1 we see that there are differences in the identified hyperparameters for approximating the gas output curves. The largest differences are found in the values for regularization term λ in the grid search results, where the highest value 1.00×10^{-4}, found for W3, is two orders of magnitude larger than the lowest, 1.00×10^{-6}, found for all wells except W3 and W7. Thus the grid search indicates that the W3 and W7 NNs are more prone to overfitting the data. However, for the randomized search, all λ terms are of the same order of magnitude. Here, the largest values for the regularization term are found for wells W5 and W6. Based on these results it is difficult to draw any conclusions regarding which well NN is more likely to overfit its data.

The dropout rate p generally yields the best performance for values in the range of $0.033 - 0.079$. The complexity of a NN increases with the number of hidden layers L and neurons in these layers N. Both grid and randomized search find that complex architectures

Table 10.1: Hyperparameter search results for gas output.

			Hyperparameters			
Well	**Search Type**	**Avg. Test Score**	λ	p	L	N
W1	Grid	-4.66	1.00×10^{-6}	0.050	2	40
	Randomized	-4.71	1.49×10^{-5}	0.033	2	39
W2	Grid	-5.57	1.00×10^{-6}	0.050	2	40
	Randomized	-5.79	6.34×10^{-5}	0.013	2	30
W3	Grid	-6.58	1.00×10^{-4}	0.050	2	40
	Randomized	-6.75	2.98×10^{-5}	0.063	2	28
W4	Grid	-7.16	1.00×10^{-6}	0.050	2	40
	Randomized	-7.27	4.26×10^{-5}	0.035	2	33
W5	Grid	-7.24	1.00×10^{-6}	0.050	2	40
	Randomized	-7.11	7.18×10^{-5}	0.037	2	36
W6	Grid	-6.76	1.00×10^{-6}	0.050	2	40
	Randomized	-6.82	7.93×10^{-5}	0.012	2	29
W7	Grid	-6.65	1.00×10^{-5}	0.050	2	40
	Randomized	-7.45	6.13×10^{-5}	0.079	2	35

Table 10.2: Hyperparameter search results for oil output.

			Hyperparameters			
Well	**Search Type**	**Avg. Test Score**	λ	p	L	N
W1	Grid	-6.35	1.00×10^{-2}	0.050	2	20
	Randomized	-6.22	9.33×10^{-6}	0.014	2	37
W2	Grid	-4.67	1.00×10^{-4}	0.050	2	40
	Randomized	-4.69	6.93×10^{-5}	0.026	2	31
W3	Grid	-10.94	1.00×10^{-5}	0.150	2	20
	Randomized	-8.72	1.48×10^{-5}	0.260	2	7
W4	Grid	-5.88	1.00×10^{-2}	0.050	2	20
	Randomized	-5.96	3.73×10^{-5}	0.067	2	10
W5	Grid	-7.71	1.00×10^{-2}	0.050	2	20
	Randomized	-7.08	9.5×10^{-5}	0.074	2	5
W6	Grid	-7.78	1.00×10^{-2}	0.050	2	20
	Randomized	-7.36	1.86×10^{-5}	0.012	2	6
W7	Grid	-8.71	1.00×10^{-3}	0.050	2	20
	Randomized	-7.64	9.17×10^{-5}	0.110	2	9

Table 10.3: Top results from hyperparameter search over complex architectures for gas output. The first two rows show results for increased depth and the last two rows show results for increased width.

				Hyperparameters			
	Well	**Search Type**	**Avg. Test Score**	λ	p	L	N
Expanded Depth	W1	Grid	-4.76	1.00×10^{-6}	0.050	4	20
		Randomized	-4.58	3.88×10^{-6}	0.034	3	11
	W2	Grid	-5.53	1.00×10^{-6}	0.050	3	40
		Randomized	-5.50	4.89×10^{-6}	0.031	2	23
Expanded Width	W1	Grid	-4.63	1.00×10^{-6}	0.050	2	70
		Randomized	-4.86	3.39×10^{-6}	0.047	1	79
	W2	Grid	-5.57	1.00×10^{-6}	0.050	2	60
		Randomized	-6.51	3.05×10^{-6}	0.032	1	79

produce the best performance. Consequently, these results suggest better test scores may be achieved if we expand the search space with respect to architecture parameters. We conduct an additional search over strictly more complex architectures to see if significant improvements to the test score can be obtained for two selected wells, W1 and W2. We search over an increase in width, i.e., $1-2$ hidden layers with $40-80$ neurons each, and an increase in depth, i.e., $3-4$ hidden layers with $10-40$ neurons each. The top expanded grid and randomized search results are presented in Table 10.3. Comparing the depth and width test scores to Table 10.1, we see that the best improvement is less than 5%, and the test score is worse in half of the cases. If we reformulate the NN as a MILP, adding hidden layers and neurons leads to an increase in the number of integer variables and constraints. Thus, we conclude that for our application, 2 hidden layers offers sufficient complexity for our well models for the gas output.

The oil output results in Table 10.2 exhibit a larger spread than for the gas output. Notably, for some wells, we observe stricter regularization in the best performing results. Since dropout has the effect of regularization, we expect that setting a large λ term and a low rate p produces somewhat comparable results to a small λ and a high p. However, from the results, we see that high dropout rates and large regularization terms yield similar test scores to low dropout rates and small regularization terms. From these results, it seems NN performance is more dependent on architecture than regularization intensity in the oil output case. In such cases, we may be more interested in searching over parameters which affect the test scores to a greater extent, e.g., by setting regularization intensity using heuristic rules and instead including the learning rate in the search space. However, it is also possible that the parameter space we include in the search space is too limited to capture the dropout rates and regularization terms that yield the actual best performance. With respect to the bias-variance trade-off discussed in Section 4.1.1, we may be in a situation where the search space only covers a small region of the trade-off curve, yielding similar test scores in both extreme ends.

The use of heuristic rules to determine certain NN hyperparameters is a strategy to avoid the combinatorial explosion of the grid and randomized searches in high dimensions. It is

impossible to verify optimality of the determined hyperparameters, and in light of the discussion above, alternative search spaces are interesting to examine. This topic is discussed in Chapter 12.

Once the best performing hyperparameters are identified for each well and phase, we are able to train the NN well models.

10.2.2 Neural Network Well Models

The result of a fully trained well model as described in Chapter 7 is plotted in figures 10.4 and 10.5 for well W7 for the oil and gas outputs, respectively. The data points shown in the figures are discounted using discount Algorithm A.1 with a grid of size 100×1 along the choke and output axes, and $\alpha = 1.5$. This means that the plot includes at most 100 data points. In this case, the amount of data points is reduced from $1{,}308$ to 33. The time spent training a network on the original data points is 502.69 seconds. After discounting, training time is reduced to 37.82 seconds. Thus, as a side effect of applying the discounting algorithm, the NN training is considerably quicker. Training times for different interval sizes in the discounting algorithm for well W7 are found in Appendix A. The optimal fidelity of the discount algorithm is case dependent since data characteristics determine how aggressively we wish to phase out old measurements. Determining this fidelity thus relies on intimate knowledge of the relevant case. For the case in this thesis, we use a 100×1 discount grid for all wells.

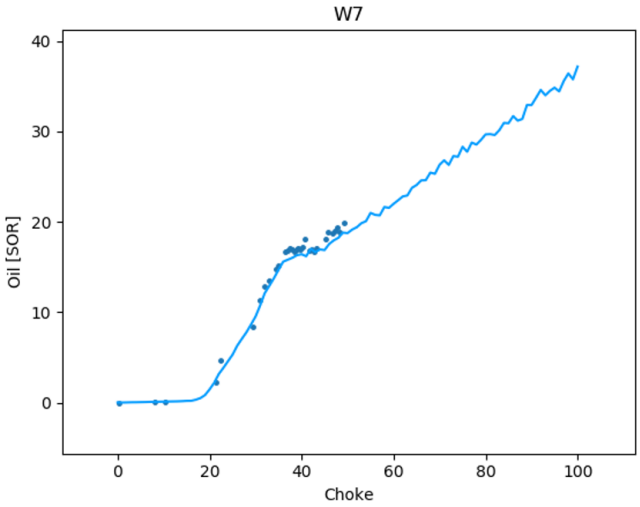

Figure 10.4: NN Well model for the oil output W7. Data points are discounted by discount algorithm A.1 and represented as dark blue dots in the figure. The solid blue line represents the estimated mean.

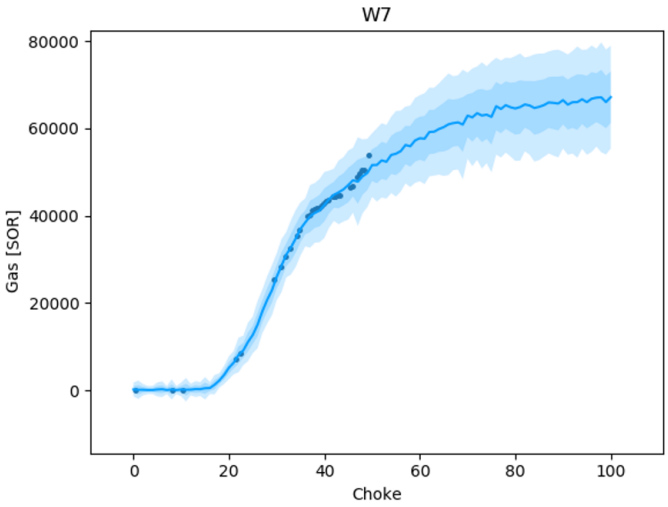

Figure 10.5: NN Well model for the gas output W7. Data points are discounted by Algorithm A.1 and represented as dark blue dots in the figure. The solid blue line represents the estimated mean, while each blue shade corresponds to a standard deviation.

Reformulating our well model NNs as MILPs in the optimization model leads to a slight change in how the NN estimates its mean output. Prior to reformulation, MC dropout is used to approximate the mean over several forward passes as in Equation 4.24. However, applying dropout during forward passes is complicated in the MILP reformulation of the NN. Thus, we instead perform one forward pass through the piecewise linear NN MILP without dropout, scaling outputs as described in Section 4.2.6, to calculate the mean output. If there is a large difference between the estimated mean using MC dropout and the estimated mean of the single forward pass, we lose accuracy in the MILP reformulation. With this in mind, we now examine the difference between the two approaches to calculating the mean output.

In the following, we denote the estimated mean for choke level $x \in X$ by $\mu_{\text{MC}}(x)$ when using MC dropout, and $\mu_{\text{off}}(x)$ using a single forward pass without dropout. Figure 10.6 shows the deviation between $\mu_{\text{MC}}(x)$ and $\mu_{\text{off}}(x)$, for NNs trained with dropout rates $p_1 = 0.05$, $p_2 = 0.15$ and $p_3 = 0.26$ for well W3. Here, the dark blue dots represent the well measurement data points. Calculations for the mean absolute deviation are shown in the top row in Table 10.4. We include p_3 since it is the highest dropout rate identified by the hyperparameter search for oil output for W3 in Section 10.2.1. For all three rates, the mean absolute deviation between the two curves over the entire choke domain is less than 0.3 SOR. Our findings confirm results by Srivastava et al. (2014), who empirically observe small deviations for simple NN architectures.

Two regions of interest are identified in the plots. The highest observed choke value in the measurement data for W3 is $x_{\text{max}} = 50.50$, and the first region of interest consists of choke values $x \in \{\overline{X} \subset X \,|\, x > x_{\text{max}}\}$. For all three dropout rates, $\mu_{\text{MC}}(x)$ exhibits

increased volatility in this region. This is a result of different realizations of the random dropout masks leading to more variation in the NN output for regions in which no data is observed. Consequently, the deviation between the two calculations for the mean is larger in this region, as is seen in the bottom row of Table 10.4. Nonetheless, the mean absolute deviation is still small compared to the total output, and we deem this gap to be acceptable for choke values $x \in \overline{X}$.

The second region of interest is located around the lowest choke setting for which the estimated mean oil output takes a positive value. This setting is different for $\mu_{\text{MC}}(x)$ and $\mu_{\text{off}}(x)$, with the difference increasing with higher dropout rates. Consequently, a higher choke setting is required before positive production is observed when using $\mu_{\text{off}}(x)$. Since the gap quickly closes as the choke level increases, this deviation occurs solely when wells are marginally producing. We conclude that the deviation in this area is acceptable in our application since it is present only in a very contained region.

In summary, we find that the deviation of the estimated mean that occurs when we no longer apply dropout in a NN reformulated as a MILP is insignificant in the relevant case of this thesis.

Table 10.4: Mean absolute deviation of MC dropout mean and NN prediction without dropout. The top row shows the calculation over the entire input domain, while the bottom row shows the calculations for the region of the input domain in which choke values exceed the highest observed measurement data point.

Dropout rate	**0.05**	**0.15**	**0.26**
Avg. deviation, $x \in X$	0.26 SOR	0.14 SOR	0.19 SOR
Avg. deviation, $x \in \overline{X}$	0.33 SOR	0.18 SOR	0.21 SOR

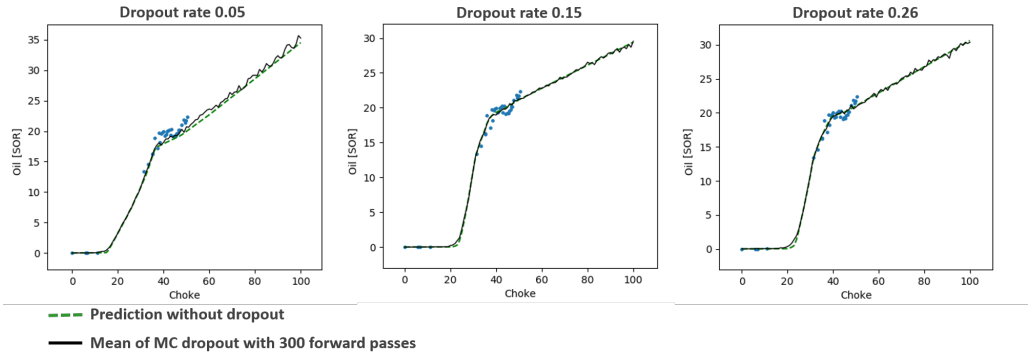

Figure 10.6: Mean of MC dropout compared to NN prediction without dropout for well W3.

10.3 Robust Optimization Study

In this section, we present results from the stochastic optimization model applied to the initial cases defined in Section 9.3. We use the penalty approaches introduced in Section 8.4 to compare solutions for different model settings. First, we examine results from directly and simultaneously applying all suggested changes in the optimal solutions. Results from applying operational changes sequentially in the Recourse Algorithm are then presented before we discuss and compare the approaches.

10.3.1 Stochastic Optimization Results

We now present results from evaluating the solutions of our robust stochastic model applied to initial cases A, B and C. For each initial case and number of scenarios, the model solution is evaluated in 200 different scenarios that we treat as distinct ground truths. These true scenarios are kept out of the optimization model. We note that in this section, all operational changes suggested by the model are assumed to be implemented simultaneously when we evaluate solutions.

In the following, we show an excerpt of results aiming to give the reader insight into how solutions evolve as we increase the number of scenarios in the problem. Full results for a wider range of scenario sizes are found for all cases and models in Appendix B. Tables 10.5, 10.6 and 10.7 show results for cases A, B and C, respectively. Each table includes results for both the Markov Weighted scenario models and the Factor scenario models. The # *Infeasible* column denotes the number of ground truths (out of 200) in which the model solution is infeasible. The average objective value under each penalty approach is shown. Note that the WS solution is equal for both scenario model types in all cases.

Table 10.5: Initial case A: The EEV, WS and stochastic optimization model objective values using Markov Weighted scenarios and Factor scenarios.

	Markov Weighted Scenarios			Factor Scenarios		
Model/Scenarios	**# Infeasible**	**Strict Penalty**	**Reversion Penalty**	**# Infeasible**	**Strict Penalty**	**Reversion Penalty**
EEV	56	91.82	123.71	37	101.15	122.22
WS	0	134.18	134.18	0	134.18	134.18
5	64	92.41	126.22	0	116.16	116.16
15	22	118.23	129.86	0	116.16	116.16
30	11	125.44	131.25	0	115.88	115.88
50	9	125.74	130.50	0	115.78	115.78
400	0	127.86	127.86	0	108.57	108.57

Some observations are made from the results for the three initial cases. First, we see that the Factor model behaves strangely for case B, producing identical solutions that are infeasible in all ground truth scenarios for any scenario number. In Section 8.3.2 it is shown that providing a known point, i.e., a choke setting with corresponding gas output, leads to a single possible Factor representation for the relevant well. Thus the Factor model reduces to a deterministic model in case B except for well W2, since this is the only well that is switched off prior to optimization. However, since the specification of case B disallows turning on wells that are not initially producing, uncertainty is eliminated from the

Table 10.6: Initial case B: The EEV, WS and stochastic optimization model objective values using Markov Weighted scenarios and Factor scenarios.

	Markov Weighted Scenarios			Factor Scenarios		
Model/Scenarios	**# Infeasible**	**Strict Penalty**	**Reversion Penalty**	**# Infeasible**	**Strict Penalty**	**Reversion Penalty**
EEV	190	6.00	98.21	200	0	100.93
WS	0	120.37	120.37	0	120.37	120.37
5	113	48.18	99.18	200	0	100.93
15	62	72.57	101.52	200	0	100.93
30	14	94.97	100.58	200	0	100.93
50	4	98.73	100.50	200	0	100.93
400	1	97.81	98.61	200	0	100.93

Table 10.7: Initial case C: The EEV, WS and stochastic optimization model objective values using Markov Weighted scenarios and Factor scenarios.

	Markov Weighted Scenarios			Factor Scenarios		
Model/Scenarios	**# Infeasible**	**Strict Penalty**	**Reversion Penalty**	**# Infeasible**	**Strict Penalty**	**Reversion Penalty**
EEV	166	22.55	99.43	169	19.52	93.47
WS	0	131.86	131.86	0	131.86	131.86
5	122	48.28	108.27	99	61.82	111.24
15	63	83.08	114.22	2	110.90	111.85
30	25	102.93	115.35	2	110.90	111.85
50	14	107.86	114.70	1	106.65	107.08
400	7	108.31	111.64	0	100.24	100.24

problem. The resulting solution is therefore always identical. This solution is infeasible because the deterministic Factor curves fail to represent the ground truth scenarios accurately. Case B highlights the weakness of the Factor scenario generation procedure when known points are specified. We conclude that the Factor model is not a suitable model for case B.

For cases A and C, the Factor model produces solutions that are feasible in all 200 ground truth scenarios even with a modest number of scenarios. However, the obtained objective value is generally lower for the Factor model than for the Markov Weighted once the number of scenarios is sufficiently high. With 400 scenarios, the Markov Weighted model obtains objective values that are 17.8% and 11.4% higher than the Factor model for cases A and C respectively, when applying the switch-off penalty. This suggests that optimal solutions of the Factor model are conservative, avoiding infeasibility in ground truth scenarios at the cost of optimal objective values. We note that this confirms the result in Section 10.1.1. For case A under the Reversion penalty, the Factor model obtains a higher objective value for the EEV than for the robust stochastic model using any number of scenarios, yielding a negative VSS. This suggests that when subject to certain penalties, the conservatism of the Factor model detracts value from the decision maker rather than adding it.

Generally, case C leads to a higher number of infeasible settings than cases A and B for nearly all scenario sizes and scenario generation procedures (except for the unsuited

Factor model for case B). In case C all wells are shut off prior to optimization, and we are uncertain about all regions of the well curves. For cases A and B, the generated scenarios intersect known points for initially producing wells, and the well models are less uncertain in close proximity to such known points. Thus we expect case C to be generally more uncertain, which is confirmed by the higher number of infeasible settings. We also see that there is a large difference in mean objective value for the Strict and Reversion/Switch-Off penalty for models with less than 100 scenarios, except for the Factor model for case A. This difference is a function of the number of infeasible solutions, since these do not contribute to the average objective value in the Strict penalty approach. As the number of infeasible solutions decreases with a higher number of scenarios, the difference between the Strict and Switch-Off/Reversion penalties diminishes.

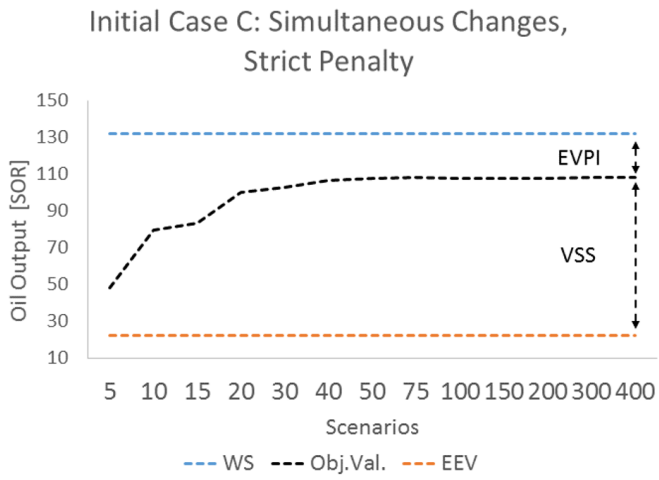

Figure 10.7: VSS and EVPI for a number of scenario sizes for the Strict penalty for initial case C.

