NTNU
Norwegian University of
Science and Technology

Master's thesis

2019

I Gusti Agung Gede Angga

Master's thesis

NTNU
Norwegian University of
Science and Technology
Faculty of Engineering
Department of Geoscience and Petroleum

I Gusti Agung Gede Angga

Automated Decision Support Methodologies for Field Development: The Safari Field Case

July 2019

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

Automated Decision Support Methodologies for Field Development: The Safari Field Case

I Gusti Agung Gede Angga

Petroleum Engineering
Submission date: July 2019
Supervisor: Professor Milan Stanko

Norwegian University of Science and Technology
Department of Geoscience and Petroleum

To whoever supports me during my stay in Trondheim.

Summary

In planning a field development, there are many field design features to be decided upon. The company's development team has a responsibility to determine a configuration of those options which maximizes the value of the company's asset. Automated field development is a cutting-edge methodology to screen and search in an automated manner for the best combination of field design features under a given set of constraints.

This thesis presents an automated field development methodology which is based on mathematical optimization. The methodology is designed for a synthetic (fictitious) case called "Safari". The Safari field is characterized by having multiple reservoir units, where the reservoirs are non-communicating and have unique properties.

In this study, there are two main optimization problems, i.e., maximization of the plateau duration and maximization of the Net Present Value (NPV). The optimization problems include production & injection rates, drilling schedule, and recovery mechanism as the decision variables. Some constraints concerning production, injection, and drilling are also considered in the optimization. Formulations of the optimization problems adapt 2 main ideas, i.e. (1) to use production potential curves as the proxy model of the production system and (2) to use multi-dimensional PWL approximations for representing some non-linear functions. According to the optimization results, a better objective value is obtained when the drilling schedule and recovery mechanism are also defined as the decision variables in addition to fluid rates.

Several evaluations have been performed to make the formulations of the optimization problems and their implementations in AMPL more efficient and more accurate. By adapting the evaluations' results, the optimization process becomes much faster. Furthermore, uncertainties of the optimization results have been quantified through uncertainty analyses. Three uncertain parameters are considered, i.e., in-place, development & operational cost, and oil price. To conduct an uncertainty analysis, two approaches have been studied, i.e., using Latin Hypercube Sampling (LHS) method and using a probability tree. Uncertainty analysis using a probability tree is preferred because it is quicker and produces similar results as the other approach.

Preface

The thesis is submitted in partial fulfillment of the requirements for the MSc degree in Reservoir Engineering & Petrophysics at Department of Geoscience and Petroleum - Norwegian University of Science and Technology (NTNU). In addition, this thesis is a continuation of previous works done for the specialization project (Angga, 2018).

I wish to express my sincere gratitude to my supervisor, Professor Milan Edward Wolf Stanko, for introducing me to the topic, sharing his knowledge, giving me excellent guidance, and dedicating himself to this project. It is such an honor being supervised and working with him.

This study would not have been completed without the support of Aker Solutions, especially Thomas Førde. I would like to thank him for providing a data set and giving meaningful suggestions. My appreciation also goes to Professor Eduardo Camponogara, Professor Lars Struen Imsland, and Thiago Lima Silva for helping me understand the concept of mathematical optimization and resolve some practical issues. Finally, I would like to thank PhD candidate Diana Gonzales for her inputs to this study.

Trondheim, July 2019
I Gusti Agung Gede Angga

Table of Contents

Summary i

Preface ii

Table of Contents 3

List of Tables 6

List of Figures 9

1 Introduction 11

1.1 Background 11
1.2 Objective 12
1.3 Overview of Safari Field 12

2 Theoretical Framework 15

2.1 Fundamentals of Production Potential Curve 15
2.1.1 Definition of Production Potential 15
2.1.2 Construction of Production Potential Curve 17
2.1.3 Planning the Production Schedule Using the Production Potential Curve 17
2.1.4 Characteristics of Production Potential Curve 19
2.2 Mathematical Optimization 20
2.2.1 Linear Programming 20
2.2.2 Algorithms to Solve Linear Programming 21
2.2.3 Integer Programming 22
2.2.4 Mixed-Integer Linear Programming 23
2.2.5 Algorithms to Solve IP or MILP 24
2.3 Piecewise Linear (PWL) Approximation 26
2.3.1 One-Dimensional PWL Approximation 26
2.3.2 Two-Dimensional PWL Approximation 28

2.3.3 Three-Dimensional PWL Approximation 33
2.4 Latin Hypercube Sampling (LHS) 34

3 Maximizing the Plateau Duration 37
3.1 Formulation of the Optimization Problem 38
3.1.1 Parameters 39
3.1.2 Sets 43
3.1.3 Variables 44
3.1.4 Objective Function 47
3.1.5 Operational Constraints 47
3.1.6 Drilling Constraints 48
3.1.7 Consistency Constraints 49
3.1.8 Integer & Binary Constraints 51
3.1.9 Constraints Induced by PWL Approximations 51
3.2 Implementation in AMPL 54
3.3 Results & Discussion 55

4 Maximizing the NPV 59
4.1 Formulation of the Optimization Problem 60
4.1.1 Parameters 60
4.1.2 Sets 63
4.1.3 Variables 63
4.1.4 Objective Function 64
4.1.5 Additional Constraints 65
4.2 Implementation in AMPL 69
4.3 Results & Discussion 69

5 Improving the Computational Efficiency & Accuracy 73
5.1 Reducing the Number of Breakpoints for PWL Approximation 73
5.1.1 Reducing the Number of N_p Breakpoints 74
5.1.2 Reducing the Number of N_{op} Breakpoints 77
5.2 Selecting the N_p Breakpoints 78
5.3 Determining Gas Production Rate 80
5.4 Choosing the Numerical Integration Technique for Calculating the Cumulative 84
5.4.1 Cumulative Gas Production 84
5.4.2 Cumulative Oil Production 85
5.5 Optimizing the Parallel Computing 87
5.6 Choosing the Solver Type 89
5.7 Configuring the Solver Options 90
5.8 Analyzing the Running Time 92

6 Quantifying the Uncertainties 95
6.1 Proof of Concept 95
6.2 The Change of Production Potential Curve with In-Place 98
6.3 Uncertain Parameters 100

6.4 Modifications of the Mathematical Formulation 102
6.5 Optimization Problem & Implementation in AMPL 104
6.6 Uncertainty Analysis Using LHS Method 106
6.7 Uncertainty Analysis Using Probability Tree 108
6.8 An Empirical Equation to Estimate the Optimal NPV 111
6.9 Rule of Thumb for Maximizing the NPV 113

7 Conclusion 115

Bibliography 116

Appendix 119

A AMPL Code for Maximizing the Plateau Duration 121
A.1 CASE-1 121
A.1.1 Model File (CASE-1_Model.mod) 121
A.1.2 Data File (CASE-1_Data.dat) 124
A.1.3 Run File (CASE-1_Run.run) 124
A.2 REF-CASE-1 125
A.2.1 Model File (REF-CASE-1_Model.mod) 125
A.2.2 Data File (REF-CASE-1_Data.dat) 127
A.2.3 Run File (REF-CASE-1_Run.run) 127

B AMPL Code for Maximizing the NPV 129
B.1 CASE-2 129
B.1.1 Model File (CASE-2_Model.mod) 129
B.1.2 Data File (CASE-2_Data.dat) 132
B.1.3 Data File (CASE-2_Econ.dat) 133
B.1.4 Run File (CASE-2_Run.run) 133
B.2 REF-CASE-2 134
B.2.1 Model File (REF-CASE-2_Model.mod) 134
B.2.2 Data File (REF-CASE-2_Data.dat) 137
B.2.3 Data File (REF-CASE-2_Econ.dat) 137
B.2.4 Run File (REF-CASE-2_Run.run) 138

C AMPL Code for Uncertainty Analysis 139
C.1 Model File (CASE-1xx_Model.mod) 139
C.2 Data File (CASE-1xx_Data.dat) 142
C.3 Data File (CASE-1xx_Econ.dat) 142
C.4 Files Related to CASE-101 143
C.4.1 Data File (CASE-101_Uncertainty.dat) 143
C.4.2 Run File (CASE-101_Run.run) 143
C.5 Run File (MASTER_Run.run) 144

List of Tables

1.1 Initial reservoir fluid properties (Angga, 2018) 13
1.2 Reservoir rock properties (Angga, 2018) 13
1.3 Well properties (Angga, 2018) 14

3.1 Comparisons of objective function, decision variables, and parameters between CASE-1 and REF-CASE-1 38
3.2 Maximum number of gas and water injectors for each reservoir 40
3.3 2D PWL approximation to represent a non-linear function $RM = f(zN_{gi}, zN_{wi})$ 40
3.4 Number of breakpoints for 3D PWL approximation which represents a non-linear function $q_{opp} = f(N_p, N_{op}, RM)$ 41
3.5 3D PWL approximation to represent a non-linear function $q_{opp} = f(N_p, N_{op}, RM)$ 42
3.6 Sets of N_p breakpoints used for 3D PWL approximation 43
3.7 Sets of N_{op} breakpoints used for 3D PWL approximation 44
3.8 Sets of RM breakpoints used for 3D PWL approximation 44
3.9 Comparisons of optimization results between CASE-1 and REF-CASE-1 55

4.1 Comparisons of objective function, decision variables, and parameters between CASE-2 and REF-CASE-2 59
4.2 Fixed expenditure for exploration activities (Angga, 2018) 61
4.3 Fixed expenditure for abandonment activities (Angga, 2018) 61
4.4 All the coefficients in Equation 4.2 (Angga, 2018) 62
4.5 All the coefficients in Equation 4.3 (Angga, 2018) 62
4.6 All the constant and the coefficients in Equation 4.4 (Angga, 2018) 63
4.7 Comparisons of optimization results between CASE-2 and REF-CASE-2 69

5.1 Objective function, decision variable, and parameters of the optimization problem used in the reduction of $n_{bp\_N_p}$ study 74
5.2 List of cases for the reduction of $n_{bp\_N_p}$ study 74
5.3 Comparison of the maximum RE between case 4 to case 9 76
5.4 List of cases for the reduction of $n_{bp\_N_{op}}$ study 77

5.5 Comparison of the maximum RE between case 11 to case 16 78
5.6 Objective function, decision variables, and parameters of the optimization problem used in the determination of q_g study 81
5.7 Average RE for all reservoirs and the field 83
5.8 Objective function, decision variables, and parameters of the optimization problem used in the evaluation of the numerical integration 85
5.9 Objective function, decision variables, and parameter of the optimization problem used in the evaluation of parallel computing 88
5.10 List of cases for the evaluation of parallel computing 88
5.11 Comparison of objective value and relative gap between case 24 to case 27 88
5.12 List of cases for the evaluation of solver type 89
5.13 Comparison of objective value and relative gap between case 28 to case 31 90
5.14 Values for "mipemphasis" and their descriptions (IBM, 2019a) 90
5.15 Values for "parallelmode" and their descriptions (IBM, 2019b) 91
5.16 List of cases for the evaluation of solver options 91
5.17 Comparison of objective value and relative gap between case 32 to case 43 92
5.18 List of cases for the runtime analysis 92
5.19 Comparison of objective value, best bound, and relative gap between case 44 to case 47 93

6.1 Objective function, decision variables, and parameter of the optimization problem used in the uncertainty analysis 104

List of Figures

1.1 Location of Safari field and its reservoirs (Angga, 2018) 13

2.1 An illustration of field production potential. The solid line indicates the profile of field production potential, while the dots represent the profile of field production rate (Angga, 2018). 16
2.2 An example of production potential curve (Angga, 2018) 16
2.3 An illustration of generating production potential curve (Angga, 2018) . . 17
2.4 An illustration of planning the production schedule using the production potential curve 18
2.4 An illustration of planning the production schedule using the production potential curve (cont.) 19
2.5 Visualization of an LP (Cormen et al., 2009) 21
2.6 Illustration of simplex algorithm and interior-point method (Hoffmann, 2014) 22
2.7 Visualization of IP and its LP relaxation (Vanderbei, 2008) 23
2.8 An illustration of branch-and-bound algorithm (Camponogara, 2018) . . . 24
2.9 An illustration of cutting plane algorithm (NTNU, 2019) 25
2.10 A non-linear function and its PWL function (Hoffmann, 2014) 27
2.11 An illustration of two-dimensional PWL grid (Hoffmann, 2014) 29
2.12 Visualization of $\eta_{x,i}$ and $\eta_{y,j}$ 31
2.13 An illustration of SOS2 constraint for both sets $\{\eta_{x,i}, i = 1, 2, \ldots, 4\}$ and $\{\eta_{y,j}, j = 1, 2, \ldots, 4\}$ when carrying out a linear interpolation 31
2.14 The procedure of generating samples using LHS method 35
2.14 The procedure of generating samples using LHS method (cont.) 36

3.1 Layout of the production system of Safari field for FPSO development alternative (Angga, 2018) 37
3.2 Visualization of the objective function 47
3.3 An illustration of a non-linear function $RM = f(zN_{gi}, zN_{wi})$ 53
3.4 Workflow of AMPL implementation for CASE-1 54

3.5 Comparison of recovery mechanisms between CASE-1 and REF-CASE-1 56
3.6 Comparison of drilling schedule between CASE-1 and REF-CASE-1 . . . 56
3.7 Comparison of q_o and q_{opp} between CASE-1 and REF-CASE-1 57
3.8 Comparison of production & injection profiles between CASE-1 and REF-CASE-1 57

4.1 Workflow of AMPL implementation for CASE-2 69
4.2 Comparison of recovery mechanisms between CASE-2 and REF-CASE-2 70
4.3 Comparison of drilling schedule between CASE-2 and REF-CASE-2 . . . 71
4.4 Comparison of q_o and q_{opp} between CASE-2 and REF-CASE-2 71
4.5 Comparison of production & injection profiles between CASE-2 and REF-CASE-2 72
4.6 Comparison of DCF & its cumulative between CASE-2 and REF-CASE-2 72

5.1 An illustration of N_p breakpoints for a case with 10 N_p breakpoints . . . 75
5.2 Relationship between the number of N_p breakpoints and the optimization runtime 76
5.3 An illustration of a set of N_p breakpoints used to reproduce the production potential curves of Løve reservoir with any number of oil producers and any recovery mechanism 78
5.4 An illustration of wisely-selected N_p breakpoints used to reproduce the production potential curves of Løve reservoir 79
5.5 The reproduced G_p-N_p and W_p-N_p curves using 10 equally-spaced N_p breakpoints 80
5.6 The reproduced G_p-N_p and W_p-N_p curves using 10 wisely-selected N_p breakpoints 80
5.7 2D PWL approximation to represent a non-linear function $q_g = f(q_o, GOR)$ 82
5.8 Comparison of the gas production profile obtained from both methods . . 83
5.9 Comparison of the optimization runtime required by both methods 83
5.10 Comparison of gas production profile between the case applying backward rectangular integration technique and the case adapting trapezoidal integration technique 85
5.11 Comparison of the optimization outcomes between case 21 to case 23 . . 87
5.12 Comparison of drilling schedule between case 46 and case 47 93
5.13 Comparison of production & injection profile between case 46 and case 47 93

6.1 Relationship between the plateau rate and the NPV (Angga, 2018) 95
6.2 NPV, PV_{ar}, PV_{tcao}, and their first derivatives vs. q_{p_F} (base case) . . . 97
6.3 NPV, PV_{ar}, PV_{tcao}, and their first derivatives vs. q_{p_F} (CapEx is doubled) 98
6.4 The changes of production potential curve and G_p-N_p curve with in-place 99
6.5 An approach to evaluate how far the production potential curve being stretched or squeezed when the in-place changes 100
6.6 PDF & CDF of the normal distribution 102
6.7 PDF & CDF of the uniform distribution 102
6.8 Workflow of AMPL implementation of the optimization problem used in the uncertainty analysis 105

6.9 Visualization of master run file 106
6.10 Distribution of the optimal NPV (from LHS method) 107
6.11 Distributions of the optimization results (from LHS method) 108
6.12 Determination of the output values 109
6.13 Discrete probability distribution which approximates the normal distribution 109
6.14 Discrete probability distribution which approximates the uniform distribution 109
6.15 The possible outcomes for the uncertain parameters and its corresponding probabilities 110
6.16 Distribution of the optimal NPV (from probability tree) 111
6.17 Distributions of the optimization results (from probability tree) 111
6.18 Relationship between PV_{ac} and N_{pu} 112
6.19 Distribution of F_D 113
6.20 Distribution of I_R 113

Chapter 1

Introduction

1.1 Background

In planning a field development, there are many field design features to be decided upon, e.g., type of offshore structure, recovery mechanism, production schedule, number of wells & its drilling schedule (Haldorsen, 1996). Decisions on those features often have impacts on the economy of the field development. Therefore, the company's development team has a responsibility to determine a configuration of those options which maximizes the value of the company's asset.

However, most of the development team's outcomes are suboptimal. The reason is that only a few selected combinations and cases are evaluated due to time constraints. In consequence of this conventional approach, there are many combinations or cases which potentially have higher economic value are unstudied.

Automated field development is a cutting-edge methodology to screen and search in an automated manner for the best combination of field design features under a given set of constraints. Such methodology is currently under development at SUBPRO, i.e., a research center for subsea production & processing (SUBPRO, 2018). The project is carried out by PhD student Diana Gonzalez and supervised by Professor Milan Stanko. Up to date, the developed methodology is capable of determining the production schedule and drilling schedule which yield the highest NPV (Net Present Value) through mathematical optimization.

The progress of the project has been presented to the oil industry and successfully gain attention from several companies; one of the interested company is Aker Solutions. They have proposed to test the methodology for a synthetic (fictitious) case called "Safari". The Safari field is characterized by multiple reservoir units, where the reservoirs are non-communicating and have unique properties.

In the specialization project, an automated field development methodology is established for Safari field (Angga, 2018). Some concepts of SUBPRO's automated development methodology are adapted, e.g., using production potential curves as the proxy model of the production system, using piecewise-linear models to represent non-linear functions, etc. The developed methodology is successful in prolonging the plateau duration of Safari field. It is done by optimizing the production splitting between the reservoirs. The optimization problem established in the specialization project is rather simple, where only the oil production rates are defined as the decision variables. In addition, only a few constraints are included for the optimization.

1.2 Objective

The primary objective of the thesis work is to further develop the automated field development methodology for Safari field. The further developments will be focused to:

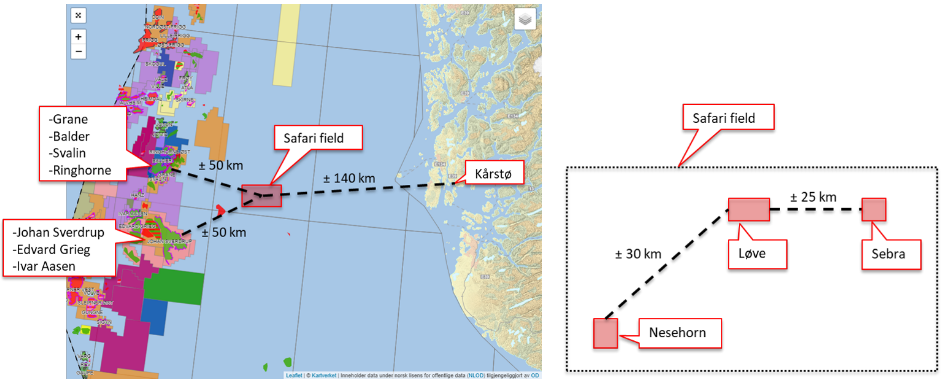
1. formulate & solve a more complex optimization problem which maximizes the plateau duration. What is meant by more complex is that the optimization includes more decision variables (e.g., production & injection rates, drilling schedule, and recovery mechanism) and considers more constraints (e.g., production constraints, injection constraints, and drilling constraints).
2. formulate & solve another optimization problem which maximizes the Net Present Value (NPV) instead of the plateau duration.
3. make the formulations of the optimization problems and the optimization process not only more efficient but also more accurate
4. analyze the effects of uncertainties in the optimization results. The uncertainty analyses are expected to take into account the uncertainties of in-place and oil price in addition to the uncertainty of development & operational cost.

1.3 Overview of Safari Field

Safari field is an offshore field located east of Johan Sverdrup and Grane fields and west of Haugesund (Figure 1.1a). Water depth in the field location is approximately 120 m. The field consists of three reservoirs, i.e. Løve, Nesehorn, and Sebra. Location of all reservoirs is shown in Figure 1.1b.

All the reservoirs are undersaturated oil reservoirs. Løve is characterized by nearly circular reservoir shape, highest oil-in-place, and heavier oil. Nesehorn and Sebra are almost identical in term of reservoir properties, except Sebra has much less oil-in-place but has higher permeability. All production wells will be completed with gas-lift as the artificial lift method. Fluid properties, rock properties, and well properties of the Safari field are summarized in Table 1.1, Table 1.2, and Table 1.3, respectively.

Development of the field will be commenced from the 1st 2019. Facility construction and installation are planned to be ready in four years, and the field first oil is expected to be in 2023. The field is scheduled for abandonment in 2040, after 18 years of production.



(a) Safari field (NPD, 2019)

(b) Reservoirs in Safari field

Figure 1.1: Location of Safari field and its reservoirs (Angga, 2018)

Table 1.1: Initial reservoir fluid properties (Angga, 2018)

Parameters	**Løve**	**Nesehorn**	**Sebra**	**Unit**
Oil density	933.99	850	850	kg/m^3
Gas density	0.8	0.75	0.75	sp. gravity
Solution GOR	50	150	150	sm^3/sm^3
Saturation pressure	174.4	257.3	257.3	bara
Reservoir pressure	280	280	280	bara
Reservoir temperature	80	80	80	^{o}C
STOOIP	75	55	13	M sm^3
Aquifer volume	30	20	20	M sm^3

Table 1.2: Reservoir rock properties (Angga, 2018)

Parameters	**Løve**	**Nesehorn**	**Sebra**	**Unit**
Permeability	250	250	450	mD
Porosity	0.18	0.18	0.18	
Reservoir thickness	50	50	50	m
Irreducible water saturation	0.25	0.25	0.25	
Residual oil saturation	0.25	0.25	0.25	
$k_{rw}(1-S_{or})$	0.8	0.8	0.8	
$k_{ro}(S_{wirr})$	0.8	0.8	0.8	
Corey exponent for water	1.5	1.5	1.5	
Corey exponent for oil	1.5	1.5	1.5	

Table 1.3: Well properties (Angga, 2018)

Parameters	**Løve**	**Nesehorn**	**Sebra**	**Unit**
Water depth	120	120	120	m
Well MD - TVD	3500 - 2500	3500 - 2500	3500 - 2500	m
Wellbore radius	0.12	0.12	0.12	m
Well drainage radius	800	800	800	m
Skin factor	+5	+5	+5	
Tubing ID	0.124	0.124	0.124	m
Artificial lift	gas lift (GL)	GL	GL	
GL valve depth	3000	3000	3000	m
GL gas gravity	0.7	0.7	0.7	sp. gravity
Max GL inj. rate	400	400	400	1000 sm^3/d

Chapter 2

Theoretical Framework

This chapter is presented to provide an understanding of the concepts and theories that are relevant to this study.

2.1 Fundamentals of Production Potential Curve

2.1.1 Definition of Production Potential

One can define the field production potential, $q_{pp}(t)$, as the maximum rate of preferred production phase (the maximum gas rate for a gas field and the maximum oil rate for an oil field) that a field can produce at a particular time. This maximum production rate is often achieved when the controllable components of the production system, such as choke, gas-lift injection rate, and so on, are operated at the optimal condition.

An illustration of field production potential is provided in Figure 2.1. To fully understand the definition of production potential, let us consider the following example based on Figure 2.1. At the beginning of the year 2026, the field production potential is around 1500 sm^3/d. This number simply pinpoints the upper limit of the field production rate at that time. However, in the illustration, the field is only produced at its plateau rate (1000 sm^3/d).

The production potential of a producing field typically declines over time. The production potential may step up if some modifications are made to the field, for instance, drilling new production wells, installing multi-phase booster, etc. The decline of production potential over time is often caused by the dynamic changes of inflow performance (IPR) and lift performance (VLP) during the life of the field (Stanko, 2019).

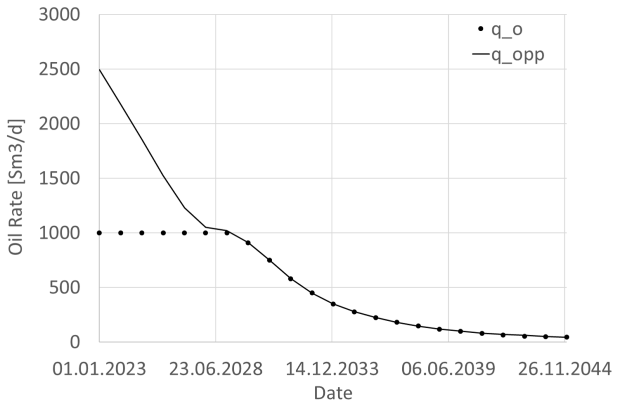


Figure 2.1: An illustration of field production potential. The solid line indicates the profile of field production potential, while the dots represent the profile of field production rate (Angga, 2018).

As previously discussed, the changes of IPR and VLP from time to time lead to the decline of production potential. Instead of time, the changes of IPR and VLP are in reality dependent mostly on the amount of fluid that has been produced. Thus, it can be deduced that the production potential at a given point in time, $q_{pp}(t)$, primarily depends on the cumulative production up to that point in time, $N_p(t)$ (Stanko, 2019).

Production potential curve is defined as a curve which relates the production potential with the cumulative production. An example of production potential curve is given in Figure 2.2. It is important to note that a single production potential curve represents a particular production system, i.e., one particular reservoir produced with a particular number of wells and a particular recovery mechanism. Besides that, the time-dependency of production potential is successfully eliminated in that production potential curve.

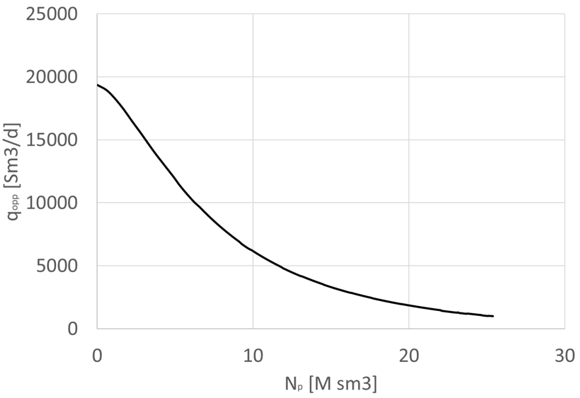


Figure 2.2: An example of production potential curve (Angga, 2018)

2.1.2 Construction of Production Potential Curve

To generate a production potential curve, one has to initially create the model of a production system. After the model is established, reservoir and network simulations are performed simultaneously to obtain the profile of production potential (q_{pp} vs. t). In the simulations, one has to ensure that the field will always produce the maximum rate, i.e., by ensuring the optimal operating conditions (e.g. optimal choke opening, optimal gas-lift injection rate, etc.) and by discarding all production constraints (e.g. maximum liquid rate, maximum gas rate, erosional velocity, etc.).

The following step is to compute the profile of cumulative production over time (N_p vs t) based on the profile of production potential (q_{pp} vs. t). The cumulative production at a particular time is defined as:

$$N_p(t) = \int_0^t q_{pp}(t) \cdot dt \tag{2.1}$$

Finally, the production potential curve is obtained by plotting the production potential q_{pp} against the cumulative production N_p. Here, the production potential curve is provided as a collection of points. An illustration of generating the production potential curve is given in Figure 2.3.

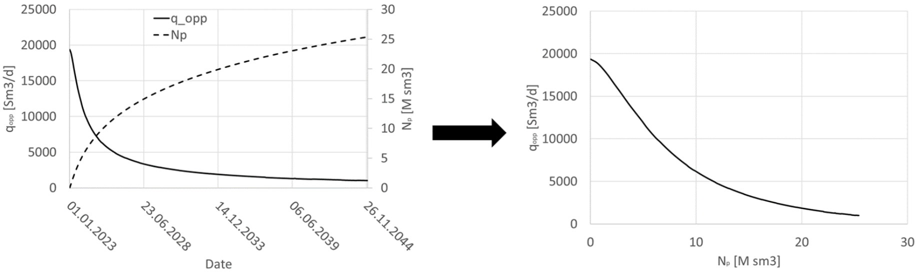


Figure 2.3: An illustration of generating production potential curve (Angga, 2018)

2.1.3 Planning the Production Schedule Using the Production Potential Curve

Planning the production schedule using the production potential curve is a straightforward routine. At any time, the production rate $q(t_i)$ can be any number as long as it fulfills the following condition:

$$0 \leq q(t_i) \leq q_{pp}\Big(N_p(t_i)\Big) \tag{2.2a}$$

where

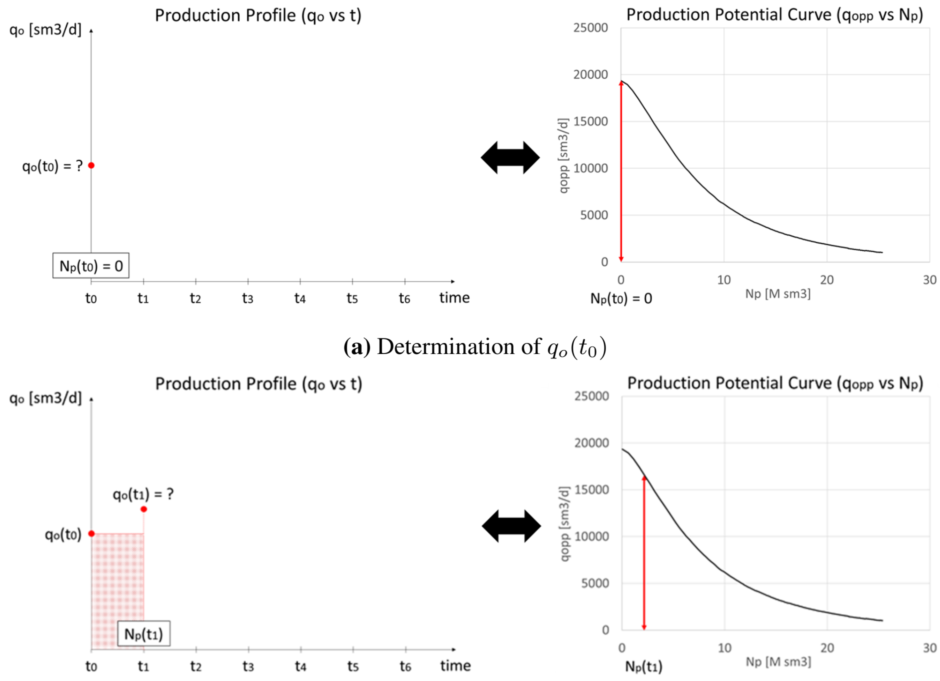
$$N_p(t_i) = \int_0^{t_i} q(t) \cdot dt \tag{2.2b}$$

If one discretizes the timespan of the production period, the cumulative production at time t_i can be approximated using the rectangular integration technique as the following:

$$N_p(t_i) = N_p(t_{i-1}) + q(t_{i-1}) \cdot (t_i - t_{i-1}) \tag{2.2c}$$

To get a better understanding, let us consider an illustration provided in Figure 2.4. Initially (time t_0), we know that there is no oil has been produced. Thus, by referring to the production potential curve, the production rate at time t_0 can be chosen within a range of 0 - 19000 sm^3/d (as indicated by the red line in Figure 2.4a).

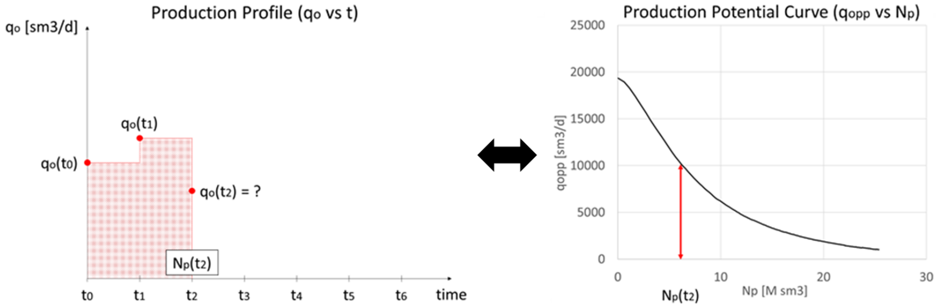
Once $q_o(t_0)$ has been selected, the cumulative production at time t_1 can be estimated by applying Equation 2.2c. Note that the numerical integration technique in this equation assumes that $q_o(t_0)$ remains constant until time t_1. Afterward, again by referring to the production potential curve, the production rate at time t_1 is selected between 0 and 16000 sm^3/d (as indicated by the red line in Figure 2.4b). The procedure is carried out until the end of the defined field lifetime.



(a) Determination of $q_o(t_0)$

(b) Determination of $q_o(t_1)$

Figure 2.4: An illustration of planning the production schedule using the production potential curve



(c) Determination of $q_o(t_2)$

Figure 2.4: An illustration of planning the production schedule using the production potential curve (cont.)

Angga (2018) has validated the utilization of the production potential curve for planning the production schedule. The validation is accomplished by comparing the results obtained from the production potential approach against the results retrieved from a full reservoir-network simulation. It is found that the production potential approach provides very similar results as the full simulation run.

2.1.4 Characteristics of Production Potential Curve

Some characteristics of the production potential curve are presented as the following (Angga, 2018). These characteristics work for most cases, but not always.

1. If there is no abrupt human intervention on either the reservoir or the production system, the production potential curve is continuous. These interventions consist of performing well stimulation, introducing pressure support, implementing EOR, adding more wells, modifying well completion, changing tubing/pipeline diameter, switching artificial-lift method, varying separator pressure, etc.

2. Reservoir pressure, water cut (WC), gas-oil ratio (GOR), and injection-production ratio can be expressed as functions of cumulative oil production (N_p). With these properties, cumulative production/injection of water & gas (W_p, G_p, W_i, G_i) can also be expressed as functions of cumulative oil production.

3. If a field has a fully-connected reservoir, the field production potential is unique for a given field cumulative production.

4. If the field is composed of two or more non-communicating reservoirs, the production potential of a reservoir is solely dependent on the cumulative production of that reservoir. In addition, the field production potential is not unique for a given field cumulative production. It depends on the production schedule of each reservoir.

5. If the field is composed of two or more non-communicating reservoirs and the production from all reservoirs are tied to a non-fixed pressure node, the production potential of each reservoir is dependent on the production of the other reservoirs.