The Markov Weighted VSS and EVPI for initial case C under the Strict and Switch-Off penalty are plotted for various numbers of scenarios in Figures 10.7 and 10.8, respectively. Identical plots for cases A and B are found in Appendix B. Clearly, different penalty approaches lead to different values for the VSS. While the VSS for 400 scenarios with the Strict penalty is 80.72, the Switch-Off penalty with the same number of scenarios yields a VSS of 6.77. The large difference in VSS highlights the importance of carefully selecting a suitable penalty to infeasible solutions in a robust optimization model. If the penalty does not reflect the true utility of the decision maker, we end up with a poor understanding of model performance.

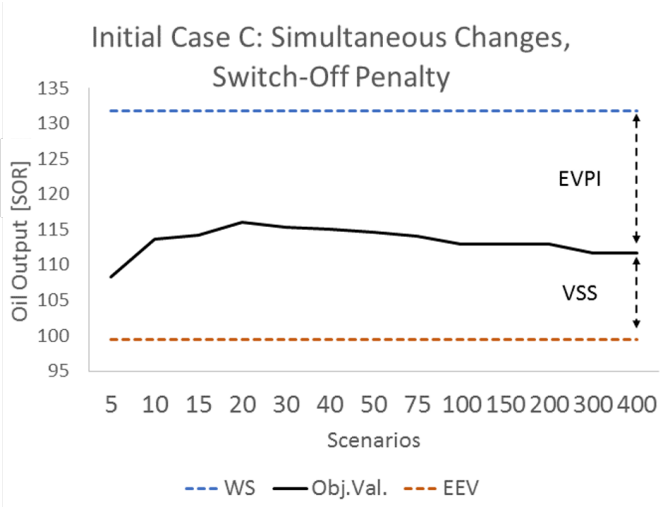

Figure 10.8: VSS and EVPI for a number of scenario sizes for the Switch-Off penalty for initial case C.

10.3.2 Recourse Algorithm Results

We first examine the objective values attained by the RA and the behavior of the VSS and EVPI. Then, we look at an example of an iteration performed by the algorithm to study its behavior. In light of the weaknesses of the Factor scenario model identified in the previous section, throughout this section we present results for the RA using Markov Weighted scenarios only.

Markov Weighted Scenarios for Initial Cases

As in the previous section, for each initial case and number of scenarios, the model solution is evaluated in 200 different scenarios that we treat as distinct ground truths. We use the same 200 ground truths as for the standard stochastic robust model, in the interest of comparing solutions in Section 10.3.3. These true scenarios are kept out of the optimization model, but are gradually learned by the RA as changes are implemented.

Table 10.8 shows results from running the RA for a selected number of scenario sizes for all three initial cases. As in the previous section, complete tables with results for a wider range of scenario sizes are found in Appendix B. The # *Infeasible* column for cases A and C denotes the number of ground truths in which the RA is forced to revert one or more changes due to visiting an infeasible state. Note that as discussed in Section 8.5, in case B an implemented change is not reverted even though the resulting solution is infeasible if the model is still moving toward feasibility. The *# End States Infeasible* column for case B therefore represents the number of ground truth scenarios in which the RA failed to reach feasibility after expending all its permitted changes. Thus, while for cases A and C infeasible moves are reverted as the algorithm iterates, it is appropriate to penalize the

RA in case B whenever it fails to reach feasibility.

Results from applying the Strict and Switch-Off penalties to the RA in case B are shown in Table 10.9. The difference between the penalty-free RA average objective value and the objective value under the Switch-Off penalty is less than 3.5% for all numbers of scenarios, and is negligible for models with 20 or more scenarios since the number of infeasible states is 2 or less for these models. Considering the model with 15 scenarios, the RA fails to reach a feasible end state in more than 10% of the ground truths. However, the difference between the penalty-free objective value and the Switch-Off penalty is only 0.47%, lower than we expect with a relatively high infeasibility rate. This indicates that in the cases where the RA ends in an infeasible state, it breaches the gas export capacity constraint by a marginal amount. However, this result is not obvious from considering the *# End States Infeasible* column in Table 10.9. We conclude that accounting not only for the number of times the end state is infeasible, but the amount by which the gas capacity constraints are breached may provide clearer insight in cases where we start in an initially infeasible state and move towards feasibility.

For case B, the number of infeasible states visited is not strictly decreasing with an increasing number of scenarios. Whether or not the RA reaches feasibility in case B typically depends on how conservative its final suggested operational change is. As we add scenarios, the solution path of the RA changes, i.e., which wells it changes and the order of these wells. As a consequence, the final well to change also varies with the number of scenarios, and the number of infeasible states may increase as scenarios are added to the model. Nonetheless, we see a clear trend in the results, with the RA stabilizing at 1 or 2 failures to reach feasibility for problems with 30 scenarios or more.

Table 10.8: Recourse algorithm results for Markov Weighted scenarios.

	Initial Case A: Under Cap.		**Initial Case B: Over Cap.**		**Initial Case C: Zero**	
Model/ Scenarios	**# Infeasible Moves**	**Avg. Objective Value**	**# End States Infeasible**	**Avg. Objective Value**	**# Infeasible Moves**	**Avg. Objective Value**
EEV	56	125.18	78	104.77	183	109.20
WS	0	134.18	0	120.37	0	131.86
5	60	126.96	15	102.94	119	112.79
15	12	130.47	22	106.10	46	117.64
30	11	131.05	0	104.25	15	116.69
50	8	129.83	2	102.61	13	115.88
400	2	128.20	1	104.29	4	112.49

Figure 10.9 gives a visual representation of the development of the EVPI and VSS of the RA result for case C as the number of scenarios increases. Identical figures are found for cases A and B in Appendix B. The RA objective value is volatile for models with few scenarios since the number of infeasible states visited at first decreases quickly as we increase the number of scenarios above 5. As the number of scenarios increases, the objective value stabilizes at 112.49 SOR, yielding a VSS of 3.29.

In tables 10.8 and 10.9, the maximum of the VSS (corresponding to the maximum average objective value) is not found when solving the model with the maximum number of scenar-

Table 10.9: Penalized recourse algorithm results for Markov Weighted scenarios, case B.

Initial Case B Penalized				
Model/ Scenarios	**# End States Infeasible**	**Avg. Objective Value**	**Strict Penalty**	**Switch-Off Penalty**
EEV	78	104.77	64.9	101.14
WS	0	120.37	N/A	N/A
5	15	102.94	95.15	101.47
15	22	106.10	94.47	105.60
30	0	104.25	104.25	104.25
50	2	102.61	101.60	102.49
400	1	104.29	103.52	104.27

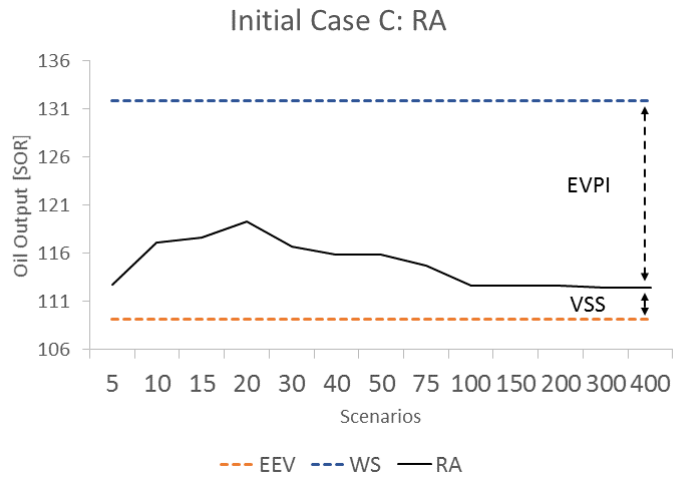

Figure 10.9: VSS and EVPI for a number of scenario sizes for initial case C.

ios, 400, for any of the initial cases A, B or C. For the number of scenarios that maximize the objective value and thus the VSS, the model performs changes that visit infeasible states in 11, 29 and 27 ground truth scenarios for cases A, B (Switch-Off penalty) and C, respectively. Consequently, the maximum objective value comes at the risk of stressing the production system with numerous infeasible states. This result is expected in a robust formulation, where including few scenarios in the model leads to an incomplete representation of the uncertainty in the problem and weaker robustness against feasibility.

Similar to the results in Section 10.3.1, we find that the RA produces a higher number of infeasible states for case C than what is found for cases A and B. Again, this is due to the model being uncertain about the entirety of the gas curve for all wells prior to optimizing, since no initially known points are specified in case C.

We defer a more thorough discussion of the RA results to Section 10.3.3, where obtained objective values are compared to the results from implementing changes simultaneously.

Single RA Run

We now examine a single run of the RA for case B with 25 scenarios in the optimization model. For convenience, we remind the reader that the total gas export capacity in this case is $225{,}000$ SOR, the individual well gas production capacity is $54{,}166$ SOR and the number of allowed operational changes is 3. In the initial state prior to optimization, the total gas output is $250{,}387$ SOR.

In Figure 10.10, each optimization step in the RA is represented by a box, while triangles indicate the flow of the algorithm. Each box contains information on the initial choke levels prior to the optimization, the changes suggested by the optimization, and the choke levels after implementing a single change. The well for which the RA chooses to implement a change is selected according to the priority lists shown at the top of the figure, and is in each step marked with a thick border in the *Change* column under *Model Solution*. The priority list for increasing the choke settings is based on sorting wells from lowest average GOR to highest, while the list for choke decrease is the reverse of the aforementioned. The intermediate results after implementing the selected change are shown in the top half of each box and include which well is changed, whether or not the solution is feasible and any constraint breaches, the new oil output, the expected gas output before the change is implemented and the realized gas output after the implementation. The choke values after the implementation, dubbed *New State Chokes*, are then carried on to the next optimization step as *Initial State Chokes*. Choke values are marked with red in infeasible solutions and green in feasible solutions. The example run is never close to breaching individual gas capacity constraints, and in the interest of avoiding clutter in Figure 10.10, individual well gas outputs are not shown.

We now consider the details of the RA run in Figure 10.10.

Step 1. In the first optimization step, the model is solved with 3 allowed changes. A negative change is suggested for wells W7 and W5, while a positive change is suggested for well W3. The RA prioritizes negative changes first, and since W7 ranks second and W5 ranks sixth in the choke decrease priority list, W7 is selected as the well to change. After implementing the change, the realized gas output is $5{,}447$ SOR lower than expected. At this point, the state is still infeasible by approximately $6{,}000$ SOR.

Step 2. As a result of the lower than expected gas output, the second optimization solution recommends decreasing the choke level for well W6 instead of well W5. The suggested increase for well W3 remains the same as in the suggested solution in the first step, i.e., an increase of 18.00. Again, the RA prioritizes the negative change first, selecting W6 as the well to change. The planned increase in W3 is justified by the model expecting a feasible state upon decreasing the choke for W6, which is observed by an expected gas output in step 2 of $223{,}488$ SOR, about $1{,}600$ SOR lower than the total gas capacity. However, after performing the choke decrease, well W6 produces almost $7{,}000$ SOR more gas than expected. The realized gas output yields a new state that is still infeasible by about $5{,}000$ SOR.

Step 3. Faced with a higher than expected gas output that yields an infeasible state, the RA suggests as its final change a decrease in the choke level for well W3 by 9.22 instead of the positive change it originally suggested in steps 1 and 2. This reduction yields only slightly more gas output than expected in the final state, and the result is a solution that is feasible by about $1{,}800$ SOR.

The final oil output rate the RA achieves in this example run is 97.67 SOR. If all 3 changes from the solution suggested in step 1 are implemented simultaneously, the resulting state is feasible, with an oil output rate of 106.51 SOR and a total gas output rate of $216{,}314$ SOR. In other words, implementing 3 changes simultaneously in this case leads to both lower gas output and higher oil output. When the RA learns the true production output of well W7 in step 1, the uncertainty regarding all other wells remains unchanged. With its limited amount of 25 scenarios, the model has a poor representation of the uncertainty in well W6, causing it to underestimate the rate of gas this well produces. Thus, even though the decision to change W6 instead of W5 is more informed than the decision to implement all changes simultaneously, the final result is significantly worse. Clearly, discovering information about the true performance of wells is potentially of little value if the underlying uncertainty is not properly represented in the model. In Section 10.3.3, we compare results for the RA with simultaneous changes for a wider range of scenario sizes.

We note that in a setting with perfect information, the implemented change in each step corresponds to a suggested change from the solution of the first step optimization since the realized gas output always equals the expected gas output. Thus, a WS model always follows its first solution and the RA provides no added value in such cases.

Well Priority List Increase Choke							
Priority	1	2	3	4	5	6	7
Well	W1	W5	W4	W3	W6	W7	W2

Well Priority List Decrease Choke							
Priority	1	2	3	4	5	6	7
Well	W2	W7	W6	W3	W4	W5	W1

Changes Available: 3

Action: change W7 [53.43 ---> 30.17]

Result infeasible - total gas constraint breached

Oil output	113.39
Expected gas output	236 694.00
Realized gas output	231 247.11

Well	Initial State Chokes	Model Solution Suggested Chokes	Model Solution Change	New State Chokes
W1	72.98	72.98	-	72.98
W2	0.00	0.00	-	0.00
W3	40.00	58.00	18.00	40.00
W4	22.52	22.52	-	22.52
W5	47.84	29.27	-18.57	53.23
W6	62.60	62.61	-	62.61
W7	53.43	30.17	-13.26	30.17

Changes Available: 2

Action: change W6 [62.61 ---> 38.82]

Result infeasible - total gas constraint breached

Oil output	103.98
Expected gas output	223 487.85
Realized gas output	230 391.54

Well	Initial State Chokes	Model Solution Suggested Chokes	Model Solution Change	New State Chokes
W1	72.98	72.98	-	72.98
W2	0.00	0.00	-	0.00
W3	40.00	58.00	18.00	40.00
W4	22.52	22.52	-	22.52
W5	53.23	53.23	-	53.23
W6	62.61	38.82	-23.79	38.82
W7	30.17	30.17	-	30.17

Changes Available: 1

Action: change W3 [40.0 ---> 30.78]

Result feasible

Oil output	97.67
Expected gas output	222 984.29
Realized gas output	223 209.92

Well	Initial State Chokes	Model Solution Suggested Chokes	Model Solution Change	New State Chokes
W1	72.98	72.98	-	72.98
W2	0.00	0.00	-	0.00
W3	40.00	30.78	-9.22	30.78
W4	22.52	22.52	-	22.52
W5	53.23	53.23	-	53.23
W6	38.82	38.82	-	38.82
W7	30.17	30.17	-	30.17

Figure 10.10: A single run of the RA for case B. The optimization model includes 25 scenarios. Well priority lists decide which of the suggested well chokes are changed. The solutions of the optimization model change as the RA learns more about the true production of the wells. Infeasible solutions are colored in red, while feasible solutions are colored in green.

10.3.3 Stochastic Optimization Results Comparison

We now compare results for the simultaneous change penalty approaches with the RA. Since the EEV is different for these approaches, we cannot directly compare the VSS. However, since the same 200 ground truths are used to evaluate all models in this study, we are able to compare the obtained objective value of the approaches directly. We note that this is equivalent to comparing the EVPI since the WS solution is equal for all three approaches, i.e., the Strict and Reversion/Switch-Off penalties and the RA. In Section 10.3.2, RA solutions for case B are penalized whenever infeasible in the ground truth scenario. For convenience, in the following discussion with respect to RA results for case B, we consider the results obtained under the Switch-Off penalties. Since the difference in the RA objective value under the Strict and Switch-Off penalties for case B is found to be 1% or less for models with 30 or more scenarios in the previous section, this distinction has an insignificant impact on the relevant discussion.

In the following, all results for all model types pertain to the models using Markov Weighted scenarios since important weaknesses with the Factor scenarios are identified in Section 10.3.1.

Figures 10.11, 10.12 and 10.13 show the comparison of objective values for case A, B and C respectively. In all figures, the blue and green solid lines denote the objective values obtained when implementing changes simultaneously, penalized with the Strict and Reversion/Switch-Off penalties, respectively. The solid black line denotes the objective value obtained with the RA.

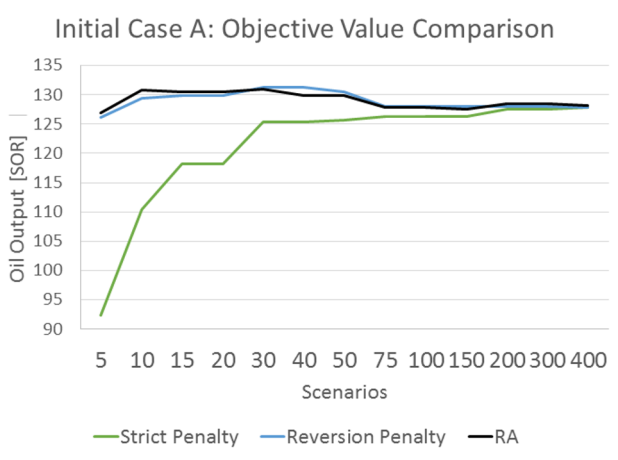

Figure 10.11: The obtained average objective values of the RA, and the Strict and Reversion penalties for initial case A.

In all cases, the number of infeasible solutions is typically high for models with few scenarios regardless of whether simultaneous changes or the RA are used. The objective value

under the Strict penalty is low as a consequence but approaches the Reversion/Switch-Off penalty and RA as the number of scenarios increases in all cases. The gap between the Strict penalty and the Switch-Off penalty is higher for case C than for A and B for models with a high number of scenarios. For cases A and B, the gap is 0.00 and 0.47 at its lowest, respectively, while for case C the lowest gap is 3.32. The reason for this is the higher degree of uncertainty in case C, noted in Section 10.3.1 for the robust model implementing changes simultaneously, and in Section 10.3.2 for the RA.

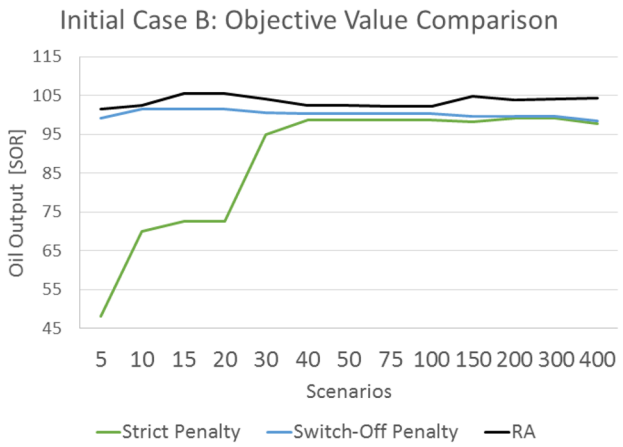

Figure 10.12: The obtained average objective values of the RA, and the Strict and Switch-Off penalties for initial case B.

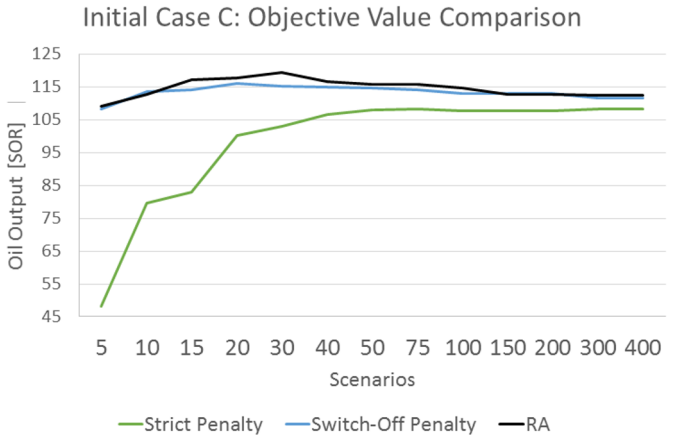

Figure 10.13: The obtained average objective values of the RA, and the Strict and Switch-Off penalties for initial case B.

In cases A and C, regions exist for which the average objective value obtained by the RA is lower than the value obtained by implementing all changes simultaneously using the Reversion or Switch-Off penalties. These regions are mainly located where models contain less than 200 scenarios. Making decisions based on new information in the RA leads to unstable results for models with insufficient numbers of scenarios, confirming results from Section 10.3.2, where the value of learning true well outputs is found to depend on the ability of the model to represent uncertainty in the problem.

In all three initial cases, the objective values are more volatile for lower numbers of scenarios and seem to stabilize for models with 200 scenarios or more. The relative performance of the three approaches is different depending on the number of scenarios we select as our sample. In the previous sections, it is shown that we are not always able to avoid infeasibility even with the maximum number of scenarios. We find it reasonable to use the performance of the approaches with 400 scenarios as the benchmark since we optimize in a robust setting and these models provide the highest robustness against infeasibility. However, a decision maker with a higher appetite for risk may prefer to set a different benchmark.

Table 10.10 shows the objective values for the three approaches with 400 scenarios in the models, for case A, B and C. The RA is higher in all initial cases, with values 0.27%, 5.74% and 0.76% higher than the Reversion/Switch-Off penalties for cases A, B and C respectively. Compared to the Strict penalty, the RA achieves an improvement of 0.27%, 6.60% and 3.86%.