2.2 Mathematical Optimization

Mathematical optimization is defined as the mathematical process of finding the best solutions -a combination of decision variables- that maximize or minimize the objective function while honoring a defined set of constraints. There are three main components of an optimization problem, i.e., objective function, decision variables, and constraints. The objective function is basically an equation to be maximized or minimized through the optimization. The decision variables are the quantities to be varied and decided upon in order to maximize or minimize the objective function. The constraints declare the limitation on the decisions to be made. The type of optimization problems and the common algorithms for solving the optimization problems are explained briefly in the following subsections.

2.2.1 Linear Programming

Definition

One defines Linear Programming (LP) as the optimization of a linear objective function subject to a set of linear constraints. The standard form of LP is expressed as follows (Cormen et al., 2009):

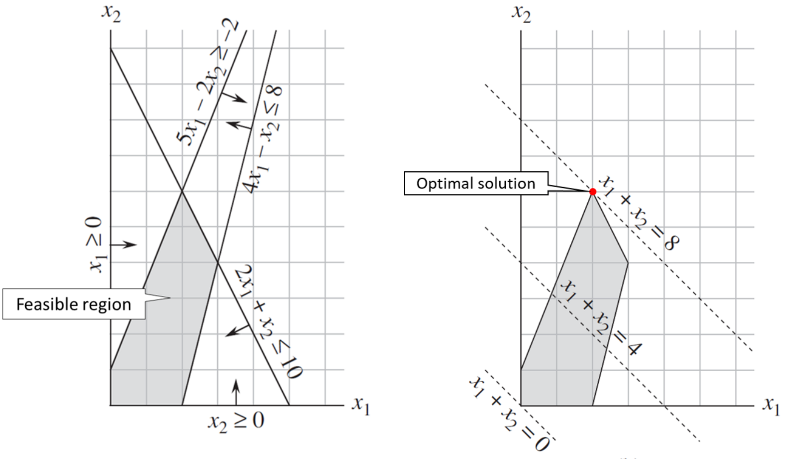
$$\begin{aligned} \text{maximize} \quad & c^T x && \text{(2.3a)} \\ \text{subject to} \quad & Ax \leq b && \text{(2.3b)} \\ & x \geq 0 && \text{(2.3c)} \end{aligned}$$

where c and x are vectors in $\mathbb{R}^n$, b is a vector in $\mathbb{R}^m$, and A is an $m \times n$ matrix.

Feasible Solution, Feasible Region, and Optimal Solution

Feasible solution refers to a set of values for the decision variables which satisfies the entire constraints included in the optimization problem. A collection of feasible solutions forms a region called feasible region. For an LP, the feasible region is always a convex polytope (Hoffmann, 2014). With linear objective function and convex feasible region, the optimal solution of an LP is always located at the boundary of the feasible region (e.g., a vertex, a line segment, or a plane). In addition, linear objective function and convex feasible region ensure the global optimality of the optimal solution.

An illustration of an LP with two decision variables (x_1 and x_2) is provided in Figure 2.5. As shown in Figure 2.5a, every constraint of the LP is depicted with a straight line and its direction. The intersection of the semispaces creates the feasible region as indicated by the grey area. Figure 2.5b points out the location of the optimum solution. Correspond to the previous explanation, the optimal solution lies on the vertex of the feasible region.



(a) Feasible region

(b) Optimal solution

Figure 2.5: Visualization of an LP (Cormen et al., 2009)

Existence of the Optimal Solution

There are two conditions for LP to have no optimal solution, i.e., it has no feasible solution -often called infeasible problem- or it has feasible solutions with an extremely large value of the objective function -generally called unbounded problem. For infeasible problem, there is no solution that fulfills all the constraints. This occurs when an LP has two contradictory constraints. On the other hand, the unbounded problem happens when the upper bound for the value of the objective function does not exist. This means that the value of the objective function can grow to an arbitrarily large value.

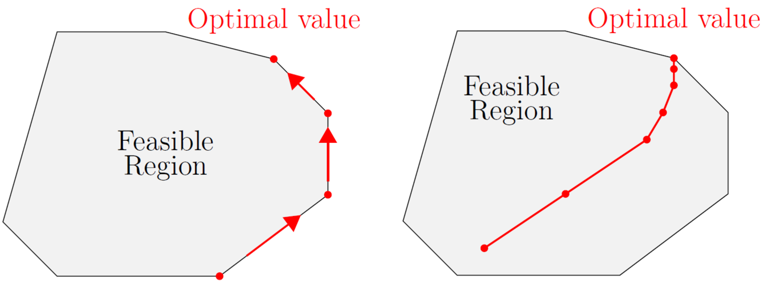
2.2.2 Algorithms to Solve Linear Programming

Simplex Algorithm

Dantzig (1951) develops the simplex algorithm to solve an LP. The algorithm starts from one vertex of the feasible region. In each iteration, the algorithm evaluates the value of the objective function of the neighboring vertices and then moves the incumbent solution to the adjacent vertex with the highest value of the objective function. The iterations continue until it arrives at a vertex in which all the neighboring vertices have smaller values of the objective function. The simplex algorithm finally returns this vertex as the optimal solution of the LP. An illustration of the simplex algorithm is shown in Figure 2.6a.

Interior-Point Method

The interior-point method is another algorithm commonly used to solve an LP. The algorithm is developed by Karmarkar (1984). In contrary to the simplex algorithm, the interior-point method finds the optimal solution by traveling through the interior of the feasible region. An illustration of the interior-point method is provided in Figure 2.6b.



(a) Simplex algorithm

(b) Interior-point method

Figure 2.6: Illustration of simplex algorithm and interior-point method (Hoffmann, 2014)

2.2.3 Integer Programming

Definition

Integer programming (IP) is a kind of optimization problem in which all the decision variables are restricted to be integers. Generally, IP refers to integer linear programming (ILP) in which the objective function and the constraints are linear. The standard form of IP is expressed as follows (Camponogara, 2018):

$$\begin{aligned} \text{maximize} \quad & c^T x && (2.4a)\\ \text{subject to} \quad & Ax \leq b && (2.4b)\\ & x \geq 0 && (2.4c)\\ & x \in \mathbb{Z}^n && (2.4d) \end{aligned}$$

where c is vectors in $\mathbb{R}^n$, b is a vector in $\mathbb{R}^m$, and A is an $m \times n$ matrix.

Feasible Solutions & LP Relaxation

An illustration of an IP with two integer variables x_1 and x_2 is provided in Figure 2.7. In this illustration, the constraints are represented by the solid black lines, while the objective function is represented by the solid red line. The feasible solutions for this IP are indicated by the black dots, and the optimal solution is indicated by the blue dot (objective value = 68).

LP relaxation of an IP is obtained by replacing the integrity constraints (in this illustration, $x_1, x_2 \in \mathbb{Z}$) with appropriate continuous constraints ($x_1, x_2 \in \mathbb{R}$). In the illustration, the feasible region of the LP relaxation is indicated by the darker grey area, and the optimal solution of the LP relaxation is indicated by the green dot (objective value = 68.33). Note that in LP relaxation, we perform an optimization over a larger set of solutions. Therefore, the optimal solution of LP relaxation is always better than or equal to the optimal solution of the initial IP (Trevisan, 2011).

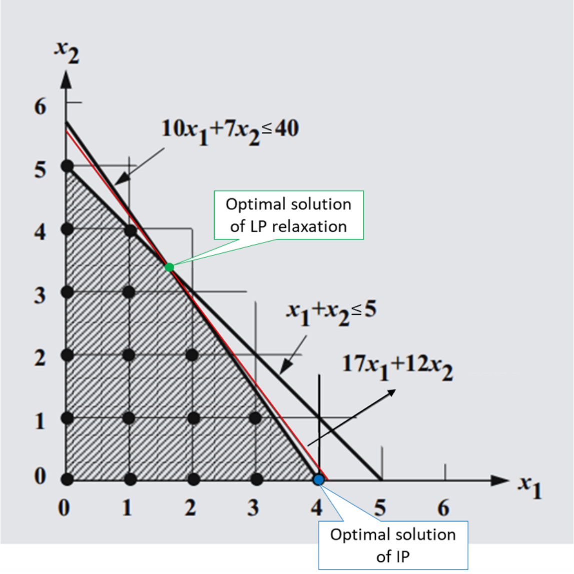


Figure 2.7: Visualization of IP and its LP relaxation (Vanderbei, 2008)

2.2.4 Mixed-Integer Linear Programming

Definition

Mixed-integer linear programming (MILP) is an LP in which some of the variables are constrained to be integers while the rests are continuous variables. The standard form of MILP is expressed as follows (Camponogara, 2018):

$$\begin{aligned} \text{maximize} \quad & c^T x && \text{(2.5a)} \\ \text{subject to} \quad & Ax \leq b && \text{(2.5b)} \\ & x = (x_C, x_I) \geq 0 && \text{(2.5c)} \\ & x_C \in \mathbb{R}^k && \text{(2.5d)} \\ & x_I \in \mathbb{Z}^{n-k} && \text{(2.5e)} \end{aligned}$$

where c is vectors in $\mathbb{R}^n$, b is a vector in $\mathbb{R}^m$, and A is an $m \times n$ matrix.

2.2.5 Algorithms to Solve IP or MILP

Branch-and-Bound Algorithm

The branch-and-bound algorithm is developed by Land and Doig (1960) for solving an IP or MILP. The concept of this algorithm is the divide-and-conquer strategy. Let us consider an IP or MILP problem P with a feasible region S. At first, the algorithm divides the the problem P into an equivalent set of sub-problems $\{SP_k\}$. The sub-problem SP_k is defined on a feasible region S_k, with $\bigcup_k S_k = S$. After solving the sub-problems, the optimal solution for problem P is obtained by comparing the optimal solutions for sub-problems SP_k. The divisions of problem P are carried out iteratively such that the sub-problems SP_k are easier to solve. In addition, some sub-problems are eliminated if they can be proven that they will not produce the optimal solution for problem P.

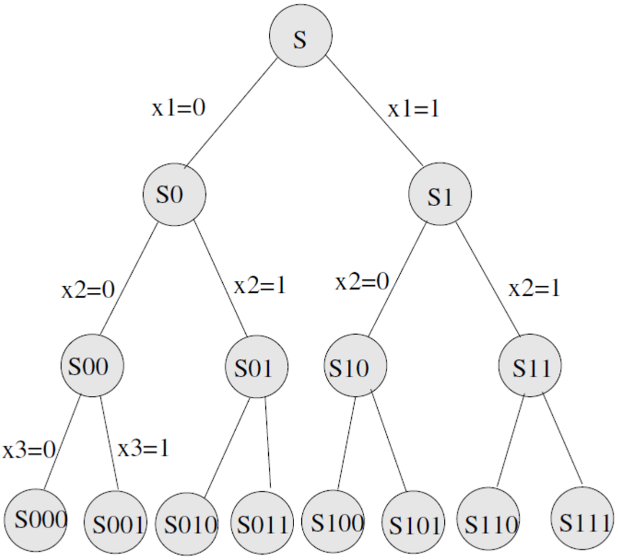


Figure 2.8: An illustration of branch-and-bound algorithm (Camponogara, 2018)

An illustration of the branch-and-bound algorithm is provided in Figure 2.8. In this example, there are three binary variables (x_1, x_2, and x_3) to be decided. In addition, Figure 2.8 illustrates a complete tree enumeration, which means that it sketches all possible tree branches. In practice, the complete tree enumeration is not preferred due to the efficiency issue. Therefore, to improve the efficiency, one needs to prune the sub-problems that are proven not to produce the optimal solution for the initial problem. The rules for cutting the tree branches are listed as follows:

1. by optimality → the optimal solution of sub-problem SP_k has been found, which means the solution cannot be improved by further decomposing the formulation and adding bounds.
2. by bounding → the optimal solution of sub-problem SP_k relaxation is worse than the best-known solution for problem P.
3. by infeasibility → the sub-problem SP_k has no feasible solution.

Cutting Plane Algorithm

To illustrate the cutting plane algorithm, let us consider an IP, as shown in Figure 2.9. In this illustration, there are two integer variables to be decided (y_1 and y_2). The IP is defined on a feasible region bounded by the solid green line. The optimal solution for LP relaxation of the IP is indicated by the point y^R.

The convex hull of the feasible integer solutions is indicated by the dashed black line. By having the convex hull, one can find the optimal solution for the IP by solving an LP defined on the convex hull. In this illustration, the optimal solution for the IP is indicated by the point y^Z.

To approximate the convex hull, one needs to generate valid and non-trivial inequalities. For example, two inequalities which are depicted by the red and the blue lines in Figure 2.9 have been introduced into the initial IP. As shown in the figure, these inequalities cut the feasible region of the initial IP, and make it tighter towards the convex hull. Note that the optimal solution for LP relaxation of the new IP (the initial IP with two additional inequalities) is equal to the optimal solution of the initial IP, i.e., the point y^Z.

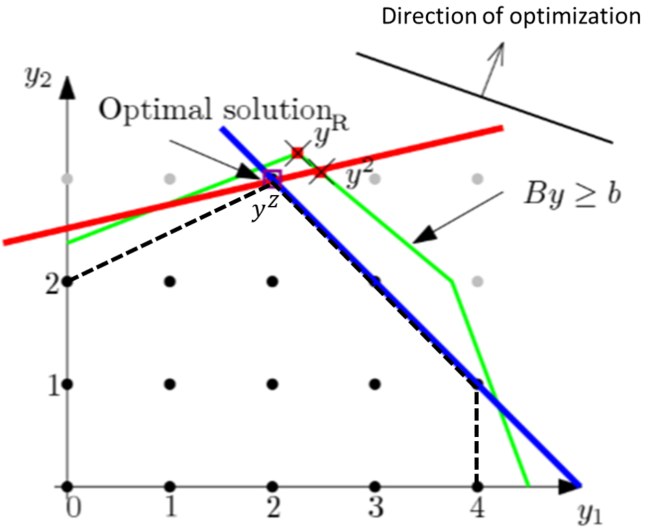


Figure 2.9: An illustration of cutting plane algorithm (NTNU, 2019)

A few methods that have been developed to generate valid inequalities of an IP or MILP are listed as follows (Cornuejols, 2007):

- Chavatal-Gomory cuts for IP (Gomory, 1963)
- Gomory cuts for MILP
- Mixed-integer rounding inequalities
- Lift-and-project
- Cover inequalities
- Split and intersection cuts

2.3 Piecewise Linear (PWL) Approximation

Non-linear behaviours appear in countless real-world problems. An optimization problem which involves non-linear function(s) is called Non-Linear Programming (NLP). Today, this class of optimization problem often remains difficult to solve. Due to the irregularity behaviour of the non-linear function, some challenges emerge when solving an NLP, such as difficulties to verify the global optimal, high running time, and so on.

PWL approximation is a well-known yet powerful approach to solve an NLP. The fundamental of PWL is to replace the non-linear function with a set of linear functions. A function that containts a set of linear functions is called as PWL function. Using PWL approximation, one can transform an NLP into an MILP, which is easier to solve.

2.3.1 One-Dimensional PWL Approximation

Let us consider a one-dimensional non-linear function $f(x)$. This function is denoted by the solid black line in Figure 2.10. The function is defined on domain $\mathcal{D}$, where $\mathcal{D} = \{x \in [x_1, x_4],\ x \in \mathbb{R}\}$. A PWL function that approximates function $f(x)$ is represented by the dotted blue line in Figure 2.10. To produce this PWL function, one has to divide the domain $\mathcal{D}$ into several breakpoints. In this example, those breakpoints are x_1, x_2, x_3, and x_4. Linear functions are then generated for every two consecutive breakpoints. The PWL function $g(x)$ for this example is expressed as follows:

$$g(x) = \begin{cases} g_{1,2}(x), & \text{if } x \in [x_1, x_2],\ x \in \mathbb{R} \\ g_{2,3}(x), & \text{if } x \in [x_2, x_3],\ x \in \mathbb{R} \\ g_{3,4}(x), & \text{if } x \in [x_3, x_4],\ x \in \mathbb{R} \end{cases} \tag{2.6}$$

where $g_{i,i+1}(x)$ is a linear function valid for any x within $[x_i, x_{i+1}]$.

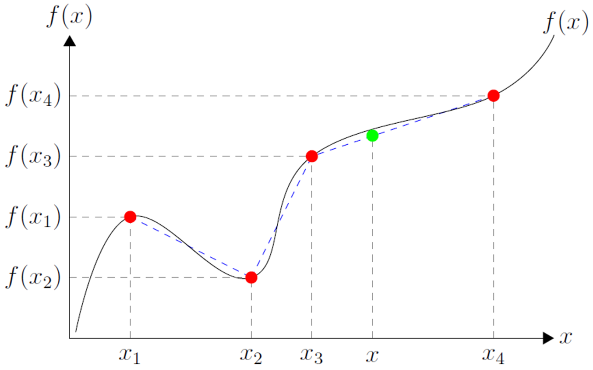


Figure 2.10: A non-linear function and its PWL function (Hoffmann, 2014)

The fundamental of constructing the linear function $g_{i,i+1}(x)$ is a linear interpolation between point $(x_i, f(x_i))$ and $(x_{i+1}, f(x_{i+1}))$. Hoffmann (2014) showed that $g_{i,i+1}(x)$ can be expressed as follows:

$$g_{i,i+1}(x) = \lambda_i f(x_i) + \lambda_{i+1} f(x_{i+1}) \tag{2.7a}$$

where

$$x = \lambda_i x_i + \lambda_{i+1} x_{i+1} \tag{2.7b}$$
$$1 = \lambda_i + \lambda_{i+1} \tag{2.7c}$$
$$\lambda_i, \lambda_{i+1} \geq 0 \tag{2.7d}$$
$$\lambda_i, \lambda_{i+1} \in \mathbb{R} \tag{2.7e}$$

In Equation 2.7, λ_i can be interpreted as the weighting coefficient of breakpoint-i.

By combining Equation 2.7 with Equation 2.6, the PWL function $g(x)$ can be expressed as follows:

$$g(x) = \lambda_1 f(x_1) + \lambda_2 f(x_2) + \lambda_3 f(x_3) + \lambda_4 f(x_4) \tag{2.8a}$$

where

$$x = \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_4 + \lambda_4 x_4 \tag{2.8b}$$
$$1 = \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 \tag{2.8c}$$
$$\lambda = \{\lambda_1, \lambda_2, \lambda_3, \lambda_4\} \geq 0 \tag{2.8d}$$
$$\lambda = \{\lambda_1, \lambda_2, \lambda_3, \lambda_4\} \in \mathbb{R}^4 \tag{2.8e}$$

To ensure that the linear interpolation only involves two consecutive breakpoints, we have another condition for the set λ, i.e. at most two elements of set λ are positive and the two positive elements must be consecutive in the ordered set. For example, when approximating the green dot within Figure 2.10, only λ_3 and λ_4 are greater than zero, while λ_1 and λ_2 equal to zero.

Generalization of the previous example defines the standard expression of one-dimensional PWL approximation for a non-linear function $f(x)$ as follows:

$$f(x) \approx g(x) \tag{2.9a}$$

$$\approx \sum_{i=1}^{N} \lambda_i \cdot f(x_i) \tag{2.9b}$$

where N is the number of breakpoints and

$$x = \sum_{i=1}^{N} \lambda_i \cdot x_i \tag{2.9c}$$

$$1 = \sum_{i=1}^{N} \lambda_i \tag{2.9d}$$

$$\lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_N\} \geq 0 \tag{2.9e}$$

$$\lambda \in \mathbb{R}^N \tag{2.9f}$$

$$\lambda \text{ is SOS2} \tag{2.9g}$$

The SOS2 (Special Ordered Set type 2) condition for set λ verifies that:

1. at most two elements of set λ are positive

2. if two elements are positive, they must be consecutive in the ordered set, let us say λ_i and λ_{i+1}.

There is another condition for set λ, i.e., SOS1 (Special Ordered Set type 1). This condition restricts that at most one element of set λ is positive. If one uses SOS1 condition for set λ, the input for the PWL function $g(x)$ must be located at one of the breakpoints. For the previous example, x must be the element of a set $\{x_1, x_2, x_3, x_4\}$.

2.3.2 Two-Dimensional PWL Approximation

Let $f(x, y)$ be a two-dimensional non-linear function defined on a domain $\mathcal{D} = \{x \in [x_1, x_4],\ y \in [y_1, y_4],\ x, y \in \mathbb{R}\}$ (see Figure 2.11). To make a PWL function $g(x, y)$ that models $f(x, y)$, one needs to define some breakpoints in x-direction and y-direction. In this illustration, the value of $f(x, y)$ is known for a set of breakpoints $\{B_{i,j} = (x_i, y_j), i = 1, 2, \ldots, 4, j = 1, 2, \ldots, 4\}$ (see Figure 2.11). Linear functions are then generated for every combination of two consecutive breakpoints in x-direction and two consecutive breakpoints in y-direction. The PWL function $g(x, y)$ for this example is expressed as follows:

$$g(x,y) = \begin{cases} g_{1,2,1,2}(x,y), & \text{if } x \in [x_1, x_2],\ y \in [y_1, y_2],\ \{x, y\} \in \mathbb{R}^2 \\ g_{2,3,1,2}(x,y), & \text{if } x \in [x_2, x_3],\ y \in [y_1, y_2],\ \{x, y\} \in \mathbb{R}^2 \\ g_{3,4,1,2}(x,y), & \text{if } x \in [x_3, x_4],\ y \in [y_1, y_2],\ \{x, y\} \in \mathbb{R}^2 \\ g_{1,2,2,3}(x,y), & \text{if } x \in [x_1, x_2],\ y \in [y_2, y_3],\ \{x, y\} \in \mathbb{R}^2 \\ g_{2,3,2,3}(x,y), & \text{if } x \in [x_2, x_3],\ y \in [y_2, y_3],\ \{x, y\} \in \mathbb{R}^2 \\ g_{3,4,2,3}(x,y), & \text{if } x \in [x_3, x_4],\ y \in [y_2, y_3],\ \{x, y\} \in \mathbb{R}^2 \\ g_{1,2,3,4}(x,y), & \text{if } x \in [x_1, x_2],\ y \in [y_3, y_4],\ \{x, y\} \in \mathbb{R}^2 \\ g_{2,3,3,4}(x,y), & \text{if } x \in [x_2, x_3],\ y \in [y_3, y_4],\ \{x, y\} \in \mathbb{R}^2 \\ g_{3,4,3,4}(x,y), & \text{if } x \in [x_3, x_4],\ y \in [y_3, y_4],\ \{x, y\} \in \mathbb{R}^2 \end{cases} \tag{2.10}$$

where $g_{i,i+1,j,j+1}(x, y)$ is a linear function valid for any x within $[x_i, x_{i+1}]$ and y within $[y_j, y_{j+1}]$. For example, $g_{1,2,1,2}(x, y)$ is valid for any x within $[x_1, x_2]$ and y within $[y_1, y_2]$. The domain of this linear function is visualized by the grey area in Figure 2.11.

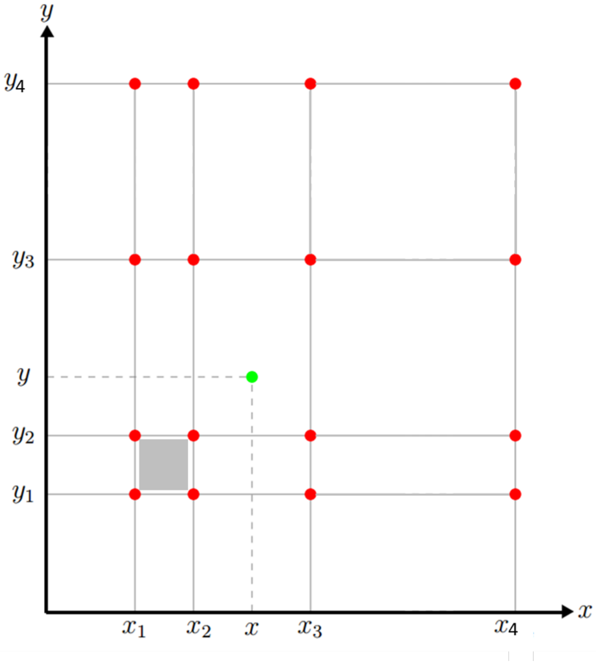


Figure 2.11: An illustration of two-dimensional PWL grid (Hoffmann, 2014)

Similarly, the fundamental of constructing the linear function $g_{i,i+1,j,j+1}(x, y)$ is a linear interpolation. For two-dimensional linear interpolation, four breakpoints are employed, i.e., $B_{i,j}$; $B_{i+1,j}$; $B_{i,j+1}$; and $B_{i+1,j+1}$. Hoffmann (2014) showed that $g_{i,i+1,j,j+1}(x, y)$

can be expressed as follows:

$$g_{i,i+1,j,j+1}(x,y) = \lambda_{i,j} f(B_{i,j}) + \lambda_{i+1,j} f(B_{i+1,j}) + \lambda_{i,j+1} f(B_{i,j+1}) + \lambda_{i+1,j+1} f(B_{i+1,j+1}) \tag{2.11a}$$

where

$$x = \lambda_{i,j} \cdot x_i + \lambda_{i+1,j} \cdot x_{i+1} + \lambda_{i,j+1} \cdot x_i + \lambda_{i+1,j+1} \cdot x_{i+1} \tag{2.11b}$$

$$y = \lambda_{i,j} \cdot y_j + \lambda_{i+1,j} \cdot y_j + \lambda_{i,j+1} \cdot y_{j+1} + \lambda_{i+1,j+1} \cdot y_{j+1} \tag{2.11c}$$

$$1 = \lambda_{i,j} + \lambda_{i+1,j} + \lambda_{i,j+1} + \lambda_{i+1,j+1} \tag{2.11d}$$

$$\lambda_{i,j}; \lambda_{i+1,j}; \lambda_{i,j+1}; \lambda_{i+1,j+1} \geq 0 \tag{2.11e}$$

$$\lambda_{i,j}; \lambda_{i+1,j}; \lambda_{i,j+1}; \lambda_{i+1,j+1} \in \mathbb{R} \tag{2.11f}$$

In Equation 2.11, $\lambda_{i,j}$ can be interpreted as the weighting coefficient of breakpoint $B_{i,j}$.

By combining Equation 2.11 with Equation 2.10, the PWL function $g(x, y)$ can be expressed as follows:

$$g(x,y) = \sum_{i=1}^{4} \sum_{j=1}^{4} \lambda_{i,j} \cdot f(x_i, y_j) \tag{2.12a}$$

where

$$x = \sum_{i=1}^{4} \sum_{j=1}^{4} \lambda_{i,j} \cdot x_i \tag{2.12b}$$

$$y = \sum_{i=1}^{4} \sum_{j=1}^{4} \lambda_{i,j} \cdot y_j \tag{2.12c}$$

$$1 = \sum_{i=1}^{4} \sum_{j=1}^{4} \lambda_{i,j} \tag{2.12d}$$

$$\lambda = \{\lambda_{i,j}, i = 1, 2, \ldots, 4, j = 1, 2, \ldots, 4\} \geq 0 \tag{2.12e}$$

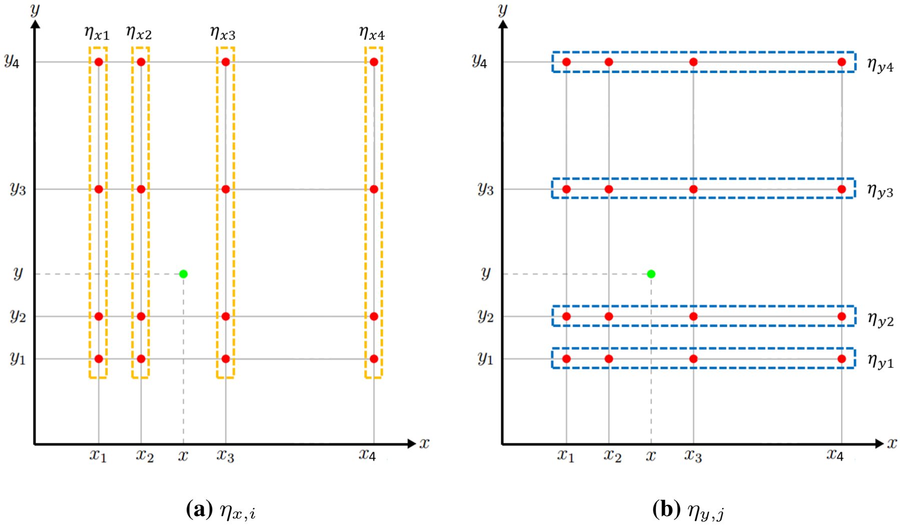
$$\lambda \in \mathbb{R}^{4 \times 4} \tag{2.12f}$$

In addition, we define sets $\{\eta_{x,i}, i = 1, 2, \ldots, 4\}$ and $\{\eta_{y,j}, j = 1, 2, \ldots, 4\}$ as follows:

$$\eta_{x,i} = \sum_{j=1}^{4} \lambda_{i,j}, \quad \forall i \in \{1, 2, \ldots, 4\} \tag{2.12g}$$

$$\eta_{y,j} = \sum_{i=1}^{4} \lambda_{i,j}, \quad \forall j \in \{1, 2, \ldots, 4\} \tag{2.12h}$$

One can find the visualization of both terms $\eta_{x,i}$ and $\eta_{y,j}$ in Figure 2.12. To ensure that the linear interpolation only involves two consecutive breakpoints in x-direction and two consecutive breakpoints in y-direction (i.e., breakpoints $B_{i,j}$, $B_{i+1,j}$, $B_{i,j+1}$, and $B_{i+1,j+1}$), both sets $\{\eta_{x,i}, i = 1, 2, \ldots, 4\}$ and $\{\eta_{y,j}, j = 1, 2, \ldots, 4\}$ must be SOS2.



(a) $\eta_{x,i}$ **(b)** $\eta_{y,j}$

Figure 2.12: Visualization of $\eta_{x,i}$ and $\eta_{y,j}$

To clarify the previous statement, let consider an illustration given in Figure 2.13. In this example, we want to estimate the value of $f(x, y)$ at the green dot shown in the figure. When performing a linear interpolation, we know that we only want to involve the adjacent breakpoints indicated by the purple dots. This means that we need only λ of these breakpoints may be greater than zero, while λ of the other breakpoints equal to zero. Consequently, we require only $\eta_{x,2}$, $\eta_{x,3}$, $\eta_{y,2}$, and $\eta_{y,3}$ are positive. This condition is attained when both sets $\{\eta_{x,i}\}$ and $\{\eta_{y,j}\}$ are SOS2.

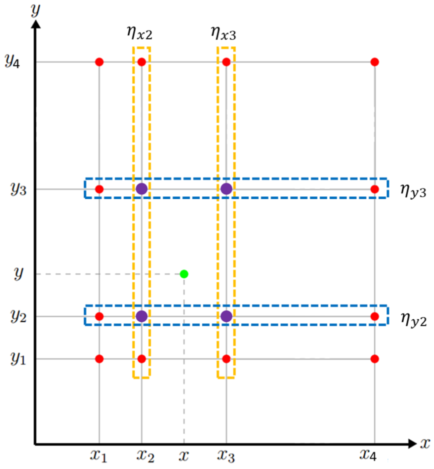


Figure 2.13: An illustration of SOS2 constraint for both sets $\{\eta_{x,i}, i = 1, 2, \ldots, 4\}$ and $\{\eta_{y,j}, j = 1, 2, \ldots, 4\}$ when carrying out a linear interpolation

If one uses SOS1 constraint for the sets $\{\eta_{x,i}\}$ or $\{\eta_{y,j}\}$, the inputs for the PWL function $g(x, y)$ are restricted to the breakpoints used for the PWL approximation. For example, let us consider the set $\{\eta_{y,j}\}$ to be SOS1. This constraint imposes that the input y for $g(x, y)$ can only be selected from the breakpoints defined for the y-direction. In the previous illustration, y must be the element of a set $\{y_1, y_2, y_3, y_4\}$.

Generalization of the previous example defines the standard expression of two-dimensional PWL approximation for a non-linear function $f(x, y)$ as follows:

$$f(x,y) \approx g(x,y) \tag{2.13a}$$

$$\approx \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \lambda_{i,j} \cdot f(x_i, y_j) \tag{2.13b}$$

where N_x and N_y are the numbers of breakpoints in x-direction and in y-direction, respectively. In addition, the following conditions must be fulfilled.

$$x = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \lambda_{i,j} \cdot x_i \tag{2.13c}$$

$$y = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \lambda_{i,j} \cdot y_j \tag{2.13d}$$

$$1 = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \lambda_{i,j} \tag{2.13e}$$

$$\lambda = \{\lambda_{i,j}, i = 1, 2, \ldots, N_x, j = 1, 2, \ldots, N_y\} \geq 0 \tag{2.13f}$$

$$\lambda \in \mathbb{R}^{N_x \times N_y} \tag{2.13g}$$

$$\eta_{x,i} = \sum_{j=1}^{N_y} \lambda_{i,j}, \quad \forall i \in \{1, 2, \ldots, N_x\} \tag{2.13h}$$

$$\eta_{y,j} = \sum_{i=1}^{N_x} \lambda_{i,j}, \quad \forall j \in \{1, 2, \ldots, N_y\} \tag{2.13i}$$

$$\eta_x = \{\eta_{x,i}, i = 1, 2, \ldots, N_x\} \text{ is SOS2} \tag{2.13j}$$

$$\eta_y = \{\eta_{y,j}, j = 1, 2, \ldots, N_y\} \text{ is SOS2} \tag{2.13k}$$

2.3.3 Three-Dimensional PWL Approximation

Let $f(x, y, z)$ be a three-dimensional non-linear function defined on a domain $\mathcal{D} = \{x \in [x_1, x_{N_x}],\ y \in [y_1, y_{N_y}],\ z \in [z_1, z_{N_z}],\ x, y, z \in \mathbb{R}\}$. The concept of three-dimensional PWL approximation is pretty much the same as the two-dimensional PWL approximation. Assume that the value of $f(x, y, z)$ is known for a set of breakpoints $\{B_{i,j,k} = (x_i, y_j, z_k), i = 1, 2, \ldots, N_x, j = 1, 2, \ldots, N_y, k = 1, 2, \ldots, N_z\}$. The standard expression of three-dimensional PWL approximation for a non-linear function $f(x, y, z)$ is provided as follows:

$$f(x,y,z) \approx g(x,y,z) \tag{2.14a}$$

$$\approx \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \sum_{k=1}^{N_z} \lambda_{i,j,k} \cdot f(x_i, y_j, z_k) \tag{2.14b}$$

where N_x, N_y, and N_z are the numbers of breakpoints in x-direction, in y-direction, and in z-direction, respectively. In addition, the following conditions must be fulfilled.

$$x = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \sum_{k=1}^{N_z} \lambda_{i,j,k} \cdot x_i \tag{2.14c}$$

$$y = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \sum_{k=1}^{N_z} \lambda_{i,j,k} \cdot y_j \tag{2.14d}$$

$$z = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \sum_{k=1}^{N_z} \lambda_{i,j,k} \cdot z_k \tag{2.14e}$$

$$1 = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \sum_{k=1}^{N_z} \lambda_{i,j,k} \tag{2.14f}$$

$$\lambda = \{\lambda_{i,j,k}, i = 1, 2, \ldots, N_x, j = 1, 2, \ldots, N_y, k = 1, 2, \ldots, N_z\} \geq 0 \tag{2.14g}$$

$$\lambda \in \mathbb{R}^{N_x \times N_y \times N_z} \tag{2.14h}$$

$$\eta_{x,i} = \sum_{j=1}^{N_y} \sum_{k=1}^{N_z} \lambda_{i,j,k}, \quad \forall i \in \{1, 2, \ldots, N_x\} \tag{2.14i}$$

$$\eta_{y,j} = \sum_{i=1}^{N_x} \sum_{k=1}^{N_z} \lambda_{i,j,k}, \quad \forall j \in \{1, 2, \ldots, N_y\} \tag{2.14j}$$

$$\eta_{z,k} = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \lambda_{i,j,k}, \quad \forall k \in \{1, 2, \ldots, N_z\} \tag{2.14k}$$

$$\eta_x = \{\eta_{x,i}, i = 1, 2, \ldots, N_x\} \text{ is SOS2} \tag{2.14l}$$

$$\eta_y = \{\eta_{y,j}, j = 1, 2, \ldots, N_y\} \text{ is SOS2} \tag{2.14m}$$

$$\eta_z = \{\eta_{z,k}, j = 1, 2, \ldots, N_z\} \text{ is SOS2} \tag{2.14n}$$

2.4 Latin Hypercube Sampling (LHS)

In uncertainty analysis, sampling is used to generate possible values from probability distributions. Each set of samples represents a possible combination for the inputs. Sampling technique affects not only the quality of the uncertainty analysis, but also the time necessitated to complete the analysis.