The large improvement in objective values from the RA when compared to implementing changes simultaneously for case B warrants a discussion. In case B, we start in an infeasible point due to the right-hand side of a constraint shifting in the negative direction. The nature of this shift determines how realistically the RA models a real setting. The RA requires letting the system settle in a steady state between each single implemented change. Since this process may take hours in a real oil field, the RA implicitly assumes the system is able to spend considerable time in an, albeit decreasingly, infeasible state as it sequentially moves towards feasibility. If the nature of the constraint shift is dramatic and requires feasibility to be achieved as fast as possible, the RA is ill-suited to the task. In such cases, the observed gain in objective value is not realistically achievable. Nonetheless, for less dramatic shifts, e.g., planned maintenance, the results indicate that updating model with observed data can yield significant performance gains. However, in Section 8.5 it is noted that since the RA learns the entire true well output curve of a well for which an operational change is performed, RA results are an optimistic estimate of the performance we expect to see in a real application. If a more realistic learning process is modeled, it is unclear from our results whether or not the performance gain from the RA is significant. Considering case A, where the average (optimistic) gain is only 0.27%, we may expect this gain to be insignificant in a realistic setting. Modifications to the learning process of the RA are discussed in more detail in Chapter 12.

Table 10.10: Objective value comparison for the penalty approaches and the RA for 400 scenarios included in the models.

	Initial Case		
Approach	**A**	**B**	**C**
Strict Penalty	127.86	97.81	108.31
Reversion/Switch-Off Penalty	127.86	98.61	111.64
RA	128.20	104.27	112.49

10.4 Multi-objective Programming

In this section, we show the results from the MOP, implemented as described in Section 8.6. The MOP is a deterministic model which seeks to find solutions where the oil production is maximized, and the total gas uncertainty is minimized. The problem is solved for the three initial cases introduced in Section 9.3.

The objective values of the solutions are visualized as a Pareto front for cases A and B, with the visualization for case C found in Appendix C. Exhaustive data for the Pareto fronts is also found in Appendix C. Such a Pareto front serves as a set of options for the production engineer. Depending on the current risk aversion of the engineer and other situational factors, he or she may prefer solutions from different regions of the Pareto front. For each visualization, we present five plots. These plots represent the problem solved with $1-5$ permitted changes and allow us to examine how the Pareto front varies with the number of changes we wish to perform. Here, the y-axis represents the confidence level of a model that its solution is feasible with respect to the total gas capacity constraint. The calculation for this confidence is shown in 8.6. Note that we represent confidence in percentage units throughout this section.

10.4.1 MOP: Case A

In case A, we start in an operating point in which we are close to, but below the maximum gas capacity constraints. In Figure 10.14, we see that for any number of permitted changes, the solution is guaranteed to be feasible for oil output levels slightly above 105 SOR. This is expected since the output of oil is 105.66 SOR in the initial setting of this scenario.

The maximum achievable oil output is relatively similar for all solutions permitting 2 or more changes, with the largest difference being 3.1% between 5 and 2 permitted changes. Almost any desired level of oil output is achievable with only a few allowed changes as a consequence. However, the confidence of the model that the resulting solution is feasible differs significantly for lower and higher numbers of permitted changes. The confidence levels for 3, 4 and 5 changes follow each other closely for oil outputs lower than 125, after which differences appear, and the model with 5 changes is the most confident. Clearly, 2 allowed changes produces significantly lower confidence levels than $3-5$ changes for large regions of the oil output rates. Uncertainty estimates from the well models explain this behavior. In general, moving a well choke setting far from a previously observed value

is associated with high uncertainty since we do not know how the well output behaves in such regions. In the case considered in this study, output measurements do not exist for choke settings above 60. Consequently, increasing oil production by changing only a few wells forces the model to implement choke settings for which the output rate estimates exhibit high degrees of uncertainty. Higher confidence is obtained if the model is allowed to perform minor changes for several wells, keeping choke levels closer to previously observed measurements where uncertainty estimates are lower.

The sharp drop in confidence for the model with 2 changes, from 65.75% to 25.58% between an oil output of 128.46 and 129.83, corresponds to a change in the configuration of wells that are producing in the solution. Specifically, at this point the model begins producing from well W2, inducing a significantly higher risk of infeasibility. This indicates that well W2 is a well with high levels of uncertainty associated with the oil output rate it produces in the solution. The qualitative properties of solutions in the Pareto front are discussed in more detail in Section 10.4.5.

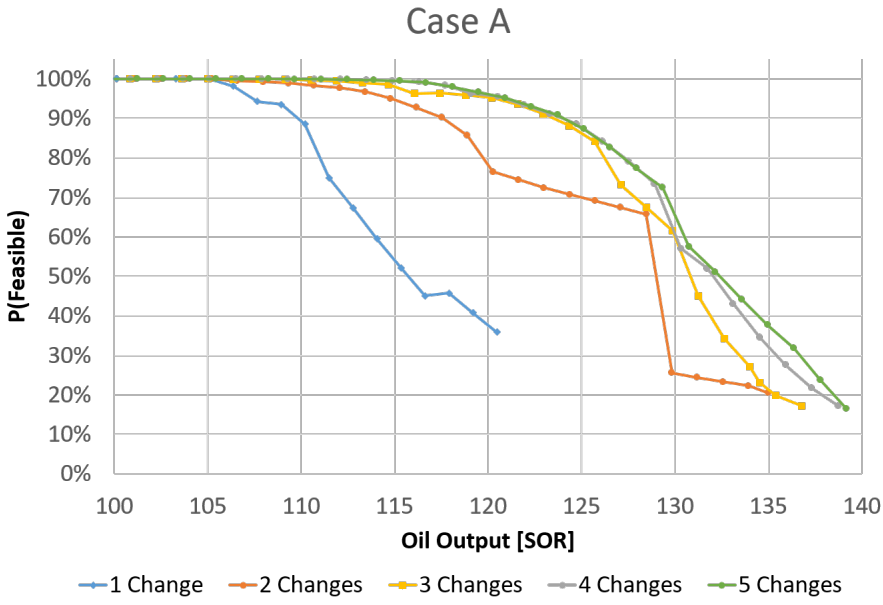

Figure 10.14: Pareto front for case A and 1-5 allowed changes.

10.4.2 MOP: Case B

The initial operating point in case B is an infeasible one, where the total gas capacity constraint is breached. In Figure 10.15 the model with 1 permitted change yields a maximum confidence level of 97.95%. Thus, the model is unable to guarantee feasibility when only a single change is permitted. For the models with 2 or more changes allowed, smaller differences in confidence levels are observed when compared to the plot for case A in

Section 10.4.1. The maximum difference between these models is found for an oil output of 111.11, where the model with 5 changes allowed is 8.34% more confident in its solution than the model with 2 changes allowed. The plots for 4 and 5 changes are identical, indicating that 4 changes are the maximum amount needed to obtain the most confident solution for any desired oil output level. Thus, we observe diminishing returns on the increase in confidence gained by allowing more changes.

The initial oil output is 124.22 in case B. The MOP results show that the maximum probability of reaching a feasible state without incurring a loss in the oil output in this case is 18.18%, obtained for the model with 4 or 5 changes (since these produce equivalent fronts). Thus, a decision maker who wishes to maintain the initial oil output level in the event of a reduction in the total gas capacity, observes that doing so entails that the probability of a constraint breach is over 80%.

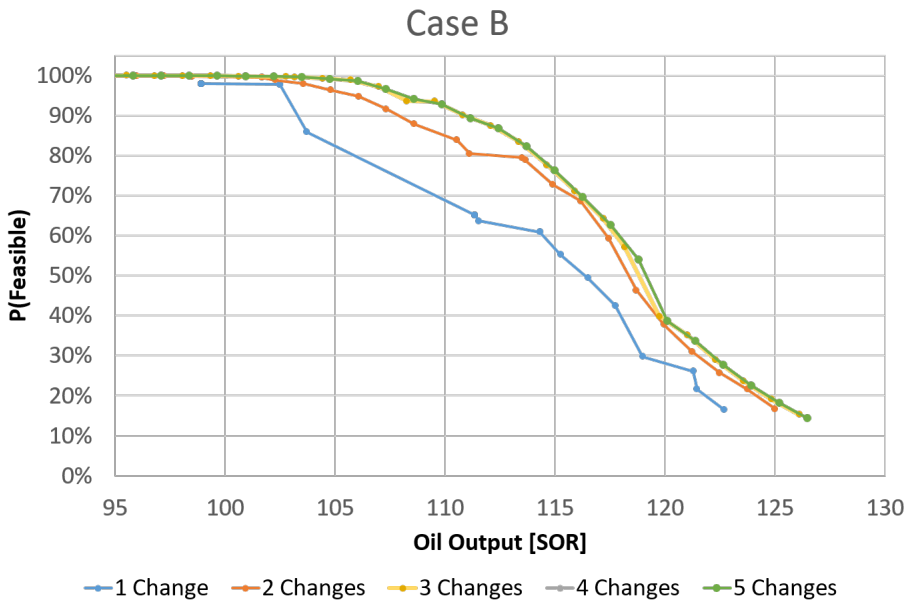

Figure 10.15: Pareto front for case B and 1-5 allowed changes.

10.4.3 MOP: Case C

In this case, all models with $1-4$ permitted changes produce solutions that are always, or nearly always feasible, while models with $5-7$ changes produce identical Pareto fronts. The figure that shows the results for case C is therefore not discussed in detail but is included in Appendix C.

10.4.4 Comparing MOP and RA Solutions

When considering the MOP Pareto front, an optimal robust solution is found for the point on the front yielding the highest oil production while still guaranteeing $P(Feasible) = 1$. In Section 8.6, we simplify the MOP calculations for $P(Feasible)$ by disregarding the probability of breaching the individual well gas capacity constraints. The subsequent loss of accuracy in the calculation of $P(Feasible)$ depends on whether the relevant case is constrained more tightly by the total gas capacity constraint than the individual capacity constraints for each well. If we produce gas at a rate close to individual gas constraints in the optimal solution, the MOP estimate of $P(Feasible)$ is not realistically achievable for a robust formulation considering both constraint types. We therefore note that in the following discussion, the MOP confidence level is artificially high when compared to the RA since the latter is robust against both constraint breaches while the former is not.

We now compare results for case C. RA and MOP uncertainty estimates are straightforward to compare when all regions of well output curves are uncertain, i.e., when no initially known points are specified. Figure 10.16 shows RA and MOP solutions for 6 changes in case C, with results for RA labeled according to the number of scenarios the model includes. The optimal robust solution on the MOP line is marked with a red dot, corresponding to a 100% confidence of feasibility for an oil output of 112.01 SOR. The probability of feasibility, $P(Feasible)$, is for the RA calculated as the number of feasible solutions it obtained in Table 10.8 relative to the 200 ground truth scenarios, with the corresponding oil output of the RA found in the same table.

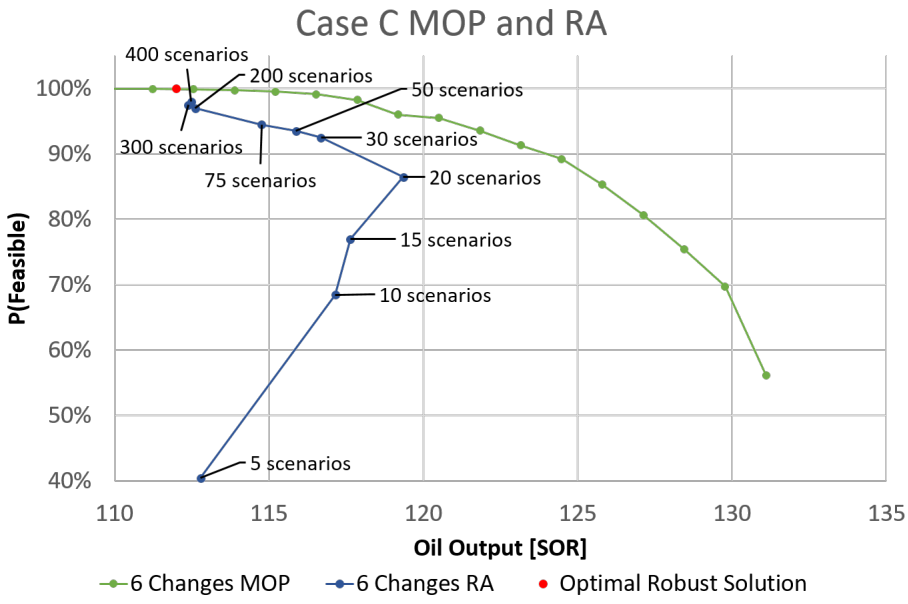

Figure 10.16: Pareto front for case C and 6 allowed changes, with RA solutions from Table 10.8

With 5 scenarios, the RA is feasible in 81 out of 200 ground truth scenarios, corresponding to a confidence level of 40.50% and oil output of 112.79 SOR. The number of infeasible solutions produced quickly drops as we increase the number of scenarios, with 20 scenarios yielding feasible solutions in 173 out of 200 ground truths for a confidence level of 86.50%, and increased oil output of 119.35 SOR. Thus, the RA solution with 20 scenarios dominates all solutions obtained for models with fewer scenarios. As we further increase the number of scenarios, the RA produces more conservative solutions which offer higher confidence of feasibility at the cost of lower oil outputs. Except for the solution for 300 scenarios which is dominated by the solution for 400 scenarios, the RA results form a Pareto front for models with $20 - 400$ scenarios. We note that the oil output of the RA depends on the sample of scenarios included in the problem, particularly for small scenario samples. Thus, the bias of the scenario sample causes large variations in the oil output, and the behavior of models with few scenarios may vary significantly for different samples. That is, for repeated experiments we expect large variations in the behavior of the RA solutions for scenario sizes $5 - 100$. For models with more than 100 scenarios where bias is lower, confidence levels and oil outputs are approximately equal.

The RA solution does indeed seem to approach the optimal robust solution in the MOP Pareto front for models with 200 scenarios or more. The remaining gap between the 400 scenario RA solution and the optimal solution marked by the red dot may be due to the previously noted artificially high confidence of the MOP Pareto front. We also note, as discussed in Section 10.3.2, that the RA fails to produce solutions that are feasible in all 200 ground truth scenarios even with 400 scenarios in the model in case C. Thus, it is possible the gap is closed for RA models with a higher number of scenarios.

In the comparison above, the MOP confidence is based on the estimated well model distribution. The RA includes Markov Weighted scenarios which are generated based on this distribution and a Markov assumption. Since the two models converge to approximately the same optimal solution in a robust setting, the Markov Weighted scenario generation procedure seems to be a sensible representation of different well output curve realizations given the estimated distribution. That is, the scenario generation procedure does not produce scenarios which poorly reflect our belief of the underlying distribution. Informally, we may think of this as an indication of the out-of-sample stability of the scenario generation procedure, since solutions obtained using the scenarios yield approximately the same objective values as solutions obtained using the underlying distribution.

10.4.5 Analyzing the Solutions in a Pareto Front

In this section, we investigate how the well configurations change in the solution as we adjust the value of ε_j in the ε-constrained method. We consider case A and generate the Pareto front with 2 allowed changes, similarly to the plot in Section 10.4.1. Now, for each solution point in the front, we color the point according to the combination of wells that are producing. The results are presented in in Figure 10.17. Here, each color corresponds to a unique configuration of producing wells. The choke level value for each well is variable within each color, but the configuration of which wells are active is unchanged. Table 10.11 shows which wells are active for each unique combination.

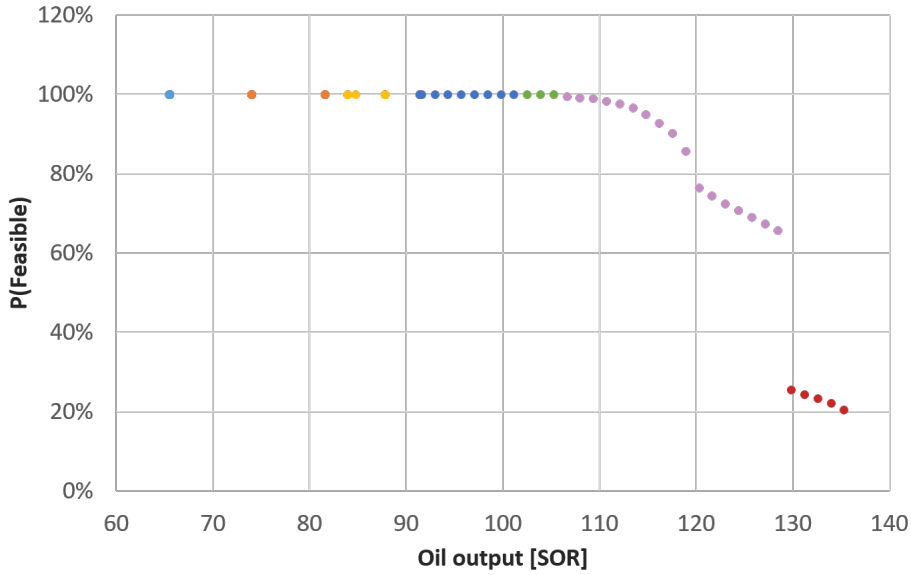

Figure 10.17: Pareto front for case A with 2 changes. The front represents the trade-off between maximizing oil output while attempting to ensure infeasibility. Each color represents one unique combination of wells producing in the solution.

Table 10.11: For each color in Figure 10.17, the check mark in each well column indicates if the well is producing or not in the configuration. The top row (blue) corresponds to the lowest oil output and the highest confidence $P(Feasible)$, while the bottom row (red) corresponds to the highest oil output and the lowest confidence $P(Feasible)$.

	Well						
Solution	**W1**	**W2**	**W3**	**W4**	**W5**	**W6**	**W7**
	✓	✓				✓	
	✓				✓	✓	
	✓	✓			✓	✓	
	✓				✓	✓	✓
	✓	✓			✓	✓	✓
	✓		✓		✓	✓	✓
	✓	✓	✓		✓	✓	✓
Initial Configuration	✓	✓			✓	✓	✓

In Table 10.11, we see that some wells are more favored than others depending on the risk aversion of the decision maker. For example, W3 is one of the first wells in which production is ceased in the solution as the constraint on oil production is loosened. This indicates that W3 has a high oil output relative to the other wells but at the cost of high gas output and/or uncertainty. Consequently, W3 produces only for high levels of oil output, i.e., when the model is forced to produce from W3 in order to satisfy the constrained oil

production. As soon as solutions satisfying the constrained oil output rate without W3 are identified, it is switched off. On the other hand, W1 and W6 are switched on in every solution in the Pareto front. This indicates that producing from W1 and W6 yields low gas outputs and corresponding variance in gas output relative to the oil output.

Chapter 11

Concluding Remarks

In this thesis, we consider the real-time optimization of an offshore petroleum production field. The problem involves maximizing oil production on a daily basis by performing minor adjustments in the field and frequently re-solving the optimization models. Historical measurement data from the field forms the basis for modelling well output functions. Two sources of uncertainty in estimated gas outputs, epistemic and aleatoric, make adhering to gas capacity constraints in the field complicated. The risk aversion of the field engineer with respect to breaching these constraints is assumed to vary with field conditions.

We formulate a robust stochastic MILP modelling a maximally risk averse field engineer, and a MOP modelling variable risk aversion. Well output functions and the corresponding uncertainty are modelled with NNs, reformulated as MILPs and incorporated into the optimization models. Two distinct scenario generation procedures, Factor and Markov Weighted, are developed for the robust problem. A recourse algorithm which aims to exploit the discovery of new data points as we implement changes in the field is presented. Finally, all components are integrated into a seamless solution pipeline and tested on a simplified representation of a real oil field.

An important contribution of this thesis is a novel well model based on NN architectures able to predict distributions for the target output. We combine this architecture with a recently developed technique for estimating model uncertainty, using MC dropout to approximate variational inference in a BNN. To our knowledge, this is the first work to utilize NNs able to capture both epistemic and aleatoric uncertainty in estimated output rates.

The technical study compares two different NN MILP representations with a model using SOS2. Results show that the SOS2 model yields the quickest solution times, while NN MILP scalability is poorer than expected. The difference is particularly evident in the case of Markov Weighted scenarios, where both formulations represent the exact same piecewise linear functions. Although alternative branching strategies may help close the gap, results indicate that the NN MILP formulation is an inefficient representation of a

piecewise linear function with a single dimensional input. However, due to the curse of dimensionality, SOS2 scales poorly in cases of higher dimensional input and further testing is needed to determine which formulation offers the superior solution times in such cases.

We solve the robust stochastic problem for three initial cases for both scenario generation procedures. The Factor scenarios suffer from substantial bias due to their inability to deviate from the shape of the mean curve, producing overly conservative solutions. Furthermore, the scenarios reduce to deterministic curves when initially known production points are specified. Due to these issues, the Factor scenarios are found to be unsuited models of well output function realizations in their current form. Results using Markov Weighted scenarios show that planning with uncertainty in the optimization models adds value compared to planning with expected values. However, the VSS strongly depends on the way infeasible solutions are penalized. If the penalty does not reflect the true utility of the decision maker, we end up with a poor understanding of model performance.