Random sampling or Monte Carlo sampling is a conventional sampling technique which generates samples from a probability distribution using random numbers. With a huge number of iterations, random sampling is able to recreate the input probability distribution. However, we may encounter a problem of clustering when only a few samples are generated (UiO, 2019). The implication of this problem is that the samples taken do not represent the input probability distribution properly, and it makes the results of the uncertainty analysis less accurate.

Numerous techniques have been developed to resolve the drawback of random sampling. One of those technique is Latin Hypercube sampling (LHS), which generates controlled random samples. This sampling method can recreate the input probability distribution through fewer iterations when compared to the random sampling method (UiO, 2019). The fundamental idea of LHS in mitigating the clustering problem is stratification of the input probability distribution. Sampling is then forced to represent values in each strata, meaning that the sampling is forced to recreate the input probability distribution.

To have a better understanding on how LHS technique works, let us consider an example. In this example, LHS technique is adapted to generate 5 sets of samples for 2 independent variables, i.e. x and y. In addition, these variables are assumed to have normal distributions. The procedure of generating samples using LHS method is provided as follows:

1. The first step of LHS method is stratification of the input probability distribution. In this step, the cumulative probability curves are divided into equal intervals on the cumulative probability scale (0 to 1). The number of intervals is equal to the number of samples to be generated. Illustration of this step is provided in Figure 2.14a. The blue and the green numbers in this figure are the index of the intervals

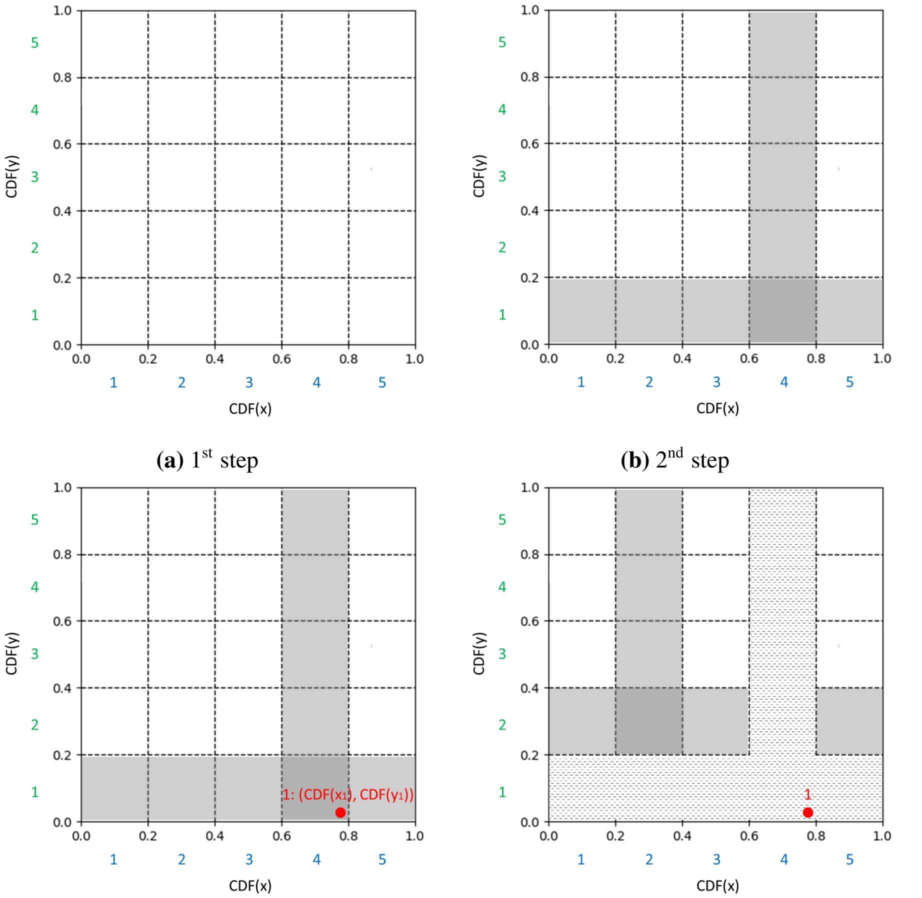
2. For the first iteration, we can randomly select an interval to draw a sample for each variable. In this example, variable x is sampled from interval #4, while variable y is sampled from interval #1 (see Figure 2.14b).

3. After that, a set of samples is randomly taken from the selected intervals (see Figure 2.14c). Up to this stage, remember that we have sampled only the cumulative probabilities of variable x and y.

4. For the second iteration, we again randomly select an interval to draw a sample for each variable. However, the intervals that have been selected in the previous iterations can not be chosen anymore. In this example, both variable x and variable y are sampled from interval #2 (see Figure 2.14d).

5. Similar to the 3rd step, another set of samples is randomly taken from the selected intervals (see Figure 2.14e).

6. The 4th and the 5th steps are repeated until we get 5 sets of samples (see Figure 2.14f).
7. In the previous steps, we have generated 5 combinations of cumulative probabilities for variable x and y. The final step of LHS is to convert the sampled cumulative probabilities using the corresponding cumulative probability curves (see Figure 2.14g and Figure 2.14h).



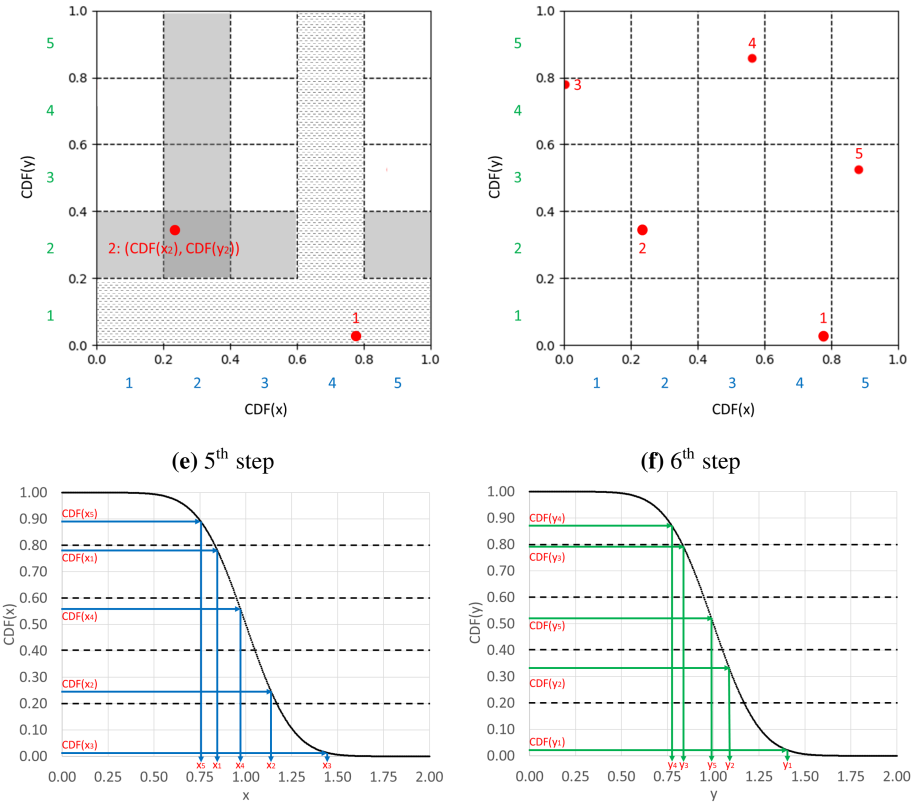
(a) 1st step

(b) 2nd step

(c) 3rd step

(d) 4th step

Figure 2.14: The procedure of generating samples using LHS method



(e) 5th step

(f) 6th step

(g) 7th step

(h) 7th step

Figure 2.14: The procedure of generating samples using LHS method (cont.)

Chapter 3

Maximizing the Plateau Duration

There are a few possible development alternatives for the Safari field (Angga, 2018), i.e., FPSO, gravity-based structure (GBS), PDQ platform based on a fixed steel jacket, tie-back to Grane oil field, and subsea-to-beach. However, in this study, the optimization algorithm is only developed for FPSO alternative.

As we know, the Safari field consists of three non-communicating reservoirs, i.e. Løve, Nesehorn, and Sebra. For the development with FPSO, the production from each reservoir is tied to a constant-pressure separator at FPSO (see Figure 3.1). With these features, the plateau duration of Safari field can be prolonged by adjusting the production splitting between the three non-communicating reservoirs (Angga, 2018).

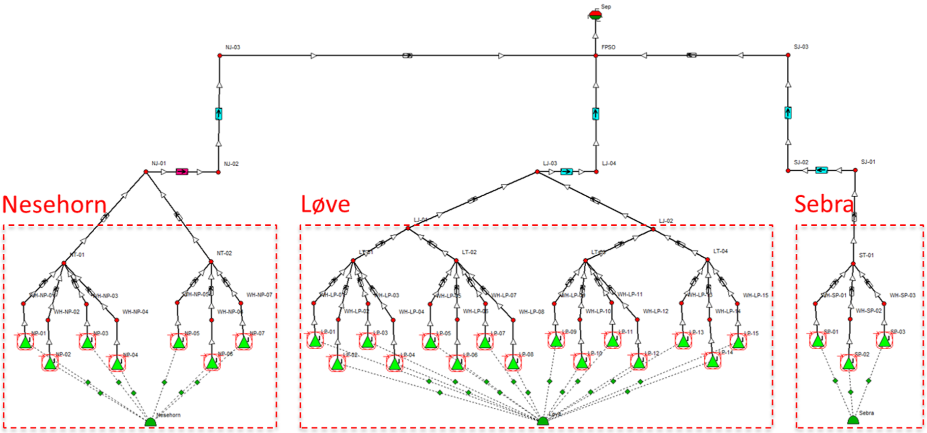


Figure 3.1: Layout of the production system of Safari field for FPSO development alternative (Angga, 2018)

An optimization problem which maximizes the plateau duration (t_p) will be discussed in this chapter. In this optimization problem, fluid flow rates, drilling schedule, and recovery mechanism are defined as the variables to be decided upon. In addition, some constraints (e.g., production constraints, injection constraints, and drilling constraints) are taken into account in the optimization. This optimization problem is referred to as CASE-1.

The results of the optimization problem are then compared to the results of the reference case (REF-CASE-1). The reference case has the same objective function as CASE-1, but it only considers the fluid flow rates as the decision variables. Drilling schedule and recovery mechanism are inputs for the reference case, and their values are similar to the ones used for the specialization project (Angga, 2018). Since CASE-1 has more variables to play with, it is expected that CASE-1 will provide a longer plateau duration than the reference case. Brief comparisons between CASE-1 and REF-CASE-1 are shown in Table 3.1.

Table 3.1: Comparisons of objective function, decision variables, and parameters between CASE-1 and REF-CASE-1

		CASE-1	**REF-CASE-1**
Objective Function		Max: t_p	Max: t_p
Production & Injection Rates	q_o	variable	variable
	q_g	variable	variable
	q_w	variable	variable
	q_{gi}	variable	variable
	q_{wi}	variable	variable
Drilling Schedule	N_{op}	variable	parameter
	N_{gi}	variable	parameter
	N_{wi}	variable	parameter
Recovery Mechanism	RM	variable	parameter

3.1 Formulation of the Optimization Problem

In this optimization, production potential curves are used to bound the oil production rate of a particular reservoir at a particular time. There are some constraints reflecting that the oil production rate (q_o) has to be less than or equal to the oil production potential (q_{opp}). The oil production potential (q_{opp}) is a non-linear function that depends on the cumulative oil production (N_p), the number of oil producer (N_{op}), and the recovery mechanism (RM). This non-linear function makes the initial formulation of the optimization problem is categorized as an NLP. Today, this class of optimization problem often remains difficult to solve. Therefore, PWL approximation is implemented to transform the optimization problem from an NLP into an MILP which is easier to solve. There are some well-developed algorithms to solve an MILP efficiently (see subsection 2.2.5).

Mathematical formulation of CASE-1 is discussed in the following subsections.

3.1.1 Parameters

Parameters are the constant quantities that affect the outcome of a mathematical object. In this formulation, they represent the inputs required for the optimization. The parameters involved in this formulation are listed, defined, and specified as follows:

- n_r
  definition: Number of reservoirs
  value: 3
  comment: Løve, Nesehorn, Sebra

- n_p
  definition: Number of points resulted from discretizing the field lifetime
  value: 18
  comment: The field lifetime is specified from 1st January 2023 until 1st January 2040. With a timestep size of 1 year, the field lifetime time is discretized into 18 points.

- t_{up}
  definition: Number of operational days per year (unit: days/year)
  value: 365

- $q_{o\_F\_plateau}$
  definition: Desired oil plateau rate for the field (unit: sm^3/d)
  value: 20000

- $\hat{N}_{op\_pd}$
  definition: Maximum number of pre-drilled oil producers
  value: 4
  comment: This number is decided by Aker Solutions beforehand.

- $\hat{N}_{i\_pd}$
  definition: Maximum number of pre-drilled injectors (gas injector or water injector)
  value: 1
  comment: This number is decided by Aker Solutions beforehand.

- $\hat{N}_{wt\_py}$
  definition: Maximum number of wells (producers & injectors) drilled each year (unit: wells/year)
  value: 4
  comment: As informed by Aker Solutions, the drilling & completion time is approximately 3 months/well, regardless of the well type. With one rig assigned for the field, at maximum four wells can be drilled each year.

- $\hat{N}_{gi,i}, \quad \forall i \in R$
  definition: Maximum number of gas injectors drilled at reservoir-i
  value: see Table 3.2
  comment: This number is decided by Aker Solutions beforehand.

- $\hat{N}_{wi,i}, \quad \forall i \in R$
  definition: Maximum number of water injectors drilled at reservoir-i
  value: See Table 3.2
  comment: This number is decided by Aker Solutions beforehand.

Table 3.2: Maximum number of gas and water injectors for each reservoir

i	$\hat{N}_{gi,i}$	$\hat{N}_{wi,i}$
1 (Løve)	1	3
2 (Nesehorn)	0	2
3 (Sebra)	0	1

- $\overline{zN_{gi}}_{,m,n}, \quad \forall m \in \{1,2\} \quad \text{and} \quad \forall n \in \{1,2\}$
  definition: The value of zN_{gi} at breakpoint (m,n)
  value: see Table 3.3
  comment: 2D PWL approximation is implemented to represent a non-linear function $RM = f(zN_{gi}, zN_{wi})$. In this PWL approximation, index-m indicates the breakpoints for zN_{gi}, while index-n indicates the breakpoints for zN_{wi}.

- $\overline{zN_{wi}}_{,m,n}, \quad \forall m \in \{1,2\} \quad \text{and} \quad \forall n \in \{1,2\}$
  definition: The value of zN_{wi} at breakpoint (m,n)
  value: see Table 3.3

- $\overline{RM_{ds}}_{,m,n}, \quad \forall m \in \{1,2\} \quad \text{and} \quad \forall n \in \{1,2\}$
  definition: The value of RM, which indicates the recovery mechanism, at breakpoint (m,n)
  value: see Table 3.3
  comment: The recovery mechanism depends on the value of zN_{gi} and zN_{wi}. Note that every combination of zN_{gi} and zN_{wi} reflects a unique type of recovery mechanism.

Table 3.3: 2D PWL approximation to represent a non-linear function $RM = f(zN_{gi}, zN_{wi})$

m	**n**	$\overline{zN_{gi}}_{,m,n}$	$\overline{zN_{wi}}_{,m,n}$	$\overline{RM_{ds}}_{,m,n}$
1	1	0	0	1 (Natural Depletion)
1	2	0	1	2 (Water Injection)
2	1	1	0	3 (Gas Injection)
2	2	1	1	4 (Water & Gas Injection)

- $\hat{q}_{gi\_pgi}$
  definition: Maximum gas injection rate for a gas injector (unit: 1000 sm^3/d)
  value: 4000
  comment: It is assumed that 4 million sm^3/d gas can be injected into the reservoirs by every gas injector, regardless of the required surface pressure for the injection.

- $\hat{q}_{wi\_pwi}$
  definition: Maximum water injection rate for a water injector (unit: sm^3/d)
  value: 6360
  comment: It is assumed that 6360 sm^3/d ($\approx$ 40000 bbl/d) water can be injected into the reservoirs by every water injector, regardless of the required surface pressure for the injection.

- $n_{bp\_N_p}$
  definition: Number of breakpoints for the cumulative oil production (N_p)
  value: see Table 3.4

- $n_{bp\_N_{op},i}, \quad \forall i \in R$
  definition: Number of breakpoints for the number of oil producers (N_{op}) corresponds to reservoir-i
  value: see Table 3.4

- $n_{bp\_RM,i}, \quad \forall i \in R$
  definition: Number of breakpoints for the recovery mechanism (RM) corresponds to reservoir-i
  value: see Table 3.4

Table 3.4: Number of breakpoints for 3D PWL approximation which represents a non-linear function $q_{opp} = f(N_p, N_{op}, RM)$

i	$n_{bp\_N_p}$	$n_{bp\_N_{op},i}$	$n_{bp\_RM,i}$
1 (Løve)	10	5	4
2 (Nesehorn)		4	2
3 (Sebra)		3	2

- $\overline{N_p}_{,i,j,k,l}, \quad \forall i \in R, \quad \forall j \in V_{N_p}, \quad \forall k \in V_{N_{op},i}, \quad \text{and} \quad \forall l \in V_{RM,i}$
  definition: Cumulative oil production of reservoir-i ($N_{p,i}$) at breakpoint (j, k, l) (unit: Mill. sm^3)
  value: see Table 3.5
  comment: 3D PWL approximation is implemented to represent a non-linear function $q_{opp} = f(N_p, N_{op}, RM)$. In this PWL approximation, index-j indicates the breakpoints for N_p, index-k indicates the breakpoints for N_{op}, and index-l indicates the breakpoints for RM.

- $\overline{N_{op}}_{,i,j,k,l}, \quad \forall i \in R, \quad \forall j \in V_{N_p}, \quad \forall k \in V_{N_{op},i}, \quad \text{and} \quad \forall l \in V_{RM,i}$
  definition: Number of oil producers in reservoir-i ($N_{op,i}$) at breakpoint (j, k, l)
  value: see Table 3.5

- $\overline{RM_{pp}}_{,i,j,k,l}, \quad \forall i \in R, \quad \forall j \in V_{N_p}, \quad \forall k \in V_{N_{op},i}, \quad \text{and} \quad \forall l \in V_{RM,i}$
  definition: Recovery mechanism of reservoir-i (RM_i) at breakpoint (j, k, l)
  value: see Table 3.5

- $\overline{q_{opp}}_{,i,j,k,l}$, $\forall i \in R$, $\forall j \in V_{N_p}$, $\forall k \in V_{N_{op},i}$, and $\forall l \in V_{RM,i}$
definition: Oil production potential of reservoir-i ($q_{opp,i}$) at breakpoint (j, k, l) (unit: sm^3/d)
value: see Table 3.5

Table 3.5: 3D PWL approximation to represent a non-linear function $q_{opp} = f(N_p, N_{op}, RM)$

i	**j**	**k**	**l**	$\overline{N_p}_{,i,j,k,l}$ (Mill. sm^3)	$\overline{N_{op}}_{,i,j,k,l}$	$\overline{RM_{pp}}_{,i,j,k,l}$	$\overline{q_{opp}}_{,i,j,k,l}$ (sm^3/d)
1	1	1	1	0	0	1	0
1	2	1	1	1.65	0	1	0
1	$\vdots$	1	1	$\vdots$	$\vdots$	$\vdots$	$\vdots$
1	9	1	1	23.10	0	1	0
1	10	1	1	27.61	0	1	0
1	1	2	1	0	1	1	1382.43
1	$\vdots$	$\vdots$	1	$\vdots$	$\vdots$	$\vdots$	$\vdots$
1	10	5	1	27.61	15	1	0
1	1	1	2	0	0	2	0
1	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
1	10	5	4	27.61	15	4	0
2	1	1	1	0	0	1	0
2	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
2	10	4	2	32.30	7	2	520.37
3	1	1	1	0	0	1	0
3	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
3	10	3	2	8.31	3	2	53.77

- $\overline{G_p}_{,i,j,k,l}$, $\forall i \in R$, $\forall j \in V_{N_p}$, $\forall k \in V_{N_{op},i}$, and $\forall l \in V_{RM,i}$
definition: Cumulative gas production of reservoir-i ($G_{p,i}$) at breakpoint (j, k, l) (unit: Mill. sm^3)

- $\overline{W_p}_{,i,j,k,l}$, $\forall i \in R$, $\forall j \in V_{N_p}$, $\forall k \in V_{N_{op},i}$, and $\forall l \in V_{RM,i}$
definition: Cumulative water production of reservoir-i ($W_{p,i}$) at breakpoint (j, k, l) (unit: Mill. sm^3)

- $\overline{G_i}_{,i,j,k,l}$, $\forall i \in R$, $\forall j \in V_{N_p}$, $\forall k \in V_{N_{op},i}$, and $\forall l \in V_{RM,i}$
definition: Cumulative gas injection of reservoir-i ($G_{i,i}$) at breakpoint (j, k, l) (unit: Mill. sm^3)

- $\overline{W_i}_{,i,j,k,l}$, $\forall i \in R$, $\forall j \in V_{N_p}$, $\forall k \in V_{N_{op},i}$, and $\forall l \in V_{RM,i}$
definition: Cumulative water injection of reservoir-i ($W_{i,i}$) at breakpoint (j, k, l) (unit: Mill. sm^3)

3.1.2 Sets

The sets associated in this formulation are listed and described as follows:

- $R = \{1, ..., n_r\}$
  definition: A set of reservoirs
  comment: "1" indicates Løve, "2" indicates Nesehorn, and "3" indicates Sebra.

- $P = \{1, ..., n_p\}$
  definition: A set of points of time throughout the production period
  comment: "1" represents 1st Jan 2023, "2" represents 1st Jan 2024, "3" represents 1st Jan 2025, and so on.

- $V_{N_p} = \{1, ..., n_{bp\_N_p}\}$
  definition: A set of breakpoints for N_p
  comment: As shown in Table 3.4, there are 10 N_p breakpoints for every reservoir. The N_p breakpoints for each reservoir are provided in Table 3.6.

Table 3.6: Sets of N_p breakpoints used for 3D PWL approximation

j	$\overline{N_p}_{,i,j,k,l}$ (Mill. sm^3)		
	i = 1 (Løve)	i = 2 (Nesehorn)	i = 3 (Sebra)
1	0	0	0
2	1.65	0.62	0.45
3	2.63	1.09	0.97
4	4.12	3.33	1.14
5	6.27	4.03	1.39
6	9.59	5.27	4.29
7	14.09	18.79	5.63
8	18.60	25.54	6.97
9	23.10	29.48	7.83
10	27.61	32.30	8.31

- $V_{N_{op},i} = \{1, ..., n_{bp\_N_{op},i}\}, \quad \forall i \in R$
  definition: A set of breakpoints for N_{op} corresponds to reservoir-i
  comment: As shown in Table 3.4, the number of N_{op} breakpoints varies between the reservoirs. The N_{op} breakpoints for each reservoir are provided in Table 3.7. In this 3D PWL approximation, the numbers of oil producers for Løve, Nesehorn, and Sebra can be chosen up to 15, 7, and 3, respectively. These numbers are decided by Aker Solutions beforehand.

Table 3.7: Sets of N_{op} breakpoints used for 3D PWL approximation

k	$\overline{N_{op}}_{,i,j,k,l}$		
	i = 1 (Løve)	i = 2 (Nesehorn)	i = 3 (Sebra)
1	0	0	0
2	1	1	1
3	5	4	3
4	10	7	-
5	15	-	-

- $V_{RM,i} = \{1, ..., n_{bp\_RM,i}\}, \quad \forall i \in R$
definition: A set of breakpoints for RM corresponds to reservoir-i
comment: As shown in Table 3.4, the number of RM breakpoints varies between the reservoirs. The RM breakpoints for each reservoir are provided in Table 3.8. In this 3D PWL approximation, all reservoirs can be developed with natural depletion or water injection. However, developments with gas injection or water-gas injection are only feasible for Løve. The options of recovery mechanism for each reservoir are decided by Aker Solutions beforehand.

Table 3.8: Sets of RM breakpoints used for 3D PWL approximation

l	$\overline{RM_{pp}}_{,i,j,k,l}$		
	i = 1 (Løve)	i = 2 (Nesehorn)	i = 3 (Sebra)
1	1 (Natural Depletion)	1	1
2	2 (Water Injection)	2	2
3	3 (Gas Injection)	-	-
4	4 (Water & Gas Injection)	-	-

3.1.3 Variables

In contrary to parameters, variables are defined as the varying quantities that affect the outcome of a mathematical object. In this formulation, variables can be seen as the outputs of the optimization. The variables involved in this formulation are listed and defined as follows:

- $RM_i, \quad \forall i \in R$
definition: Recovery mechanism of reservoir-i
comment: The recovery mechanism of reservoir-i is assumed constant throughout the production period. The reason for this assumption is because the production potential curves for a particular reservoir are valid only if the reservoir is produced with a constant recovery mechanism during its lifetime.

- $zN_{gi,i}, \quad \forall i \in R$
  definition: Binary variable which relates the number of gas injectors in reservoir-i with the recovery mechanism of reservoir-i.
  comment: "$zN_{gi} = 0$" when the reservoir is developed with natural depletion or water injection, while "$zN_{gi} = 1$" when the reservoir is developed with gas injection or water-gas injection.

- $zN_{wi,i}, \quad \forall i \in R$
  definition: Binary variable which relates the number of water injectors in reservoir-i with the recovery mechanism of reservoir-i.
  comment: "$zN_{wi} = 0$" when the reservoir is developed with natural depletion or gas injection, while "$zN_{wi} = 1$" when the reservoir is developed with water injection or water-gas injection.

- $N_{op,i,z}, \quad \forall i \in R$ and $\forall z \in P$
  definition: Number of oil producers in reservoir-i at time-z

- $N_{gi,i,z}, \quad \forall i \in R$ and $\forall z \in P$
  definition: Number of gas injectors in reservoir-i at time-z

- $N_{wi,i,z}, \quad \forall i \in R$ and $\forall z \in P$
  definition: Number of water injectors in reservoir-i at time-z

- $N_{wt,i,z}, \quad \forall i \in R$ and $\forall z \in P$
  definition: Total number of wells (producers & injectors) in reservoir-i at time-z

- $q_{opp,i,z}, \quad \forall i \in R$ and $\forall z \in P$
  definition: Oil production potential of reservoir-i at time-z (unit: sm^3/d)

- $q_{o,i,z}, \quad \forall i \in R$ and $\forall z \in \{2, ..., n_p\}$
  definition: Oil production rate of reservoir-i at time-z (unit: sm^3/d)

- $q_{g,i,z}, \quad \forall i \in R$ and $\forall z \in \{2, ..., n_p\}$
  definition: Gas production rate of reservoir-i at time-z (unit: 1000 sm^3/d)

- $q_{w,i,z}, \quad \forall i \in R$ and $\forall z \in \{2, ..., n_p\}$
  definition: Water production rate of reservoir-i at time-z (unit: sm^3/d)

- $q_{gi,i,z}, \quad \forall i \in R$ and $\forall z \in \{2, ..., n_p\}$
  definition: Gas injection rate of reservoir-i at time-z (unit: 1000 sm^3/d)

- $q_{wi,i,z}, \quad \forall i \in R$ and $\forall z \in \{2, ..., n_p\}$
  definition: Water injection rate of reservoir-i at time-z (unit: sm^3/d)

- $N_{p,i,z}, \quad \forall i \in R$ and $\forall z \in P$
  definition: Cumulative oil production of reservoir-i at time-z (unit: Mill. sm^3)

- $G_{p,i,z}, \quad \forall i \in R$ and $\forall z \in P$
  definition: Cumulative gas production of reservoir-i at time-z (unit: Mill. sm^3)

- $W_{p,i,z}, \quad \forall i \in R$ and $\forall z \in P$
  definition: Cumulative water production of reservoir-i at time-z (unit: Mill. sm^3)

- $G_{i,i,z}, \quad \forall i \in R \quad \text{and} \quad \forall z \in P$
  definition: Cumulative gas injection of reservoir-i at time-z (unit: Mill. sm^3)
- $W_{i,i,z}, \quad \forall i \in R \quad \text{and} \quad \forall z \in P$
  definition: Cumulative water injection of reservoir-i at time-z (unit: Mill. sm^3)
- $N_{op\_F,z}, \quad \forall z \in P$
  definition: Number of oil producers in the field at time-z
- $N_{gi\_F,z}, \quad \forall z \in P$
  definition: Number of gas injectors in the field at time-z
- $N_{wi\_F,z}, \quad \forall z \in P$
  definition: Number of water injectors in the field at time-z
- $N_{wt\_F,z}, \quad \forall z \in P$
  definition: Total number of wells (producers & injectors) in the field at time-z
- $q_{opp\_F,z}, \quad \forall z \in P$
  definition: Oil production potential of the field at time-z (unit: sm^3/d)
- $q_{o\_F,z}, \quad \forall z \in \{2, ..., n_p\}$
  definition: Oil production rate of the field at time-z (unit: sm^3/d)
- $q_{g\_F,z}, \quad \forall z \in \{2, ..., n_p\}$
  definition: Gas production rate of the field at time-z (unit: 1000 sm^3/d)
- $q_{w\_F,z}, \quad \forall z \in \{2, ..., n_p\}$
  definition: Water production rate of the field at time-z (unit: sm^3/d)
- $q_{gi\_F,z}, \quad \forall z \in \{2, ..., n_p\}$
  definition: Gas injection rate of the field at time-z (unit: 1000 sm^3/d)
- $q_{wi\_F,z}, \quad \forall z \in \{2, ..., n_p\}$
  definition: Water injection rate of the field at time-z (unit: sm^3/d)
- $N_{p\_F,z}, \quad \forall z \in P$
  definition: Cumulative oil production of the field at time-z (unit: Mill. sm^3)
- $G_{p\_F,z}, \quad \forall z \in P$
  definition: Cumulative gas production of the field at time-z (unit: Mill. sm^3)
- $W_{p\_F,z}, \quad \forall z \in P$
  definition: Cumulative water production of the field at time-z (unit: Mill. sm^3)
- $G_{i\_F,z}, \quad \forall z \in P$
  definition: Cumulative gas injection of the field at time-z (unit: Mill. sm^3)
- $W_{i\_F,z}, \quad \forall z \in P$
  definition: Cumulative water injection of the field at time-z (unit: Mill. sm^3)
- $\lambda_{i,z,j,k,l}, \quad \forall i \in R, \quad \forall z \in P, \quad \forall j \in V_{N_p}, \quad \forall k \in V_{N_{op},i}, \quad \text{and} \quad \forall l \in V_{RM,i}$
  definition: Weighting coefficient of breakpoint (j,k,l) in the 3D PWL approximation corresponds to reservoir-i at time-z

- $\eta_{N_p,i,z,j}, \quad \forall i \in R, \quad \forall z \in P, \quad \text{and} \quad \forall j \in V_{N_p}$
- $\eta_{N_{op},i,z,k}, \quad \forall i \in R, \quad \forall z \in P, \quad \text{and} \quad \forall k \in V_{N_{op},i}$
- $\eta_{RM,i,z,l}, \quad \forall i \in R, \quad \forall z \in P, \quad \text{and} \quad \forall l \in V_{RM,i}$
- $\omega_{i,m,n}, \quad \forall i \in R, \quad \forall m \in \{1,2\}, \quad \text{and} \quad \forall n \in \{1,2\}$
  definition: Weighting coefficient of breakpoint (m,n) in the 2D PWL approximation corresponds to reservoir-i
- $\tau_{zN_{gi},i,m}, \quad \forall i \in R \quad \text{and} \quad \forall m \in \{1,2\}$
- $\tau_{zN_{wi},i,n}, \quad \forall i \in R \quad \text{and} \quad \forall n \in \{1,2\}$

3.1.4 Objective Function

The objective function of the optimization problem is to maximize the plateau duration for a given plateau rate. The objective function is expressed as follows:

$$\text{minimize} \quad \sum_{z \in \{2,\ldots,n_p\}} \left(q_{o\_F\_plateau} - q_{o\_F,z}\right) \tag{3.1}$$

Visually, the objective function is aimed to minimize the grey area in Figure 3.2.

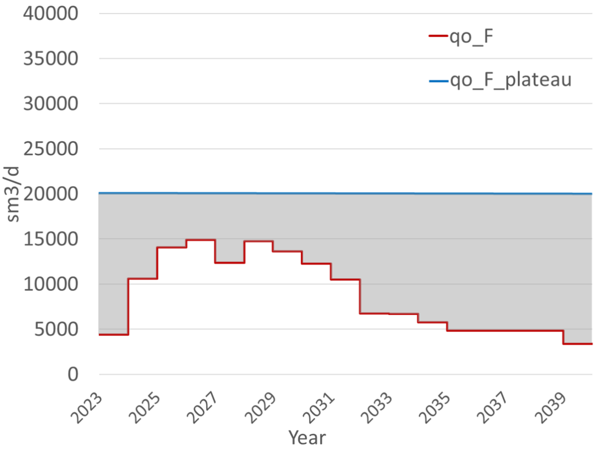


Figure 3.2: Visualization of the objective function

3.1.5 Operational Constraints

- The field should be produced at any rates no higher than the desired plateau rate:

$$q_{o\_F,z} \leq q_{o\_F\_plateau}, \quad \forall z \in \{2, \ldots, n_p\} \tag{3.2}$$

- Production potential defines the upper limit of the production rate. Thus, it must be guaranteed that the production rate at any point in time never exceeds the production potential. Since backward rectangular integration technique is utilized, the production rate $q_{o,i,z}$ is assumed to last from time t_{z-1} until t_z. It is also assumed that the production potential is monotonic between time t_{z-1} and t_z. Therefore, to assure that the production rate is always less than or equal to the production potential, the following two constraints are included:

$$q_{o,i,z} \leq q_{opp,i,z}, \quad \forall i \in R \quad \text{and} \quad \forall z \in \{2, ..., n_p\} \tag{3.3a}$$

$$q_{o,i,z} \leq q_{opp,i,z-1}, \quad \forall i \in R \quad \text{and} \quad \forall z \in \{2, ..., n_p\} \tag{3.3b}$$

- Backward rectangular integration method is also employed to estimate cumulative gas & water injection (G_i & W_i). It means that the injection rates are also assumed constant from time t_{z-1} until t_z. One needs to ensure that these injection rates do not surpass the injection capabilities, which are defined as the number of injectors available multiplied by the maximum injection rate of each injector. Since the numbers of injectors are non-decreasing over time (see subsection 3.1.6), the injection capabilities also never decrease. Consequently, to verify that the injection rates do not exceed the injection capabilities, one only needs to impose that the injection capabilities at time t_{z-1} are higher or equal to the constant injection rates from time t_{z-1} until t_z:

$$q_{gi,i,z} \leq N_{gi,i,z-1} \cdot \hat{q}_{gi\_pgi}, \quad \forall i \in R \quad \text{and} \quad \forall z \in \{2, ..., n_p\} \tag{3.4a}$$

$$q_{wi,i,z} \leq N_{wi,i,z-1} \cdot \hat{q}_{wi\_pwi}, \quad \forall i \in R \quad \text{and} \quad \forall z \in \{2, ..., n_p\} \tag{3.4b}$$

3.1.6 Drilling Constraints

- The following three constraints reflect that the numbers of wells (oil producers, gas injectors, or water injectors) are non-decreasing:

$$N_{op,i,z} - N_{op,i,z-1} \geq 0, \quad \forall i \in R \quad \text{and} \quad \forall z \in \{2, ..., n_p\} \tag{3.5a}$$

$$N_{gi,i,z} - N_{gi,i,z-1} \geq 0, \quad \forall i \in R \quad \text{and} \quad \forall z \in \{2, ..., n_p\} \tag{3.5b}$$

$$N_{wi,i,z} - N_{wi,i,z-1} \geq 0, \quad \forall i \in R \quad \text{and} \quad \forall z \in \{2, ..., n_p\} \tag{3.5c}$$

- The number of gas injectors ($N_{gi,i,z}$) is controlled by two factors, i.e., 1) the variable $zN_{gi,i}$ that depends on the recovery mechanism, and 2) the maximum number of gas injectors to drill ($\hat{N}_{gi,i}$):

$$N_{gi,i,z} \leq zN_{gi,i} \cdot \hat{N}_{gi,i}, \quad \forall i \in R \quad \text{and} \quad \forall z \in P \tag{3.6a}$$

When the reservoir is developed with natural depletion or water injection, the variable $zN_{gi,i}$ equals to zero, and it imposes $N_{gi,i,z}$ to be zero. On the other hand, when the reservoir is developed with gas injection or water-gas injection, $zN_{gi,i}$ equals to one, and thus $N_{gi,i,z}$ can be any value as long as it is no higher than the input value for $\hat{N}_{gi,i}$. Similarly, the number of water injectors ($N_{wi,i,z}$) has to satisfy the following constraint:

$$N_{wi,i,z} \leq zN_{wi,i} \cdot \hat{N}_{wi,i}, \quad \forall i \in R \quad \text{and} \quad \forall z \in P \tag{3.6b}$$

- The two following constraints are related to the pre-drilled wells, i.e., the number of wells available at the beginning of the production period is no greater than the maximum number of pre-drilled wells:

$$N_{op\_F,1} \leq \hat{N}_{op\_pd} \tag{3.7a}$$

$$N_{gi\_F,1} + N_{wi\_F,1} \leq \hat{N}_{i\_pd} \tag{3.7b}$$

- The following constraint represents that at most four wells can be drilled & completed each year:

$$N_{wt\_F,z} - N_{wt\_F,z-1} \leq \hat{N}_{wt\_py}, \quad \forall z \in \{2, ..., n_p\} \tag{3.8}$$

3.1.7 Consistency Constraints

- At the commencement of the production period, the cumulative oil production ($N_{p,i,1}$) equals to zero. With the 3D PWL approximation, note that this constraint also imposes the cumulative production & injection of gas & water to be zero:

$$N_{p,i,1} = 0, \quad \forall i \in R \tag{3.9}$$

- The cumulative production & injection at a particular point in time is numerically calculated using the backward rectangular integration technique. In this integration method, the production & injection rates at time t_z represent the constant production & injection rates from time t_{z-1} until t_z:

$$N_{p,i,z} = N_{p,i,z-1} + \frac{t_{up} \cdot q_{o,i,z}}{10^6}, \quad \forall i \in R \quad \text{and} \quad \forall z \in \{2, ..., n_p\} \tag{3.10a}$$

$$G_{p,i,z} = G_{p,i,z-1} + \frac{t_{up} \cdot q_{g,i,z}}{10^3}, \quad \forall i \in R \quad \text{and} \quad \forall z \in \{2, ..., n_p\} \tag{3.10b}$$

$$W_{p,i,z} = W_{p,i,z-1} + \frac{t_{up} \cdot q_{w,i,z}}{10^6}, \quad \forall i \in R \quad \text{and} \quad \forall z \in \{2, ..., n_p\} \tag{3.10c}$$

$$G_{i,i,z} = G_{i,i,z-1} + \frac{t_{up} \cdot q_{gi,i,z}}{10^3}, \quad \forall i \in R \quad \text{and} \quad \forall z \in \{2, ..., n_p\} \tag{3.10d}$$

$$W_{i,i,z} = W_{i,i,z-1} + \frac{t_{up} \cdot q_{wi,i,z}}{10^6}, \quad \forall i \in R \quad \text{and} \quad \forall z \in \{2, ..., n_p\} \tag{3.10e}$$

- The total number of wells is defined as the summation of numbers of oil producers, gas injectors, and water injectors:

$$N_{wt,i,z} = N_{op,i,z} + N_{gi,i,z} + N_{wi,i,z}, \quad \forall i \in R \quad \text{and} \quad \forall z \in P \tag{3.11}$$

- Any variable corresponds for the field is determined as the summation of the relevant variable of each reservoir:

$$N_{op\_F,z} = \sum_{i \in R} N_{op,i,z}, \quad \forall z \in P \tag{3.12a}$$

$$N_{gi\_F,z} = \sum_{i \in R} N_{gi,i,z}, \quad \forall z \in P \tag{3.12b}$$

$$N_{wi\_F,z} = \sum_{i \in R} N_{wi,i,z}, \quad \forall z \in P \tag{3.12c}$$

$$N_{wt\_F,z} = \sum_{i \in R} N_{wt,i,z}, \quad \forall z \in P \tag{3.12d}$$

$$q_{opp\_F,z} = \sum_{i \in R} q_{opp,i,z}, \quad \forall z \in P \tag{3.12e}$$

$$q_{o\_F,z} = \sum_{i \in R} q_{o,i,z}, \quad \forall z \in \{2, ..., n_p\} \tag{3.12f}$$

$$q_{g\_F,z} = \sum_{i \in R} q_{g,i,z}, \quad \forall z \in \{2, ..., n_p\} \tag{3.12g}$$

$$q_{w\_F,z} = \sum_{i \in R} q_{w,i,z}, \quad \forall z \in \{2, ..., n_p\} \tag{3.12h}$$

$$q_{gi\_F,z} = \sum_{i \in R} q_{gi,i,z}, \quad \forall z \in \{2, ..., n_p\} \tag{3.12i}$$

$$q_{wi\_F,z} = \sum_{i \in R} q_{wi,i,z}, \quad \forall z \in \{2, ..., n_p\} \tag{3.12j}$$

$$N_{p\_F,z} = \sum_{i \in R} N_{p,i,z}, \quad \forall z \in P \tag{3.12k}$$

$$G_{p\_F,z} = \sum_{i \in R} G_{p,i,z}, \quad \forall z \in P \tag{3.12l}$$

$$W_{p\_F,z} = \sum_{i \in R} W_{p,i,z}, \quad \forall z \in P \tag{3.12m}$$

$$G_{i\_F,z} = \sum_{i \in R} G_{i,i,z}, \quad \forall z \in P \tag{3.12n}$$

$$W_{i\_F,z} = \sum_{i \in R} W_{i,i,z}, \quad \forall z \in P \tag{3.12o}$$

- In addition, we have to verify that:

$$\text{all variables are non-negative} \tag{3.13}$$

3.1.8 Integer & Binary Constraints

All variables that are restricted to be integers or to have 0-1 values are listed as follows:

$$RM_i \in \mathbb{Z}, \quad \forall i \in R \tag{3.14a}$$
$$zN_{gi,i} \in \{0,1\}, \quad \forall i \in R \tag{3.14b}$$
$$zN_{wi,i} \in \{0,1\}, \quad \forall i \in R \tag{3.14c}$$
$$N_{op,i,z} \in \mathbb{Z}, \quad \forall i \in R \quad \text{and} \quad \forall z \in P \tag{3.14d}$$
$$N_{gi,i,z} \in \mathbb{Z}, \quad \forall i \in R \quad \text{and} \quad \forall z \in P \tag{3.14e}$$
$$N_{wi,i,z} \in \mathbb{Z}, \quad \forall i \in R \quad \text{and} \quad \forall z \in P \tag{3.14f}$$
$$N_{wt,i,z} \in \mathbb{Z}, \quad \forall i \in R \quad \text{and} \quad \forall z \in P \tag{3.14g}$$
$$N_{op\_F,z} \in \mathbb{Z}, \quad \forall z \in P \tag{3.14h}$$
$$N_{gi\_F,z} \in \mathbb{Z}, \quad \forall z \in P \tag{3.14i}$$
$$N_{wi\_F,z} \in \mathbb{Z}, \quad \forall z \in P \tag{3.14j}$$
$$N_{wt\_F,z} \in \mathbb{Z}, \quad \forall z \in P \tag{3.14k}$$

3.1.9 Constraints Induced by PWL Approximations

- The oil production potential (q_{opp}) is non-linearly dependent on the cumulative oil production (N_p), the number of oil producers (N_{op}), and the recovery mechanism (RM). To reformulate the problem into an MILP, 3D PWL approximation is used. The fundamental and the standard expression of 3D PWL approximation are provided in section 2.3. In this formulation, the 3D PWL approximation is used to estimate the production potential of each reservoir-i at each time-z. The implementation of the 3D PWL approximation is provided as follows:

$$N_{p,i,z} = \sum_{j \in V_{N_p}} \sum_{k \in V_{N_{op},i}} \sum_{l \in V_{RM,i}} \lambda_{i,z,j,k,l} \cdot \overline{N_p}_{,i,j,k,l}, \quad \forall i \in R \quad \text{and} \quad \forall z \in P \tag{3.15a}$$

$$N_{op,i,z} = \sum_{j \in V_{N_p}} \sum_{k \in V_{N_{op},i}} \sum_{l \in V_{RM,i}} \lambda_{i,z,j,k,l} \cdot \overline{N_{op}}_{,i,j,k,l}, \quad \forall i \in R \quad \text{and} \quad \forall z \in P \tag{3.15b}$$

$$RM_i = \sum_{j \in V_{N_p}} \sum_{k \in V_{N_{op},i}} \sum_{l \in V_{RM,i}} \lambda_{i,z,j,k,l} \cdot \overline{RM_{pp}}_{,i,j,k,l}, \quad \forall i \in R \quad \text{and} \quad \forall z \in P \tag{3.15c}$$

$$q_{opp,i,z} = \sum_{j\in V_{N_p}} \sum_{k\in V_{N_{op},i}} \sum_{l\in V_{RM,i}} \lambda_{i,z,j,k,l} \cdot \overline{q_{opp}}_{,i,j,k,l}, \quad \forall i \in R \quad \text{and} \quad \forall z \in P \tag{3.15d}$$

$$1 = \sum_{j\in V_{N_p}} \sum_{k\in V_{N_{op},i}} \sum_{l\in V_{RM,i}} \lambda_{i,z,j,k,l}, \quad \forall i \in R \quad \text{and} \quad \forall z \in P \tag{3.15e}$$

$$\eta_{N_p,i,z,j} = \sum_{k\in V_{N_{op},i}} \sum_{l\in V_{RM,i}} \lambda_{i,z,j,k,l}, \quad \forall i \in R, \quad \forall z \in P, \quad \text{and} \quad \forall j \in V_{N_p} \tag{3.15f}$$

$$\eta_{N_{op},i,z,k} = \sum_{j\in V_{N_p}} \sum_{l\in V_{RM,i}} \lambda_{i,z,j,k,l}, \quad \forall i \in R, \quad \forall z \in P, \quad \text{and} \quad \forall k \in V_{N_{op},i} \tag{3.15g}$$

$$\eta_{RM,i,z,l} = \sum_{j\in V_{N_p}} \sum_{k\in V_{N_{op},i}} \lambda_{i,z,j,k,l}, \quad \forall i \in R, \quad \forall z \in P, \quad \text{and} \quad \forall l \in V_{RM,i} \tag{3.15h}$$

$$\{\eta_{N_p,i,z,j}, \quad j \in V_{N_p}\} \text{ is SOS2}, \quad \forall i \in R, \quad \forall z \in P \tag{3.15i}$$

$$\{\eta_{N_{op},i,z,k}, \quad k \in V_{N_{op},i}\} \text{ is SOS2}, \quad \forall i \in R, \quad \forall z \in P \tag{3.15j}$$

$$\{\eta_{RM,i,z,l}, \quad l \in V_{RM,i}\} \text{ is SOS1}, \quad \forall i \in R, \quad \forall z \in P \tag{3.15k}$$

In the implementation of 3D PWL approximation above, the set $\{\eta_{RM,i,z,l}, \quad l \in V_{RM,i}\}$ is restricted to be SOS1. The reason of applying this constraint is to ensure that the variable RM_i is selected among the RM breakpoints (see Table 3.8). Again, these RM breakpoints represent the feasible recovery mechanism options for each reservoir.

- The cumulative gas production (G_p) is found to have a non-linear dependency on the cumulative oil production (N_p) and the recovery mechanism (RM). For sure, one can establish another 2D PWL approximation to estimate G_p. However, instead of doing that, 3D PWL approximation is again implemented to represent the function $G_p = f(N_p, N_{op}, RM)$. Note that, if the 3D PWL approximation has similar sets of breakpoints as the 3D PWL approximation for $q_{opp} = f(N_p, N_{op}, RM)$, one can use the same variable $\lambda_{i,z,j,k,l}$ to compute G_p, as expressed below. This approach is considered more efficient since fewer variables are involved.

$$G_{p,i,z} = \sum_{j\in V_{N_p}} \sum_{k\in V_{N_{op},i}} \sum_{l\in V_{RM,i}} \lambda_{i,z,j,k,l} \cdot \overline{G_p}_{,i,j,k,l}, \quad \forall i \in R \quad \text{and} \quad \forall z \in P \tag{3.16a}$$

In similar fashion, W_p, G_i, W_i are determined as follows:

$$W_{p,i,z} = \sum_{j\in V_{N_p}} \sum_{k\in V_{N_{op},i}} \sum_{l\in V_{RM,i}} \lambda_{i,z,j,k,l} \cdot \overline{W_p}_{,i,j,k,l}, \quad \forall i \in R \quad \text{and} \quad \forall z \in P \tag{3.16b}$$

$$G_{i,i,z} = \sum_{j\in V_{N_p}} \sum_{k\in V_{N_{op},i}} \sum_{l\in V_{RM,i}} \lambda_{i,z,j,k,l} \cdot \overline{G_i}_{,i,j,k,l}, \quad \forall i \in R \quad \text{and} \quad \forall z \in P \tag{3.16c}$$

$$W_{i,i,z} = \sum_{j \in V_{N_p}} \sum_{k \in V_{N_{op},i}} \sum_{l \in V_{RM,i}} \lambda_{i,z,j,k,l} \cdot \overline{W}_{i,i,j,k,l}, \quad \forall i \in R \quad \text{and} \quad \forall z \in P \tag{3.16d}$$

- In this formulation, there is another non-linear function involved, i.e., $RM = f(zN_{gi}, zN_{wi})$. The non-linear function is illustrated in Figure 3.3. To have an MILP problem, 2D PWL approximation is employed. For each reservoir-i, the 2D PWL approximation is used to represent the relationship between the recovery mechanism (RM) and the values of variable zN_{gi} & zN_{wi}. The implementation of the 2D PWL approximation is provided as follows:

$$zN_{gi,i} = \sum_{m \in \{1,2\}} \sum_{n \in \{1,2\}} \omega_{i,m,n} \cdot \overline{zN_{gi}}_{,m,n}, \quad \forall i \in R \tag{3.17a}$$

$$zN_{wi,i} = \sum_{m \in \{1,2\}} \sum_{n \in \{1,2\}} \omega_{i,m,n} \cdot \overline{zN_{wi}}_{,m,n}, \quad \forall i \in R \tag{3.17b}$$

$$RM_i = \sum_{m \in \{1,2\}} \sum_{n \in \{1,2\}} \omega_{i,m,n} \cdot \overline{RM_{ds}}_{,m,n}, \quad \forall i \in R \tag{3.17c}$$

$$1 = \sum_{m \in \{1,2\}} \sum_{n \in \{1,2\}} \omega_{i,m,n}, \quad \forall i \in R \tag{3.17d}$$

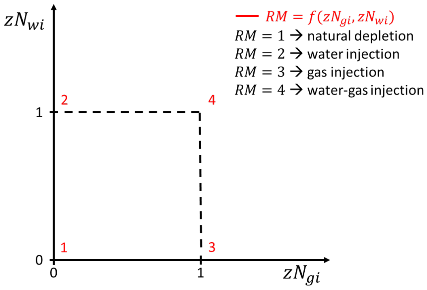
$$\tau_{zN_{gi},i,m} = \sum_{n \in \{1,2\}} \omega_{i,m,n}, \quad \forall i \in R \quad \text{and} \quad \forall m \in \{1,2\} \tag{3.17e}$$

$$\tau_{zN_{wi},i,n} = \sum_{m \in \{1,2\}} \omega_{i,m,n}, \quad \forall i \in R \quad \text{and} \quad \forall n \in \{1,2\} \tag{3.17f}$$

$$\{\tau_{zN_{gi},i,m}, \quad m = 1,2\} \text{ is SOS1}, \quad \forall i \in R \tag{3.17g}$$

$$\{\tau_{zN_{wi},i,n}, \quad n = 1,2\} \text{ is SOS1}, \quad \forall i \in R \tag{3.17h}$$

As expressed above, the sets $\{\tau_{zN_{gi},i,m}, \quad m = 1,2\}$ and $\{\tau_{zN_{wi},i,n}, \quad n = 1,2\}$ are restricted to be SOS1. These constraints aim to ensure that the values of variable zN_{gi} and zN_{wi} are chosen among the breakpoints (see Table 3.3), i.e., either zero or one.



zN_{wi}
$RM = f(zN_{gi}, zN_{wi})$
RM = 1 → natural depletion
RM = 2 → water injection
RM = 3 → gas injection
RM = 4 → water-gas injection
2
4
1
1
3
0
0
1
zN_{gi}

Figure 3.3: An illustration of a non-linear function $RM = f(zN_{gi}, zN_{wi})$

3.2 Implementation in AMPL

A Mathematical Programming Language, or commonly abbreviated as AMPL, is a programming language designed to solve a wide range of optimization problems (AMPL, 2019a). AMPL closely resembles the symbolic algebraic notation that many modelers use to describe mathematical programs. This makes it convenient to formulate and solve an optimization problem in AMPL. In addition, AMPL offers various powerful solvers, such as CPLEX, Gurobi, etc., to solve many classes of mathematical optimization.

In this study, the optimization problem CASE-1 is implemented in AMPL. The AMPL implementation is divided into four modules/files, i.e., the model file, the data file, the PWL table file, and the run file (see Figure 3.4).

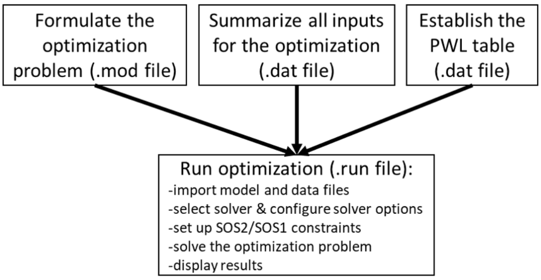

Formulate the optimization problem (.mod file)
Summarize all inputs for the optimization (.dat file)
Establish the PWL table (.dat file)
Run optimization (.run file):
-import model and data files
-select solver & configure solver options
-set up SOS2/SOS1 constraints
-solve the optimization problem
-display results

Figure 3.4: Workflow of AMPL implementation for CASE-1

The optimization problem is formulated in the model file. Here, the parameters, the sets, the variables, the objective function, and all the constraints are clearly stated. The model file for CASE-1 is given in Appendix A.1.1. In the data file, all optimization parameters -except the ones that are related to the production potential- are specified. Some of those parameters are n_r, n_p, t_{up}, $q_{o\_F\_plateau}$, drilling considerations, & injection limitations. The data file for CASE-1 is provided in Appendix A.1.2. A PWL table is an input to any software that performs PWL approximation. In this implementation, the PWL table file contains a PWL table to carry out 3D PWL approximations for q_{opp}, G_p, W_p, G_i, and W_i. An illustration of the PWL table is presented in Table 3.5. In addition, the PWL table file includes the numbers of breakpoints involved in the 3D PWL approximation.

The run file can be considered as the main file in the AMPL implementation. Once this run file is executed, the model file, the data file, and the PWL table file are loaded. After that, the run file decides on which solver to be used. The selected solver can be customized by configuring the solver options. Description of each solver option can be found in AMPL (2019b) and Gurobi (2019). After choosing the solver & configuring the solver options, the run file declares the SOS2 & SOS1 constraints for some sets. Subsequently, the run file commands to solve the optimization problem. Finally, the optimization results are displayed and extracted into an output file (.out file). The run file for CASE-1 is provided in Appendix A.1.3.

In addition, AMPL implementation for the reference case is provided in Appendix A.2.

3.3 Results & Discussion

Before discussing the optimization results, let us discuss some terms commonly used in optimization:

- Objective value → the value of objective function under a particular solution.
- Best integer solution → a feasible solution that has been found so far to give the highest (or the lowest) objective value. One can interpret the best integer solution as the current best solution.
- Best integer → the objective value of the best integer solution.
- Best bound → the best objective value that a feasible solution could potentially have, based on information the solver has discovered so far. With the best bound, one can say that there might be a better solution than the best integer solution, but if there is, it will not have an objective value better than the best bound.
- Relative gap → the difference between the best bound and the best integer relative to the best integer. Mathematically, it is calculated as follows:

$$\text{Relative gap} = \frac{|\text{best bound} - \text{best integer}|}{|\text{best integer}|} \tag{3.18}$$

  The optimality of a solution is proven when the best integer and the best bound evaluate the same value, i.e., the solver could prove a relative gap of 0%.

Table 3.9: Comparisons of optimization results between CASE-1 and REF-CASE-1

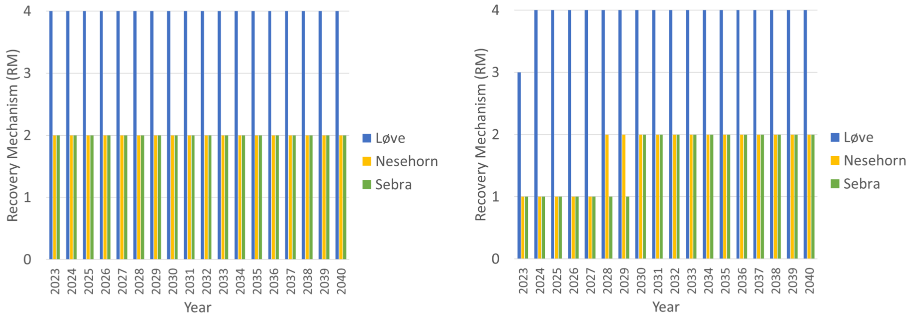
	CASE-1	**REF-CASE-1**
Objective value	190421	201293
Relative gap	1.53%	0.00%
Runtime (sec.)	43210	0.23
Runtime (hr.)	12.00	0.00
Number of variables	8113	3814
Number of constraints	2488	1986

As shown in Table 3.9, the best integer solution for CASE-1 gives an objective value of 190421. This best integer solution is obtained after running the optimization for 12 hours. The relative gap of 1.53% indicates that there might exist a better feasible solution which could lower objective value up to 187515. Of course, one can reduce this relative gap by extending the optimization runtime. Unfortunately, due to the time limitation to complete this project, we have to accept the best feasible solution found so far.

For the reference case, relative gap of 0.00% implies that the optimal solution has been found. This optimal solution is obtained extremely quick since we include fewer variables. The objective value of the optimal solution is 201293. When we compare this number to the one belongs to CASE-1, it can be seen that CASE-1 produces a lower objective value than REF-CASE-1. The lower objective value simply means a longer plateau duration. This finding is consistent with what we have expected, i.e., the plateau duration can be prolonged further by involving more decision variables.

Some optimization results are presented below. More detailed drilling schedule (e.g. for oil producers, gas injectors, and water injectors) and production profile (e.g. for each reservoir) are also obtained, but they are not presented here.

Figure 3.5 compares the recovery mechanisms obtained from CASE-1 and the ones inputted for REF-CASE-1. As shown in the figures, the recovery mechanisms in both cases are similar, i.e. water-gas injection for Løve and water injection for Nesehorn and Sebra.

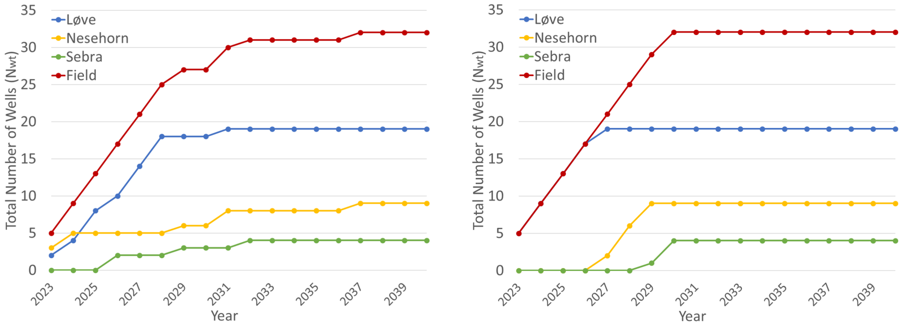


(a) CASE-1

(b) REF-CASE-1

Figure 3.5: Comparison of recovery mechanisms between CASE-1 and REF-CASE-1

Drilling schedule resulted from CASE-1 and the one used in REF-CASE-1 are depicted in Figure 3.6. In CASE-1, a longer plateau duration is attained by developing Løve and Nesehorn simultaneously, and then followed by Sebra. On the other hand, for the reference case, the reservoirs are developed sequentially, i.e. started with Løve, followed by Nesehorn, and ended by Sebra.

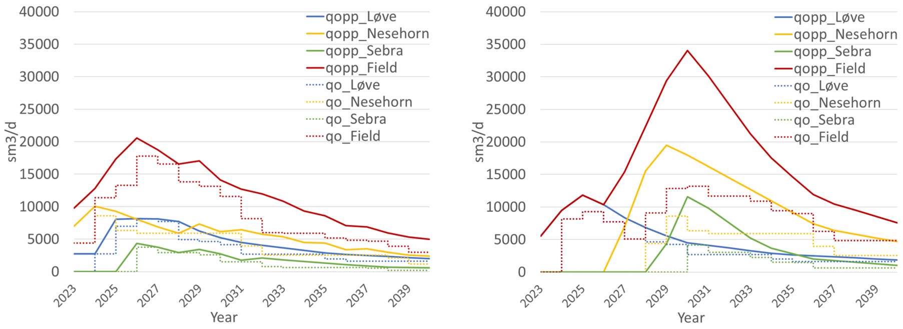


(a) CASE-1

(b) REF-CASE-1

Figure 3.6: Comparison of drilling schedule between CASE-1 and REF-CASE-1

Figure 3.7 compares q_o and q_{opp} for both cases. As shown in the figures, the oil production rates are always no greater than the oil production potential, which means the rates are feasible to deliver. In addition, the increases of production potential at some points in time are resulted from the drilling of new oil producers.

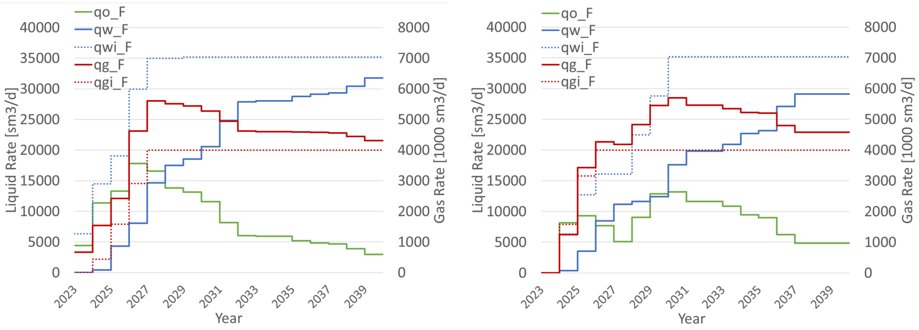


(a) CASE-1

(b) REF-CASE-1

Figure 3.7: Comparison of q_o and q_{opp} between CASE-1 and REF-CASE-1

The production & injection profiles for both cases are presented in Figure 3.8. By observing the injection rates, it is found that the injection constraints (Equation 3.4a and Equation 3.4b) are active in both cases. For Løve, the gas injection rate hits the gas injection capability, while for Nesehorn and Sebra, there are limitations regarding the water injection. Those active injection constraints justify why the oil production rates never reach the desired plateau rate, even though the production potential is sometimes way higher than the desired plateau rate. If we observe Figure 3.6a, those active injection constraints also justify why we drill fewer wells per year from 2028 onward. When the injection constraints are active, note that increasing the production potential by increasing the number of oil producers will not contribute anything to the production profile.



(a) CASE-1

(b) REF-CASE-1

Figure 3.8: Comparison of production & injection profiles between CASE-1 and REF-CASE-1

Chapter 4

Maximizing the NPV

In this chapter, an optimization problem which maximizes the Net Present Value (NPV) will be discussed. This optimization problem is basically an extension of the optimization problem presented in chapter 3. Fluid flow rates, drilling schedule, and recovery mechanism are the decision variables in this optimization problem. Similarly, few limitations in production, injection, and drilling are taken into account. This optimization problem is referred as CASE-2.

Another optimization problem called REF-CASE-2 is also established as a reference to compare the results of CASE-2. The reference case has an identical objective function as CASE-2, yet it only involves the fluid flow rates as the variables to decide. The reference case requires drilling schedule and recovery mechanism as optimization inputs, and those values are obtained from the specialization project (Angga, 2018). Brief comparisons between CASE-2 and REF-CASE-2 are given in Table 4.1.

Table 4.1: Comparisons of objective function, decision variables, and parameters between CASE-2 and REF-CASE-2

		CASE-2	**REF-CASE-2**
Objective Function		Max: NPV	Max: NPV
Production & Injection Rates	q_o	variable	variable
	q_g	variable	variable
	q_w	variable	variable
	q_{gi}	variable	variable
	q_{wi}	variable	variable
Drilling Schedule	N_{op}	variable	parameter
	N_{gi}	variable	parameter
	N_{wi}	variable	parameter
Recovery Mechanism	RM	variable	parameter

4.1 Formulation of the Optimization Problem

Similar to the optimization problem presented in chapter 3, some non-linear functions, such as $q_{opp} = f(N_p, N_{op}, RM)$ and $RM = f(zN_{gi}, zN_{wi})$, are involved in this optimization. Therefore, PWL approximations are again employed to convert the optimization problem into an MILP. The mathematical formulation of CASE-2 will be discussed in the following subsections.

4.1.1 Parameters

Except for the desired plateau rate $q_{o\_F\_plateau}$, all parameters presented in subsection 3.1.1 are relevant for CASE-2. A few new parameters required to compute NPV are introduced. Those additional parameters are listed, defined, and specified as follows:

- P_o
  definition: Oil price (unit: USD/bbl)
  value: 60
  comment: The oil price is assumed constant throughout the production period.

- XR
  definition: Exchange rate (unit: NOK/USD)
  value: 8.5
  comment: A fixed exchange rate is used to convert any revenue or expense in USD into an equivalent amount in NOK.

- VC
  definition: Volume conversion constant (unit: bbl/m^3)
  value: 6.29

- d
  definition: Discount rate
  value: 12%
  comment: This number is provided by Aker Solutions.