The recourse algorithm outperforms the stochastic robust model by 0.27% 5.74%, and 0.76%, for cases A, B and C respectively, when using the softest punishment on infeasible solutions obtained from the robust model. However, due to simplifications in its implementation the RA represents an optimistic estimate of the value we expect to gain from learning true well output functions. Furthermore, the RA assumes that it is always possible to let the production system settle in a steady state before implementing operational changes. This is not realistic in certain settings where reaching feasibility as quickly as possible is the top priority. Thus, we expect performance gains from applying the RA in a real application to be lower than reported in the computational study.

The results from the MOP optimization yield Pareto fronts maximizing the probability of feasibility for a given oil output. Constraining the number of operational changes in the problem increases the risk of infeasibility. Higher confidence levels are obtained by allowing numerous smaller changes since this allows wells to operate in regions of relatively low uncertainty. Analyses of the well configurations of different solutions in the Pareto front reveal insights into well uncertainty characteristics. Specifically, results show how certain well configurations are preferable to others depending on the desired oil output.

An optimistic estimate of the optimal robust solution is located in the MOP Pareto front, maximizing oil output while guaranteeing feasibility. When comparing RA and MOP results, we find that the models converge to approximately the same optimal robust solution. This indicates that the Markov Weighted scenario generation procedure properly represents our belief of the underlying distribution of the stochastic variables in the problem. Thus, an informal indication of out-of-sample stability is obtained.

In conclusion, the novel well models proposed in this thesis offer a flexible representation of epistemic and aleatoric uncertainty. Incorporating these models into the RTO problems facilitates decision support for a production engineer who is concerned with maximizing oil production while considering uncertainty. In particular, the models developed in this thesis enable the decision maker to identify optimal solutions for any level of risk aversion.

Chapter 12

Future Research

In this chapter, we give ideas and suggestions for further research topics. Several challenges with the models proposed in this thesis are identified in the computational study which require closer examination. In the following, we keep the discussion at a conceptual level.

The well models developed in this thesis provide the estimated distribution of the output rate given any setting of the input decision variables. These models provide flexible tools applicable to a variety of stochastic problems. A natural extension of the robust stochastic model in this thesis is modeling both oil and gas output rates as uncertain. Additionally, numerous alternative optimization applications exist that make use of the acquired distributions. For instance, it is possible to minimize the expected value of any gas capacity constraint breach by considering conditional value at risk (CVaR). A multi-stage stochastic model including recourse decisions is another possible formulation that further makes use of the uncertainty estimates of the well models.

In Section 8.5, we discuss the way in which operational changes are implemented in a petroleum production field. After performing one or more changes, the production system is typically allowed to settle in a steady state. Upon reaching this steady state, we are in a position to approximate the output rates of the system in its new setting. The RA presented in this thesis models this process but updates the optimization model with true information regarding the entire production curve of the well for which a change was made. In reality, we have at this time only observed a single new point of the true production curve of the well. Furthermore, the initially known point for this well now represents a slightly outdated data point, since conditions in the field may change while the system settles in its new steady state. An interesting topic for further research is to generate new scenarios once the system has moved to a new steady state. In the new scenarios, the current operating point with corresponding output rates is treated as the known point. The previous operating point is treated as uncertain, but less so than unobserved parts of the well curve since we have a relatively recent measurement data point for this input setting. Scenar-

ios are allowed to deviate slightly from the previous point, e.g., half a standard deviation. This method potentially allows a closer approximation of reality than what is possible in the current implementation of the RA.

The case application in this thesis is based on a single dimensional input, i.e., the choke level. In Section 10.1.1, the SOS2 formulation for Markov Weight scenarios offers considerably better scalability than NNs reformulated as MILPs. However, for cases with higher input dimensionality, SOS2 requires grid approximations which may quickly become computationally intractable. In such cases, NN MILPs likely scale better due to the fact that increasing the dimension of the input merely involves adding another input neuron to the NN. The increase in integer variables and constraints in the MILP reformulation of the NN is in this case relatively modest. Nonetheless, the identified scaling problems of the NN MILP formulation with Markov Weighted scenarios need to be addressed if the optimization models are to be applied to large problem instances with high dimensionality input. Relevant research topics include an effective branching strategy and methods for calculating bounds on the objective value. Alternative formulations of NNs as MILPs are also of interest.

In Section 4.3.3 we use a loss function based on the log-likelihood of the predicted output and corresponding variance. The log-likelihood assumes that the errors of the function generating our data are normally distributed. In a real application, this assumption may not be realistic. Extensive work exists regarding the use of NNs to predict intervals containing the target variable rather than point estimates, known as the field of predictive intervals (PIs). Several methods and architectures do not rely on assumptions regarding the distribution of the target function, with interesting works including Khosravi et al. (2011a), Khosravi et al. (2011b) and Pearce et al. (2018). We suggest conducting a feasibility study to examine if such NN architectures are suitable for integration in a MILP optimization framework. PIs potentially provide superior flexibility since no assumptions regarding the underlying distribution are made.

In Section 8.2.1, we use heuristic rules and standard values in the literature to determine a subset of the NN well model hyperparameters. Consequently, the grid and randomized searches conducted in Section 10.2.1 are only conducted for parameters with regularization effects, i.e., the L2 regularization term λ and the dropout rate p, and architecture parameters, i.e., the number of hidden layers L and the number of neurons N in each such layer. We are not able to verify whether or not the defined search space lets us identify the hyperparameters yielding the best performance. Thus, further studies examining results from searching over different sets of hyperparameters are of great interest, for instance, examining the effect of varying the learning rate parameter.

Bibliography

Alarcón, G. A., Torres, C. F., Gómez, L. E., 2002. Global optimization of gas allocation to a group of wells in artificial lift using nonlinear constrained programming. Journal of Energy Resources Technology 124, 262–268.

Bahadori, A., Ayatollahi, S., Moshfeghian, M., 2001. Simulation and optimization of continuous gas lift system in aghajari oil field. Society of Petroleum Engineers.

Bergstra, J., Bengio, Y., 2012. Random search for hyper-parameter optimization. Journal of Machine Learning Research 13 (Feb), 281–305.

Bieker, H. P., 2007. Topics in offshore oil production optimization using real-time data. Ph.D. thesis, Norwegian University of Science and Technology.

Bieker, H. P., Slupphaug, O., Johansen, T., 2007. Real-time production optimization of oil and gas production systems: A technology survey.

Bishop, C. M., 1994. Mixture density networks.

Bishop, C. M., 1995. Neural networks for pattern recognition. Oxford university press.

Blundell, C., Cornebise, J., Kavukcuoglu, K., Wierstra, D., 2015. Weight uncertainty in neural networks. arXiv preprint arXiv:1505.05424.

Breiman, L., Spector, P., 1992. Submodel selection and evaluation in regression. the x-random case. International Statistical Review 60 (3), 291–319.

Brown, K. E., 1982. Overview of artificial lift systems. Journal of Petroleum Technology 34 (10), 2384–2396.

Buitrago, S. B., Rodriguez, E., Espin, D., 04 1996. Global optimization techniques in gas allocation for continuous flow gas lift systems SPE 35616, 375–383.

Buntine, W. L., 1991. Bayesian back-propagation. Complex systems 5 (6), 603–643.

Changkong, V., Haimes, Y. Y., 1983. Multiobjective decision making: Theory and methodology. In: North-Holland Series in System Science and Engineering. Vol. 8. Elsevier Science Publishing Co New York NY.

Cheng, C.-H., Nührenberg, G., Ruess, H., 2017. Maximum resilience of artificial neural networks. In: International Symposium on Automated Technology for Verification and Analysis. Springer, pp. 251–268.

Chollet, F., et al., 2015. Keras.

Dahl, E., Corneliussen, S., Couput, J.-P., Dykesteen, E., Frøysa, K.-E., Malde, E., Moestue, H., Moksnes, P. O., Scheers, L., Tunheim, H., 2005. Handbook of Multiphase Flow Metering, 2nd Edition. The Norwegian Society for Oil and Gas Measurement and The Norwegian Society of Chartered Technical and Scientific Professionals.

de Freitas, J. F. G., 2003. Bayesian methods for neural networks. Ph.D. thesis, University of Cambridge.

Ding, A. A., He, X., Mar 2003. Backpropagation of pseudo-errors: neural networks that are adaptive to heterogeneous noise. IEEE Transactions on Neural Networks 14 (2), 253–262.

Dutta-Roy, K., Kattapuram, J., 1997. A new approach to gas-lift allocation optimization. Society of Petroleum Engineers, 685–691.

Dybowski, R., Roberts, S. J., 2001. Confidence intervals and prediction intervals for feed-forward neural networks. Cambridge University Press, p. 298–326.

E. Hinton, G., Srivastava, N., Krizhevsky, A., Sutskever, I., R. Salakhutdinov, R., 07 2012. Improving neural networks by preventing co-adaptation of feature detectors arXiv.

Fang, W. Y., Lo, K. K., 1996. A generalized well-management scheme for reservoir simulation. SPE Reservoir Engineering, 116–120.

Fischetti, M., Jo, J., 2017. Deep neural networks as 0-1 mixed integer linear programs: A feasibility study. arXiv preprint arXiv:1712.06174.

Gal, Y., 2016. Uncertainty in deep learning. Ph.D. thesis, University of Cambridge.

Gal, Y., Ghahramani, Z., 2015. Dropout as a Bayesian approximation: Insights and applications. In: Deep Learning Workshop, ICML.

Gharbi, R., Elsharkawy, A. M., et al., 1997. Neural network model for estimating the pvt properties of middle east crude oils. In: Middle East Oil Show and Conference. Society of Petroleum Engineers.

Graves, A., 2011. Practical variational inference for neural networks. In: Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F., Weinberger, K. Q. (Eds.), Advances in Neural Information Processing Systems 24. Curran Associates, Inc., pp. 2348–2356.

Grimstad, B., 10 2015. Daily production optimization for subsea production systems. Ph.D. thesis, Norwegian University of Science and Technology.

Grimstad, B., Foss, B., Heddle, R., Woodman, M., 2016. Global optimization of multi-phase flow networks using spline surrogate models. Computers Chemical Engineering 84 (Supplement C), 237 – 254.
URL http://www.sciencedirect.com/science/article/pii/S0098135415002835

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

Gurobi, 2014. Inc.,"gurobi optimizer reference manual," 2015. Google Scholar.

Hamedi, H., Rashidi, F., Khamehchi, E., 2011. A novel approach to the gas-lift allocation optimization problem. Petroleum Science and Technology 29 (4), 418–427.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning. Springer.

Heskes, T., 10 1997. Practical confidence and prediction intervals 9.

Hinton, G. E., Van Camp, D., 1993. Keeping the neural networks simple by minimizing the description length of the weights. In: Proceedings of the sixth annual conference on Computational learning theory. ACM, pp. 5–13.

International Energy Agency, 2018. Energy balance flows. https://www.iea.org/statistics/ieaenergyatlas/, accessed: 2018-06-06.

Kanu, E. P., 1981. Economic approach to oil production and gas allocation in continuous gas lift. Journal of Petroleum Technology 33 (10), 1887–1892.

Kendall, A., Gal, Y., 2017. What uncertainties do we need in bayesian deep learning for computer vision? CoRR abs/1703.04977.
URL http://arxiv.org/abs/1703.04977

Kennedy, M. P., Chua, L. O., 1988. Neural networks for nonlinear programming. IEEE Transactions on Circuits and Systems 35 (5), 554–562.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., Tang, P. T. P., 2016. On large-batch training for deep learning: Generalization gap and sharp minima. arXiv preprint arXiv:1609.04836.

Khishvand, M., Khamehchi, E., Nokandeh, N. R., 2015. A nonlinear programming approach to gas lift allocation optimization. Energy Sources, Part A: Recovery, Utilization, and Environmental Effects 37 (5), 453–461.

Khor, C. S., Elkamel, A., 11 2017. Production systems optimization methods for petroleum fields.

Khosravi, A., Nahavandi, S., Creighton, D., Atiya, A. F., Sept 2011a. Comprehensive review of neural network-based prediction intervals and new advances. IEEE Transactions on Neural Networks 22 (9), 1341–1356.

Khosravi, A., Nahavandi, S., Creighton, D., Atiya, A. F., 2011b. Lower upper bound estimation method for construction of neural network-based prediction intervals. IEEE Transactions on Neural Networks 22 (3), 337–346.

King, A. J., Wallace, S. W., 2012. Modeling with stochastic programming. Springer Science & Business Media.

Kingma, D. P., Ba, J., 2014. Adam: A method for stochastic optimization. CoRR abs/1412.6980.
URL http://arxiv.org/abs/1412.6980

Kingma, D. P., Salimans, T., Welling, M., 2015. Variational dropout and the local reparameterization trick. In: Advances in Neural Information Processing Systems. pp. 2575–2583.

Kingma, D. P., Welling, M., 2013. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

Kosmidis, V. D., Perkins, J. D., Pistikopoulos, E. N., 2004. Optimization of well oil rate allocations in petroleum fields. Ind. Eng. Chem. Res. 43 (14), 3513–3527.

Kosmidis, V. D., Perkins, J. D., Pistikopoulos, E. N., 2005. A mixed integer optimization formulation for the well scheduling problem on petroleum fields. Computers and Chemical Engineering 29 (1), 1523–1541.

Kullback, S., Leibler, R. A., 1951. On information and sufficiency. The annals of mathematical statistics 22 (1), 79–86.

Lampinen, J., Vehtari, A., 2001. Bayesian approach for neural networks—review and case studies. Neural networks 14 (3), 257–274.

Lang, Z. X., Horne, R. N., 1983. Optimum production scheduling using reservoir simulators: A comparison of linear programming and dynamic programming techniques. Society of Petroleum Engineers.

Leiras, A., Ribas, G., Hamacher, S., Elkamel, A., 2011. Literature review of oil refineries planning under uncertainty. International Journal of Oil, Gas and Coal Technology 4 (2), 156–173.

Louizos, C., Welling, M., 2016. Structured and efficient variational deep learning with matrix gaussian posteriors. In: International Conference on Machine Learning. pp. 1708–1716.

Mach, J., Proano, E., Brown, K. E., 1979. A nodal approach for applying systems analysis to the flowing and artificial lift oil or gas well. SPE 8025.

MacKay, D. J., 1992. A practical bayesian framework for backpropagation networks. Neural computation 4 (3), 448–472.

Malvik, A. G., Witzøe, B., 2017. A piecewise linear real-time optimization model with multivariate adaptive regression splines and neural networks. Project report, TIØ4500 - Managerial Economics and Operations Research.

Manne, A. S., 1960. On the job-shop scheduling problem. Operations Research 8 (2), 219–223.

Martinez, E. R., Moreno, W. J., Moreno, J. A., Maggiolo, R., 1994. Application of genetic algorithm on the distribution of gas-lift injection. Societ of Petroleum Engineers.

Misener, R., Gounaris, C. E., Floudas, C. A., 2009. Global optimization of gas lifting operations: A comparative study of piecewise linear formulations. Ind. Eng. Chem. Res. 48 (13), 6098–6104.

Mora, O., Startzman, R. A., Saputelli, L., 2005. Maximizing net present value in mature gas-lift fields. Society of Petroleum Engineers.

Morken, M. L., Sandberg, P. T., 6 2016. Modelling and optimization of real-time petroleum production. Master's thesis, Norwegian University of Science and Technology, Trondheim, Norway.

Neal, R. M., 1995. Bayesian learning for neural networks. Ph.D. thesis, Toronto, Ont., Canada, Canada, aAINN02676.

Neiro, S. M., Pinto, J. M., 2004. A general modeling framework for the operational planning of petroleum supply chains. Computers & Chemical Engineering 28 (6), 871 – 896, fOCAPO 2003 Special issue.

Neiro, S. M. S., Pinto, J. M., 2003. Supply chain optimization of petroleum refinery complexes. In: Proceedings FOCAPO 2003 (Eds. I.E. Grossmann and C.M. McDonald. pp. 59–72.

Nemhauser, G., Wolsey, L., 1988. Integer and Combinatorial Optimization. John Wiley & Sons.

Nishikiori, N., Redner, R. A., Doty, D., Schmidt, Z., 1995. An improved mehtod for gas lift allocation optmization. Journal of Energy Resources Technology 117 (1), 87–92.

Nix, D. A., Weigend, A. S., 1994. Estimating the mean and variance of the target probability distribution. In: International Conference on Neural Networks. Orlando, FL, USA.

Papadopoulos, G., Edwards, P. J., Murray, A. F., Nov 2001. Confidence estimation methods for neural networks: a practical comparison. IEEE Transactions on Neural Networks 12 (6), 1278–1287.

Pearce, T., Zaki, M., Brintrup, A., Neely, A., 02 2018. High-quality prediction intervals for deep learning: A distribution-free, ensembled approach. arXiv:1802.07167.
URL https://arxiv.org/abs/1802.07167

Prékopa, A., 2013. Stochastic programming. Vol. 324. Springer Science & Business Media.

PROSPER, 2015. PROSPER Brochure. Petroleum Experts Ltd.

PwC, 2017. 2017 oil and gas trends: Adjusting business models to a period of recovery. PwC Strategy&.

Rashid, K., Bailey, W., Couet, B., 06 2012. A survey of methods for gas-lift optimization 2012.

Rasmussen, C. E., 2004. Gaussian processes in machine learning. In: Advanced lectures on machine learning. Springer, pp. 63–71.

Ray, T., Sarker, R., 2006. Multiobjective evolutionary approach to the solution of gas lift optimization problems. In: 2006 IEEE International Conference on Evolutionary Computation. pp. 3182–3188.

Redden, D. J., Sherman, G. T. A., Blann, J. R., 1974. Optimizing gas-lift systems. Society of Petroleum Engineers of AIME.

Rezende, D. J., Mohamed, S., Wierstra, D., 2014. Stochastic backpropagation and approximate inference in deep generative models. arXiv preprint arXiv:1401.4082.

Rissanen, J., 1986. Stochastic complexity and modeling. The annals of statistics, 1080–1100.

Rodriguez-Vazquez, A., Dominguez-Castro, R., Rueda, A., Huertas, J. L., Sanchez-Sinencio, E., 1990. Nonlinear switched capacitor'neural'networks for optimization problems. IEEE Transactions on Circuits and Systems 37 (3), 384–398.

Roe-Fortmann, S., 2012. Understanding the bias-variance tradeoff. http://scott.fortmann-roe.com/docs/BiasVariance.html, accessed: 2017-10-12.

Ruz-Hernandez, J. A., Salazar-Mendoza, R., de la C., G. J., Garcia-Hernandez, R., Shelomov, E., 2010. An Approach Based on Neural Networks for Gas Lift Optimization. Springer Berlin Heidelberg, pp. 207–224.

Saputelli, L., Malki, H., Canelon, J., Nikolaou, M., 2002. A critical overview of artificial neural network applications in the context of continuous oil field optimization. SPE Annual Technical Conference and Exhibition, 29 September-2 October, San Antonio, Texas.

Schlumberger Limited, 2017. Oilfield glossary. `http://www.glossary.oilfield.slb.com/Terms/r/reservoir.aspx`, accessed: 2017-11-13.

Sequeira, S. E., Graells, M., Puigjaner, L., 2002. Real-time evolution for on-line optimization of continuous processes. Industrial & Engineering Chemistry Research 41 (7), 1815–1825.

Shokir, E. M. E.-M., Hamed, M. M. B., Ibrahim, A. E.-S. B., Mahgoub, I., 2017. Gas lift optimization using artificial neural network and integrated production modeling. Energy & Fuels 31 (9), 9302–9307.

Silva, T. L., Camponogara, E., 2014. A computational analysis of multidimensional piecewise-linear models with applications to oil production optimization. European Journal of Operational Research 232 (3), 630–642.

Simmons, W., 08 1972. Optimizing continuous flow gas lift wells. pt. 1-2 45:8.

Simon, F. Y.-P., et al., 1988. Integer linear programming neural networks for job-shop scheduling. In: Neural Networks, 1988., IEEE International Conference on. IEEE, pp. 341–348.

Sra, S., Nowozin, S., Wright, S. J., 2011. Optimization for Machine Learning. The MIT Press.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research 15 (1), 1929–1958.

SSB Statistikkbanken, 2018. Utenrikshandel med varer - tabell 08800 og 09226. https://www.ssb.no/statistikkbanken/selecttable/hovedtabellHjem.asp?KortNavnWeb=muh&CMSSubjectArea=utenriksokonomi&checked=true, accessed: 2018-06-06.

Stoisits, R. F., Batesole, E. C., Champion, J. H., Park, D. H., 1992. Application of non-linear adaptive modeling for rigorous representation of production facilities in reservoir simulaton. Society of Petroleum Engineers, 425–434.

Stoisits, R. F., Crawford, K. D., MacAllister, D. J., McCormack, M. D., Lawal, A. S., Ogbe, D. O., 1999. Production optimization at the kuparuk river field utilizing neural networks and genetic algorithms. Society of Petroleum Engineers, 1–12.

Stoisits, R. F., Scherer, P. W., Schmidt, S. E., 1994. Gas optimization at the kuparuk river field. Society of Petroleum Engineers, 35–42.

Stutzle, T., Dorigo, M., Aug 2002. A short convergence proof for a class of ant colony optimization algorithms. IEEE Transactions on Evolutionary Computation 6 (4), 358–365.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R., 2013. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199.