- $A_p, \quad \forall p \in \{1, 2, \ldots, 4\}$
  definition: Fixed exploration expenditure at time-p
  value: see Table 4.2
  comment: A cost proxy model that is developed in the specialization project (Angga, 2018) is adapted for NPV calculation. In that proxy model, the exploration expenditure (ExpEx) is modeled as a fixed cost. The subscript p in A_p indicates when the ExpEx is made (see Table 4.2 for the details).

Table 4.2: Fixed expenditure for exploration activities (Angga, 2018)

p	A_p (Mill. NOK)	Date of Expenditure
1	665	1st Jan 2019
2	240	1st Jan 2020
3	380	1st Jan 2021
4	366	1st Jan 2022

- B
  definition: Drilling cost per well (unit: Mill. NOK/well)
  value: 480
  comment: In the cost proxy model, the drilling cost is assumed constant for any well type (producer/injector), any reservoir, and anytime the drilling commenced.

- $C_p, \quad \forall p \in \{1, 2, \ldots, 4\}$
  definition: Fixed abandonment expenditure at time-p
  value: see Table 4.3
  comment: In the cost proxy model, the abandonment expenditure (AbEx) is expressed as follows:

  $$\text{AbEx} = \text{fixed cost} + f(N_{op\_F,n_p}) + g(N_{gi\_F,n_p}, N_{wi\_F,n_p}) \tag{4.1a}$$

  The term "fixed cost" in Equation 4.1a is composed by parameters C_1, C_2, C_3, and C_4. The subscript p in C_p also indicates when the AbEx is made (see Table 4.3 for the details).

Table 4.3: Fixed expenditure for abandonment activities (Angga, 2018)

p	C_p (Mill. NOK)	Date of Expenditure
1	192	1st Jan 2019
2	449	1st Jan 2020
3	449	1st Jan 2021
4	192	1st Jan 2022

- $D_q, \quad \forall q \in \{1, 2\}$
  definition: D_1 is the abandonment cost of an oil producer (unit: Mill. NOK/oil producer), while D_2 reflects the abandonment cost for every injector (unit: Mill. NOK/injector)
  value: D_1 and D_2 are equal to 12 and 5, respectively
  comment: In Equation 4.1a, the AbEx is also influenced by two linear functions, i.e., $f(N_{op\_F,n_p})$ and $g(N_{gi\_F,n_p}, N_{wi\_F,n_p})$. Those functions are defined as fol-

lows:

$$f(N_{op\_F,n_p}) = D_1 \cdot N_{op\_F,n_p} \tag{4.1b}$$

$$g(N_{gi\_F,n_p}, N_{wi\_F,n_p}) = D_2 \cdot (N_{gi\_F,n_p} + N_{wi\_F,n_p}) \tag{4.1c}$$

where N_{op_F,n_p}, N_{gi_F,n_p}, and N_{wi_F,n_p} are the number of oil producers, gas injectors, and water injectors, respectively, at the end of the field lifetime (time-z $= n_p$).

- $E_r, \quad \forall r \in \{1, 2\}$
  definition: Coefficients of a linear function that estimates the capital cost of subsea equipment
  value: see Table 4.4
  comment: In the cost proxy model, the capital expenditure (CapEx) for subsea equipment is estimated as follows:

$$\text{Subsea CapEx} = E_1 \cdot N_{t\_F} + E_2 \cdot N_{x\_F} \tag{4.2}$$

where N_{t_F} and N_{x_F} are the total number of subsea templates and subsea Xmas trees in the field, respectively.

Table 4.4: All the coefficients in Equation 4.2 (Angga, 2018)

r	E_r	Unit
1	484	Mill. NOK/template
2	50	Mill. NOK/Xmas tree

- $F_s, \quad \forall s \in \{1, 2, 3\}$
  definition: Coefficients of a linear function that determines the capital cost of an offshore structure
  value: see Table 4.5
  comment: In the cost proxy model, the capital expenditure (CapEx) for the offshore structure is expressed as follows:

$$\text{Topside CapEx} = F_1 \cdot \hat{q}_{o\_F} + F_2 \cdot \hat{q}_{g\_F} + F_3 \cdot \hat{q}_{w\_F} \tag{4.3}$$

where $\hat{q}_{o\_F}$, $\hat{q}_{g\_F}$, and $\hat{q}_{w\_F}$ are maximum oil, gas, and water production rate of the field, respectively. Since the study is only focused on the development with FPSO, the numbers presented in Table 4.5 are relevant for FPSO alternative.

Table 4.5: All the coefficients in Equation 4.3 (Angga, 2018)

s	F_s	Unit
1	0.10914	Mill. NOK/(sm^3/d)
2	0.23848	Mill. NOK/(1000 sm^3/d)
3	0.22074	Mill. NOK/(sm^3/d)

- $G_t, \quad \forall t \in \{1, 2, \ldots, 5\}$
  definition: Constant and coefficients of a linear function that computes the operating expense for a particular year
  value: see Table 4.6
  comment: In the cost proxy model, the operating expense (OpEx) for a particular year is expressed as follows:

  $$\text{OpEx} = G_1 + G_2 \cdot N_{op\_F} + G_3 \cdot \dot{q}_{o\_F} + G_4 \cdot \dot{q}_{g\_F} + G_5 \cdot \dot{q}_{w\_F} \tag{4.4}$$

  where N_{op_F} is the number of oil producers in the field at that year, while $\dot{q}_{o\_F}$, $\dot{q}_{g\_F}$, and $\dot{q}_{w\_F}$ are the average production rate of oil, gas, and water, respectively, of the field at that year. Similarly, the numbers provided in Table 4.6 correspond for the development with FPSO.

Table 4.6: All the constant and the coefficients in Equation 4.4 (Angga, 2018)

t	G_t	Unit
1	400	Mill. NOK
2	4	Mill. NOK/(oil producer)
3	0.007388	Mill. NOK/(sm^3/d)
4	0.018289	Mill. NOK/(1000 sm^3/d)
5	0.022829	Mill. NOK/(sm^3/d)

4.1.2 Sets

The sets defined in this formulation are exactly the same as the sets defined in subsection 3.1.2.

4.1.3 Variables

In addition to the variables presented in subsection 3.1.3, there are other variables involved in this formulation. Those additional variables are needed for NPV calculation, and they are listed and defined as follows:

- PV_e
  definition: Present value of the exploration expenditure (unit: Mill. NOK)
- PV_{dp}
  definition: Present value of the drilling expenditure for the pre-drilled wells (unit: Mill. NOK)
- PV_a
  definition: Present value of the abandonment expenditure (unit: Mill. NOK)
- $N_{t,i}, \quad \forall i \in R$
  definition: Number of subsea templates in reservoir-i

- N_{t_F}
  definition: Total number of subsea templates in the field

- N_{x_F}
  definition: Total number of subsea Xmas trees in the field

- PV_{sc}
  definition: Present value of the capital expenditure for the subsea equipment (unit: Mill. NOK)

- $\hat{q}_{o\_F}$
  definition: Maximum oil production rate of the field (unit: sm^3/d)

- $\hat{q}_{g\_F}$
  definition: Maximum gas production rate of the field (unit: 1000 sm^3/d)

- $\hat{q}_{w\_F}$
  definition: Maximum water production rate of the field (unit: sm^3/d)

- PV_{tc}
  definition: Present value of the capital expenditure for the offshore structure (unit: Mill. NOK)

- PV_{ap}
  definition: Present value of all expenditures made prior to the first oil date, i.e., at 1st Jan 2023 (unit: Mill. NOK)

- $PV_{d,z}, \quad \forall z \in \{1, 2, \ldots, (n_p - 1)\}$
  definition: Present value of the drilling expenditure at time-z (unit: Mill. NOK)

- $PV_{o,z}, \quad \forall z \in \{2, 3, \ldots, n_p\}$
  definition: Present value of the operating expense at time-z (unit: Mill. NOK)

- $PV_{r,z}, \quad \forall z \in \{2, 3, \ldots, n_p\}$
  definition: Present value of the revenue at time-z (unit: Mill. NOK)

- $DCF_z, \quad \forall z \in P$
  definition: Present value of the cash flow at time-z (unit: Mill. NOK)

- NPV
  definition: Net present value (unit: Mill. NOK)

4.1.4 Objective Function

The objective function of the optimization problem is to maximize the net present value (NPV). The objective function is expressed as follows:

$$\text{maximize} \quad NPV \tag{4.5}$$

It is important to note that the reference date for NPV calculation is 1st Jan 2019.

4.1.5 Additional Constraints

The formulation of CASE-2 uses all the constraints presented in chapter 3, except for the one related to the desired plateau rate (Equation 3.2). In addition to that, there are several other constraints involved in this formulation. Those constraints are important for NPV calculation, and they are discussed as follows:

Exploration Expenditure

The costs proxy model estimates the future values of the expenditures. Therefore, when computing the present value, one needs to discount the future values with the specified discount rate. In this study, the present date refers to 1st Jan 2019. Thus, the present value of the ExpEx is found as follows:

$$PV_e = \sum_{p \in \{1,2,\ldots,4\}} \frac{A_p}{(1+d)^{(p-1)}} \tag{4.6}$$

The exponent $p-1$ denotes the number of periods (in our case, the number of years) between the date of expenditure and the present date.

Drilling Expenditure

- As informed by Aker Solutions, all the pre-drilled wells are drilled in 2022, and the associated DrillEx is made at 1st Jan 2022. Consequently, the present value of the DrillEx for the pre-drilled wells is computed as follows:

  $$PV_{dp} = \frac{B \cdot N_{wt\_F,1}}{(1+d)^3} \tag{4.7}$$

  The variable $N_{wt\_F,1}$ points out the number of pre-drilled wells in the field.

- The drilling cost of all wells drilled within a particular year must be paid in the beginning of that year. For instance, the drilling cost of all wells drilled in 2023 must be paid at 1st Jan 2023. Therefore, the present value of the DrillEx at time-z is determined as follows:

  $$PV_{d,z} = \frac{B \cdot (N_{wt\_F,z+1} - N_{wt\_F,z})}{(1+d)^{(z+3)}}, \quad \forall z \in \{1,2,\ldots,(n_p-1)\} \tag{4.8}$$

  The term $(N_{wt\_F,z+1} - N_{wt\_F,z})$ reflects the number of wells drilled between time-z and z+1. The DrillEx for these wells must be made at time-z. Similarly, the exponent $z+3$ denotes the number of periods between the date of expenditure and the present date.

Abandonment Expenditure

The present value of the AbEx is calculated as follows:

$$PV_a = \left(\sum_{p \in \{1,2,\ldots,4\}} \frac{C_p}{(1+d)^{(p-1)}} \right) + D_1 \cdot N_{op\_F,n_p} + D_2 \cdot (N_{gi\_F,n_p} + N_{wi\_F,n_p}) \tag{4.9}$$

where N_{op_F,n_p}, N_{gi_F,n_p}, and N_{wi_F,n_p} are the number of oil producers, gas injectors, and water injectors, respectively, at the end of the field lifetime (time-z $= n_p$). In the equation above, the leftmost term represents the present value of the fixed abandonment expenditure, whereas the other two terms represent the present value of well plugging and abandonment (P&A) cost. Since the expenditure for well P&A is made at once at 1$^{\text{st}}$ Jan 2019, the last two terms are not discounted.

Subsea Capital Expenditure

- As decided by Aker Solutions, all the oil producers are drilled from 4-slot subsea templates. As consequence, the number of subsea templates varies, depending on the number of oil producers. To find out the number of subsea templates required in each reservoir, the following constraint is included:

$$N_{op,i,n_p} \leq 4 \cdot N_{t,i}, \quad \forall i \in R \tag{4.10a}$$

  where N_{op,i,n_p} is the number of oil producers in reservoir-i at the end of the field lifetime (time-z $= n_p$). Since we have an optimization problem to maximize NPV, the optimization will eliminate the unnecessary cost. Therefore, as the number of subsea templates affects the subsea CapEx, the optimization will give the smallest value of $N_{t,i}$ such that the constraint above is satisfied. With the number of subsea templates in each reservoir, the number of all subsea templates in the field (N_{t_F}) is simply determined as follows:

$$N_{t\_F} = \sum_{i \in R} N_{t,i} \tag{4.10b}$$

- Unlike the oil producers, all the injectors are drilled from different locations, separated from the oil producers and from the other injectors. With this subsea configuration, each injector is equipped with subsea Xmas tree. Therefore, the number of subsea Xmas trees in the field (N_{x_F}) is obtained as follows:

$$N_{x\_F} = N_{gi\_F,n_p} + N_{wi\_F,n_p} \tag{4.11}$$

- In the cost proxy model, the subsea CapEx has linear dependency on N_{t_F} and N_{x_F} (see Equation 4.2). Accordingly, the present value of subsea CapEx is determined as follows:

$$PV_{sc} = \sum_{p \in \{1,2,\ldots,4\}} \frac{E_1 \cdot N_{t\_F} + E_2 \cdot N_{x\_F}}{4 \cdot (1+d)^{(p-1)}} \tag{4.12}$$

  The term "4" in the denominator emerges because the subsea CapEx is evenly distributed from 2019 until 2022 (4 years).

Topside Capital Expenditure

- In the cost proxy model, the topside CapEx depends on the maximum oil, gas, and water production rate of the field (see Equation 4.3). Those maximum productions rates are obtained by imposing the following constraints:

$$q_{o\_F,z} \leq \hat{q}_{o\_F}, \quad \forall z \in \{2, 3, \dots, n_p\} \tag{4.13a}$$

$$q_{g\_F,z} \leq \hat{q}_{g\_F}, \quad \forall z \in \{2, 3, \dots, n_p\} \tag{4.13b}$$

$$q_{w\_F,z} \leq \hat{q}_{w\_F}, \quad \forall z \in \{2, 3, \dots, n_p\} \tag{4.13c}$$

  The idea of including the constraints above to determine the maximum oil, gas, and water production rates is similar to the idea of determining $N_{t,i}$.

- With those maximum oil, gas, and water production rates, the present value of topside CapEx is obtained by the following expression:

$$PV_{tc} = \sum_{p \in \{1,2,\dots,4\}} \frac{F_1 \cdot \hat{q}_{o\_F} + F_2 \cdot \hat{q}_{g\_F} + F_3 \cdot \hat{q}_{w\_F}}{4 \cdot (1+d)^{(p-1)}} \tag{4.14}$$

  Similarly, we have "4" in the denominator because the topside CapEx is equally divided into 4 years of construction & installation period, i.e., from 2019 until 2022.

All Expenditures Prior to the First Oil Date

The variable PV_{ap} summarizes all expenditures made before the field enters the production period, i.e., before 1st Jan 2023. One can determine the variable PV_{ap} using the following expression:

$$PV_{ap} = PV_e + PV_{dp} + PV_a + PV_{sc} + PV_{tc} \tag{4.15}$$

Operating Expense

Referring to the cost proxy model for OpEx (see Equation 4.4), the present value of the OpEx in a particular year is obtained as follows:

$$PV_{o,z} = \frac{G_1 + G_2 \cdot N_{op\_F,z-1} + G_3 \cdot q_{o\_F,z} + G_4 \cdot q_{g\_F,z} + G_5 \cdot q_{w\_F,z}}{(1+d)^{(z+3)}}, \quad \forall z \in \{2, 3, \dots, n_p\} \tag{4.16}$$

In the equation above, z is the element of a set $\{2, 3, \dots, n_p\}$. This is because the OpEx for a particular year incurs at the end of that year. For example, the OpEx for year 2023 incurs at 1st Jan 2024.

In Equation 4.16, the term $G_2 \cdot N_{op\_F,z-1}$ represents the well maintenance cost. For a particular year (i.e., from time $z-1$ until z), this well maintenance is only performed to the oil producers that are available in the beginning of that year, i.e. at time $z-1$. This is the reason of having subscript $z-1$ for N_{op_F}.

In the cost proxy model for OpEx (see Equation 4.4), the coefficients G_3, G_4, and G_5 are multiplied to the average field production rate of oil, gas, and water, respectively, at a particular year. Since we applies backward rectangular integration technique for computing the cumulative production, the production rate at time z reflects the constant production rate from time $z-1$ until z. In consequence, as shown in Equation 4.16, those average production rates are substituted with the production rates at time z.

Revenue

In this optimization, the revenue only comes from the sales of oil production. The present value of the revenue in a particular year is given as follows:

$$PV_{r,z} = \frac{P_o \cdot XR \cdot VC \cdot (N_{p\_F,z} - N_{p\_F,z-1})}{(1+d)^{(z+3)}}, \quad \forall z \in \{2, 3, \ldots, n_p\} \tag{4.17}$$

The term $(N_{p\_F,z} - N_{p\_F,z-1})$ in the equation above represents the volume oil production in a particular year, i.e., from time $z-1$ until time z. As for the OpEx, z is the element of a set $\{2, 3, \ldots, n_p\}$ because the revenue in a particular year is acquired at the end of that year.

Cash Flow & NPV

- The discounted values of the cash flow of various points in time are computed using the following equations. These equations simply imply the balance between revenue, OpEx, and DrillEx.

$$DCF_1 = -PV_{d,1} \tag{4.18a}$$

$$DCF_z = PV_{r,z} - PV_{o,z} - PV_{d,z}, \quad \forall z \in \{2, 3, \ldots, (n_p - 1)\} \tag{4.18b}$$

$$DCF_{n_p} = PV_{r,n_p} - PV_{o,n_p} \tag{4.18c}$$

- The NPV is finally determined using the following equation:

$$NPV = -PV_{ap} + \sum_{z \in P} DCF_z \tag{4.19}$$

It is important to note that the variables DCF_z and NPV are the only variables that are not constrained to have non-negative values.

Integrality Constraints

To have physically logical optimization results, some of the variables are imposed to be integers:

$$N_{t,i} \in \mathbb{Z}, \quad \forall i \in R \tag{4.20a}$$

$$N_{t\_F} \in \mathbb{Z} \tag{4.20b}$$

$$N_{x\_F} \in \mathbb{Z} \tag{4.20c}$$

4.2 Implementation in AMPL

The implementation of optimization problem CASE-2 in AMPL is very much alike to the AMPL implementation of CASE-1 (see section 3.2). As shown in Figure 4.1, an extra file (marked by the red box) is established. This file contains the additional inputs required for NPV calculation, i.e., the parameters defined in subsection 4.1.1. The AMPL implementations for both CASE-2 and REF-CASE-2 are provided in Appendix B.1 and Appendix B.2, respectively.

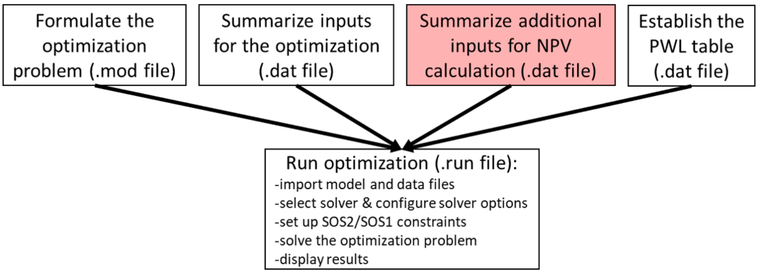

Formulate the optimization problem (.mod file)
Summarize inputs for the optimization (.dat file)
Summarize additional inputs for NPV calculation (.dat file)
Establish the PWL table (.dat file)
Run optimization (.run file):
-import model and data files
-select solver & configure solver options
-set up SOS2/SOS1 constraints
-solve the optimization problem
-display results

Figure 4.1: Workflow of AMPL implementation for CASE-2

4.3 Results & Discussion

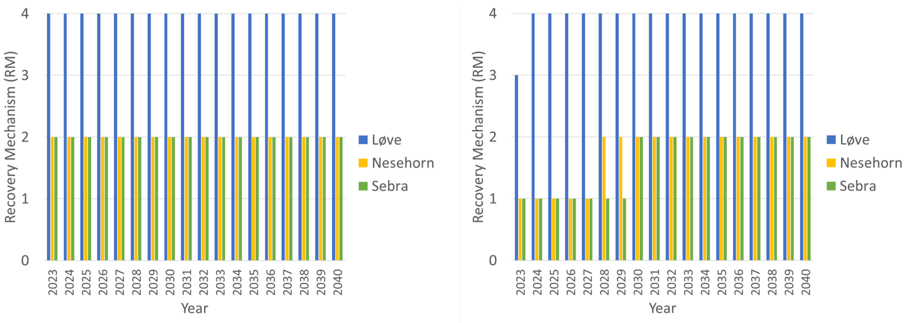
Comparisons of optimization results between CASE-2 and REF-CASE-2 are provided in Table 4.7. As shown in this table, CASE-2 produces 84.44% higher NPV than the reference case. This fact is in accordance to the previous conclusion, i.e., the objective value can be further improved by considering more decision variables. Referring to the value of relative gap, the solution obtained in CASE-2 is not guaranteed as the optimal solution because there might exist other solutions which could enhance the NPV up to 37482 Mill. NOK (16.88% higher than the current best NPV). On the other side, the relative gap for REF-CASE-2 points out that the optimal solution has been found and the objective value cannot be improved anymore.

Table 4.7: Comparisons of optimization results between CASE-2 and REF-CASE-2

	CASE-2	**REF-CASE-2**
Objective value (Mill. NOK)	32068	17387
Relative gap	16.88%	0.00%
Runtime (sec.)	43413	0.13
Runtime (hr.)	12	0.00
Number of variables	8197	3898
Number of constraints	2603	2101

For CASE-2, the optimization process is intentionally stopped after 12 hours. Once it is stopped, the relative gap is still fairly high (16.88%). A further investigation finds out that, after approximately one hour of the simulation run, the best integer is no longer increasing, while the best bound progresses very slowly. One can indeed extend the optimization run to lessen the relative gap until it reaches a desired value.

Comparison of recovery mechanism between CASE-2 and REF-CASE-2 is presented in Figure 4.2. As shown in Figure 4.2a, the optimization figures out that, for maximizing the NPV, Løve should be developed with water-gas injection ($RM = 4$), while the production of Nesehorn and Sebra should be supported by water injection ($RM = 2$). For the reference case, we are fortunate since we input this optimal configuration of recovery mechanisms.



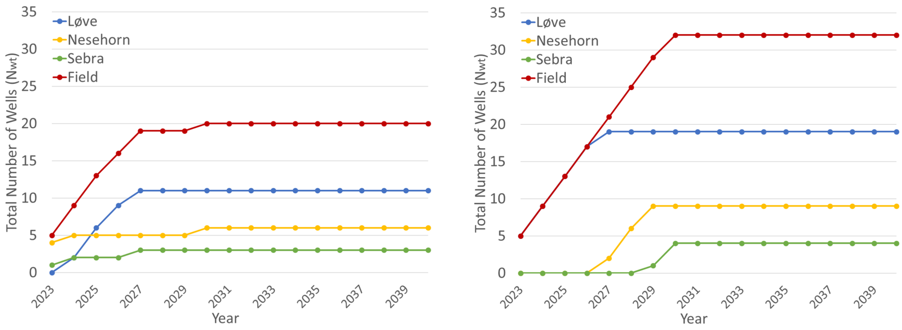
(a) CASE-2

(b) REF-CASE-2

Figure 4.2: Comparison of recovery mechanisms between CASE-2 and REF-CASE-2

Figure 4.3 compares the drilling schedule between CASE-2 and REF-CASE-2. According to Figure 4.3a, the optimization suggests developing all the reservoirs simultaneously in order to maximize the NPV. It contradicts to the input drilling schedule for REF-CASE-2, where the reservoirs are developed sequentially.

Another meaningful recommendation when including drilling schedule as decision variable is to drill fewer wells ($N_{wt} = 20$ in CASE-2 & $N_{wt} = 32$ in REF-CASE-2). The reason behind is, when maximizing the NPV, the optimization will rule out the unnecessary cost. Based on the production & injection profile of CASE-2, it is found that the field production is limited by the processing capacity of produced water and by the water injection capability. As consequence, drilling an extra oil producer will give no impact on the production profile, and is a waste of money. The optimization process thus avoids to drill an excessive number of oil producers.

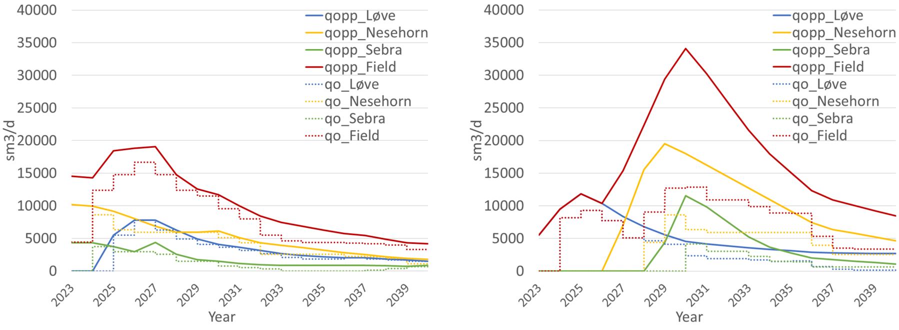


(a) CASE-2 **(b)** REF-CASE-2

Figure 4.3: Comparison of drilling schedule between CASE-2 and REF-CASE-2

The comparison of q_o and q_{opp} between CASE-2 and REF-CASE-2 is depicted in Figure 4.4. As shown in Figure 4.4a, the difference between q_o and q_{opp} is small in CASE-2. In contrast, q_{opp} is way higher than q_o in the reference case. This huge discrepancy between q_o and q_{opp} in REF-CASE-2 is because of the excessive number of oil producers and restricted production (due to some production & injection constraints).

Furthermore, if one compares the q_{o_F} profile obtained in both cases, it is found that the CASE-2 has a higher production in the early production period. This kind of production profile typically provides a higher NPV because the revenues are discounted less. Therefore, it can be inferred that including the drilling schedule as the decision variable makes it possible to have a higher production in the early production period, which generally leads to a higher NPV.

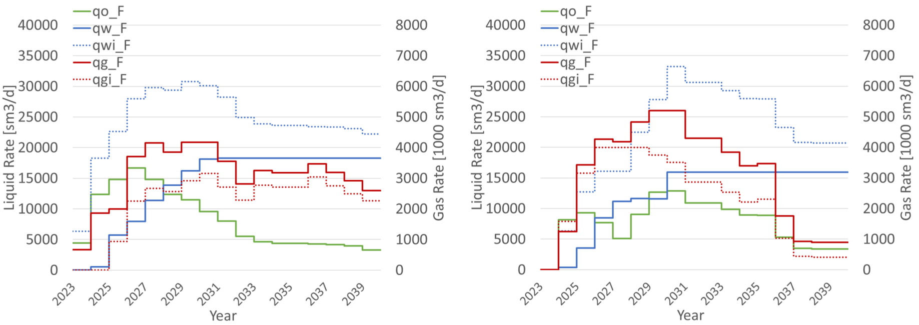


(a) CASE-2 **(b)** REF-CASE-2

Figure 4.4: Comparison of q_o and q_{opp} between CASE-2 and REF-CASE-2

If one observes the production & injection profile of each reservoir in both cases, it is found that the production of Nesehorn and Sebra is restrained by water injection capability. In addition, the field production is bounded by the processing capacity of produced water. This active constraint is identified from the flat q_{w_F} in Figure 4.5. Increasing the processing capacity of produced water ($\hat{q}_{w\_F}$) is not favored because of two factors, i.e.:

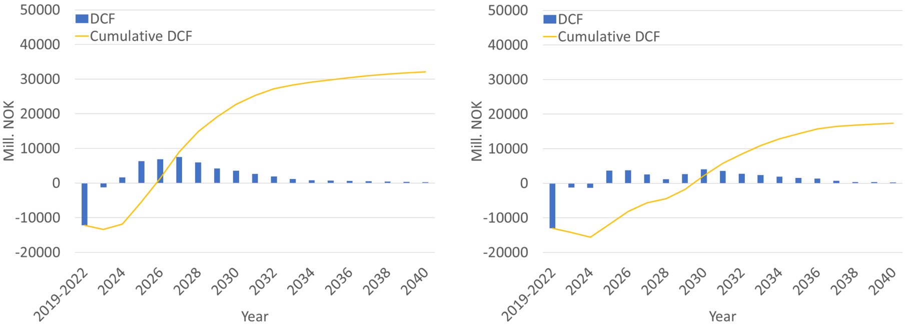
- It increases CapEx which significantly lowers the NPV since the CapEx incurs in the beginning of the field development.
- It only slightly increases the field oil production rate and the increment of oil production occurs at the late period of the field development, meaning that the increase of oil production has minimal contribution on the NPV.



(a) CASE-2 **(b)** REF-CASE-2

Figure 4.5: Comparison of production & injection profiles between CASE-2 and REF-CASE-2

The comparison of DCF and its cumulative between CASE-2 and REF-CASE-2 is provided in Figure 4.6. In addition to 84.44% higher NPV, CASE-2 also produces a shorter payout time (POT) than the reference case. As shown in the figure, it takes approximately 11 years to return the initial investment in REF-CASE-2, while the payout in CASE-2 happens only after 7 years (36.36% shorter). By comparing the NPV and the POT, it is worthwhile to optimize not only the fluid flow rates, but also the drilling schedule and the recovery mechanism as in CASE-2.



(a) CASE-2 **(b)** REF-CASE-2

Figure 4.6: Comparison of DCF & its cumulative between CASE-2 and REF-CASE-2

Chapter 5

Improving the Computational Efficiency & Accuracy

Several efforts have been deployed to make the formulations of the optimization problems and their implementations in AMPL not only more efficient but also more accurate. These efforts will be discussed in this chapter. The results of these efforts have actually been adapted to the formulations and the AMPL implementations presented in chapter 3 and chapter 4.

5.1 Reducing the Number of Breakpoints for PWL Approximation

In PWL approximation, the number of breakpoints correlates with the number of variables. The more breakpoints involved means the more variables to decide, and thus leads to a longer optimization process. One surely can reduce the number of breakpoints to shorten the optimization runtime. However, if one uses too few breakpoints, the PWL function might not be able to represent the non-linear behaviours of the original function, and it causes inaccuracy of optimization results. Thus, the objective of this section is to reduce the number of breakpoints so that it reduces the optimization runtime, while maintains the accuracy of the optimal solution. In this study, 3D PWL approximation is used to represent a non-linear function $q_{opp} = f(N_p, N_{op}, RM)$. The reduction of the number of breakpoints is applied for N_p breakpoints and N_{op} breakpoints.

5.1.1 Reducing the Number of N_p Breakpoints

Methodology

This study is implemented using a simple optimization problem, i.e., to maximize the plateau duration by adjusting only the oil production rate. The objective function, decision variable, and parameters of the optimization problem are summarized in Table 5.1. As presented in Table 5.2, seven cases are generated and run. The difference between these cases is only the number of N_p breakpoints ($n_{bp\_N_p}$).

Table 5.1: Objective function, decision variable, and parameters of the optimization problem used in the reduction of $n_{bp\_N_p}$ study

Objective Function		Max: t_p
Production & Injection Rates	q_o	variable
	q_g	not included
	q_w	not included
	q_{gi}	not included
	q_{wi}	not included
Drilling Schedule	N_{op}	parameter
	N_{gi}	not included
	N_{wi}	not included
Recovery Mechanism	RM	parameter

Table 5.2: List of cases for the reduction of $n_{bp\_N_p}$ study

Case	$n_{bp\_N_p}$
3	30
4	25
5	20
6	15
7	10
8	7
9	5

In this study, the N_p breakpoints are separated with an equal interval. The spacing between two consecutive N_p breakpoints depends on the number of N_p breakpoints, and is computed with the following expression:

$$\Delta N_p = \frac{\text{Max } N_p}{n_{bp\_N_p} - 1} \tag{5.1}$$

For example, when $n_{bp\_N_p}$ equals to ten, the N_p breakpoints for the scenario Løve reservoir produced with 15 oil producers and water-gas injection are illustrated in Figure 5.1. The red line in this figure represents the PWL function generated using those 10 N_p breakpoints. The process of generating a PWL table with a reduced number of N_p breakpoints is accomplished using Excel VBA.

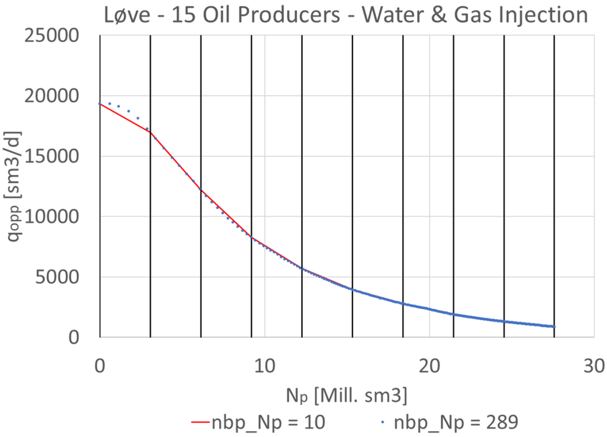


Figure 5.1: An illustration of N_p breakpoints for a case with 10 N_p breakpoints

Results

The more N_p breakpoints involved, the better the PWL function represents the original non-linear function, and the more accurate the optimization results are. Since Case 3 uses more N_p breakpoints than the other cases, the results obtained from this case are considered as the most accurate results, and defined as the references for comparison study.

To have a quantitative comparison, Relative Error (RE) is calculated. This RE indicates how far the results of a particular case deviate from the results of the reference case. The expression to compute RE is given as follows:

$$RE_{i,z}(\text{case-x}) = \frac{|q_{o,i,z}(\text{case-x}) - q_{o,i,z}(\text{ref. case})|}{q_{o,i,z}(\text{ref. case})} \tag{5.2}$$

where $q_{o,i,z}(\text{case-x})$ is the oil production rate of reservoir-i at time-z that is obtained in case-x.

The maximum RE of all cases are summarized in Table 5.3. As shown in the table, RE generally goes up when fewer N_p breakpoints are used. If we want to have RE no higher than 10%, the number of N_p breakpoints can be reduced up to 10.

Table 5.3: Comparison of the maximum RE between case 4 to case 9

Case	nbp_Np	Maximum RE [ref: nbp_Np = 30]				Meet criteria?
		Løve	Nesehorn	Sebra	Field	
4	25	0.46%	6.61%	0.98%	3.19%	Yes
5	20	0.62%	2.07%	6.30%	1.12%	Yes
6	15	0.67%	3.24%	2.60%	1.60%	Yes
7	10	1.19%	3.96%	7.13%	2.51%	Yes
8	7	3.69%	4.65%	12.93%	3.69%	No
9	5	8.30%	9.18%	19.27%	8.30%	No

The relationship between the number of N_p breakpoints and the optimization runtime is presented in Figure 5.2. As shown in the figure, the runtime exponentially increases when the PWL approximation utilizes more N_p breakpoints. By reducing the number of N_p breakpoints from 30 to 10, we manage to make the optimization process 150 times faster.

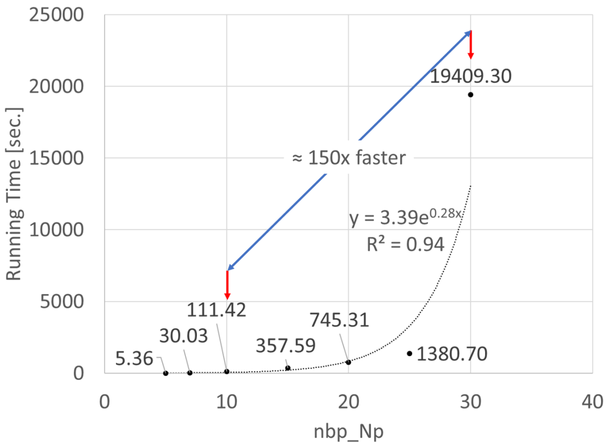


Figure 5.2: Relationship between the number of N_p breakpoints and the optimization runtime

A similar study is also carried out for another optimization problem. The optimization problem has the objective function to maximize the plateau duration by configuring not only the oil production rate but also the drilling schedule of the oil producers. It is found that the optimization results of a case that uses 10 N_p breakpoints are not significantly different from the optimization results of another case that utilizes more N_p breakpoints. According to this finding, it is assumed that the number of N_p breakpoints can be safely reduced to 10 N_p breakpoints in any optimization problems.

5.1.2 Reducing the Number of N_{op} Breakpoints

Methodology

This study is implemented using an optimization problem identical to the one presented in subsection 5.1.1. As presented in Table 5.4, seven cases are generated and run. The difference between these cases is only the number of N_{op} breakpoints ($n_{bp\_N_{op}}$). The table also presents the N_{op} breakpoints correspond to each reservoir in all cases.

Table 5.4: List of cases for the reduction of $n_{bp\_N_{op}}$ study

Case	$n_{bp\_N_{op},i}$		
	$i = 1$ (Løve)	$i = 2$ (Nesehorn)	$i = 3$ (Sebra)
10	16 $\{0, 1, 2, 3, \ldots, 14, 15\}$	8 $\{0, 1, 2, 3, \ldots, 6, 7\}$	4 $\{0, 1, 2, 3\}$
11	9 $\{0, 1, 3, 5, \ldots, 13, 15\}$	5 $\{0, 1, 3, 5, 7\}$	3 $\{0, 1, 3\}$
12	7 $\{0, 1, 4, 7, 10, 13, 15\}$	4 $\{0, 1, 4, 7\}$	3 $\{0, 1, 3\}$
13	5 $\{0, 1, 5, 10, 15\}$	4 $\{0, 1, 4, 7\}$	3 $\{0, 1, 3\}$
14	4 $\{0, 1, 8, 15\}$	3 $\{0, 1, 7\}$	3 $\{0, 1, 3\}$
15	3 $\{0, 1, 15\}$	3 $\{0, 1, 7\}$	3 $\{0, 1, 3\}$
16	2 $\{0, 15\}$	2 $\{0, 7\}$	2 $\{0, 3\}$

Results

Case 10 is assumed to produce the most accurate results because the case uses more N_{op} breakpoints than the other cases. Referring to the results of this case, RE is calculated using Equation 5.2. The maximum RE of all cases are summarized in Table 5.5. As shown in the table, RE generally increases when the PWL approximation involves fewer N_{op} breakpoints. With a cutoff for the maximal RE of 10%, the number of N_{op} breakpoints can be reduced to 5 for Løve, 4 for Nesehorn, and 3 for Sebra.

The optimization runtime is not relevant for comparison since this study is implemented in an easy optimization problem. In each case, the optimal solution is found rapidly, and no conclusion can be derived based on the relationship between the number of N_{op} breakpoints and the optimization runtime. However, it is believed that the reduction of the number of N_{op} breakpoints will give a huge reduction of optimization runtime in a more complex optimization problem.

Table 5.5: Comparison of the maximum RE between case 11 to case 16

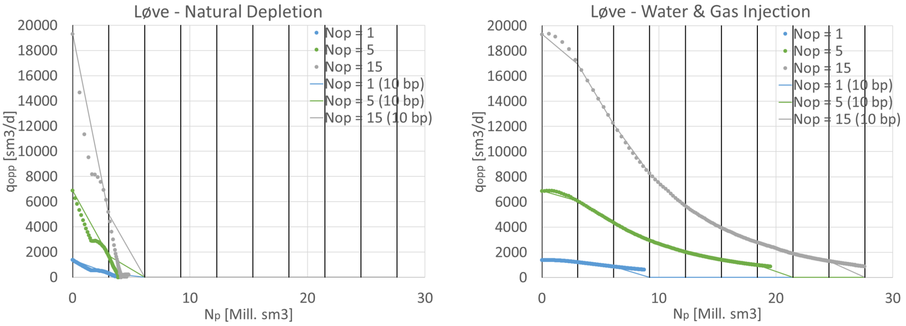
Case	Maximum RE [ref: Case 10]				Meet criteria?
	Løve	Nesehorn	Sebra	Field	
11	0.38%	1.69%	1.03%	0.84%	Yes
12	0.00%	3.41%	4.87%	1.61%	Yes
13	1.16%	3.41%	4.70%	1.69%	Yes
14	1.91%	8.65%	18.41%	4.81%	No
15	5.64%	8.65%	19.93%	5.64%	No
16	6.97%	17.81%	50.17%	7.96%	No

5.2 Selecting the N_p Breakpoints

In the previous section, we have talked about the optimum number of N_p breakpoints. This section discusses how to select the N_p breakpoints, whether to separate them with an equal interval or to manually select them.

Methodology

In the 3D PWL approximation, a unique set of N_p breakpoints is applied to all possible scenarios (all combinations of N_{op} and RM) for a particular reservoir. For clarification, let us see Figure 5.3. In this illustration, the same set of N_p breakpoints is used to reproduce the production potential curves of Løve reservoir with any number of oil producers and any recovery mechanism. In the figure, the dots represent the original production potential curves, while the solid lines represent the reproduced production potential curves with 10 N_p breakpoints.



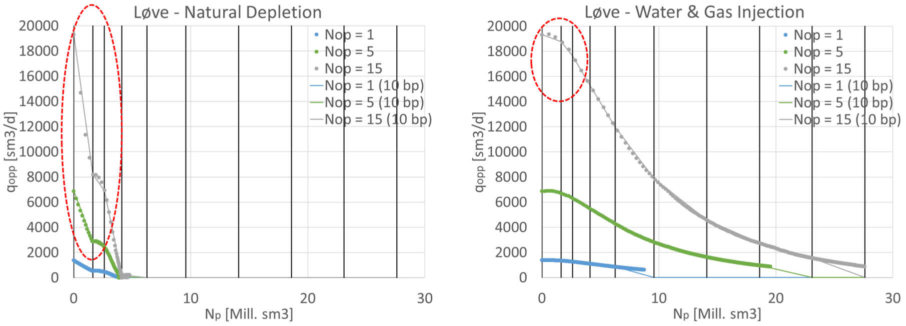
(a) Natural depletion

(b) Water-gas injection

Figure 5.3: An illustration of a set of N_p breakpoints used to reproduce the production potential curves of Løve reservoir with any number of oil producers and any recovery mechanism

In Figure 5.3, the production potential curves are reproduced with 10 N_p breakpoints, which is the optimum number of N_p breakpoints as discussed in the previous section. These N_p breakpoints are separated with equal spacing. As shown in the figure, the reproduced production potential curves with these N_p breakpoints do not fit the original production potential curves.

The N_p breakpoints are not necessary to be separated with equal spacing, and they can instead be selected manually to have better reproduction of the production potential curves. An illustration of wisely-selected N_p breakpoints is provided in Figure 5.4. The non-uniform distances between the N_p breakpoints are aimed to capture the characteristics of the production potential curves of the natural depletion scenarios. The red circles in Figure 5.4 indicate that the reproduced production potential curves with the wisely-selected N_p breakpoints have a better match with the original production potential curves.



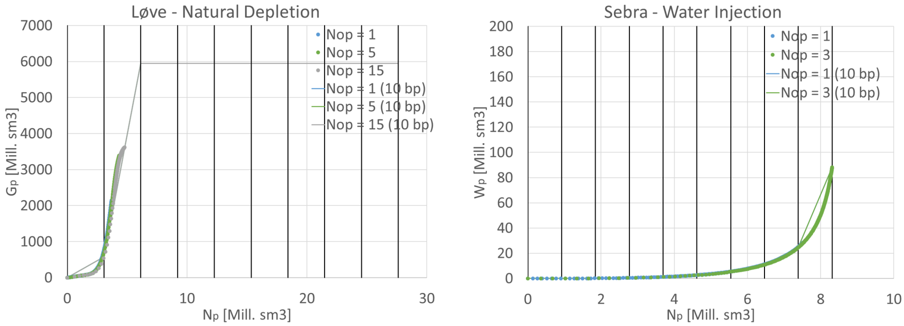
(a) Natural depletion

(b) Water-gas injection

Figure 5.4: An illustration of wisely-selected N_p breakpoints used to reproduce the production potential curves of Løve reservoir

The chosen N_p breakpoints are also utilized to replicate the G_p-N_p, W_p-N_p, G_i-N_p, and W_i-N_p curves. Therefore, to have a good reproduction of those curves, we have to carefully select the N_p breakpoints.

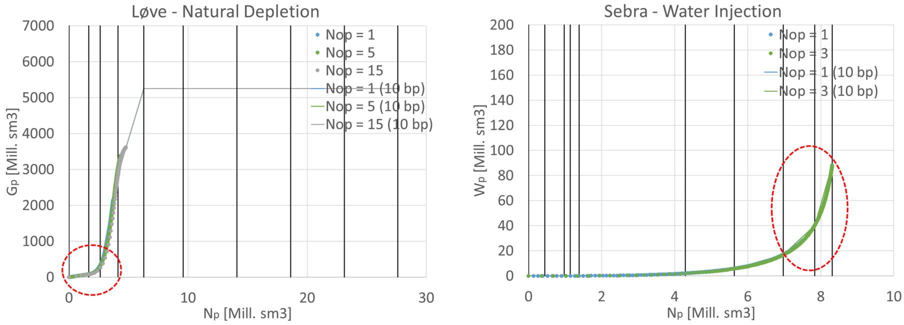
Figure 5.5 depicts the G_p-N_p and W_p-N_p curves that are reconstructed with 10 equally-spaced N_p breakpoints, whereas Figure 5.6 shows the reproduced G_p-N_p and W_p-N_p curves using 10 wisely-selected N_p breakpoints. By comparing these figures, we can see that the wisely-selected N_P breakpoints give a better reproduction of the G_p-N_p and W_p-N_p curves (as indicated by the red circles in Figure 5.6).



(a) G_p-N_p curve of Løve reservoir produced with natural depletion

(b) W_p-N_p curve of Sebra reservoir produced with water injection

Figure 5.5: The reproduced G_p-N_p and W_p-N_p curves using 10 equally-spaced N_p breakpoints



(a) G_p-N_p curve of Løve reservoir produced with natural depletion

(b) W_p-N_p curve of Sebra reservoir produced with water injection

Figure 5.6: The reproduced G_p-N_p and W_p-N_p curves using 10 wisely-selected N_p breakpoints

As previously discussed, a better reproduction of the production potential curves and cumulative-N_p curves is attained by wisely-selecting the N_p breakpoints. This better reproduction makes the optimization results more accurate (lower RE).

5.3 Determining Gas Production Rate

This section discusses two methods for determining the gas production rate (q_g). The first method uses 2D interpolation, while the other one utilizes the G_p-N_p curve. The more efficient method is then adapted for the optimization problems presented in chapter 3 and chapter 4.

Methodology

Evaluation of both methods is carried out using a simple optimization problem. The objective function, decision variables, and parameters of the optimization problem are summarized in Table 5.6. There is no constraint regarding the gas production rate. Therefore, the gas production rate is only determined based on the oil production rate.

Table 5.6: Objective function, decision variables, and parameters of the optimization problem used in the determination of q_g study

Objective Function		Max: t_p
Production & Injection Rates	q_o	variable
	q_g	variable
	q_w	not included
	q_{gi}	not included
	q_{wi}	not included
Drilling Schedule	N_{op}	parameter
	N_{gi}	not included
	N_{wi}	not included
Recovery Mechanism	RM	parameter

In the first method, the procedure for determining the gas production rate is provided as follows:

1. Estimate the gas-oil ratio of reservoir-i at time-z ($GOR_{i,z}$) using GOR-N_p curves
   comment: For a particular reservoir, $GOR_{i,z}$ is controlled by $N_{p,i,z}$ and RM_i. Therefore, we can estimate $GOR_{i,z}$ using GOR-N_p curves. The equation to compute $GOR_{i,z}$ is given as follows:

$$GOR_{i,z} = \sum_{j \in V_{N_p}} \sum_{k \in V_{N_{op},i}} \sum_{l \in V_{RM,i}} \lambda_{i,z,j,k,l} \cdot \overline{GOR}_{,i,j,k,l}, \quad \forall i \in R \quad \text{and} \quad \forall z \in P \tag{5.3}$$

   The equation above is alike to Equation 3.16a, which is used to calculate $G_{p,i,z}$ based on G_p-N_p curves.

2. Compute $q_{g,i,z}$ using 2D PWL approximation
   comment: By definition, $q_{g,i,z}$ can be obtained by multiplying $q_{o,i,z}$ with $GOR_{i,z}$. However, introducing this constraint into the formulation makes the optimization problem becomes an MINLP (since we have a multiplication of variables). To retain the optimization problem as an MILP, 2D PWL approximation is applied to represent the non-linear function $q_g = f(q_o, GOR)$. Visualization of the 2D interpolation is provided in Figure 5.7. The additional constraints induced by the 2D PWL approximation are fundamentally the same as Equation 2.13 presented in subsection 2.3.2.

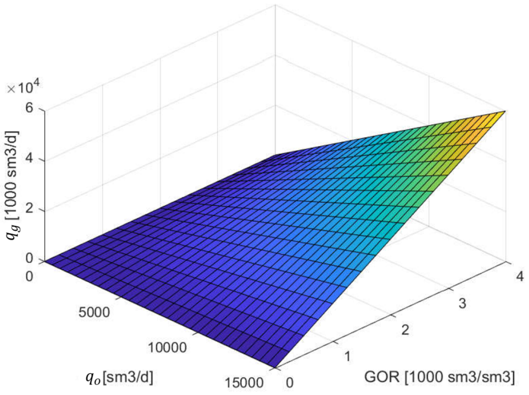


Figure 5.7: 2D PWL approximation to represent a non-linear function $q_g = f(q_o, GOR)$

In the second approach, the gas production rate is obtained by following the series of steps:

1. Estimate $G_{p,i,z}$ using G_p-N_p curves
comment: $G_{p,i,z}$ is found to have a non-linear dependency on $N_{p,i,z}$ and RM_i, so we can estimate G_p using G_p-N_p curves. G_p is computed with the following expression:

$$G_{p,i,z} = \sum_{j \in V_{N_p}} \sum_{k \in V_{N_{op},i}} \sum_{l \in V_{RM,i}} \lambda_{i,z,j,k,l} \cdot \overline{Gp}_{,i,j,k,l}, \quad \forall i \in R \quad \text{and} \quad \forall z \in P \tag{5.4}$$

2. Using $G_{p,i,z}$, estimate $q_{g,i,z}$ based on backward rectangular integration technique
comment: In backward rectangular integration technique, $G_{p,i,z}$ is computed as follows:

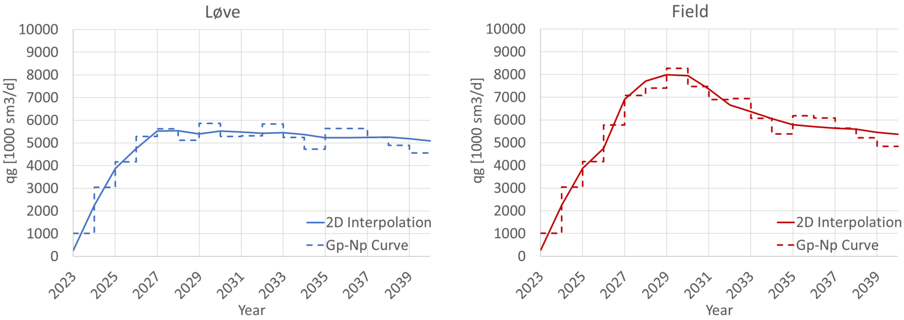
$$G_{p,i,z} = G_{p,i,z-1} + \frac{t_{up} \cdot q_{g,i,z}}{10^3}, \quad \forall i \in R \quad \text{and} \quad \forall z \in \{2, ..., n_p\} \tag{5.5a}$$

In equivalent, the equation above is expressed as follows:

$$q_{g,i,z} = \frac{(G_{p,i,z} - G_{p,i,z-1}) \cdot 10^3}{t_{up}}, \quad \forall i \in R \quad \text{and} \quad \forall z \in \{2, ..., n_p\} \tag{5.5b}$$

Results

Figure 5.8 compares the gas production profile provided by both methods. At a glance, they look similar. To have a quantitative comparison, RE is computed with Equation 5.2. The results from the second method are used as the reference since they are considered more accurate. Erroneous outcomes of the first method might originate from improper reproduction of the GOR-N_p curves with the selected 10 N_p breakpoints. The average RE for all reservoirs and the field are presented in Table 5.7. According to this table, it is concluded that both methods produce very similar outcomes.



(a) Gas production profile of Løve reservoir

(b) Gas production profile of the field

Figure 5.8: Comparison of the gas production profile obtained from both methods

Table 5.7: Average RE for all reservoirs and the field

	Average RE [ref: results of method 2]
Løve	6.00%
Nesehorn	5.73%
Sebra	8.92%
Field	5.28%

Figure 5.9 compares the optimization runtime required by both methods. As shown in this figure, the optimization problem can be solved considerably quicker when the second method is applied. Based on the average RE and the optimization runtime, the second method is inferred more efficient, and thus adapted for more complex optimization problems as presented in chapter 3 and chapter 4. In addition, the idea of the second method is applied to determine q_w, q_{gi}, and q_{wi}.

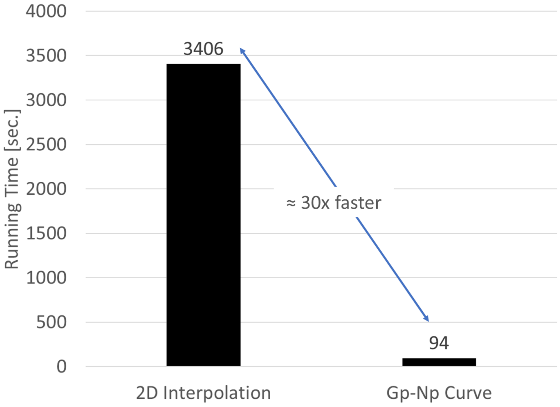


Figure 5.9: Comparison of the optimization runtime required by both methods

5.4 Choosing the Numerical Integration Technique for Calculating the Cumulative

Intuitively, trapezoidal integration technique would provide more accurate results than rectangular integration technique. However, in our main optimization problems (presented in chapter 3 and chapter 4), the cumulative production & injection at a particular point in time is numerically calculated using the backward rectangular integration technique. The reason of applying this integration technique instead of the trapezoidal integration technique is explained in this section.

5.4.1 Cumulative Gas Production

Methodology

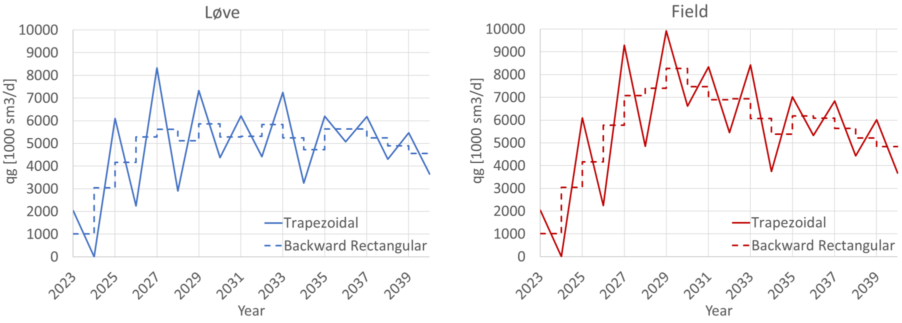
The study is conducted using the same optimization problem as in section 5.3. For the case with backward rectangular integration technique, Equation 5.5a is used. In contrary, for the case with trapezoidal integration technique, Equation 5.5a is replaced with the following equation:

$$G_{p,i,z} = G_{p,i,z-1} + \frac{t_{up} \cdot \frac{(q_{g,i,z-1}+q_{g,i,z})}{2}}{10^3}, \quad \forall i \in R \quad \text{and} \quad \forall z \in \{2, ..., n_p\} \qquad (5.6)$$

Results

Comparison of gas production profile between the case applying backward rectangular integration technique and the case adapting trapezoidal integration technique is presented in Figure 5.10. As shown in this figure, the gas production rate oscillates severely in the case employing the trapezoidal integration technique. A further investigation finds out that the oscillations occur because we have an under-determined system, i.e., fewer equations than unknowns. Referring to Equation 5.6, note that we use $n_p - 1$ equations ($z \in \{2, ..., n_p\}$) to determine the gas production rate of n_p points in time.

The oscillation issue does not emerge when the backward rectangular integration technique is applied. It is because we use $n_p - 1$ equations to determine the gas production rate of $n_p - 1$ points in time (see Equation 5.5a). This integration technique is thus preferred to compute the gas cumulative production. For the same reason, the cumulative water production (W_p) and the cumulative injection of gas and water (G_i & W_i) are also computed using the backward rectangular integration technique.



(a) Gas production profile of Løve reservoir

(b) Gas production profile of the field

Figure 5.10: Comparison of gas production profile between the case applying backward rectangular integration technique and the case adapting trapezoidal integration technique

5.4.2 Cumulative Oil Production

Methodology

The study is implemented using a simple optimization problem, i.e., to maximize the plateau period of a single reservoir (Løve) production. The production of Løve is supported with water injection. Fifteen oil producers and a water injector are available from the beginning of the production period. Maximization of the plateau duration is achieved by configuring the oil production and water injection rates. The objective function, decision variables, and parameters of the optimization problem are summarized in Table 5.8.

Table 5.8: Objective function, decision variables, and parameters of the optimization problem used in the evaluation of the numerical integration

Objective Function		Max: t_p
Production & Injection Rates	q_o	variable
	q_g	not included
	q_w	not included
	q_{gi}	not included
	q_{wi}	variable
Drilling Schedule	N_{op}	parameter
	N_{gi}	not included
	N_{wi}	parameter
Recovery Mechanism	RM	parameter

In this study, three cases are examined. Details of the cases are given as follows:

1. Case 21: Compute N_p using trapezoidal integration technique and ignore the water injection constraint
2. Case 22: Compute N_p using trapezoidal integration technique and include the water injection constraint
3. Case 23: Compute N_p using backward rectangular integration technique and include the water injection constraint

The equations which compute N_p with both integration techniques and the equation which represents the water injection limitation are provided as follows:

- Computation of N_p with trapezoidal integration technique

$$N_{p,z} = N_{p,z-1} + \frac{t_{up} \cdot \frac{(q_{o,z-1}+q_{o,z})}{2}}{10^6}, \quad \forall z \in \{2, ..., n_p\} \tag{5.7}$$

- Computation of N_p with backward rectangular integration technique

$$N_{p,z} = N_{p,z-1} + \frac{t_{up} \cdot q_{o,z}}{10^6}, \quad \forall z \in \{2, ..., n_p\} \tag{5.8}$$

- Constraint regarding the water injection

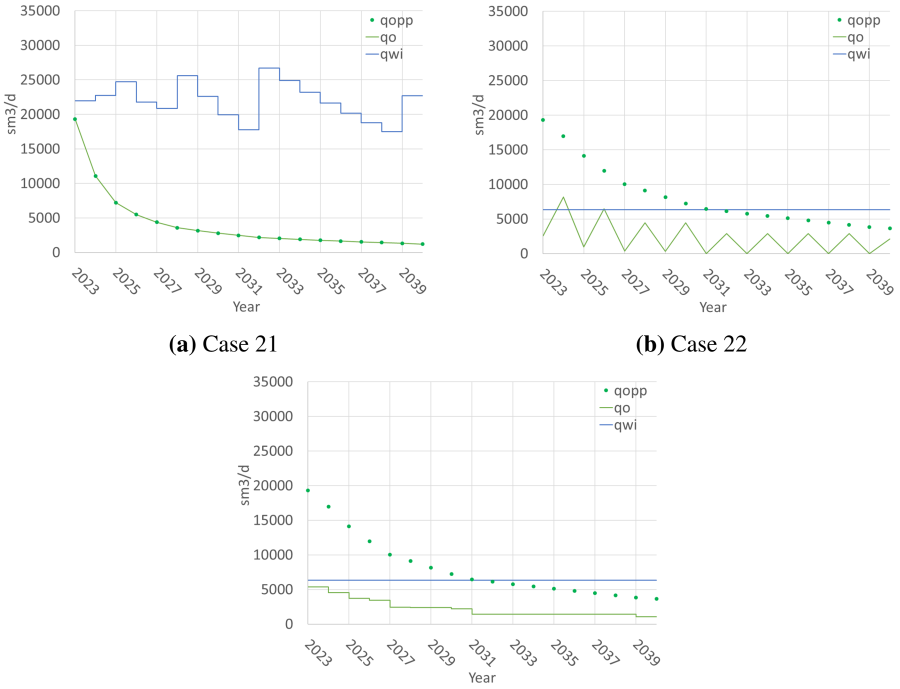
$$q_{wi,z} \leq N_{wi,z-1} \cdot 6360, \quad \forall z \in \{2, ..., n_p\} \tag{5.9}$$

Results

Figure 5.11 compares the optimization outcomes (q_{opp}, q_o, and q_{wi} profile) of the three cases. As shown in Figure 5.11a, the trapezoidal integration technique works well when the injection constraint is not active. The fact that the reservoir is always produced at its potential makes sense since the production is not restricted by any production or injection constraints.

In case 22, the oil production rate is found to oscillate. As shown in Figure 5.11b, q_{wi} is always at its upper limit (6360 sm^3/d), and it can be seen as "something that we already know". With the profile of q_{wi}, $W_i(t)$ can be easily determined. Subsequently, $N_p(t)$ is obtained using the W_i-N_p curves. With the profile of N_p, q_o is eventually determined using Equation 5.7. Referring to the equation, note that we use $n_p - 1$ equations ($z \in \{2, ..., n_p\}$) to determine the oil production rate of n_p points in time. It means we have an under-determined system. This system makes the solution of a linear system non-unique, and it is responsible for the oscillation occurrence in case 22.

The oscillation issue is mitigated once the backward rectangular integration method is used (see Figure 5.11c). This integration technique is thus applied to compute the oil cumulative production.



(a) Case 21

(b) Case 22

(c) Case 23

Figure 5.11: Comparison of the optimization outcomes between case 21 to case 23

5.5 Optimizing the Parallel Computing

In parallel computing, a large problem is divided into smaller ones so that each processing element can solve the smaller problem simultaneously with the other processing elements. Parallel computing often helps to reduce computational time. In general, the more the CPU cores involved, the shorter the runtime is. This section discusses the optimum number of CPU cores for solving our main optimization problems.

Methodology

The study is carried out using an optimization problem which maximizes the plateau duration. In this optimization problem, the production & injection rates and the drilling schedule are the decision variables, while the recovery mechanism is the parameter. The objective function, decision variables, and parameter of the optimization problem are summarized in Table 5.9. The input values for RM of Løve, Nesehorn, and Sebra are 4, 2, and 2, respectively. Based on a study which implements a brute force method, this combination of recovery mechanism would maximize the plateau duration as well as the NPV.

Table 5.9: Objective function, decision variables, and parameter of the optimization problem used in the evaluation of parallel computing

Objective Function		Max: t_p
Production & Injection Rates	q_o	variable
	q_g	variable
	q_w	variable
	q_{gi}	variable
	q_{wi}	variable
Drilling Schedule	N_{op}	variable
	N_{gi}	variable
	N_{wi}	variable
Recovery Mechanism	RM	parameter

Four cases which use different number of CPU cores are generated and run (see Table 5.10). With hyper-threading, a single physical CPU core appears as two logical CPU cores (threads) to an operating system. Hyper-threading permits the two logical CPU cores to share physical execution resources, and thus can speed things up.

Table 5.10: List of cases for the evaluation of parallel computing

Case	**Number of Cores**	**Number of Threads**
24	16	32
25	8	16
26	4	8
27	2	4

In AMPL, the way to specify the number of threads to use is by modifying an solver option called "threads". The value assigned to this option defines the maximum number of threads for any of the parallel algorithms.

Running the optimization process until a desired relative gap (0.01%) is achieved requires a lot of time. Therefore, to have an objective comparison, the optimization run is terminated once it reaches 12 hours.

Results

Table 5.11: Comparison of objective value and relative gap between case 24 to case 27

Case	**Objective Value**	**Relative Gap**	**Runtime (hr.)**
24	190947	5.91%	12
25	190421	3.24%	12
26	190421	1.55%	12
27	190421	2.42%	12

As shown in Table 5.11, the objective value is the same for case 25 to case 27. A smaller relative gap in case 26 indicates a quicker optimization process. Accordingly, solving the optimization problem with 4 cores (8 threads) is preferred.

The objective value of case 24 is the highest among the other cases, meaning that the solution obtained in this case is suboptimal compared to the solutions obtained in other cases. Slower optimization process when more CPU cores are involved might be due to the inefficient partition of the problem in AMPL or the process within the hardware.

The conclusion of this study is assumed to hold for the optimization problem which maximizes the NPV.

5.6 Choosing the Solver Type

There are many solvers available to solve an MIPL problem. In this section, an investigation is carried out to evaluate the performance of two extensively used solvers, i.e. CPLEX and Gurobi. In the end, the better solver for each optimization problem is decided.

Methodology

The study is conducted using two optimization problems, i.e., maximization of the plateau duration and to maximization of the NPV. The optimization problems have the same decision variables and parameter as listed in Table 5.9.

In this study, four cases are considered (see Table 5.12). The differences between the cases are only the objective function and the solver to use. It is important to mention that the solvers use mostly its default settings when performing the optimization process. Similarly to have a fair comparison, the optimization process is stopped after 12 hours.

Table 5.12: List of cases for the evaluation of solver type

Case	Objective Function	Solver
28	Max t_p	CPLEX
29	Max t_p	Gurobi
30	Max NPV	CPLEX
31	Max NPV	Gurobi

Results

As shown in Table 5.12, both case 28 and case 29 produce the same objective value. However, case 29 provides a slightly lower relative gap than case 28, meaning that the optimization process is quicker. Based on these results, Gurobi solver is preferred to solve the optimization problem which maximizes the plateau duration.

Identical objective values are obtained from case 30 and case 31. A smaller relative gap in case 30 than in case 31 indicates a quicker optimization process. It is thus concluded that CPLEX solver performs better in the optimization problem of NPV maximization.

Table 5.13: Comparison of objective value and relative gap between case 28 to case 31

Case	Objective Value	Relative Gap	Runtime (hr.)
28	190421	1.55%	12
29	190421	1.53%	12
30	32068	16.88%	12
31	32068	20.03%	12

5.7 Configuring the Solver Options

Modifying the solver options can affect not only the accuracy of the solution but also the time to seek the solution. The objective of this section is to find out the configuration of the solver options which can speed up the optimization process.

Methodology

The study is carried out using an optimization problem which maximizes the NPV. The optimization problem has the same decision variables and parameter as listed in Table 5.9. The study only uses the NPV maximization problem because we will use this optimization problem for uncertainty analysis, where many similar optimization problems need to be solved. Therefore, having a quicker optimization process is essential for uncertainty analysis.

As figured out in section 5.6, CPLEX solver performs better for the NPV maximization problem. Two CPLEX options are evaluated in this study, i.e., "mipemphasis" & "parallelmode". "mipemphasis" controls trade-offs between speed, feasibility, optimality, and moving bounds in MIP (IBM, 2019a), while "parallelmode" sets the parallel optimization mode (IBM, 2019b). The values for "mipemphasis" & "parallelmode" and their descriptions are given in Table 5.14 and Table 5.15.

Table 5.14: Values for "mipemphasis" and their descriptions (IBM, 2019a)

Value	Description
0	Balance optimality and feasibility (default)
1	Emphasize feasibility over optimality
2	Emphasize optimality over feasibility
3	Emphasize moving best bound

Table 5.15: Values for "parallelmode" and their descriptions (IBM, 2019b)

Value	Description
-1	Enable opportunistic parallel search mode
0	Automatic: let CPLEX decide whether to invoke deterministic or opportunistic search (default)
1	Enable deterministic parallel search mode

Twelve cases are generated and run in this study (see Table 5.16. These cases combine the possible values for "mipemphasis" and "parallelmode". To have an unbiased comparison, the runtime of all cases is limited for 1 hour.

Table 5.16: List of cases for the evaluation of solver options

Case	Value of "mipemphasis"	Value of "parallelmode"
32	0	0
33	0	-1
34	0	1
35	1	0
36	1	-1
37	1	1
38	2	0
39	2	-1
40	2	1
41	3	0
42	3	-1
43	3	1

Results

Table 5.17 summarizes objective value and relative gap obtained by every case. By comparing the objective value and the relative gap, it can be said that the configuration of CPLEX options in case 33 ("mipemphasis" = default & "parallelmode" = opportunistic mode) accelerates the optimization process. The opportunistic setting speeds up the optimization process because it entails less synchronization between threads and consequently may provide better performance (IBM, 2019b). However, in opportunistic mode, the actual optimization may differ from run to run, including the solution time itself and the path traveled in the search (IBM, 2019b). For this reason, it is preferred to run the CPLEX solver with its default settings.