Tank, D., Hopfield, J., 1986. Simple'neural'optimization networks: An a/d converter, signal decision circuit, and a linear programming circuit. IEEE transactions on circuits and systems 33 (5), 533–541.

the Norwegian Ministry of Petroleum and Energy, the Norwegian Petroleum Directorate, 2018. Conversion. https://www.norskpetroleum.no/en/calculator/about-energy-calculator/, accessed: 2018-05-29.

Villalobos-Arias, M., Coello Coello, C. A., Hernández-Lerma, O., Sep 2006. Asymptotic convergence of metaheuristics for multiobjective optimization problems. Soft Computing 10 (11), 1001–1005.

Walia, A. S., 2017. Types of optimization algorithms used in neural networks and ways to optimize gradient descent. https://towardsdatascience.com/types-of-optimization-algorithms-used-in-neural-networks-\and-ways-to-optimize-gradient-95ae5d39529f, accessed: 2018-05-28.

Wang, P., 3 2003. Development and applications of production optimization techniques for petroleum fields. Ph.D. thesis, Stanford University, USA.

Weiss, J., Scherer, P. W., Schmidt, S. E., 1994. Large-scale facility expansion evaluations at the kuparuk river field. Society of Petroleum Engineers, 35–42.

Xia, Y., 1996. A new neural network for solving linear and quadratic programming problems. IEEE transactions on neural networks 7 (6), 1544–1548.

Zak, S. H., Upatising, V., Hui, S., 1995. Solving linear programming problems with neural networks: a comparative study. IEEE Transactions on Neural Networks 6 (1), 94–104.

Zerafat, M. M., Ayatollahi, S., Roosta, A. A., 2009. Genetic algorithms and ant colony approach for gas-lift allocation optimization. Journal of the Japan Petroleum Institute 52 (3), 102–107.

Zhang, J., Kim, N.-H., Lasdon, L., 1985. An improved successive linear programming algorithm. Journal Management Science 31 (10), 1312–1331.

Zhang, X.-S., 2013. Neural networks in optimization. Vol. 46. Springer Science & Business Media.

Appendix A

Well Model Appendix

In this appendix, supplementary content regarding the well models used in this thesis is supplied.

A.1 Discounting Overlapping Data Points

In the well measurements, there are cases where a well has several data points with significantly varying production output values for approximately the same input values. For examples of this phenomenon, see Section 9.2. Potential reasons for this is that the measurements are conducted far apart in time or for different back pressures. If such cases are left untreated, regression may place excessive weight on a few outdated input values and behave undesirably.

To overcome the obstacle of overlapping data points, a discount algorithm is implemented which handles the case in which time differences are the cause of the varying output measurements for similar input settings. A variety of discounting approaches exist, the two most common being hyperbolic and exponential discounting. In exponential discounting, some attribute of interest is chosen and points are discounted with respect to this attribute by multiplying with an exponential factor.

In a data set consisting of measurements from petroleum production, we find it reasonable to assume that recent measurements describe well performance more accurately than old ones. Thus, we discount with respect to the time stamp of each data point. The factor by which we discount a point is given by $e^{-\alpha \frac{\bar{t}-t_k}{\Delta t}}$, where α is a user specified parameter and the fraction $\frac{\bar{t}-t_k}{\Delta t}$ is the relative age of data point k with time stamp t_k compared to the newest data point with time stamp $\bar{t}$. The fraction takes values from 0 to 1 due to the divisor Δt, which represents the time difference between the oldest and newest data point. The parameter α controls the rate at which points are discounted. That is, the higher the value of α, the less weight we place on old data points.

Algorithm 3: Algorithm for consolidating data points using a discount rule.

Input : A target phase p, either oil or gas.
A set $\mathcal{D} = \{(X_1, Y_1), \dots, (X_N, Y_N)\}$ of N data rows. $X_n = \{c_n, t_n\}$ where c_n is the choke level and t_n is the start time of well test n. $Y_n = \{p_n\}$ where p_n is the target output rate of well test n.
A discounting factor α.
A number of intervals I and J for the input and output dimensions, respectively.

Output: A consolidated data set $\hat{\mathcal{D}} = \{(\hat{X}_1, \hat{Y}_1), \dots, (\hat{X}_M, \hat{Y}_M)\}$ of $M \leq N$ data rows.

$\hat{\mathcal{D}} \leftarrow \emptyset$
$c^* \leftarrow \frac{max_{n=1\dots N}(c_n) - min_{n=1\dots N}(c_n)}{I}$
$p^* \leftarrow \frac{max_{n=1\dots N}(p_n) - min_{n=1\dots N}(p_n)}{J}$
for $i \leftarrow 1$ **to** I **do**
  $\overline{\mathcal{D}}^i \subseteq \mathcal{D} \leftarrow \{(X_k, Y_k) \in \mathcal{D} \mid ic^* \leq c_k < (i+1)c^*\}$;
  for $j \leftarrow 1$ **to** J **do**
    $\overline{\mathcal{D}}^{ij} \subseteq \overline{\mathcal{D}}^i \leftarrow \{(X_k, Y_k) \in \overline{\mathcal{D}}^i \mid jp^* \leq p_k < (j+1)p^*\}$;
    if $|\overline{\mathcal{D}}^{ij}| \geq 2$ **then**
      $\bar{t} \leftarrow max_{(X_k,Y_k)\in\overline{\mathcal{D}}^{ij}}(t_k)$;
      $\underline{t} \leftarrow min_{(X_k,Y_k)\in\overline{\mathcal{D}}^{ij}}(t_k)$;
      $\Delta t \leftarrow \bar{t} - \underline{t}$;
      $\lambda^{SUM} \leftarrow 0$;
      $\overline{c}^{ij}, \overline{t}^{ij}, \overline{p}^{ij} \leftarrow 0$;
      for $(X_k, Y_k) \in \overline{\mathcal{D}}^{ij}$ **do**
        $\lambda_k \leftarrow e^{-\alpha\frac{\bar{t}-t_k}{\Delta t}}$;
        $\lambda^{SUM} \leftarrow \lambda^{SUM} + \lambda_k$;
        $\overline{c}^{ij} \leftarrow \overline{c}^{ij} + \lambda_k c_k$;
        $\overline{t}^{ij} \leftarrow \overline{t}^{ij} + \lambda_k t_k$;
        $\overline{p}^{ij} \leftarrow \overline{p}^{ij} + \lambda_k p_k$;
      end
      $\overline{X}^{ij} \leftarrow \{\frac{\overline{c}^{ij}}{\lambda^{SUM}}, \frac{\overline{t}^{ij}}{\lambda^{SUM}}\}$;
      $\overline{Y}^{ij} \leftarrow \{\frac{\overline{p}^{ij}}{\lambda^{SUM}}\}$;
      $\hat{\mathcal{D}} \leftarrow \hat{\mathcal{D}} \cup \{(\overline{X}^{ij}, \overline{Y}^{ij})\}$;
    else if $|\overline{\mathcal{D}}^{ij}| = 1$ **then**
      $\hat{\mathcal{D}} \leftarrow \hat{\mathcal{D}} \cup \overline{\mathcal{D}}^{ij}$;
  end
end

Figure A.1: Discounting algorithm

The discounting algorithm, shown in Figure A.1, works as follows. The input domain space is first divided into a user specified number of intervals of equal length in line 2. The algorithm then loops through each of the intervals. The set of points belonging to the current interval is defined in line 4. If an interval contains more than one data point, a single point replaces the original points in the interval. The discount factor of each original point is calculated in line 12. The replacing point is then constructed by taking the discount weighted mean of the points within the interval in lines $19 - 20$. Note that setting $\alpha = 0$ amounts to taking the mean of all data points in each interval, whereas setting $\alpha > 10$ effectively amounts to exclusively weighting the newest data point.

A.2 Hyperparameter Search

This section lists the top 3 hyperparameter results for grid and randomized search, for each well and phase.

Table A.1: Hyperparameter grid search results for oil output.

			Hyperparameters			
Well	**Rank**	**Avg. Test Score**	λ	p	L	N
W1	1	-6.35	1.00×10^{-2}	0.05	2	20
	2	-7.76	1.00×10^{-5}	0.05	2	20
	3	-11.54	1.00×10^{-6}	0.05	2	20
W2	1	-4.67	1.00×10^{-4}	0.05	2	40
	2	-4.68	1.00×10^{-5}	0.05	1	40
	3	-4.69	1.00×10^{-6}	0.05	1	40
W3	1	-10.94	1.00×10^{-5}	0.15	2	20
	2	-12.67	1.00×10^{-5}	0.05	1	20
	3	-13.89	1.00×10^{-3}	0.05	1	20
W4	1	-5.88	1.00×10^{-2}	0.05	2	20
	2	-5.94	1.00×10^{-3}	0.05	2	20
	3	-5.96	1.00×10^{-3}	0.15	2	40
W5	1	-7.71	1.00×10^{-2}	0.05	2	20
	2	-14.27	1.00×10^{-2}	0.05	2	20
	3	-16.88	1.00×10^{-6}	0.15	2	20
W6	1	-7.78	1.00×10^{-2}	0.05	2	20
	2	-8.61	1.00×10^{-5}	0.05	2	20
	3	-21.08	1.00×10^{-4}	0.05	2	20
W7	1	-8.71	1.00×10^{-3}	0.05	2	20
	2	-15.90	1.00×10^{-2}	0.05	2	20
	3	-17.34	1.00×10^{-3}	0.25	1	20

A.3 Well Model Plots

This section contains figures of fully trained well models for all wells in the case of this thesis.

Table A.2: Hyperparameter randomized search results for oil output.

			Hyperparameters			
Well	**Rank**	**Avg. Test Score**	λ	p	L	N
W1	1	-6.22	9.33×10^{-6}	0.014	2	37
	2	-8.61	6.62×10^{-5}	0.035	1	20
	3	-10.19	7.52×10^{-5}	0.018	2	19
W2	1	-4.69	6.93×10^{-5}	0.026	2	31
	2	-4.73	2.81×10^{-5}	0.041	1	25
	3	-4.74	5.87×10^{-5}	0.042	2	35
W3	1	-8.72	1.48×10^{-5}	0.26	2	7
	2	-11.54	8.49×10^{-5}	0.055	2	10
	3	-11.58	7.05×10^{-5}	0.37	1	11
W4	1	-5.96	3.73×10^{-5}	0.067	2	10
	2	-5.97	5.56×10^{-5}	0.15	1	5
	3	-6.17	7.08×10^{-5}	0.36	1	12
W5	1	-7.08	8.20×10^{-5}	0.060	1	9
	2	-8.99	2.34×10^{-5}	0.33	1	5
	3	-9.69	9.50×10^{-5}	0.074	2	5
W6	1	-7.36	1.86×10^{-5}	0.12	2	6
	2	-7.93	4.36×10^{-5}	0.087	2	8
	3	-8.44	5.16×10^{-5}	0.25	2	5
W7	1	-7.64	9.17×10^{-5}	0.11	2	9
	2	-7.92	1.67×10^{-5}	0.14	1	7
	3	-8.47	5.28×10^{-5}	0.018	2	11

Table A.3: Hyperparameter grid search results for gas output.

			Hyperparameters			
Well	**Rank**	**Avg. Test Score**	λ	p	L	N
W1	1	-4.66	1.00×10^{-6}	0.05	2	40
	2	-5.14	1.00×10^{-5}	0.05	1	20
	3	-5.26	1.00×10^{-4}	0.05	1	20
W2	1	-5.57	1.00×10^{-6}	0.05	2	40
	2	-6.16	1.00×10^{-5}	0.05	2	40
	3	-6.32	1.00×10^{-6}	0.05	2	20
W3	1	-6.58	1.00×10^{-4}	0.05	2	40
	2	-6.65	1.00×10^{-6}	0.05	2	40
	3	-6.75	1.00×10^{-5}	0.05	2	40
W4	1	-7.16	1.00×10^{-6}	0.05	2	40
	2	-8.32	1.00×10^{-5}	0.05	1	20
	3	-8.33	1.00×10^{-4}	0.05	1	20
W5	1	-7.24	1.00×10^{-6}	0.05	2	40
	2	-8.37	1.00×10^{-5}	0.05	1	20
	3	-8.65	1.00×10^{-4}	0.05	1	20
W6	1	-6.76	1.00×10^{-6}	0.05	2	40
	2	-7.65	1.00×10^{-5}	0.05	1	20
	3	-7.63	1.00×10^{-4}	0.05	1	20
W7	1	-6.65	1.00×10^{-5}	0.05	2	40
	2	-7.59	1.00×10^{-6}	0.05	2	40
	3	-7.59	1.00×10^{-4}	0.05	2	40

Table A.4: Hyperparameter randomized search results for gas output.

			Hyperparameters			
Well	**Rank**	**Avg. Test Score**	λ	p	L	N
W1	1	-4.71	1.50×10^{-5}	0.033	2	39
	2	-4.71	4.40×10^{-5}	0.14	2	39
	3	-4.73	9.25×10^{-5}	0.033	2	33
W2	1	-5.79	6.34×10^{-5}	0.013	2	30
	2	-6.29	1.60×10^{-5}	0.061	2	26
	3	-6.35	6.16×10^{-5}	0.14	2	28
W3	1	-6.75	2.98×10^{-5}	0.063	2	28
	2	-7.70	5.12×10^{-5}	0.034	1	31
	3	-7.73	5.41×10^{-5}	0.39	2	17
W4	1	-7.27	4.26×10^{-5}	0.035	2	33
	2	-7.63	2.81×10^{-5}	0.059	2	23
	3	-8.35	2.91×10^{-5}	0.25	1	29
W5	1	-7.11	7.18×10^{-5}	0.037	2	36
	2	-9.23	7.66×10^{-5}	0.11	1	6
	3	-9.07	4.52×10^{-5}	0.17	1	13
W6	1	-6.82	7.93×10^{-5}	0.012	2	29
	2	-7.36	4.38×10^{-5}	0.10	2	30
	3	-7.68	9.45×10^{-5}	0.15	1	37
W7	1	-7.45	6.13×10^{-5}	0.079	2	35
	2	-7.52	8.11×10^{-5}	0.068	2	31
	3	-7.60	8.69×10^{-5}	0.065	2	30

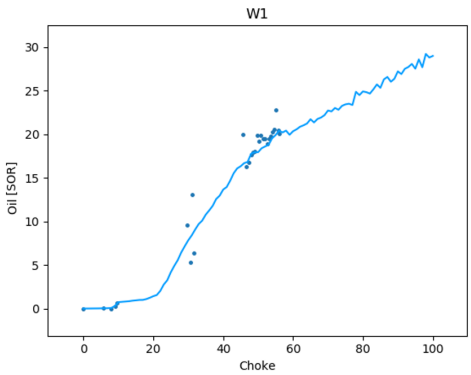

Figure A.2: Well model for oil production of well W1.

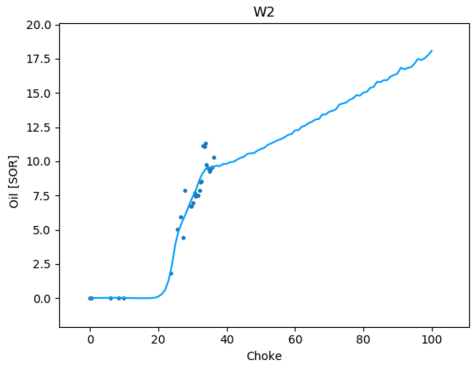

Figure A.3: Well model for oil production of well W2.

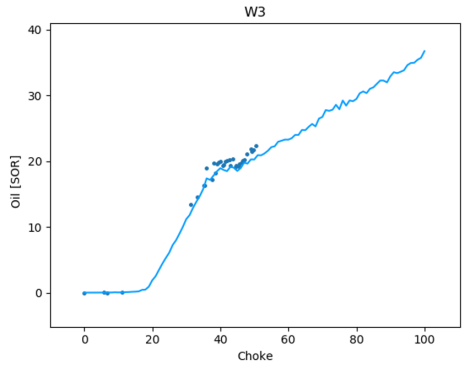

Figure A.4: Well model for oil production of well W3.

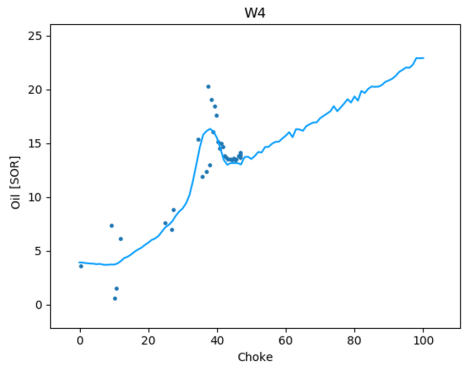

Figure A.5: Well model for oil production of well W4.

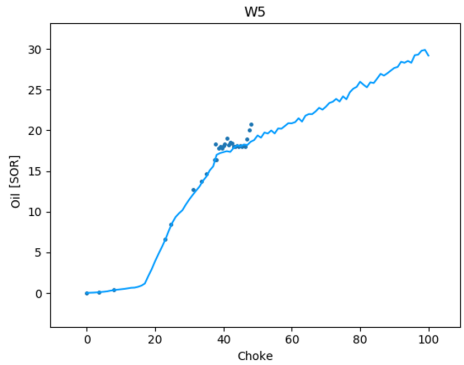

Figure A.6: Well model for oil production of well W5.

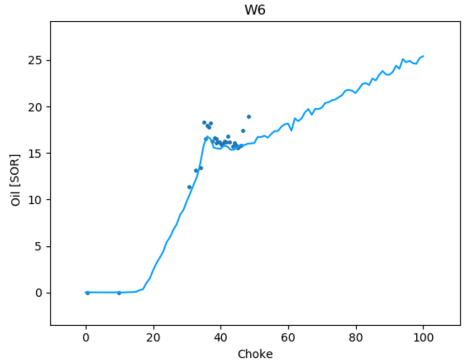

Figure A.7: Well model for oil production of well W6.

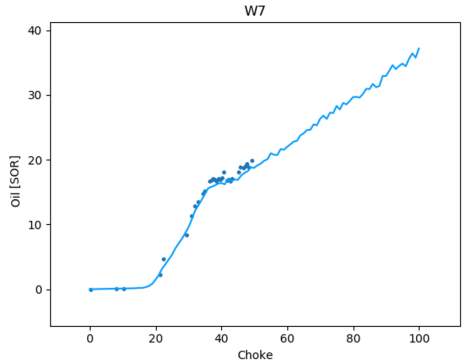

Figure A.8: Well model for oil production of well W7.

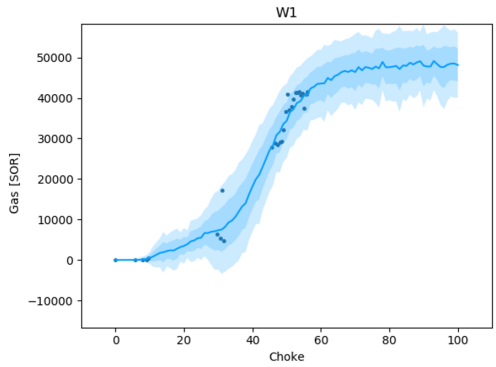

Figure A.9: Well model for gas production of well W1.

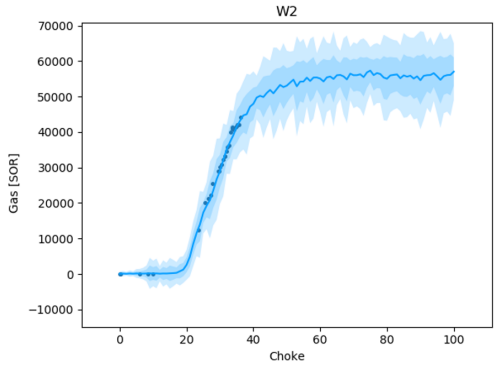

Figure A.10: Well model for gas production of well W2.

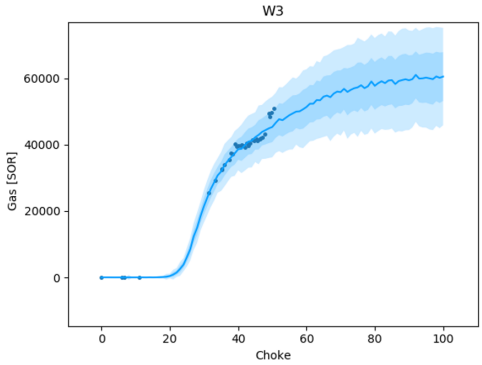

Figure A.11: Well model for gas production of well W3.

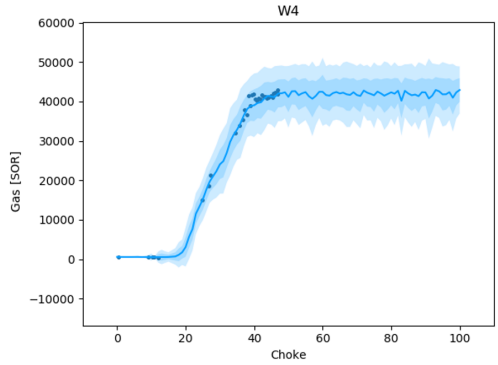

Figure A.12: Well model for gas production of well W4.