Table 5.17: Comparison of objective value and relative gap between case 32 to case 43

Case	Objective Value	Relative Gap	Runtime (hr.)
32	31868	22.04%	1
33	31957	19.49%	1
34	31868	22.03%	1
35	25676	114.83%	1
36	22634	140.01%	1
37	24973	120.88%	1
38	30916	23.81%	1
39	29946	26.20%	1
40	30916	23.74%	1
41	25538	62.66%	1
42	31374	32.26%	1
43	26583	56.39%	1

5.8 Analyzing the Running Time

The main intention of this section is to see how the best integer solution improves with an extended optimization runtime.

Methodology

The study is accomplished using two optimization problems identical to the ones presented in section 5.6. As shown in Table 5.18, four cases are evaluated. The differences between the cases are only the objective function and the limit of optimization runtime.

Table 5.18: List of cases for the runtime analysis

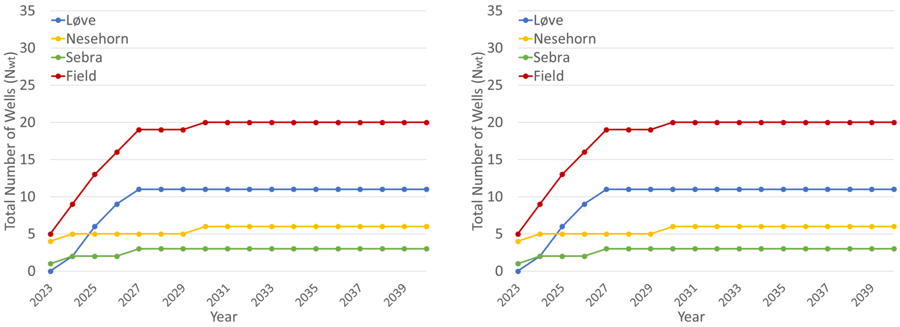
Case	Objective Function	Runtime Limit (hr.)
44	Max t_p	12
45	Max t_p	1
46	Max NPV	12
47	Max NPV	1

Results

As shown in Table 5.19, the objective value does not get better by prolonging the optimization runtime from 1 hour to 12 hours. In term of best bound, extending the optimization runtime manages to slightly move the best bound of the NPV maximization problem, but it has no contribution to the other optimization problem. A further investigation finds out that the cases with a longer optimization runtime deliver exactly the same results (production profile, injection profile, and drilling schedule) as the cases with a shorter optimization process. For instances, let us see Figure 5.12 and Figure 5.13.

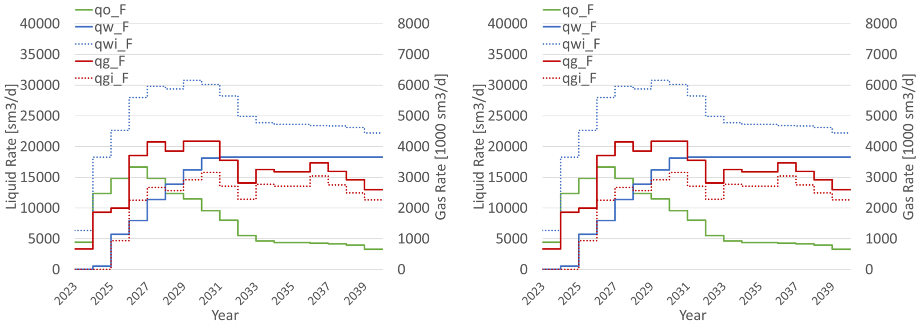
Table 5.19: Comparison of objective value, best bound, and relative gap between case 44 to case 47

Case	Objective Value	Best Bound	Relative Gap	Runtime (hr.)
44	190421	187515	1.53%	12
45	190421	187515	1.53%	1
46	32068	37482	16.88%	12
47	32068	38684	20.63%	1



(a) Case 46 **(b)** Case 47

Figure 5.12: Comparison of drilling schedule between case 46 and case 47



(a) Case 46 **(b)** Case 47

Figure 5.13: Comparison of production & injection profile between case 46 and case 47

Based on the runtime analysis, it is assumed that running the optimization algorithm for 1 hour will provide the optimal solution to the optimization problem. This assumption is beneficial for uncertainty analysis because many similar optimization problems need to be solved.

Chapter 6

Quantifying the Uncertainties

Uncertainties are a relevant topic to discuss when it comes to planning a field development. In our case, uncertainties of the optimization results mostly originate from the imperfect information used as the optimization inputs. This chapter is focused on discussing how the uncertainties in in-place, development & operational cost, and oil price affect the optimization results.

6.1 Proof of Concept

In the specialization project, it is shown how the NPV varies with the plateau rate (see Figure 6.1). Referring to the figure, the plateau rate which produces the highest NPV is considered as the optimal plateau rate. This section will discuss a concept proposed by Stanko (2019) regarding the optimal plateau rate and also the proof of the concept.

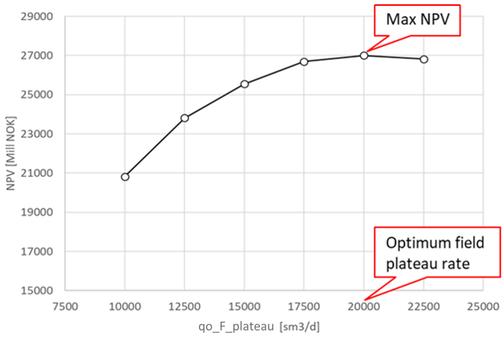


Figure 6.1: Relationship between the plateau rate and the NPV (Angga, 2018)

Concept

NPV is found by subtracting the present value of all the revenue with the present value of all the development cost. In our cost proxy model, the development cost is made up of six elements, i.e., ExpEx, AbEx, subsea CapEx, topside CapEx, DrillEx, and OpEx. Therefore, NPV is calculated as follows:

$$NPV = PV_{ar} - PV_e - PV_a - PV_{sc} - PV_{tc} - PV_{ad} - PV_{ao} \tag{6.1}$$

where PV_{ar}, PV_{ad}, and PV_{ao} are the present value of all revenues, all DrillEx, and all OpEx, respectively. PV_{ar}, PV_{ad}, and PV_{ao} are computed as follows:

$$PV_{ar} = \sum_{z=2}^{n_p} PV_{r,z} \tag{6.2a}$$

$$PV_{ad} = PV_{dp} + \sum_{z=1}^{n_p-1} PV_{d,z} \tag{6.2b}$$

$$PV_{ao} = \sum_{z=2}^{n_p} PV_{o,z} \tag{6.2c}$$

As observed in Figure 6.1, the maximum NPV corresponds to the point when:

$$\frac{\partial NPV}{\partial q_{p\_F}} = 0 \tag{6.3}$$

where q_{p_F} is the field plateau rate. By substituting Equation 6.1 into Equation 6.3, the following equation is acquired:

$$\frac{\partial PV_{ar}}{\partial q_{p\_F}} - \frac{\partial PV_e}{\partial q_{p\_F}} - \frac{\partial PV_a}{\partial q_{p\_F}} - \frac{\partial PV_{sc}}{\partial q_{p\_F}} - \frac{\partial PV_{tc}}{\partial q_{p\_F}} - \frac{\partial PV_{ad}}{\partial q_{p\_F}} - \frac{\partial PV_{ao}}{\partial q_{p\_F}} = 0 \tag{6.4}$$

In the cost proxy model, ExpEx is modeled as a fixed cost. Furthermore, AbEx, subsea CapEx, and Drillex only depend on the number of wells and its drilling schedule. Since the number of wells and the drilling schedule are fixed inputs for the study conducted in the specialization project, AbEx, subsea CapEx, and DrillEx are identical for any plateau rate. For these reasons, Equation 6.4 can be simplified as follows:

$$\frac{\partial PV_{ar}}{\partial q_{p\_F}} - \frac{\partial PV_{tc}}{\partial q_{p\_F}} - \frac{\partial PV_{ao}}{\partial q_{p\_F}} = 0 \tag{6.5}$$

Let us define PV_{tcao} (the present value of the topside CapEx and the total OpEx) as:

$$PV_{tcao} = Pv_{tc} + PV_{ao} \tag{6.6}$$

With the definition, Equation 6.5 can be written as:

$$\frac{\partial PV_{ar}}{\partial q_{p\_F}} - \frac{\partial PV_{tcao}}{\partial q_{p\_F}} = 0 \tag{6.7a}$$

or equivalently expressed as:

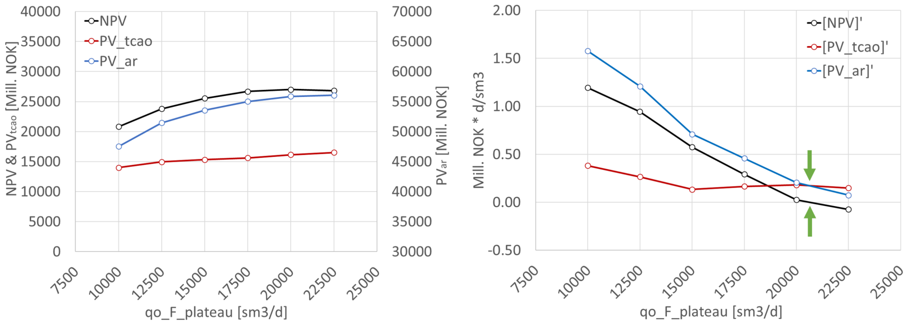
$$\frac{\partial PV_{ar}}{\partial q_{p\_F}} = \frac{\partial PV_{tcao}}{\partial q_{p\_F}} \tag{6.7b}$$

According to Equation 6.7b, it can be said that **the maximum NPV occurs at a plateau rate where the tangents of PV_{ar}-q_{p_F} and PV_{tcao}-q_{p_F} curves have the same slope.**

Proof

The relationships between NPV-PV_{ar}-PV_{tcao} and plateau rate are depicted in Figure 6.2a. As shown in the figure, the NPV curve (black line) indicates a maximum when q_{p_F} is around 20000 sm^3/d. Both PV_{ar} curve (blue curve) and PV_{tcao} curve (red curve) monotonously increase with the plateau rate. The growth of PV_{ar} slows down as the plateau rate rises. On the contrary, PV_{tcao} seems to grow linearly with the plateau rate. If we carefully observe the plateau rate which maximizes the NPV, the tangents of both PV_{ar} and PV_{tcao} curves have the same slope at this point.

To have a quantitative explanation, $\frac{NPV}{q_{p\_F}}$, $\frac{PV_{ar}}{q_{p\_F}}$, and $\frac{PV_{tcao}}{q_{p\_F}}$ are estimated numerically using central difference method, and the results are plotted in Figure 6.2b. As shown in the figure, the NPV is maximized when NPV' reaches zero (as indicated by the green arrows). At this plateau rate, the PV'_{ar} and PV'_{tcao} curves intersect each other, meaning that the slopes of the tangents of both curves are equal. This corresponds to what has been described earlier.



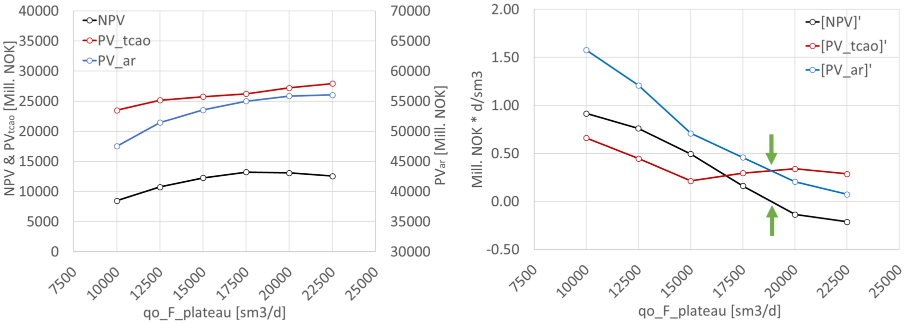
(a) NPV, PV_{ar}, PV_{tcao} vs. q_{p_F}

(b) NPV', PV'_{ar}, PV'_{tcao} vs. q_{p_F}

Figure 6.2: NPV, PV_{ar}, PV_{tcao}, and their first derivatives vs. q_{p_F} (base case)

The concept is also tested to another case where the CapEx is double. The relationships between NPV-PV_{ar}-PV_{tcao} and plateau rate are provided in Figure 6.3a. As shown in the figure, doubling the CapEx makes the PV_{tcao} curve is shifted up. As a result, the NPV curve is shifted down. In this case, the optimal plateau rate is found lower than the base case, i.e., around 17500 sm^3/d. Corresponds to the concept, both PV_{ar} and PV_{tcao} curves have the same gradient at this plateau rate.

The first derivatives of NPV, PV_{ar}, and PV_{tcao} with respect to q_{p_F} are presented in Figure 6.3b. As shown in the figure, doubling the CapEx makes the PV'_{tcao} curve to shift upward, and so the PV'_{tcao} curve cuts the PV'_{ar} curve at a lower plateau rate.



(a) NPV, PV_{ar}, PV_{tcao} vs. q_{p_F}

(b) NPV', PV'_{ar}, PV'_{tcao} vs. q_{p_F}

Figure 6.3: NPV, PV_{ar}, PV_{tcao}, and their first derivatives vs. q_{p_F} (CapEx is doubled)

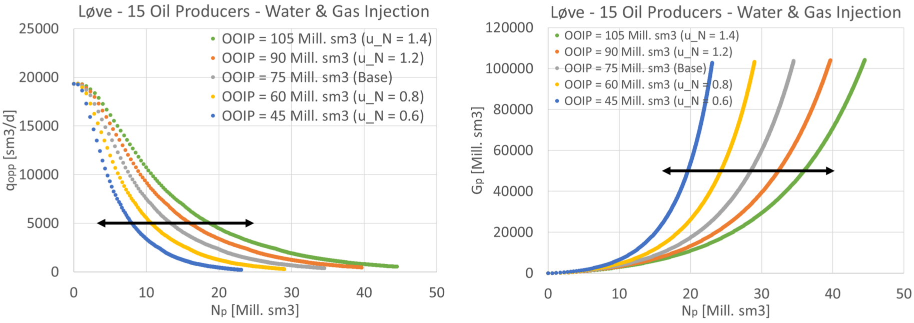
6.2 The Change of Production Potential Curve with In-Place

The uncertainty analysis carried out in this chapter includes in-place (N) as one of the uncertain parameters. Therefore, we need to know how a variation of in-place affects the production potential curve, and thus alters the optimization results. This section will talk about the change of production potential curve with in-place.

The study is carried out using one of the production scenarios, i.e., Løve reservoir that is produced with 15 oil producers and under water-gas injection recovery mechanism. As we know, the base value of Løve's in-place is 75 Mill. sm^3. Four new cases are created and simulated using GAP, i.e., a module in the Integrated Production Modelling (IPM) software that is designed to model a production network (Petroleum Experts, 2019). The difference between the cases is only the in-place of Løve reservoir, i.e., 105 Mill. sm^3, 90 Mill. sm^3, 60 Mill. sm^3, and 45 Mill. sm^3.

The production potential curves obtained from various Løve's in-places are exhibited in Figure 6.4a. In this figure, u_N represents the ratio of in-place in a particular case to the base value $\left(u_N = \frac{N}{N[Base]}\right)$. As shown in Figure 6.4a, if Løve's in-place is higher than its base value, the production potential curve would be stretched horizontally. On the other hand, the production potential curve would be squeezed in the horizontal direction if Løve's in-place is lower than its base value.

The change of G_p-N_p curve with in-place is shown in Figure 6.4b. As shown in the figure, the G_p-N_p curve is like being pulled horizontally if Løve's in-place is higher than its base value, and vice versa. This behaviour is not only discovered for G_p-N_p curve but also for W_p-N_p, G_i-N_p, and W_i-N_p curves.



(a) Production potential curve

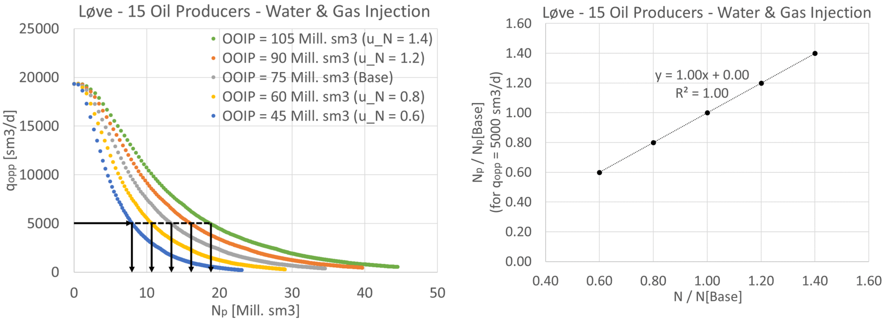
(b) G_p-N_p curve

Figure 6.4: The changes of production potential curve and G_p-N_p curve with in-place

To evaluate how far the production potential curve being stretched or squeezed when the in-place changes, the following approach is taken. Let us consider a particular q_{opp}, e.g., 5000 sm^3/d. The cumulative oil production (N_p) corresponds to this q_{opp} is then determined for each case using their production potential curve (see Figure 6.5a). $N_p[Base]$ is the N_p found from the case where the in-place equals to its base value. In this example, it is 13.35 Mill. sm^3.

After that, $\frac{N_p}{N_p[Base]}$ is computed and plotted against $\frac{N}{N[Base]}$ (see Figure 6.5b). Based on the relationship between $\frac{N_p}{N_p[Base]}$ and $\frac{N}{N[Base]}$, it can be inferred that **the change of in-place by $x\%$ from its base value would shift N_p of a particular q_{opp} by $x\%$ from its base value.**

The study has been extended to evaluate q_{opp} other than 5000 sm^3/d. It is found that the conclusion holds for any q_{opp}. There is also a study which quantifies the changes of G_p-N_p, W_p-N_p, G_i-N_p, and W_i-N_p curves with in-place. The results of this study are similar to what we have found in Figure 6.5b.



(a) N_p corresponds to a particular q_{opp} for various cases

(b) Relationship between $\frac{N_p}{N_p[Base]}$ and $\frac{N}{N[Base]}$

Figure 6.5: An approach to evaluate how far the production potential curve being stretched or squeezed when the in-place changes

6.3 Uncertain Parameters

The uncertainty analysis performed in this chapter involves 3 uncertain parameters, i.e., in-place, development & operational cost, and oil price. The details of each uncertain parameter are provided as follows:

1. **Uncertain parameter**: in-place
   Details:

   - The uncertainty of in-place is represented with u_N, which is defined as:

   $$u_N = \frac{N}{N[Base]} \tag{6.8}$$

   - The base values of Løve's in-place, Nesehorn's in-place, and Sebra's in-place are 75, 55, and 13 Mill. sm^3, respectively.

   - Changing u_N alters the in-place of all reservoirs. This simplification is made due to the time limitation in completing the uncertainty analysis. Of course, one can assign in-place uncertainty to every reservoir independently. However, it requires more cases to run, and thus takes a longer time to complete the uncertainty analysis.

   - The uncertainty of in-place is modeled with a normal distribution (see Figure 6.6a). The mean and the standard deviation of the normal distribution are 1 and 0.2, respectively.

   - The distribution of u_N refers to the in-place uncertainty of the Visund field, which is obtained from its PDO report (Norsk Hydro Produksjon A/S, 1995).

2. **Uncertain parameter**: development & operational cost
   Details:

   - The uncertainty of development & operational cost is represented with u_C, which is defined as:
   $$u_C = \frac{Cost}{Cost[Base]} \tag{6.9}$$

   - The base values of the development & operational cost are the ones obtained from the cost proxy model.

   - Changing u_C affects all the cost elements. This simplification is also made due to the time limitation in completing the uncertainty analysis.

   - The uncertainty of development & operational cost is modeled with a normal distribution (see Figure 6.6a). The mean and the standard deviation of the normal distribution are 1 and 0.2, respectively.

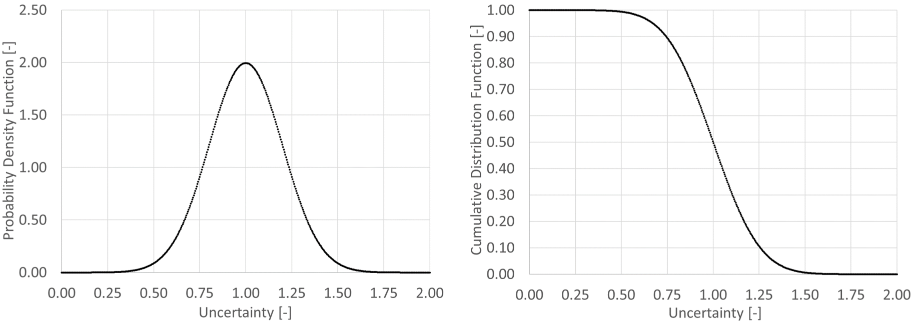
   - The distribution of u_C is typical to use for an early stage of field development (Hall and Delille, 2011).

3. **Uncertain parameter**: oil price
   Details:

   - The uncertainty of oil price is represented with u_{P_o}, which is defined as:
   $$u_{P_o} = \frac{P_o}{P_o[Base]} \tag{6.10}$$

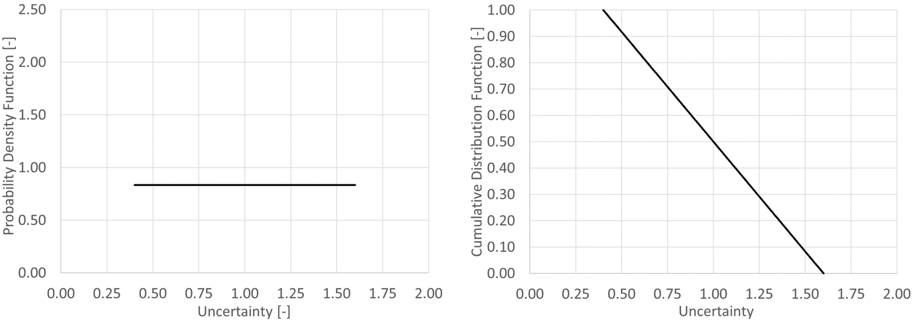
   - The base value of the oil price is 60 USD/bbl, and it is constant throughout the production period.

   - The uncertainty of oil price is modeled with a uniform distribution (see Figure 6.7a). The minimum and the maximum of the uniform distribution are 0.4 and 1.6, respectively. This distribution type is chosen to represent a higher degree of uncertainty possessed by the oil price.



(a) Probability Density Function (PDF) **(b)** Cumulative Distribution Function (CDF)

Figure 6.6: PDF & CDF of the normal distribution



(a) Probability Density Function (PDF) **(b)** Cumulative Distribution Function (CDF)

Figure 6.7: PDF & CDF of the uniform distribution

6.4 Modifications of the Mathematical Formulation

As discussed in the previous section, 3 uncertain parameters are considered for the uncertainty analysis, i.e., in-place, development & operational cost, and oil price. To include those uncertainties in our optimization problem, some modifications are made to the mathematical formulation. This section is focused to explain those modifications.

Uncertainty of In-place

As discussed in section 6.2, a chance of in-place transforms the production potential curves. One approach to include the in-place uncertainty in our optimization problem is by applying 4D PWL approximation, where the production potential (q_{opp}) is not only dependent on N_p, N_{op}, and RM, but also dependent on in-place (N). However, this approach is believed less efficient because a lot more variables are introduced, and it consequently requires a longer optimization runtime to arrive at the optimal solution.

Another approach is to use the existing PWL table, where the production potential (q_{opp}) is only determined by N_p, N_{op}, and RM. To include the in-place uncertainty into our optimization problem, the Equation 3.15a is replaced with the following equation:

$$N_{p,i,z} = u_N \cdot \left(\sum_{j \in V_{N_p}} \sum_{k \in V_{N_{op},i}} \sum_{l \in V_{RM,i}} \lambda_{i,z,j,k,l} \cdot \overline{N_p}_{,i,j,k,l} \right), \quad \forall i \in R \quad \text{and} \quad \forall z \in P \tag{6.11}$$

This modification can be made because we have verified that the shift of N_p from its base value is linearly dependent on the u_N (discussed in section 6.2). Note that the multiplier u_N in Equation 6.11 would would shift the N_p breakpoints from its base values. Since q_{opp} corresponds to each N_p breakpoint remains unchanged, applying the Equation 6.11 can be visualized as stretching or squeezing the production potential curves. One more thing, we use u_N as the multiplier because we have found that $\frac{N_p}{N_p[Base]} = \frac{N}{N[Base]} = u_N$ (see Figure 6.5b).

Uncertainty of Development & Operational Cost

Including the uncertainty of development & operational cost into our optimization problem is more straightforward. Correspond to its definition, u_C is used as a multiplier to modify the cost from its base value, i.e. the one that is estimated using the cost proxy model. The followings are the list of equations replaced along with the replacements.

- Equation 4.6 is replaced with the following equation:

$$PV_e = u_C \cdot \left(\sum_{p \in \{1,2,\ldots,4\}} \frac{A_p}{(1+d)^{(p-1)}} \right) \tag{6.12a}$$

- Equation 4.7 is replaced with the following equation:

$$PV_{dp} = u_C \cdot \left(\frac{B \cdot N_{wt\_F,1}}{(1+d)^3} \right) \tag{6.12b}$$

- Equation 4.8 is replaced with the following equation:

$$PV_{d,z} = u_C \cdot \left(\frac{B \cdot (N_{wt\_F,z+1} - N_{wt\_F,z})}{(1+d)^{(z+3)}} \right), \quad \forall z \in \{1, 2, \ldots, (n_p - 1)\} \tag{6.12c}$$

- Equation 4.9 is replaced with the following equation:

$$PV_a = u_C \cdot \left(\left(\sum_{p \in \{1,2,\ldots,4\}} \frac{C_p}{(1+d)^{(p-1)}} \right) + D_1 \cdot N_{op\_F,n_p} + D_2 \cdot (N_{gi\_F,n_p} + N_{wi\_F,n_p}) \right) \tag{6.12d}$$

- Equation 4.12 is replaced with the following equation:

$$PV_{sc} = u_C \cdot \left(\sum_{p \in \{1,2,\ldots,4\}} \frac{E_1 \cdot N_{t\_F} + E_2 \cdot N_{x\_F}}{4 \cdot (1+d)^{(p-1)}} \right) \tag{6.12e}$$

- Equation 4.14 is replaced with the following equation:

$$PV_{tc} = u_C \cdot \left(\sum_{p \in \{1,2,\ldots,4\}} \frac{F_1 \cdot \hat{q}_{o\_F} + F_2 \cdot \hat{q}_{g\_F} + F_3 \cdot \hat{q}_{w\_F}}{4 \cdot (1+d)^{(p-1)}} \right) \tag{6.12f}$$

- Equation 4.16 is replaced with the following equation:

$$PV_{o,z} = u_C \cdot \left(\frac{G_1 + G_2 \cdot N_{op\_F,z-1} + G_3 \cdot q_{o\_F,z} + G_4 \cdot q_{g\_F,z} + G_5 \cdot q_{w\_F,z}}{(1+d)^{(z+3)}} \right), \forall z \in \{2, \ldots, n_p\} \tag{6.12g}$$

Uncertainty of Oil Price

To include the oil price uncertainty, Equation 4.17 is replaced with the following equation:

$$PV_{r,z} = u_{P_o} \cdot \left(\frac{P_o \cdot XR \cdot VC \cdot (N_{p\_F,z} - N_{p\_F,z-1})}{(1+d)^{(z+3)}} \right), \quad \forall z \in \{2, 3, \ldots, n_p\} \tag{6.13}$$

In the equation above, note that we utilize u_{P_o} as a multiplier to change the oil price from its base value (60 USD/bbl).

6.5 Optimization Problem & Implementation in AMPL

The uncertainty analysis is carried out using an optimization problem which maximizes the NPV. In this optimization problem, the fluid rates and the drilling schedule are the decision variables, while the recovery mechanism is the parameter. The objective function, decision variables, and parameter of the optimization problem are summarized in Table 6.1.

Table 6.1: Objective function, decision variables, and parameter of the optimization problem used in the uncertainty analysis

Objective Function		Max: NPV
Production & Injection Rates	q_o	variable
	q_g	variable
	q_w	variable
	q_{gi}	variable
	q_{wi}	variable
Drilling Schedule	N_{op}	variable
	N_{gi}	variable
	N_{wi}	variable
Recovery Mechanism	RM	parameter

The reason of including the recovery mechanism as optimization input is to reduce the number of variables so that we have a lower relative gap after running the optimization for a defined amount of time. To ensure that the optimization produces the highest NPV, the input values for RM are carefully selected. The best combination of RM is found using brute-force approach, where all possible combinations of RM are tested and compared. According to the brute-force approach, the highest NPV is obtained from a case where Løve is developed with water-gas injection, while Nesehorn and Sebra are developed with water injection. In this uncertainty analysis, this combination of RM is adapted as optimization inputs, and it is assumed that this combination of RM always delivers the highest NPV, regardless of the change in in-place, cost, and oil price.

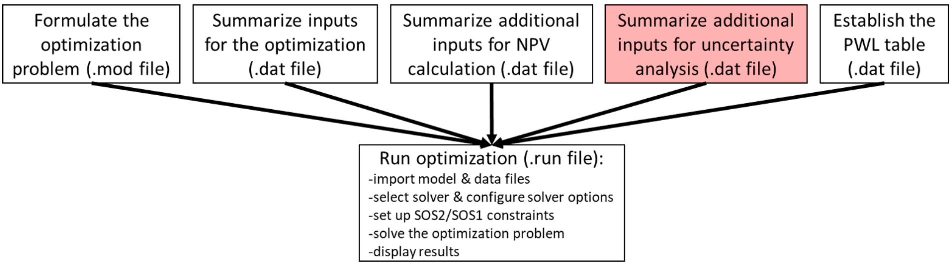

Formulate the optimization problem (.mod file)
Summarize inputs for the optimization (.dat file)
Summarize additional inputs for NPV calculation (.dat file)
Summarize additional inputs for uncertainty analysis (.dat file)
Establish the PWL table (.dat file)
Run optimization (.run file):
-import model & data files
-select solver & configure solver options
-set up SOS2/SOS1 constraints
-solve the optimization problem
-display results

Figure 6.8: Workflow of AMPL implementation of the optimization problem used in the uncertainty analysis

The AMPL implementation of the optimization problem used in the uncertainty analysis is very much similar to the AMPL implementation of CASE-2 (see section 4.2). As shown in Figure 6.8, an extra file (marked by the red box) is established. This extra file is named "Uncertainty.dat" file, and it contains the values of parameter u_N, u_C, and u_{P_o}. As we know, there are many cases to be run in the uncertainty analysis. Those cases have different input values for u_N, u_C, and u_{P_o}. Therefore, each case has its own "Uncertainty.dat" file.

Each case in the uncertainty analysis also has their own run file. The run file of a particular case will import the model file and the other data files in addition to the unique "Uncertainty.dat" file associated with that case. The model file and the data files other than "Uncertainty.dat" are non-unique for a specific case, and thus used jointly by all cases. The AMPL implementation of a case in the uncertainty analysis is provided in Appendix C.

The runtime for each case in the uncertainty analysis is restricted to not exceed 1 hour. To obtain the best solution under this limited runtime, the conclusions derived in chapter 5 are adapted, e.g., using CPLEX solver with its default settings and employing 4 cores (8 threads). In addition, according to the runtime analysis discussed in section 5.8, the best solution found after running the optimization for 1 hour is assumed to be the optimal solution of the optimization problem.

There are many cases to be run in the uncertainty analysis. Starting the optimization runs manually is an inefficient repetitive task. For this reason, a master run file is created. With this file, all the cases can be executed with a single click. This master run file works by automatically starting the optimization runs in series. Visualization of the master run file is presented in Figure 6.9, while the AMPL script for this file is provided in Appendix C.5.

As mentioned before, the optimization run of every case utilizes 4 cores. If the computer has 4 other idle cores, one can create two master run files and split the cases to be run. AMPL allows these two master run files to be executed simultaneously. This routine is beneficial because it shortens the total time required to complete the uncertainty analysis.

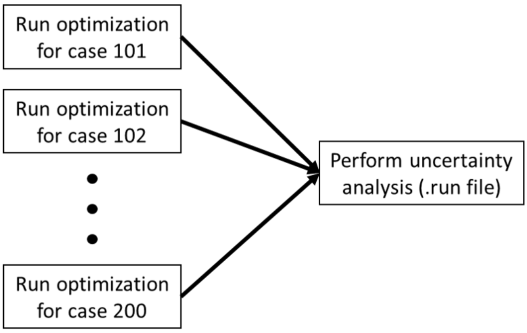

Run optimization for case 101
Run optimization for case 102
Run optimization for case 200
Perform uncertainty analysis (.run file)

Figure 6.9: Visualization of master run file

6.6 Uncertainty Analysis Using LHS Method

Methodology

The uncertainty analysis presented in this section applies LHS method to generate samples. This sampling method is preferred because it can recreate the input probability distribution through fewer iterations when compared to the random sampling method. In other words, for the same number of iterations, the LHS method will provide better accuracy of the uncertainty analysis' results.

In this uncertainty analysis, 100 sets of samples (100 combinations of u_N, u_C, and u_{P_o}) are involved. The procedure of generating these samples is almost similar to what has been presented in section 2.4. However, to simplify the sampling process, the 1st step until the 6th step are replaced with a Matlab function called "lhsdesign". This Matlab function automatically generates 100 combinations of cumulative probabilities for u_N, u_C, and u_{P_o}. The sampled cumulative probabilities are then converted to the values of u_N, u_C, and u_{P_o} using the corresponding cumulative probability curves.

Results

The results of the uncertainty analysis are presented using box plots. Box plots are typical to use for displaying the distribution of data because they give a good indication of how the values in the data are spread out. The elements of a box plot along with their definitions are provided as follows:

- P25: A number where 25% of the data exceeds this number
- P50: A number where 50% of the data exceeds this number
- P75: A number where 75% of the data exceeds this number
- Upper Whisker (UW) & Lower Whisker (LW):

$$UW = \min\left\{ \max, \left(P25 + 1.5 \cdot IQR\right)\right\} \tag{6.14a}$$

$$LW = \max\left\{ \min, \left(P75 - 1.5 \cdot IQR\right)\right\} \tag{6.14b}$$

  where $IQR = P25 - P75$

- Average

Visualization of the box plots' components is provided in Figure 6.10.

Distribution of the optimal NPV is depicted in Figure 6.10. As shown in this figure, the average and the median (P50) of the optimal NPV distribution are around 30000 Mill. NOK. In addition to that, the optimal NPV mostly varies within the positive region. Based on this optimal NPV distribution, it can be inferred that the development of the Safari field is profitable to be executed.