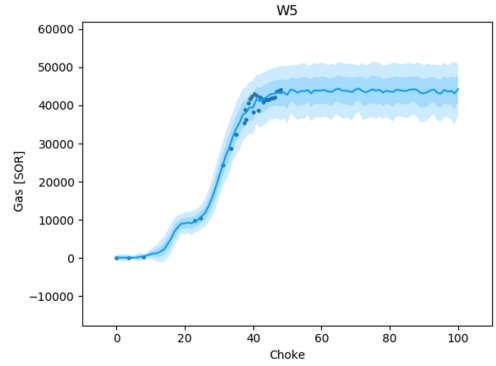

Figure A.13: Well model for gas production of well W5.

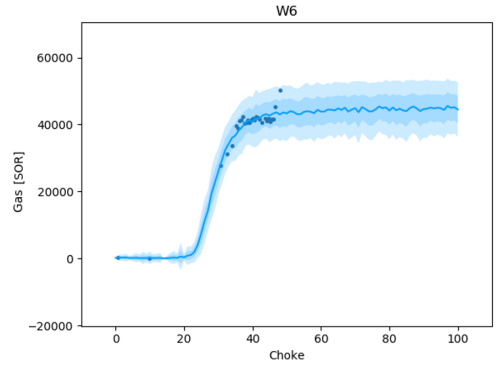

Figure A.14: Well model for gas production of well W6.

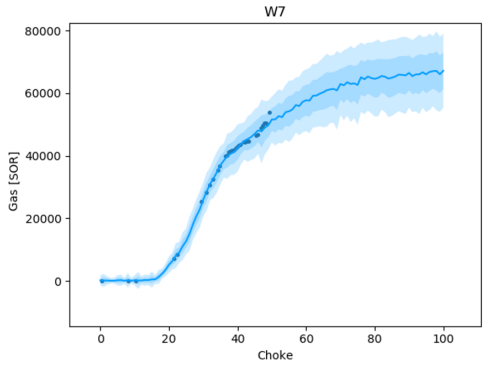

Figure A.15: Well model for gas production of well W7.

A.4 Technical Study

The following table contains the total training time for a NN for well W1 as the grid fidelity of the discounting algorithm varies.

Table A.5: Training times for different discounting parameters. W1, trained for 40,000 epochs.

Sections x-axis	Sections y-axis	Data points after discount	Seconds spent training
25	1	11	9.71
50	1	17	9.63
100	1	28	9.91
25	10	28	10.17
50	10	40	10.32
100	10	58	18.62
25	50	65	18.62
50	50	89	19.76
100	50	121	29.57
25	100	97	19.84
50	100	131	30.21
100	100	170	41.32

Appendix B

Stochastic Optimization Appendix

In this appendix, supplementary content regarding the stochastic robust model and the RA is supplied.

B.1 Stochastic Robust Optimization Model

In this Section, full results from implementing changes simultaneously with the stochastic robust model are supplied.

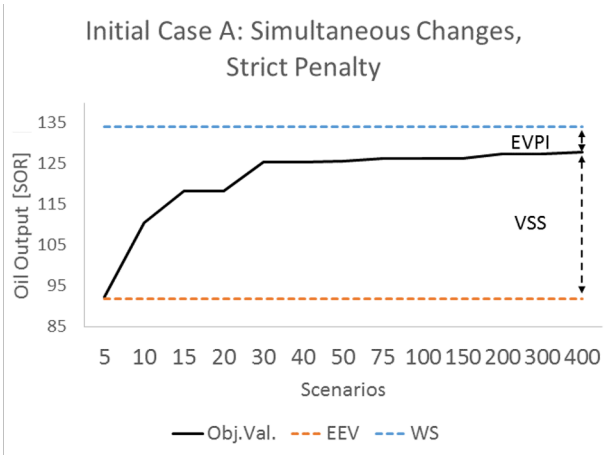

Figure B.1: VSS and EVPI for a number of scenario sizes for initial case A with Strict penalty and simultaneous implementation of changes.

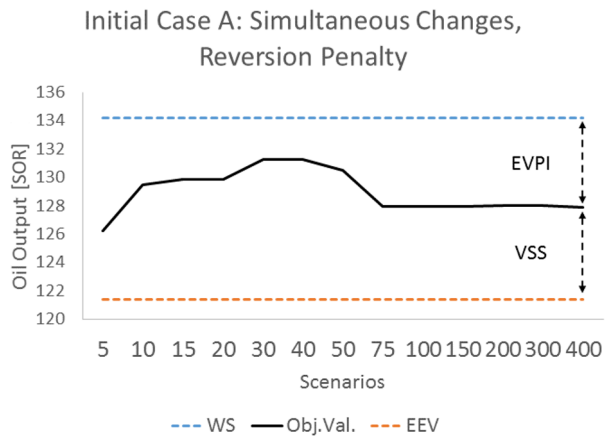

Figure B.2: VSS and EVPI for a number of scenario sizes for initial case A with Reversion penalty and simultaneous implementation of changes.

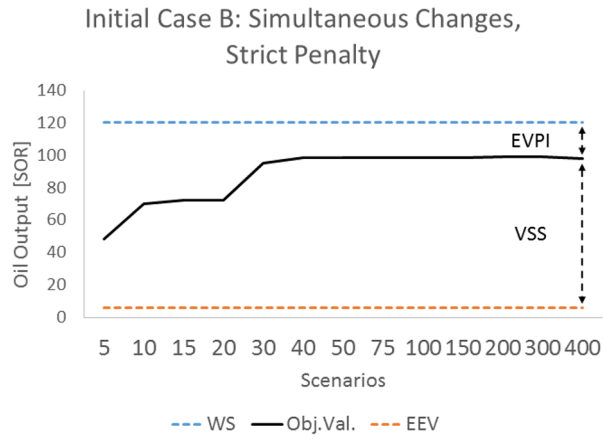

Figure B.3: VSS and EVPI for a number of scenario sizes for initial case B with Strict penalty and simultaneous implementation of changes.

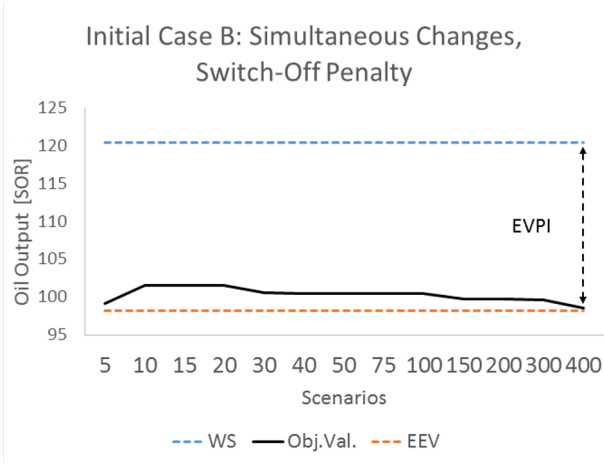

Figure B.4: VSS and EVPI for a number of scenario sizes for initial case B with Switch-Off penalty and simultaneous implementation of changes.

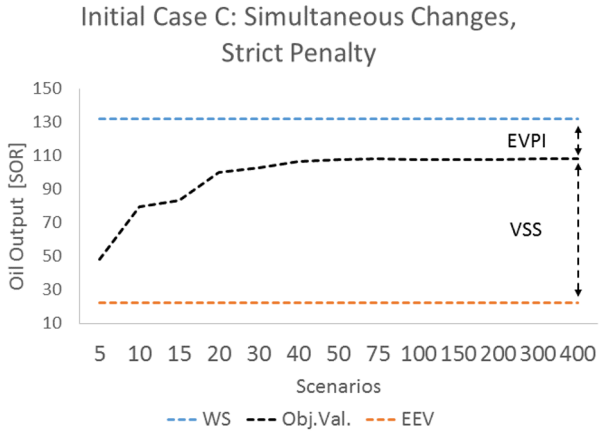

Figure B.5: VSS and EVPI for a number of scenario sizes for initial case C with Strict penalty and simultaneous implementation of changes.

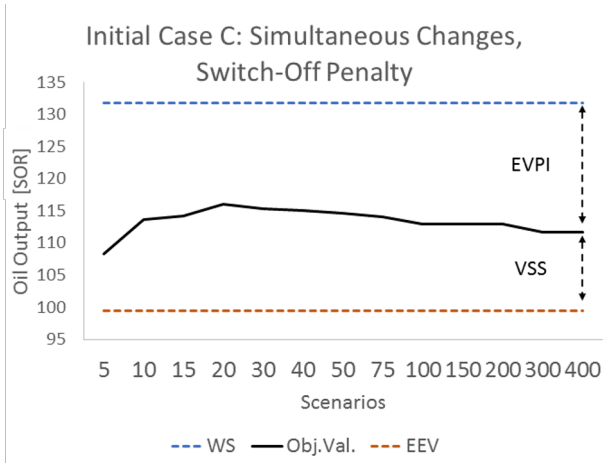

Figure B.6: VSS and EVPI for a number of scenario sizes for initial case C with Switch-Off penalty and simultaneous implementation of changes.

Table B.1: Initial case A: The EEV, WS and stochastic optimization model results using Markov Weighted scenarios.

		Average Objective Value		Final Well Choke Settings						
Model/Scenarios	**# Infeasible**	**Strict Penalty**	**Reversion Penalty**	**W1**	**W2**	**W3**	**W4**	**W5**	**W6**	**W7**
EEV	56	91.82	123.71	72.98	0.00	65.71	0.00	57.48	62.60	50.34
WS	0	134.18	134.18	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Initial State	N/A	105.66		72.98	47.07	0.00	0.00	47.84	62.6	50.34
5	64	92.41	126.22	72.98	0.00	65.71	20.00	53.23	62.61	27.65
10	36	110.46	129.48	72.98	0.00	65.71	30.82	47.84	62.60	50.34
15	22	118.23	129.86	72.98	0.00	62.39	31.35	47.84	62.60	50.34
20	22	118.23	129.86	72.98	0.00	62.39	31.35	47.84	62.60	50.34
30	11	125.44	131.25	72.98	0.00	65.71	26.60	47.84	62.60	50.34
40	11	125.44	131.25	72.98	0.00	65.71	26.60	47.84	62.60	50.34
50	9	125.74	130.50	72.98	0.00	65.71	22.83	47.84	62.60	50.34
75	3	126.36	127.94	72.98	0.00	54.63	31.09	47.84	62.60	50.34
100	3	126.36	127.94	72,98	0.00	54.63	31.09	47.84	62.60	50.34
200	1	127.49	128.02	72.98	0.00	41.09	38.64	47.84	62.60	50.34
300	1	127.49	128.02	72.98	0.00	41.09	38.64	47.84	62.60	50.34
400	0	127.86	127.86	72.98	0.00	43.36	37.41	47.84	62.60	50.34

Table B.2: Initial Case A: The EEV, WS and stochastic optimization model results using Factor scenarios.

		Average Objective Value		Final Well Choke Settings						
Model/Scenarios	**# Infeasible**	**Strict Penalty**	**Reversion Penalty**	**W1**	**W2**	**W3**	**W4**	**W5**	**W6**	**W7**
EEV	37	101.15	122.22	72.98	0.00	65.71	0.00	57.48	62.60	54.86
WS	0	134.18	134.18	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Initial State	N/A	105.66		72.98	47.07	0.00	0.00	47.84	62.6	50.34
5	0	116.16	116.16	72.98	0.00	35.84	19.61	47.84	62.60	50.34
10	0	116.16	116.16	72.98	0.00	35.84	19.61	47.84	62.60	50.34
15	0	116.16	116.16	72.98	0.00	35.84	19.61	47.84	62.60	50.34
20	0	115.88	115.88	72.98	0.00	34.71	20.66	47.84	62.60	50.34
30	0	115.88	115.88	72.98	0.00	34.71	20.66	47.84	62.60	50.34
40	0	115.78	115.78	72.98	0.00	34.32	21.01	47.84	62.60	50.34
50	0	115.78	115.78	72.98	0.00	34.32	21.01	47.84	62.60	50.34
75	0	115.59	115.59	72.98	0.00	33.52	21.78	47.84	62.60	50.34
100	0	115.43	115.43	72.98	0.00	32.44	23.02	47.84	62.60	50.34
150	0	115.43	115.43	72.98	0.00	32.44	23.02	47.84	62.60	50.34
200	0	115.43	115.43	72.98	0.00	32.44	23.02	47.84	62.60	50.34
300	0	109.08	109.08	72.98	29.69	0.00	31.03	47.84	62.60	39.00
400	0	108.57	108.57	72.98	28.91	0.00	31.03	47.84	62.60	39.00

Table B.3: Initial Case B: The EEV, WS and stochastic optimization model results using Markov Weighted scenarios.

		Average Objective Value		Final Well Choke Settings						
Model/Scenarios	**# Infeasible**	**Strict Penalty**	**Switch-Off Penalty**	**W1**	**W2**	**W3**	**W4**	**W5**	**W6**	**W7**
EEV	190	6.00	98.21	72.98	0.00	65.71	20	53.23	62.61	27.65
WS	0	120.37	120.37	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Initial State	N/A	124.23		72.98	0.00	40.00	22.52	53.23	62.61	53.43
5	113	48.18	99.18	72.98	0.00	62.16	30.00	13.31	62.61	53.43
10	68	69.94	101.60	72.98	0.00	65.71	22.52	52.84	62.61	13.36
15	62	72.57	101.52	72.98	0.00	65.05	20.00	53.23	62.61	13.36
20	62	72.57	101.52	72.98	0.00	65.05	20.00	53.23	62.61	13.36
30	14	94.97	100.58	72.98	0.00	58.32	20.00	53.23	62.61	13.95
40	4	98.73	100.50	72.98	0.00	40.00	27.73	54.12	62.61	13.36
50	4	98.73	100.50	72.98	0.00	40.00	27.73	54.12	62.61	13.36
75	4	98.73	100.50	72.98	0.00	40.00	27.73	54.12	62.61	13.36
100	4	98.73	100.50	72.98	0.00	40.00	27.73	54.12	62.61	13.36
150	3	98.38	99.79	72.98	0.00	40.00	28.03	52.77	62.61	13.36
200	1	99.27	99.74	72.98	0.00	40.46	27.18	53.23	62.61	13.36
300	1	99.11	99.58	72.98	0.00	40.62	26.86	53.23	62.61	13.36
400	1	97.81	98.61	72.98	0.00	40.62	26.86	53.23	62.61	13.36

Table B.4: Initial Case B: The EEV, WS and stochastic optimization model results using Factor scenarios.

		Average Objective Value		Final Well Choke Settings						
Model/Scenarios	**# Infeasible**	**Strict Penalty**	**Switch-Off Penalty**	**W1**	**W2**	**W3**	**W4**	**W5**	**W6**	**W7**
EEV	200	0	100.93	72.98	0.00	40.00	13.32	54.27	62.61	35.27
WS	0	120.37	120.37	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Initial State	N/A	124.23		72.98	0.00	40.00	22.52	53.23	62.61	53.43
5	200	0	100.93	72.98	0.00	40.00	13.32	54.27	62.61	35.27
10	200	0	100.93	72.98	0.00	40.00	13.32	54.27	62.61	35.27
15	200	0	100.93	72.98	0.00	40.00	13.32	54.27	62.61	35.27
20	200	0	100.93	72.98	0.00	40.00	13.32	54.27	62.61	35.27
30	200	0	100.93	72.98	0.00	40.00	13.32	54.27	62.61	35.27
40	200	0	100.93	72.98	0.00	40.00	13.32	54.27	62.61	35.27
50	200	0	100.93	72.98	0.00	40.00	13.32	54.27	62.61	35.27
75	200	0	100.93	72.98	0.00	40.00	13.32	54.27	62.61	35.27
100	200	0	100.93	72.98	0.00	40.00	13.32	54.27	62.61	35.27
150	200	0	100.93	72.98	0.00	40.00	13.32	54.27	62.61	35.27
200	200	0	100.93	72.98	0.00	40.00	13.32	54.27	62.61	35.27
300	200	0	100.93	72.98	0.00	40.00	13.32	54.27	62.61	35.27
400	200	0	100.93	72.98	0.00	40.00	13.32	54.27	62.61	35.27

Table B.5: Initial Case C: The EEV, WS and stochastic optimization model results using Markov Weighted scenarios.

		Average Objective Value		Final Well Choke Settings						
Model/Scenarios	**# Infeasible**	**Strict Penalty**	**Switch-Off Penalty**	**W1**	**W2**	**W3**	**W4**	**W5**	**W6**	**W7**
EEV	166	22.55	99.43	72.98	0.00	65.71	20	53.23	62.61	27.65
WS	0	131.86	131.86	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Initial State	N/A	0.00		0.00	0.00	00.00	0.00	0.00	0.00	0.00
5	122	48.28	108.27	72.98	0	40.00	24.24	50.87	62.61	53.43
10	69	79.75	113.67	72.98	0	40.00	26.12	50.87	57.58	50.76
15	63	83.08	114.22	72.98	0	40.00	27.24	49.01	57.58	50.76
20	32	100.31	116.08	72.98	0	40.00	33.16	48.29	57.58	37.61
30	25	102.93	115.35	72.98	0	40.00	30.70	43.54	56.51	43.21
40	17	106.64	115.01	72.98	0	42.23	29.91	43.54	54.28	42.62
50	14	107.86	114.70	72.98	0	40.00	29.77	43.54	54.28	41.86
75	12	108.22	114.09	72.98	0	40.00	30.56	43.54	51.13	41.92
100	11	107.63	112.94	72.98	0	40.00	34.23	35.78	51.13	46.22
150	11	107.61	112.91	72.98	0	40.00	34.13	35.93	51.13	45.95
200	11	107.61	112.91	72.98	0	40.00	34.13	35.93	51.13	45.95
300	7	108.31	111.64	72.98	0	40.00	30.04	36.86	51.13	45.38
400	7	108.31	111.64	72.98	0	40.00	30.04	36.86	51.13	45.38

Table B.6: Initial Case C: The EEV, WS and stochastic optimization model results using Factor scenarios.

		Average Objective Value		Final Well Choke Settings						
Model/Scenarios	**# Infeasible**	**Strict Penalty**	**Switch-Off Penalty**	**W1**	**W2**	**W3**	**W4**	**W5**	**W6**	**W7**
EEV	169	19.52	93.47	72.98	0.00	65.71	20	53.23	62.61	27.65
WS	0	131.86	131.86	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Initial State	N/A	0.00		0.00	0.00	00.00	0.00	0.00	0.00	0.00
5	99	61.82	111.24	72.98	0.00	56.24	31.03	42.56	62.1	38.37
10	80	70.86	110.67	72.98	0.00	40.40	39.80	34.79	53.51	50.34
15	2	110.90	111.85	72.98	0.00	40.40	39.80	33.93	53.51	35.27
20	2	110.90	111.85	72.98	0.00	40.40	39.80	33.93	53.51	35.27
30	2	110.90	111.85	72.98	0.00	40.40	39.80	33.93	53.51	35.27
40	1	106.70	107.13	72.98	0.00	36.31	39.80	29.69	53.51	35.27
50	1	106.65	107.08	72.98	0.00	36.31	39.80	29.75	53.29	35.27
75	0	104.25	104.25	72.98	0.00	34.89	32.82	36.04	48.1	35.52
100	0	103.64	103.64	72.98	0.00	34.89	33.32	36.04	45.93	35.52
150	0	103.16	103.16	72.98	0.00	33.91	35.44	35.00	45.21	35.52
200	0	103.16	103.16	72.98	0.00	33.91	35.44	35.00	45.21	35.52
300	0	102.15	102.15	72.98	0.00	33.91	33.53	35.00	44.97	35.52
400	0	100.24	100.24	72.98	0.00	33.91	31.03	33.93	44.97	35.27

B.2 Recourse Algorithm

In this section, full results for the RA are supplied.

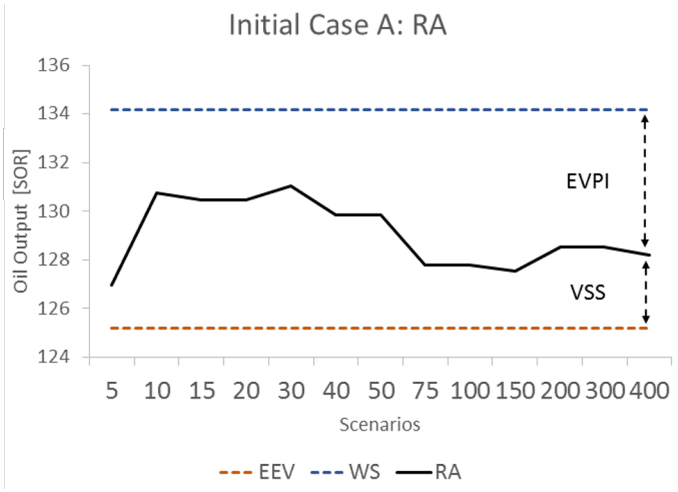

Figure B.7: VSS and EVPI of Recourse Algorithm for a number of scenario sizes for initial case A.

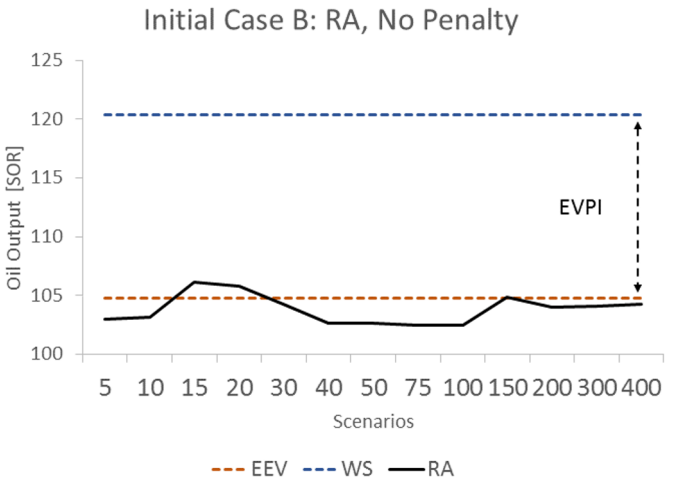

Figure B.8: VSS and EVPI of Recourse Algorithm for a number of scenario sizes for initial case B.

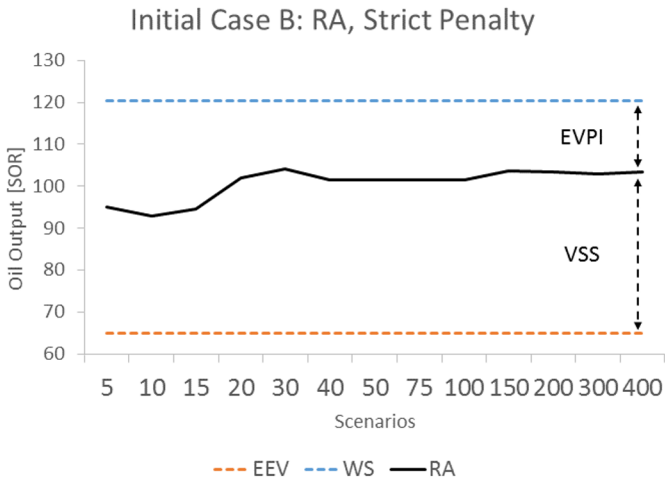

Figure B.9: VSS and EVPI of Recourse Algorithm for a number of scenario sizes for initial case B with Strict penalty.

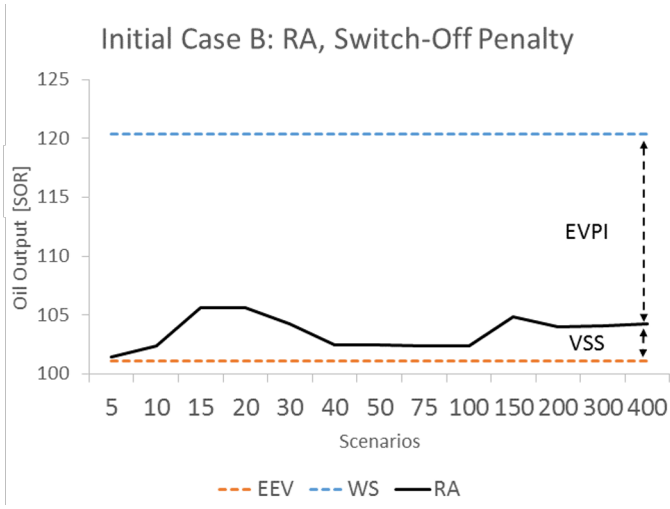

Figure B.10: VSS and EVPI of Recourse Algorithm for a number of scenario sizes for initial case B with Switch-Off penalty.

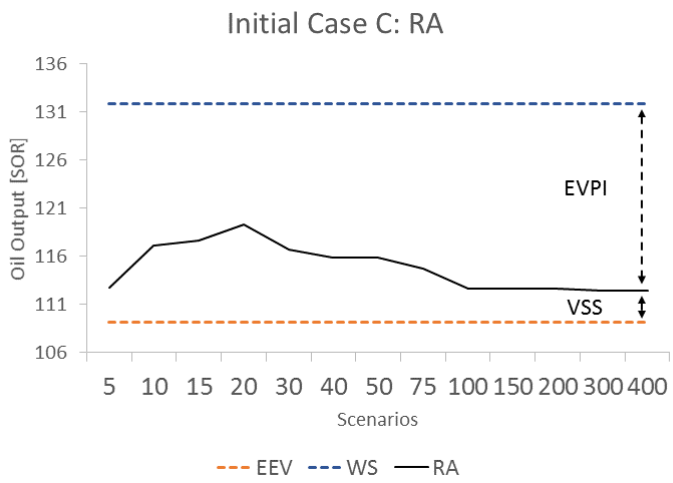

Figure B.11: VSS and EVPI of Recourse Algorithm for a number of scenario sizes for initial case C.

Table B.7: Recourse algorithm results for Markov Weighted scenarios.

	Initial Case A: Under Cap.		Initial Case B: Over Cap.		Initial Case C: Zero	
Model/Scenarios	**# Infeasible Moves**	**Avg. Objective Value**	**# End States Infeasible**	**Avg. Objective Value**	**# Infeasible Moves**	**Avg. Objective Value**
EEV	56	125.18	78	104.77	183	109.20
WS	0	134.18	0	120.37	0	131.86
5	60	126.96	15	102.94	119	112.79
10	12	130.75	19	103.11	63	117.15
15	12	130.47	22	106.10	46	117.64
20	12	130.47	7	105.83	27	119.35
30	11	131.05	0	104.25	15	116.69
40	8	129.83	2	102.63	13	115.88
50	8	129.83	2	102.61	13	115.88
75	2	127.79	2	102.47	11	114.76
100	2	127.79	2	102.47	6	112.62
150	2	127.53	2	104.86	6	112.65
200	2	128.53	1	104.03	6	112.66
300	2	128.53	2	104.12	5	112.39
400	2	128.20	1	104.29	4	112.49

Table B.8: Penalized recourse algorithm results for Markov Weighted scenarios, case B.

Model/Scenarios	**# End States Infeasible**	**Avg. Objective Value**	**Strict Penalty**	**Switch-Off Penalty**
EEV	78	104.77	64.9	101.14
WS	0	120.37	N/A	N/A
5	15	102.94	95.15	101.47
10	19	103.11	92.83	102.42
15	22	106.10	94.47	105.60
20	7	105.83	102.10	105.66
30	0	104.25	104.25	104.25
40	2	102.63	101.62	102.51
50	2	102.61	101.60	102.49
75	2	102.47	101.46	102.34
100	2	102.47	101.46	102.35
150	2	104.86	103.77	104.86
200	1	104.03	103.48	104.03
300	2	104.12	103.04	104.09
400	1	104.29	103.52	104.27

Appendix C

MOP Appendix

This appendix provides full results from solving the MOP for all initial cases.

In the following tables, *p(f)* represents P(Feasibility), *exp oil* represents the oil output in SOR in the corresponding figures and *alpha* denotes the fraction $\frac{j}{N}$ in Equation 8.24, i.e., the fraction of the optimal oil output for ε_j.

C.1 Case A

Table C.1: Data for Pareto front for case A, as shown in Figure 10.14

	1 change		2 changes		3 changes		4 changes		5 changes	
alpha	**exp oil**	**p(f)**	**exp oil**	**p(f)**	**exp oil**	**p(f)**	**exp oil**	**p(f)**	**exp oil**	**p(f)**
0.01	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	7.63	100 %
0.02	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	7.63	100 %
0.03	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	7.63	100 %
0.04	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	7.63	100 %
0.05	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	7.63	100 %
0.06	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	10.80	100 %
0.07	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	10.80	100 %
0.08	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	11.25	100 %
0.09	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	15.43	100 %
0.10	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	15.43	100 %
0.11	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	18.42	100 %
0.12	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	18.42	100 %
0.13	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	18.42	100 %
0.14	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	19.68	100 %
0.15	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	23.05	100 %
0.16	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	23.05	100 %
0.17	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	23.90	100 %
0.18	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	25.31	100 %
0.19	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	26.71	100 %
0.20	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	28.12	100 %
0.21	85.94	100 %	65.47	100 %	40.72	100 %	36.38	100 %	35.30	100 %
0.22	85.94	100 %	65.47	100 %	40.72	100 %	36.38	100 %	35.30	100 %
0.23	85.94	100 %	65.47	100 %	40.72	100 %	36.38	100 %	35.30	100 %
0.24	85.94	100 %	65.47	100 %	40.72	100 %	36.38	100 %	35.30	100 %

Table C.1 continued from previous page

	1 change		2 changes		3 changes		4 changes		5 changes	
alpha	**exp oil**	**p(f)**	**exp oil**	**p(f)**	**exp oil**	**p(f)**	**exp oil**	**p(f)**	**exp oil**	**p(f)**
0.25	85.94	100 %	65.47	100 %	40.72	100 %	36.38	100 %	35.30	100 %
0.26	85.94	100 %	65.47	100 %	40.72	100 %	39.55	100 %	36.55	100 %
0.27	85.94	100 %	65.47	100 %	40.72	100 %	39.55	100 %	37.96	100 %
0.28	85.94	100 %	65.47	100 %	40.72	100 %	39.55	100 %	39.36	100 %
0.29	85.94	100 %	65.47	100 %	40.72	100 %	40.64	100 %	40.77	100 %
0.30	85.94	100 %	65.47	100 %	53.51	100 %	47.18	100 %	42.18	100 %
0.31	85.94	100 %	65.47	100 %	53.51	100 %	47.18	100 %	46.10	100 %
0.32	85.94	100 %	65.47	100 %	53.51	100 %	47.18	100 %	46.10	100 %
0.33	85.94	100 %	65.47	100 %	53.51	100 %	47.18	100 %	46.39	100 %
0.34	85.94	100 %	65.47	100 %	53.51	100 %	47.64	100 %	47.80	100 %
0.35	85.94	100 %	65.47	100 %	53.51	100 %	49.04	100 %	49.20	100 %
0.36	85.94	100 %	65.47	100 %	53.51	100 %	50.45	100 %	50.61	100 %
0.37	85.94	100 %	65.47	100 %	53.51	100 %	51.85	100 %	52.02	100 %
0.38	85.94	100 %	65.47	100 %	53.51	100 %	53.25	100 %	53.42	100 %
0.39	85.94	100 %	65.47	100 %	61.13	100 %	54.65	100 %	54.83	100 %
0.40	85.94	100 %	65.47	100 %	61.13	100 %	56.05	100 %	56.23	100 %
0.41	85.94	100 %	65.47	100 %	61.13	100 %	57.45	100 %	57.64	100 %
0.42	85.94	100 %	65.47	100 %	61.13	100 %	60.05	100 %	59.05	100 %
0.43	85.94	100 %	65.47	100 %	61.13	100 %	60.25	100 %	60.45	100 %
0.44	85.94	100 %	65.47	100 %	61.13	100 %	61.66	100 %	61.86	100 %
0.45	85.94	100 %	65.47	100 %	64.31	100 %	63.06	100 %	63.26	100 %
0.46	85.94	100 %	65.47	100 %	64.31	100 %	64.46	100 %	64.67	100 %
0.47	85.94	100 %	65.47	100 %	64.93	100 %	65.86	100 %	66.07	100 %
0.48	85.94	100 %	73.97	100 %	71.93	100 %	67.26	100 %	67.48	100 %
0.49	85.94	100 %	73.97	100 %	71.93	100 %	68.66	100 %	68.89	100 %
0.50	85.94	100 %	73.97	100 %	71.93	100 %	70.85	100 %	70.29	100 %
0.51	85.94	100 %	73.97	100 %	71.93	100 %	71.46	100 %	71.70	100 %
0.52	85.94	100 %	73.97	100 %	71.93	100 %	72.87	100 %	73.10	100 %
0.53	85.94	100 %	73.97	100 %	73.22	100 %	74.27	100 %	74.51	100 %
0.54	85.94	100 %	73.97	100 %	74.60	100 %	75.67	100 %	75.92	100 %
0.55	85.94	100 %	81.60	100 %	75.99	100 %	77.07	100 %	77.32	100 %
0.56	85.94	100 %	81.60	100 %	77.37	100 %	78.47	100 %	78.73	100 %
0.57	85.94	100 %	81.60	100 %	78.75	100 %	79.87	100 %	80.13	100 %
0.58	85.94	100 %	81.60	100 %	80.13	100 %	81.27	100 %	81.54	100 %
0.59	85.94	100 %	81.60	100 %	81.68	100 %	82.68	100 %	82.94	100 %
0.60	85.94	100 %	83.89	100 %	82.89	100 %	84.08	100 %	84.35	100 %
0.61	85.94	100 %	83.89	100 %	84.27	100 %	85.48	100 %	85.76	100 %
0.62	85.94	100 %	84.73	100 %	85.66	100 %	86.88	100 %	87.16	100 %
0.63	85.94	100 %	87.76	100 %	87.04	100 %	88.28	100 %	88.57	100 %
0.64	85.94	100 %	87.76	100 %	88.42	100 %	89.68	100 %	89.97	100 %
0.65	85.94	100 %	91.35	100 %	89.80	100 %	91.08	100 %	91.38	100 %
0.66	85.94	100 %	91.35	100 %	91.18	100 %	92.48	100 %	92.79	100 %
0.67	85.94	100 %	91.56	100 %	92.56	100 %	93.89	100 %	94.19	100 %
0.68	93.56	100 %	92.93	100 %	93.95	100 %	95.29	100 %	95.60	100 %
0.69	93.56	100 %	94.30	100 %	95.33	100 %	96.69	100 %	97.00	100 %
0.70	93.56	100 %	95.66	100 %	96.71	100 %	98.09	100 %	98.41	100 %
0.71	93.56	100 %	97.03	100 %	98.09	100 %	99.49	100 %	99.81	100 %
0.72	93.56	100 %	98.40	100 %	99.47	100 %	100.89	100 %	101.22	100 %
0.73	93.56	100 %	99.76	100 %	100.85	100 %	102.29	100 %	102.63	100 %
0.74	100.13	100 %	101.13	100 %	102.23	100 %	103.69	100 %	104.03	100 %
0.75	100.13	100 %	102.50	100 %	103.62	100 %	105.10	100 %	105.44	100 %
0.76	100.13	100 %	103.86	100 %	105.00	100 %	106.50	100 %	106.84	100 %
0.77	100.13	100 %	105.23	100 %	106.38	100 %	107.90	100 %	108.25	100 %
0.78	100.13	100 %	106.60	99 %	107.76	100 %	109.30	100 %	109.66	100 %
0.79	103.31	100 %	107.96	99 %	109.14	100 %	110.70	100 %	111.06	100 %
0.80	103.31	100 %	109.33	99 %	110.52	100 %	112.10	100 %	112.47	100 %
0.81	103.81	100 %	110.70	98 %	111.91	99 %	113.50	100 %	113.87	100 %
0.82	105.09	100 %	112.06	98 %	113.29	99 %	114.90	99 %	115.28	100 %
0.83	106.37	98 %	113.43	97 %	114.67	99 %	116.31	99 %	116.68	99 %
0.84	107.65	94 %	114.80	95 %	116.05	96 %	117.71	98 %	118.09	98 %
0.85	108.94	94 %	116.16	93 %	117.43	96 %	119.11	96 %	119.50	97 %
0.86	110.22	89 %	117.53	90 %	118.81	96 %	120.51	96 %	120.90	95 %

Table C.1 continued from previous page

	1 change		2 changes		3 changes		4 changes		5 changes	
alpha	**exp oil**	**p(f)**	**exp oil**	**p(f)**	**exp oil**	**p(f)**	**exp oil**	**p(f)**	**exp oil**	**p(f)**
0.87	111.50	75 %	118.90	86 %	120.20	95 %	121.91	93 %	122.31	93 %
0.88	112.78	67 %	120.26	76 %	121.58	94 %	123.31	91 %	123.71	91 %
0.89	114.06	59 %	121.63	74 %	122.96	91 %	124.71	89 %	125.12	87 %
0.90	115.34	52 %	123.00	72 %	124.34	88 %	126.11	84 %	126.53	83 %
0.91	116.63	45 %	124.36	71 %	125.72	84 %	127.52	79 %	127.93	77 %
0.92	117.91	46 %	125.73	69 %	127.10	73 %	128.92	73 %	129.34	72 %
0.93	119.19	41 %	127.10	67 %	128.48	67 %	130.32	57 %	130.74	58 %
0.94	120.47	36 %	128.46	66 %	129.87	62 %	131.72	52 %	132.15	51 %
0.95	121.75	31 %	129.83	26 %	131.25	45 %	133.12	43 %	133.55	44 %
0.96	123.03	25 %	131.20	24 %	131.25	45 %	134.52	35 %	134.96	38 %
0.97	124.31	21 %	132.56	23 %	132.63	34 %	135.92	28 %	136.37	32 %
0.98	125.60	17 %	133.93	22 %	134.01	27 %	137.32	22 %	137.77	24 %
0.99	126.88	13 %	135.00	20 %	134.52	23 %	138.73	17 %	139.18	17 %

C.2 Case B

Table C.2: Data for Pareto front for case B, as shown in Figure 10.15

	1 change		2 changes		3 changes		4 changes		5 changes	
alpha	**exp oil**	**p(f)**	**exp oil**	**p(f)**	**exp oil**	**p(f)**	**exp oil**	**p(f)**	**exp oil**	**p(f)**
0.01	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	7.63	100 %
0.02	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	7.63	100 %
0.03	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	7.63	100 %
0.04	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	7.63	100 %
0.05	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	7.63	100 %
0.06	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	10.80	100 %
0.07	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	10.80	100 %
0.08	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	11.25	100 %
0.09	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	15.43	100 %
0.10	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	15.43	100 %
0.11	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	18.42	100 %
0.12	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	18.42	100 %
0.13	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	18.42	100 %
0.14	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	19.68	100 %
0.15	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	23.05	100 %
0.16	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	23.05	100 %
0.17	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	23.90	100 %
0.18	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	25.31	100 %
0.19	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	26.71	100 %
0.20	85.94	100 %	65.47	100 %	40.72	100 %	28.75	100 %	28.12	100 %
0.21	85.94	100 %	65.47	100 %	40.72	100 %	36.38	100 %	35.30	100 %
0.22	85.94	100 %	65.47	100 %	40.72	100 %	36.38	100 %	35.30	100 %
0.23	85.94	100 %	65.47	100 %	40.72	100 %	36.38	100 %	35.30	100 %
0.24	85.94	100 %	65.47	100 %	40.72	100 %	36.38	100 %	35.30	100 %
0.25	85.94	100 %	65.47	100 %	40.72	100 %	36.38	100 %	35.30	100 %
0.26	85.94	100 %	65.47	100 %	40.72	100 %	39.55	100 %	36.55	100 %
0.27	85.94	100 %	65.47	100 %	40.72	100 %	39.55	100 %	37.96	100 %
0.28	85.94	100 %	65.47	100 %	40.72	100 %	39.55	100 %	39.36	100 %
0.29	85.94	100 %	65.47	100 %	40.72	100 %	40.64	100 %	40.77	100 %
0.30	85.94	100 %	65.47	100 %	53.51	100 %	47.18	100 %	42.18	100 %
0.31	85.94	100 %	65.47	100 %	53.51	100 %	47.18	100 %	46.10	100 %
0.32	85.94	100 %	65.47	100 %	53.51	100 %	47.18	100 %	46.10	100 %
0.33	85.94	100 %	65.47	100 %	53.51	100 %	47.18	100 %	46.39	100 %
0.34	85.94	100 %	65.47	100 %	53.51	100 %	47.64	100 %	47.80	100 %
0.35	85.94	100 %	65.47	100 %	53.51	100 %	49.04	100 %	49.20	100 %
0.36	85.94	100 %	65.47	100 %	53.51	100 %	50.45	100 %	50.61	100 %
0.37	85.94	100 %	65.47	100 %	53.51	100 %	51.85	100 %	52.02	100 %
0.38	85.94	100 %	65.47	100 %	53.51	100 %	53.25	100 %	53.42	100 %
0.39	85.94	100 %	65.47	100 %	61.13	100 %	54.65	100 %	54.83	100 %

Table C.2 continued from previous page

	1 change		2 changes		3 changes		4 changes		5 changes	
alpha	**exp oil**	**p(f)**	**exp oil**	**p(f)**	**exp oil**	**p(f)**	**exp oil**	**p(f)**	**exp oil**	**p(f)**
0.40	85.94	100 %	65.47	100 %	61.13	100 %	56.05	100 %	56.23	100 %
0.41	85.94	100 %	65.47	100 %	61.13	100 %	57.45	100 %	57.64	100 %
0.42	85.94	100 %	65.47	100 %	61.13	100 %	60.05	100 %	59.05	100 %
0.43	85.94	100 %	65.47	100 %	61.13	100 %	60.25	100 %	60.45	100 %
0.44	85.94	100 %	65.47	100 %	61.13	100 %	61.66	100 %	61.86	100 %
0.45	85.94	100 %	65.47	100 %	64.31	100 %	63.06	100 %	63.26	100 %
0.46	85.94	100 %	65.47	100 %	64.31	100 %	64.46	100 %	64.67	100 %
0.47	85.94	100 %	65.47	100 %	64.93	100 %	65.86	100 %	66.07	100 %
0.48	85.94	100 %	73.97	100 %	71.93	100 %	67.26	100 %	67.48	100 %
0.49	85.94	100 %	73.97	100 %	71.93	100 %	68.66	100 %	68.89	100 %
0.50	85.94	100 %	73.97	100 %	71.93	100 %	70.85	100 %	70.29	100 %
0.51	85.94	100 %	73.97	100 %	71.93	100 %	71.46	100 %	71.70	100 %
0.52	85.94	100 %	73.97	100 %	71.93	100 %	72.87	100 %	73.10	100 %
0.53	85.94	100 %	73.97	100 %	73.22	100 %	74.27	100 %	74.51	100 %
0.54	85.94	100 %	73.97	100 %	74.60	100 %	75.67	100 %	75.92	100 %
0.55	85.94	100 %	81.60	100 %	75.99	100 %	77.07	100 %	77.32	100 %
0.56	85.94	100 %	81.60	100 %	77.37	100 %	78.47	100 %	78.73	100 %
0.57	85.94	100 %	81.60	100 %	78.75	100 %	79.87	100 %	80.13	100 %
0.58	85.94	100 %	81.60	100 %	80.13	100 %	81.27	100 %	81.54	100 %
0.59	85.94	100 %	81.60	100 %	81.68	100 %	82.68	100 %	82.94	100 %
0.60	85.94	100 %	83.89	100 %	82.89	100 %	84.08	100 %	84.35	100 %
0.61	85.94	100 %	83.89	100 %	84.27	100 %	85.48	100 %	85.76	100 %
0.62	85.94	100 %	84.73	100 %	85.66	100 %	86.88	100 %	87.16	100 %
0.63	85.94	100 %	87.76	100 %	87.04	100 %	88.28	100 %	88.57	100 %
0.64	85.94	100 %	87.76	100 %	88.42	100 %	89.68	100 %	89.97	100 %
0.65	85.94	100 %	91.35	100 %	89.80	100 %	91.08	100 %	91.38	100 %
0.66	85.94	100 %	91.35	100 %	91.18	100 %	92.48	100 %	92.79	100 %
0.67	85.94	100 %	91.56	100 %	92.56	100 %	93.89	100 %	94.19	100 %
0.68	93.56	100 %	92.93	100 %	93.95	100 %	95.29	100 %	95.60	100 %
0.69	93.56	100 %	94.30	100 %	95.33	100 %	96.69	100 %	97.00	100 %
0.70	93.56	100 %	95.66	100 %	96.71	100 %	98.09	100 %	98.41	100 %
0.71	93.56	100 %	97.03	100 %	98.09	100 %	99.49	100 %	99.81	100 %
0.72	93.56	100 %	98.40	100 %	99.47	100 %	100.89	100 %	101.22	100 %
0.73	93.56	100 %	99.76	100 %	100.85	100 %	102.29	100 %	102.63	100 %
0.74	100.13	100 %	101.13	100 %	102.23	100 %	103.69	100 %	104.03	100 %
0.75	100.13	100 %	102.50	100 %	103.62	100 %	105.10	100 %	105.44	100 %
0.76	100.13	100 %	103.86	100 %	105.00	100 %	106.50	100 %	106.84	100 %
0.77	100.13	100 %	105.23	100 %	106.38	100 %	107.90	100 %	108.25	100 %
0.78	100.13	100 %	106.60	99 %	107.76	100 %	109.30	100 %	109.66	100 %
0.79	103.31	100 %	107.96	99 %	109.14	100 %	110.70	100 %	111.06	100 %
0.80	103.31	100 %	109.33	99 %	110.52	100 %	112.10	100 %	112.47	100 %
0.81	103.81	100 %	110.70	98 %	111.91	99 %	113.50	100 %	113.87	100 %
0.82	105.09	100 %	112.06	98 %	113.29	99 %	114.90	99 %	115.28	100 %
0.83	106.37	98 %	113.43	97 %	114.67	99 %	116.31	99 %	116.68	99 %
0.84	107.65	94 %	114.80	95 %	116.05	96 %	117.71	98 %	118.09	98 %
0.85	108.94	94 %	116.16	93 %	117.43	96 %	119.11	96 %	119.50	97 %
0.86	110.22	89 %	117.53	90 %	118.81	96 %	120.51	96 %	120.90	95 %
0.87	111.50	75 %	118.90	86 %	120.20	95 %	121.91	93 %	122.31	93 %
0.88	112.78	67 %	120.26	76 %	121.58	94 %	123.31	91 %	123.71	91 %
0.89	114.06	59 %	121.63	74 %	122.96	91 %	124.71	89 %	125.12	87 %
0.90	115.34	52 %	123.00	72 %	124.34	88 %	126.11	84 %	126.53	83 %
0.91	116.63	45 %	124.36	71 %	125.72	84 %	127.52	79 %	127.93	77 %
0.92	117.91	46 %	125.73	69 %	127.10	73 %	128.92	73 %	129.34	72 %
0.93	119.19	41 %	127.10	67 %	128.48	67 %	130.32	57 %	130.74	58 %
0.94	120.47	36 %	128.46	66 %	129.87	62 %	131.72	52 %	132.15	51 %
0.95	121.75	31 %	129.83	26 %	131.25	45 %	133.12	43 %	133.55	44 %
0.96	123.03	25 %	131.20	24 %	131.25	45 %	134.52	35 %	134.96	38 %
0.97	124.31	21 %	132.56	23 %	132.63	34 %	135.92	28 %	136.37	32 %
0.98	125.60	17 %	133.93	22 %	134.01	27 %	137.32	22 %	137.77	24 %
0.99	126.88	13 %	135.00	20 %	134.52	23 %	138.73	17 %	139.18	17 %

C.3 Case C

Table C.3: Data for Pareto front for case C, as shown in Figure C.1

	1 change		2 changes		3 changes		4 changes		5-7 changes	
	exp oil	p(f)	exp oil	p(f)	exp oil	p(f)	exp oil	p(f)	exp oil	p(f)
0.01	6.54	100 %	6.54	100 %	6.54	100 %	6.54	100 %	6.54	100 %
0.02	6.54	100 %	6.54	100 %	6.54	100 %	6.54	100 %	6.54	100 %
0.03	6.54	100 %	6.54	100 %	6.54	100 %	6.54	100 %	6.54	100 %
0.04	6.54	100 %	6.54	100 %	6.54	100 %	6.54	100 %	6.54	100 %
0.05	6.54	100 %	6.54	100 %	6.54	100 %	6.54	100 %	6.62	100 %
0.06	6.54	100 %	6.54	100 %	6.54	100 %	6.90	100 %	7.95	100 %
0.07	6.54	100 %	6.54	100 %	6.54	100 %	8.05	100 %	9.27	100 %
0.08	6.54	100 %	6.54	100 %	7.22	100 %	9.20	100 %	10.59	100 %
0.09	6.54	100 %	6.54	100 %	8.13	100 %	10.35	100 %	11.92	100 %
0.10	6.54	100 %	6.54	100 %	9.03	100 %	11.50	100 %	13.24	100 %
0.11	6.54	100 %	7.18	100 %	9.93	100 %	12.66	100 %	14.57	100 %
0.12	6.54	100 %	7.84	100 %	10.84	100 %	13.81	100 %	15.89	100 %
0.13	6.54	100 %	8.49	100 %	11.74	100 %	14.96	100 %	17.22	100 %
0.14	6.54	100 %	9.14	100 %	12.64	100 %	16.11	100 %	18.54	100 %
0.15	6.54	100 %	9.79	100 %	13.55	100 %	17.26	100 %	19.86	100 %
0.16	6.54	100 %	10.45	100 %	14.45	100 %	18.41	100 %	21.19	100 %
0.17	6.54	100 %	11.10	100 %	15.35	100 %	19.56	100 %	22.51	100 %
0.18	6.54	100 %	11.75	100 %	16.26	100 %	20.71	100 %	23.84	100 %
0.19	6.54	100 %	12.41	100 %	17.16	100 %	21.86	100 %	25.16	100 %
0.20	6.54	100 %	13.06	100 %	18.06	100 %	23.01	100 %	26.48	100 %
0.21	6.54	100 %	13.71	100 %	18.97	100 %	24.16	100 %	27.81	100 %
0.22	6.54	100 %	14.36	100 %	19.87	100 %	25.31	100 %	29.13	100 %
0.23	6.54	100 %	15.02	100 %	20.77	100 %	26.46	100 %	30.46	100 %
0.24	6.68	100 %	15.67	100 %	21.67	100 %	27.61	100 %	31.78	100 %
0.25	6.96	100 %	16.32	100 %	22.58	100 %	28.76	100 %	33.11	100 %
0.26	7.24	100 %	16.98	100 %	23.48	100 %	29.91	100 %	34.43	100 %
0.27	7.52	100 %	17.63	100 %	24.38	100 %	31.06	100 %	35.75	100 %
0.28	7.79	100 %	18.28	100 %	25.29	100 %	32.21	100 %	37.08	100 %
0.29	8.07	100 %	18.93	100 %	26.19	100 %	33.36	100 %	38.40	100 %
0.30	8.35	100 %	19.59	100 %	27.09	100 %	34.51	100 %	39.73	100 %
0.31	8.63	100 %	20.24	100 %	28.00	100 %	35.66	100 %	41.05	100 %
0.32	8.91	100 %	20.89	100 %	28.90	100 %	36.81	100 %	42.38	100 %
0.33	9.19	100 %	21.55	100 %	29.80	100 %	37.97	100 %	43.70	100 %
0.34	9.46	100 %	22.20	100 %	30.71	100 %	39.12	100 %	45.02	100 %
0.35	9.74	100 %	22.85	100 %	31.61	100 %	40.27	100 %	46.35	100 %
0.36	10.02	100 %	23.51	100 %	32.51	100 %	41.42	100 %	47.67	100 %
0.37	10.30	100 %	24.16	100 %	33.41	100 %	42.57	100 %	49.00	100 %
0.38	10.58	100 %	24.81	100 %	34.32	100 %	43.72	100 %	50.32	100 %
0.39	10.86	100 %	25.46	100 %	35.22	100 %	44.87	100 %	51.65	100 %
0.40	11.13	100 %	26.12	100 %	36.12	100 %	46.02	100 %	52.97	100 %
0.41	11.41	100 %	26.77	100 %	37.03	100 %	47.17	100 %	54.29	100 %
0.42	11.69	100 %	27.42	100 %	37.93	100 %	48.32	100 %	55.62	100 %
0.43	11.97	100 %	28.08	100 %	38.83	100 %	49.47	100 %	56.94	100 %
0.44	12.25	100 %	28.73	100 %	39.74	100 %	50.62	100 %	58.27	100 %
0.45	12.53	100 %	29.38	100 %	40.64	100 %	51.77	100 %	59.59	100 %
0.46	12.81	100 %	30.03	100 %	41.54	100 %	52.92	100 %	60.92	100 %
0.47	13.08	100 %	30.69	100 %	42.45	100 %	54.07	100 %	62.24	100 %
0.48	13.36	100 %	31.34	100 %	43.35	100 %	55.22	100 %	63.56	100 %
0.49	13.64	100 %	31.99	100 %	44.25	100 %	56.37	100 %	64.89	100 %
0.50	13.92	100 %	32.65	100 %	45.16	100 %	57.52	100 %	66.21	100 %
0.51	14.20	100 %	33.30	100 %	46.06	100 %	58.67	100 %	67.54	100 %
0.52	14.48	100 %	33.95	100 %	46.96	100 %	59.82	100 %	68.86	100 %
0.53	14.75	100 %	34.60	100 %	47.86	100 %	60.97	100 %	70.19	100 %
0.54	15.03	100 %	35.26	100 %	48.77	100 %	62.12	100 %	71.51	100 %
0.55	15.31	100 %	35.91	100 %	49.67	100 %	63.28	100 %	72.83	100 %
0.56	15.59	100 %	36.56	100 %	50.57	100 %	64.43	100 %	74.16	100 %
0.57	15.87	100 %	37.22	100 %	51.48	100 %	65.58	100 %	75.48	100 %
0.58	16.15	100 %	37.87	100 %	52.38	100 %	66.73	100 %	76.81	100 %

Table C.3 continued from previous page

alpha	1 change		2 changes		3 changes		4 changes		5-7 changes	
	exp oil	p(f)	exp oil	p(f)	exp oil	p(f)	exp oil	p(f)	exp oil	p(f)
0.59	16.42	100 %	38.52	100 %	53.28	100 %	67.88	100 %	78.13	100 %
0.60	16.70	100 %	39.18	100 %	54.19	100 %	69.03	100 %	79.45	100 %
0.61	16.98	100 %	39.83	100 %	55.09	100 %	70.18	100 %	80.78	100 %
0.62	17.26	100 %	40.48	100 %	55.99	100 %	71.33	100 %	82.10	100 %
0.63	17.54	100 %	41.13	100 %	56.90	100 %	72.48	100 %	83.43	100 %
0.64	17.82	100 %	41.79	100 %	57.80	100 %	73.63	100 %	84.75	100 %
0.65	18.09	100 %	42.44	100 %	58.70	100 %	74.78	100 %	86.08	100 %
0.66	18.37	100 %	43.09	100 %	59.61	100 %	75.93	100 %	87.40	100 %
0.67	12.03	100 %	43.75	100 %	60.51	100 %	77.08	100 %	88.72	100 %
0.68	12.30	100 %	44.40	100 %	61.41	100 %	78.23	100 %	90.05	100 %
0.69	12.58	100 %	45.05	100 %	62.31	100 %	79.38	100 %	91.37	100 %
0.70	12.86	100 %	45.70	100 %	63.22	100 %	80.53	100 %	92.70	100 %
0.71	13.14	100 %	46.36	100 %	64.12	100 %	81.68	100 %	94.02	100 %
0.72	13.42	100 %	47.01	100 %	65.02	100 %	82.83	100 %	95.35	100 %
0.73	20.32	100 %	47.66	100 %	65.93	100 %	83.98	100 %	96.67	100 %
0.74	20.60	100 %	48.32	100 %	66.83	100 %	85.13	100 %	97.99	100 %
0.75	20.88	100 %	48.97	100 %	67.73	100 %	86.28	100 %	99.32	100 %
0.76	14.53	100 %	49.62	100 %	68.64	100 %	87.43	100 %	100.64	100 %
0.77	14.81	100 %	50.28	100 %	69.54	100 %	88.59	100 %	101.97	100 %
0.78	15.09	100 %	50.93	100 %	70.44	100 %	89.74	100 %	103.29	100 %
0.79	15.37	100 %	51.58	100 %	71.35	100 %	90.89	100 %	104.62	100 %
0.80	15.64	100 %	52.23	100 %	72.25	100 %	92.04	100 %	105.94	100 %
0.81	15.92	100 %	52.89	100 %	73.15	100 %	93.19	100 %	107.26	100 %
0.82	16.20	100 %	53.54	100 %	74.05	100 %	94.34	100 %	108.59	100 %
0.83	16.48	100 %	54.19	100 %	74.96	100 %	95.49	100 %	109.91	100 %
0.84	23.38	100 %	54.85	100 %	75.86	100 %	96.64	100 %	111.24	100 %
0.85	23.66	100 %	55.50	100 %	76.76	100 %	97.79	100 %	112.56	100 %
0.86	23.94	100 %	56.15	100 %	77.67	100 %	98.94	100 %	113.89	100 %
0.87	24.22	100 %	56.80	100 %	78.57	100 %	100.09	100 %	115.21	100 %
0.88	24.50	100 %	57.46	100 %	79.47	100 %	101.24	100 %	116.53	99 %
0.89	24.77	100 %	58.11	100 %	80.38	100 %	102.39	100 %	117.86	98 %
0.90	25.05	100 %	58.76	100 %	81.28	100 %	103.54	100 %	119.18	97 %
0.91	25.33	100 %	59.42	100 %	82.18	100 %	104.69	100 %	120.51	96 %
0.92	25.61	100 %	60.07	100 %	83.09	100 %	105.84	100 %	121.83	93 %
0.93	25.89	100 %	60.72	100 %	83.99	100 %	106.99	100 %	123.16	92 %
0.94	26.17	100 %	61.37	100 %	84.89	100 %	108.14	100 %	124.48	89 %
0.95	26.45	100 %	62.03	100 %	85.80	100 %	109.29	100 %	125.80	85 %
0.96	26.72	100 %	62.68	100 %	86.70	100 %	110.44	100 %	127.13	81 %
0.97	27.00	100 %	63.33	100 %	87.60	100 %	111.59	100 %	128.45	75 %
0.98	27.28	100 %	63.99	100 %	88.50	100 %	112.75	100 %	129.78	70 %
0.99	27.56	100 %	64.64	100 %	89.41	100 %	113.90	99 %	131.10	56 %

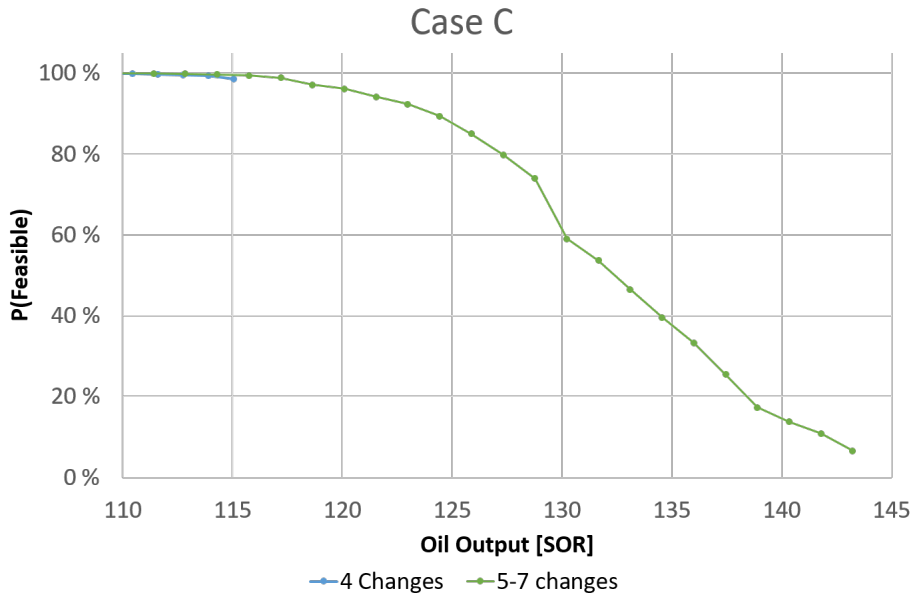

Figure C.1: Pareto front for case C and 1-7 allowed changes. 1-3 changes are always feasible and not included in the plot. See Table C.3 for the relevant data.

Appendix D

Scenario Generation Appendix

This appendix provides results from varying the hyperparameters of the Markov Weighted scenario generation procedure.

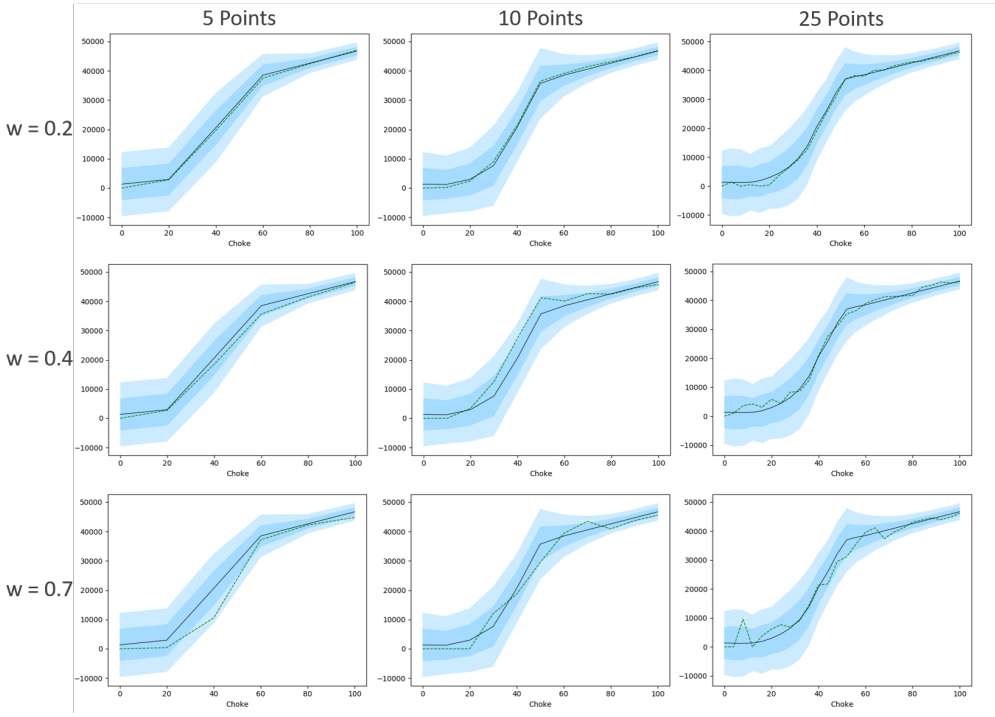

Figure D.1: Markov Weighted scenarios for a variety of weighting parameters w and number of sampling points. Scenario in dashed green line, well model mean in black solid line and 2 well model standard deviations in shades of blue.