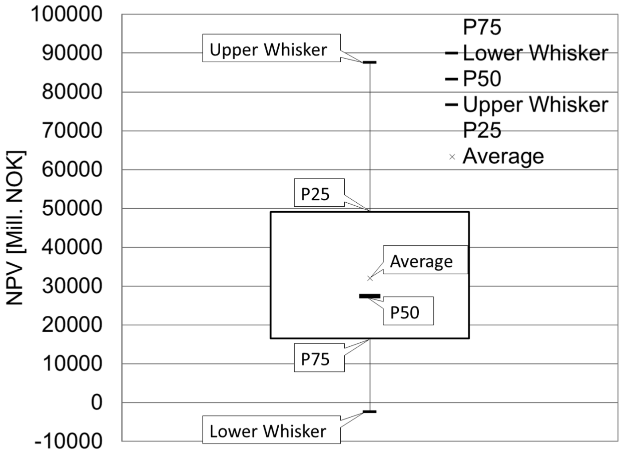
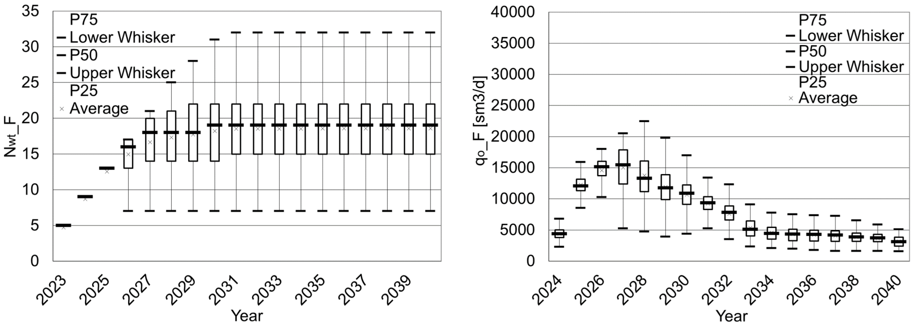


Figure 6.10: Distribution of the optimal NPV (from LHS method)

Distributions of the optimal drilling schedule (N_{wt_F}) and the optimal oil production rate (q_{o_F}) are provided in Figure 6.11. Referring to Figure 6.11a, it is found that the optimal number of wells in 50% of the cases ranges from 15 (P75) to 22 (P25) wells.



(a) The optimal drilling schedule

(b) The optimal oil production rate

Figure 6.11: Distributions of the optimization results (from LHS method)

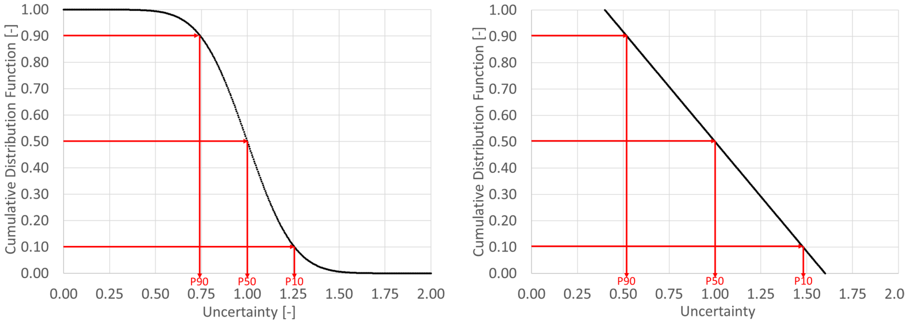
6.7 Uncertainty Analysis Using Probability Tree

Methodology

Another uncertainty analysis is carried out during the study. In this analysis, the uncertainties of the optimization results are quantified using the probability tree. In order to have an objective comparison, the inputs for the uncertainty analysis which uses the probability tree and which applies the LHS method should be equivalent. For this reason, we need to make discrete approximations of the continuous probability distributions that we used for the LHS.

McNamee and Celona (2008) present a widely used technique to discretize continuous probability distributions. The first step of the technique is to select the number of outcomes and the value of each outcome. In this analysis, we have chosen the number of outcomes to be 3. We also have chosen the values of the outcomes, i.e., P90, P50, and P10. Determinations of those values are illustrated in Figure 6.12a (for normal distribution) and Figure 6.12b (for uniform distribution).

The following step is to draw vertical lines at the output values on top of the cumulative probability curve (see the red lines in Figure 6.13 and Figure 6.14). At last, we determine the probability associated to each outcome. This is done by first shifting the green lines in Figure 6.13 and Figure 6.14 such that the regions marked with the same color have the same area. As indicated in the figures, the probability associated to an outcome is finally found.



(a) Normal distribution

(b) Uniform distribution

Figure 6.12: Determination of the output values

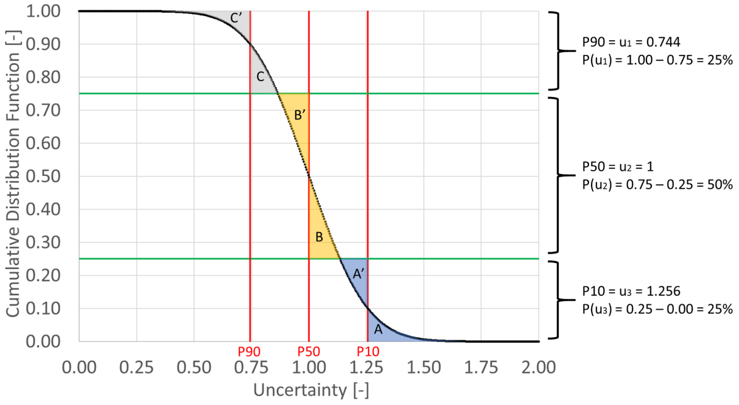


Figure 6.13: Discrete probability distribution which approximates the normal distribution

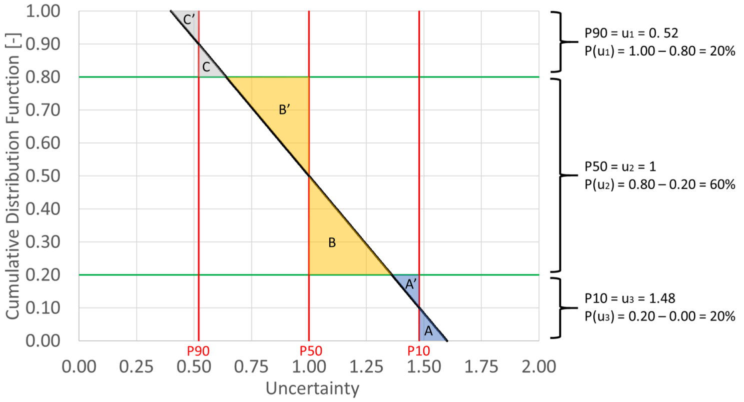


Figure 6.14: Discrete probability distribution which approximates the uniform distribution

Using the discrete probability distributions, the possible outcomes for the uncertain parameters (u_N, u_C, and u_{P_o}) and its corresponding probabilities are defined as in Figure 6.15. Because there are 3 uncertain parameters taken into account in this analysis and each uncertain parameter has 3 possible outcomes, there are 27 ($= 3^3$) cases to be evaluated.

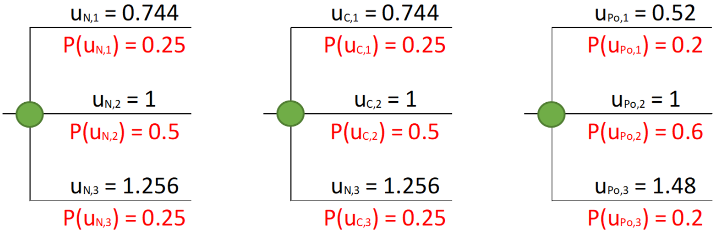


Figure 6.15: The possible outcomes for the uncertain parameters and its corresponding probabilities

Results

Similar to the previous section, the results of the uncertainty analysis are also presented using box plots. In addition to box plots, the expected values (EV) are also computed and presented. These expected values are basically probability-weighted averages. The expected value of an optimization variable y is computed as follows:

$$EV[y] = \sum_{v=1}^{3} \sum_{w=1}^{3} \sum_{x=1}^{3} y_{v,w,x} \cdot JP_{v,w,x} \tag{6.15}$$

where $y_{v,w,x}$ is the value of an optimization variable obtained from a particular case (i.e., $u_N = u_{N,v}$, $u_C = u_{C,w}$, and $u_{P_o} = u_{P_o,x}$), while $JP_{v,w,x}$ is the joint probability corresponds to that particular case. $JP_{v,w,x}$ itself is determined as follows:

$$JP_{v,w,x} = P(u_{N,v}) \cdot P(u_{C,w}) \cdot P(u_{P_o,x}), \quad \forall v, w, \text{and } x \in \{1, 2, 3\} \tag{6.16}$$

Distributions of the optimal NPV, the optimal drilling schedule (N_{wt_F}), and the optimal oil production rate (q_{o_F}) are presented in Figure 6.16, Figure 6.17a, and Figure 6.17b, respectively. If we compare these distributions to the ones we have in section 6.6, it can be observed that both uncertainty analyses produce almost similar results. The only notable difference is that the uncertainty analysis using the LHS method produces slightly wider ranges of upper whisker and lower whisker. It is reasonable because the uncertain parameters (u_N, u_C, and u_{P_o}) can vary further in the LHS method.

As mentioned earlier, there are 27 cases to be run in this probability tree analysis. The total time to complete this analysis is much shorter when compared to the uncertainty analysis using the LHS method (100 cases). Therefore, uncertainty analysis using the probability tree is preferred because it requires a shorter total runtime and produces similar results as the uncertainty analysis using the LHS method.

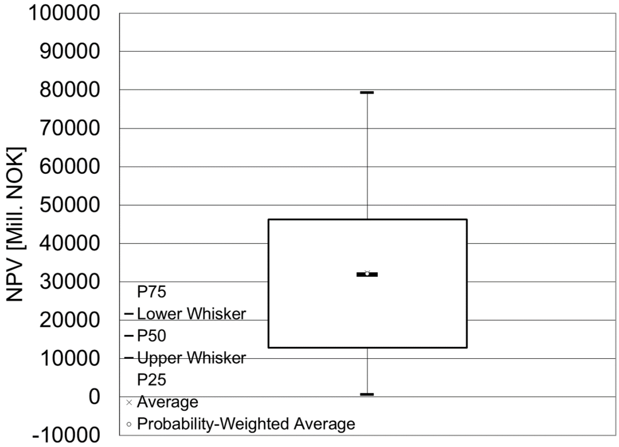
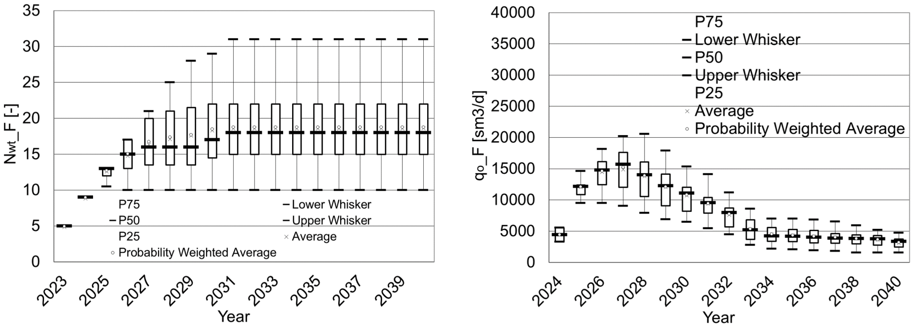


Figure 6.16: Distribution of the optimal NPV (from probability tree)



(a) The optimal drilling schedule

(b) The optimal oil production rate

Figure 6.17: Distributions of the optimization results (from probability tree)

6.8 An Empirical Equation to Estimate the Optimal NPV

In the previous uncertainty analyses, many optimization problems with different input values for u_N, u_C, and u_{P_o} have been solved. In this section, we will further evaluate the optimal values for some variables so that we can generate an empirical equation to estimate the optimal NPV.

At first, we are going to investigate how the total development & operational cost correlates with the ultimate recovery. To find out their relation, we need to determine the present value of all development & operational cost (PV_{ac}) and the ultimate recovery (N_{pu}), which are computed with the following equations:

$$PV_{ac} = PV_{ap} + \sum_{z=1}^{n_p-1} PV_{d,z} + \sum_{z=2}^{n_p} PV_{o,z} \tag{6.17}$$

$$N_{pu} = N_{p\_F,n_p} \tag{6.18}$$

The investigation is carried out using the results of 100 cases that are run in the uncertainty analysis with the LHS method (section 6.6). After determining PV_{ac} and N_{pu} in each case, the relationship between PV_{ac} and N_{pu} is acquired and it is depicted in Figure 6.18. As shown in the figure, the total development & operational cost can be estimated with a linear function of N_{pu}. There is +/-40% uncertainty of this cost estimate. This uncertainty corresponds to the input distribution for parameter u_C.

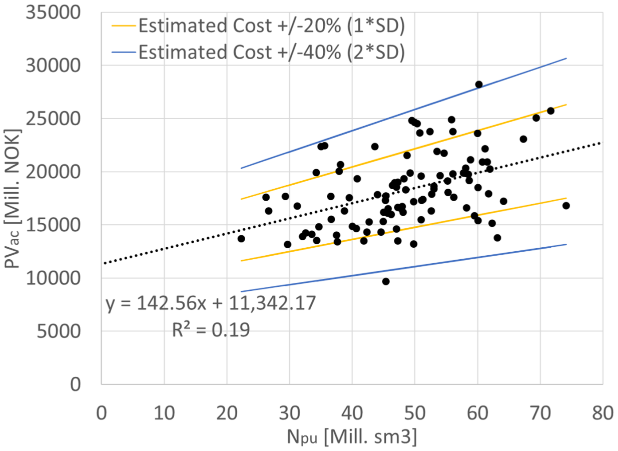


Figure 6.18: Relationship between PV_{ac} and N_{pu}

The optimal NPV can be empirically estimated using the following equation:

$$NPV = P_o \cdot XR \cdot VC \cdot N_{pu} \cdot F_D - PV_{ac} \tag{6.19}$$

where F_D is the discount factor. This quantity correlates with how fast we produce the field. A higher F_D means a higher oil production in the early production period and vice versa. By evaluating F_D in each case $\left(F_D = \frac{NPV+PV_{ac}}{P_o \cdot XR \cdot VC \cdot N_{pu}}\right)$, it is found that F_D is around 32% and it does not vary significantly (see Figure 6.19). Based on these findings, F_D in Equation 6.19 can be considered as a constant.

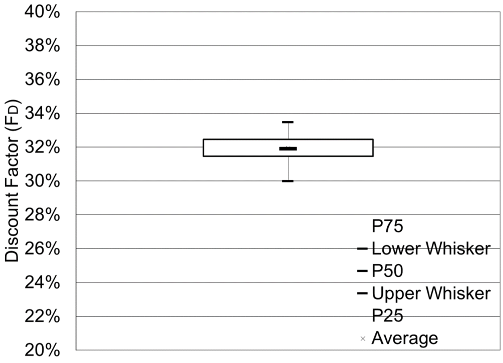


Figure 6.19: Distribution of F_D

6.9 Rule of Thumb for Maximizing the NPV

This section discusses an empirical equation which estimates the optimal oil plateau rate or the optimal oil processing capacity ($\hat{q}_{o\_F}$). As in the previous section, the empirical equation is obtained by evaluating the results of 100 cases that are involved in the uncertainty analysis using the LHS method (section 6.6).

The oil plateau rate or the oil processing capacity ($\hat{q}_{o\_F}$) which maximizes the NPV can be empirically estimated using the following equation:

$$\hat{q}_{o\_F} = \frac{I_R \cdot N_{pu} \cdot 1000000}{t_{up}} \tag{6.20}$$

where I_R is the intake ratio. By evaluating I_R in each case $\left(I_R = \frac{\hat{q}_{o\_F} \cdot t_{up}}{N_{pu} \cdot 1000000}\right)$, it is found that I_R is around 12% and it does not vary significantly (see Figure 6.20). Based on these findings, I_R in Equation 6.20 can be considered as a constant.

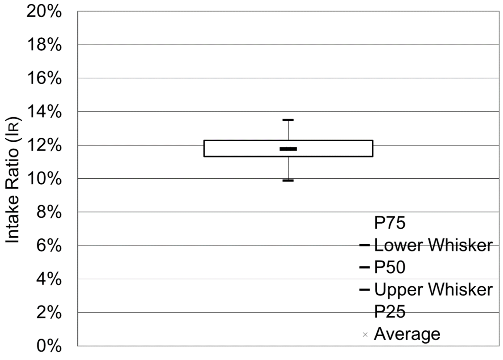


Figure 6.20: Distribution of I_R

Chapter 7

Conclusion

1. An automated field development methodology based on mathematical optimization has been developed in this study. One of the optimization problems is designed to maximize the plateau duration by seeking the best configuration of production & injection profile, drilling schedule, and recovery mechanism. Some constraints concerning production, injection, and drilling are also considered in the optimization. Formulation of the optimization problem adapts 2 main ideas, i.e. (1) to use production potential curves as the proxy model of the production system and (2) to use multi-dimensional PWL approximations for representing some non-linear functions. According to the optimization results, the plateau period can be prolonged further if the drilling schedule and recovery mechanism are also defined as decision variables in addition to the fluid rates.

2. Another optimization problem is also established in this study. Basically, this optimization problem is an extension of the previous optimization problem. However, the optimization problem maximizes the Net Present Value (NPV) instead of the plateau duration. A cost proxy model that is developed in the specialization project is adapted for making an NPV calculation. The optimization results reveal that including the drilling schedule and recovery mechanism as decision variables gives 84.44% higher NPV than just optimizing the fluid rates.

3. Several evaluations have been performed to make the formulations of the optimization problems and their implementations in AMPL more efficient and more accurate. By adapting the evaluations' results, the optimization process becomes much faster. A few recommendations from those evaluations are:

   - to reduce the number of breakpoints in the PWL approximation and to wisely select those breakpoints

   - to use G_p-N_p curve instead of 2D interpolation for determining the gas production rate

- to apply backward rectangular integration technique in estimating the cumulative production/injection
- to employ 4 CPU cores in parallel when solving the optimization problems
- to utilize CPLEX solver with its default settings when solving the NPV maximization problem

4. Uncertainties of the optimization results have been quantified through uncertainty analyses. Three uncertain parameters are considered, i.e., in-place, development & operational cost, and oil price. To conduct an uncertainty analysis, two approaches have been studied, i.e., using Latin Hypercube Sampling (LHS) method and using a probability tree. Uncertainty analysis using a probability tree is preferred because it is quicker and produces similar results as the other approach.

Bibliography

AMPL, 2019a. AMPL.
URL `https://ampl.com/products/ampl/`

AMPL, 2019b. CPLEX Options for AMPL.
URL `https://ampl.com/products/solvers/solvers-we-sell/cplex/options/`

Angga, I. G. A. G., 2018. Report for Specialization Project - Automated Decision Support Methodologies for Field Development: The Safari Field Case. NTNU.

Camponogara, E., October 2018. Lecture Notes for Mixed Integer Optimization in Energy and Oil and Gas System Course (TTK16). NTNU.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C., 2009. Introduction to Algorithms, 3rd Edition. The MIT Press.

Cornuejols, G., 2007. Valid Inequalities for Mixed Integer Linear Programs. Mathematical Programming.

Dantzig, G. B., 1951. Maximization of a linear functions in variables subject to linear inequalities. In: Koopmans, T. (Ed.), Activity Analysis of Production and Allocation. Wiley & Chapman-Hall, New York - London, pp. 339–347.

Gomory, R. E., 1963. An algorithm for integer solutions to linear programs. In: Graves, R., Wolfe, P. (Eds.), Recent Advances in Mathematical Programming. McGraw-Hill, New York, pp. 269–302.

Gurobi, 2019. AMPL - Gurobi Parameter Reference.
URL `http://www.gurobi.com/documentation/8.1/ampl-gurobi/parameters.html`

Haldorsen, H. H., 1996. Choosing between rocks, hard places and a lot more: The economic interface. In: Dore, A., Sinding-Larsen, R. (Eds.), Norwegian Petroleum Society Special Publications Volume 6: Quantification and Prediction of Hydrocarbon Resources. Elsevier, pp. 291–312.

Hall, N. A., Delille, S., 2011. Cost Estimation Challenges and Uncertainties Confronting Oil and Gas Companies. AACE International Transactions.

Hoffmann, A., June 2014. Application of Piecewise Linear Models to Short-Term Oil Production Optimization. NTNU.

IBM, 2019a. MIP Emphasis Switch.
URL https://www.ibm.com/support/knowledgecenter/SSSA5P_12.5.1/ilog.odms.cplex.help/CPLEX/Parameters/topics/MIPEmphasis.html

IBM, 2019b. Parallel Mode Switch.
URL https://www.ibm.com/support/knowledgecenter/SSSA5P_12.6.2/ilog.odms.cplex.help/CPLEX/Parameters/topics/ParallelMode.html

Karmarkar, N., 1984. A New Polynomial-Time Algorithm for Linear Programming. Combinatorica.

Land, A. H., Doig, A. G., 1960. An Automatic Method of Solving Discrete Programming Problems. Econometrica.

McNamee, P., Celona, J., October 2008. Decision Analysis for the Professional, 4th Edition. SmartOrg Inc.

Norsk Hydro Produksjon A/S, September 1995. Visund - Plan for Development and Operation.

NPD, 2019. FactMaps.
URL http://gis.npd.no/factmaps/html_21/

NTNU, 2019. MILP Algorithms: Branch-and-Bound and Branch-and-Cut.
URL https://www.itk.ntnu.no/_media/fag/fordypning/tk16/milpalgorithms.pdf

Petroleum Experts, 2019. IPM Suite.
URL http://www.petex.com/products/ipm-suite/

Stanko, M., 2019. Compendium for Field Development and Operations Course (TPG4230) - Petroleum Production System. NTNU.

SUBPRO, 2018. Annual Report 2017-18.

Trevisan, L., March 2011. Lecture Notes for Optimization and Algorithmic Paradigms Course (CS261) - Combinatorial Optimization: Exact and Approximate Algorithms. Stanford University.

UiO, 2019. Appendix A: Sampling Methods.
URL https://www.uio.no/studier/emner/matnat/math/STK4400/v05/undervisningsmateriale/Sampling%20methods.pdf

Vanderbei, R. J., 2008. Linear Programming - Foundations and Extensions, 3rd Edition. Springer.

Appendix

Appendix A

AMPL Code for Maximizing the Plateau Duration

A.1 CASE-1

A.1.1 Model File (CASE-1_Model.mod)

```
# Part 0: Index Description
# i = reservoirs
# j = breakpoints for Np
# k = breakpoints for Nop
# l = breakpoints for RM
# m = breakpoints for zNgi
# n = breakpoints for zNwi
# z = points in time during the production period
#--------------------------------------------------------------------------------

# Part 1: Declaration (var, set, param, etc)
param nr;
set R := 1..nr;
param np;
set P := 1..np;
param tup;
param qo_F_plateau;
param max_Nop_pd;
param max_Ni_pd;
param max_Nwt_py;
param max_Ngi {i in R};
param max_Nwi {i in R};
param zNgi_bar {m in {1,2}, n in {1,2}};
param zNwi_bar {m in {1,2}, n in {1,2}};
param RMds_bar {m in {1,2}, n in {1,2}};
param max_qgi_pgi;
param max_qwi_pwi;
param nbp_Np;
param nbp_Nop {i in R};
param nbp_RM {i in R};
```

```
set V_Np := 1..nbp_Np;
set V_Nop {i in R} := 1..nbp_Nop[i];
set V_RM {i in R} := 1..nbp_RM[i];
param Np_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param Nop_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param RMpp_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param qopp_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param Gp_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param Wp_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param Gi_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param Wi_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};

var RM {i in R} integer >= 0;
var zNgi {i in R} binary >= 0;
var zNwi {i in R} binary >= 0;
var Nop {i in R, z in P} integer >= 0;
var Ngi {i in R, z in P} integer >= 0;
var Nwi {i in R, z in P} integer >= 0;
var Nwt {i in R, z in P} integer >= 0;
var qopp {i in R, z in P} >= 0;
var qo {i in R, z in {2..np}} >= 0;
var qg {i in R, z in {2..np}} >= 0;
var qw {i in R, z in {2..np}} >= 0;
var qgi {i in R, z in {2..np}} >= 0;
var qwi {i in R, z in {2..np}} >= 0;
var Np {i in R, z in P} >= 0;
var Gp {i in R, z in P} >= 0;
var Wp {i in R, z in P} >= 0;
var Gi {i in R, z in P} >= 0;
var Wi {i in R, z in P} >= 0;
var Nop_F {z in P} integer >= 0;
var Ngi_F {z in P} integer >= 0;
var Nwi_F {z in P} integer >= 0;
var Nwt_F {z in P} integer >= 0;
var qopp_F {z in P} >= 0;
var qo_F {z in {2..np}} >= 0;
var qg_F {z in {2..np}} >= 0;
var qw_F {z in {2..np}} >= 0;
var qgi_F {z in {2..np}} >= 0;
var qwi_F {z in {2..np}} >= 0;
var Np_F {z in P} >= 0;
var Gp_F {z in P} >= 0;
var Wp_F {z in P} >= 0;
var Gi_F {z in P} >= 0;
var Wi_F {z in P} >= 0;
var lambda {i in R, z in P, j in V_Np, k in V_Nop[i], l in V_RM[i]} >= 0;
var eta_Np {i in R, z in P, j in V_Np} >= 0;
var eta_Nop {i in R, z in P, k in V_Nop[i]} >= 0;
var eta_RM {i in R, z in P, l in V_RM[i]} >= 0;
var omega {i in R, m in {1,2}, n in {1,2}} >= 0;
var tau_zNgi {i in R, m in {1,2}} >= 0;
var tau_zNwi {i in R, n in {1,2}} >= 0;
#--------------------------------------------------------------------------------

# Part 2: Objective Function
minimize 3_1: sum{z in {2..np}} (qo_F_plateau - qo_F[z]);
#--------------------------------------------------------------------------------

# Part 3: Constraints
subject to 3_2 {z in {2..np}}: qo_F[z] <= qo_F_plateau;
subject to 3_3a {i in R, z in {2..np}}: qo[i,z] <= qopp[i,z];
subject to 3_3b {i in R, z in {2..np}}: qo[i,z] <= qopp[i,z-1];
subject to 3_4a {i in R, z in {2..np}}: qgi[i,z] <= Ngi[i,z-1]*max_qgi_pgi;
subject to 3_4b {i in R, z in {2..np}}: qwi[i,z] <= Nwi[i,z-1]*max_qwi_pwi;
subject to 3_5a {i in R, z in {2..np}}: Nop[i,z] - Nop[i,z-1] >= 0;
subject to 3_5b {i in R, z in {2..np}}: Ngi[i,z] - Ngi[i,z-1] >= 0;
subject to 3_5c {i in R, z in {2..np}}: Nwi[i,z] - Nwi[i,z-1] >= 0;
subject to 3_6a {i in R, z in P}: Ngi[i,z] <= zNgi[i]*max_Ngi[i];
subject to 3_6b {i in R, z in P}: Nwi[i,z] <= zNwi[i]*max_Nwi[i];
subject to 3_7a: Nop_F[1] <= max_Nop_pd;
subject to 3_7b: Ngi_F[1] + Nwi_F[1] <= max_Ni_pd;
subject to 3_8 {z in {2..np}}: Nwt_F[z] - Nwt_F[z-1] <= max_Nwt_py;
subject to 3_9 {i in R}: Np[i,1] = 0;
subject to 3_10a {i in R, z in {2..np}}: Np[i,z] = Np[i,z-1] +
     tup*qo[i,z]/1000000;
subject to 3_10b {i in R, z in {2..np}}: Gp[i,z] = Gp[i,z-1] + tup*qg[i,z]/1000;
subject to 3_10c {i in R, z in {2..np}}: Wp[i,z] = Wp[i,z-1] +
     tup*qw[i,z]/1000000;
subject to 3_10d {i in R, z in {2..np}}: Gi[i,z] = Gi[i,z-1] + tup*qgi[i,z]/1000;
subject to 3_10e {i in R, z in {2..np}}: Wi[i,z] = Wi[i,z-1] +
     tup*qwi[i,z]/1000000;
subject to 3_11 {i in R, z in P}: Nwt[i,z] = Nop[i,z] + Ngi[i,z] + Nwi[i,z];
subject to 3_12a {z in P}: Nop_F[z] = sum{i in R} Nop[i,z];
subject to 3_12b {z in P}: Ngi_F[z] = sum{i in R} Ngi[i,z];
subject to 3_12c {z in P}: Nwi_F[z] = sum{i in R} Nwi[i,z];
subject to 3_12d {z in P}: Nwt_F[z] = sum{i in R} Nwt[i,z];
subject to 3_12e {z in P}: qopp_F[z] = sum{i in R} qopp[i,z];
subject to 3_12f {z in {2..np}}: qo_F[z] = sum{i in R} qo[i,z];
subject to 3_12g {z in {2..np}}: qg_F[z] = sum{i in R} qg[i,z];
subject to 3_12h {z in {2..np}}: qw_F[z] = sum{i in R} qw[i,z];
subject to 3_12i {z in {2..np}}: qgi_F[z] = sum{i in R} qgi[i,z];
subject to 3_12j {z in {2..np}}: qwi_F[z] = sum{i in R} qwi[i,z];
subject to 3_12k {z in P}: Np_F[z] = sum{i in R} Np[i,z];
subject to 3_12l {z in P}: Gp_F[z] = sum{i in R} Gp[i,z];
subject to 3_12m {z in P}: Wp_F[z] = sum{i in R} Wp[i,z];
subject to 3_12n {z in P}: Gi_F[z] = sum{i in R} Gi[i,z];
subject to 3_12o {z in P}: Wi_F[z] = sum{i in R} Wi[i,z];
subject to 3_15a {i in R, z in P}: Np[i,z] = sum{j in V_Np, k in V_Nop[i], l in
     V_RM[i]} lambda[i,z,j,k,l]*Np_bar[i,j,k,l];
subject to 3_15b {i in R, z in P}: Nop[i,z] = sum{j in V_Np, k in V_Nop[i], l in
     V_RM[i]} lambda[i,z,j,k,l]*Nop_bar[i,j,k,l];
subject to 3_15c {i in R, z in P}: RM[i] = sum{j in V_Np, k in V_Nop[i], l in
     V_RM[i]} lambda[i,z,j,k,l]*RMpp_bar[i,j,k,l];
subject to 3_15d {i in R, z in P}: qopp[i,z] = sum{j in V_Np, k in V_Nop[i], l
     in V_RM[i]} lambda[i,z,j,k,l]*qopp_bar[i,j,k,l];
subject to 3_15e {i in R, z in P}: 1 = sum{j in V_Np, k in V_Nop[i], l in
     V_RM[i]} lambda[i,z,j,k,l];
subject to 3_15f {i in R, z in P, j in V_Np}: eta_Np[i,z,j] = sum{k in V_Nop[i],
     l in V_RM[i]} lambda[i,z,j,k,l];
subject to 3_15g {i in R, z in P, k in V_Nop[i]}: eta_Nop[i,z,k] = sum{j in
     V_Np, l in V_RM[i]} lambda[i,z,j,k,l];
subject to 3_15h {i in R, z in P, l in V_RM[i]}: eta_RM[i,z,l] = sum{j in V_Np,
     k in V_Nop[i]} lambda[i,z,j,k,l];
subject to 3_16a {i in R, z in P}: Gp[i,z] = sum{j in V_Np, k in V_Nop[i], l in
     V_RM[i]} lambda[i,z,j,k,l]*Gp_bar[i,j,k,l];
subject to 3_16b {i in R, z in P}: Wp[i,z] = sum{j in V_Np, k in V_Nop[i], l in
     V_RM[i]} lambda[i,z,j,k,l]*Wp_bar[i,j,k,l];
subject to 3_16c {i in R, z in P}: Gi[i,z] = sum{j in V_Np, k in V_Nop[i], l in
     V_RM[i]} lambda[i,z,j,k,l]*Gi_bar[i,j,k,l];
subject to 3_16d {i in R, z in P}: Wi[i,z] = sum{j in V_Np, k in V_Nop[i], l in
     V_RM[i]} lambda[i,z,j,k,l]*Wi_bar[i,j,k,l];
subject to 3_17a {i in R}: zNgi[i] = sum{m in {1,2}, n in {1,2}}
     omega[i,m,n]*zNgi_bar[m,n];
subject to 3_17b {i in R}: zNwi[i] = sum{m in {1,2}, n in {1,2}}
     omega[i,m,n]*zNwi_bar[m,n];
subject to 3_17c {i in R}: RM[i] = sum{m in {1,2}, n in {1,2}}
     omega[i,m,n]*RMds_bar[m,n];
subject to 3_17d {i in R}: 1 = sum{m in {1,2}, n in {1,2}} omega[i,m,n];
subject to 3_17e {i in R, m in {1,2}}: tau_zNgi[i,m] = sum{n in {1,2}}
     omega[i,m,n];
subject to 3_17f {i in R, n in {1,2}}: tau_zNwi[i,n] = sum{m in {1,2}}
     omega[i,m,n];
```

A.1.2 Data File (CASE-1_Data.dat)

```
param nr := 3;
param np := 18;
param tup := 365;
param qo_F_plateau := 20000;
param max_Nop_pd := 4;
param max_Ni_pd := 1;
param max_Nwt_py := 4;
param: max_Ngi max_Nwi :=
#i     max_Ngi max_Nwi
1      1       3
2      0       2
3      0       1
;
param: zNgi_bar zNwi_bar RMds_bar :=
#m n   zNgi_bar zNwi_bar RMds_bar
1  1   0        0        1
1  2   0        1        2
2  1   1        0        3
2  2   1        1        4
;
param max_qgi_pgi := 4000;
param max_qwi_pwi := 6360;
```

A.1.3 Run File (CASE-1_Run.run)

```
model CASE-1_Model.mod;
data CASE-1_Data.dat;
data PWL_Table.dat;

option solver gurobi;
option gurobi_options "threads=8 timelim=43200 outlev=1 mipgap=0.0001
     nodefilestart=0.5 bestbound=1";

suffix sosno;
suffix ref;
param itersos2;
let itersos2 := 0;
for{i in R, z in P} {
  let itersos2 := itersos2-1;
  let {j in V_Np} eta_Np[i,z,j].sosno := itersos2;
  let {j in V_Np} eta_Np[i,z,j].ref := Np_bar[i,j,1,1];

  let itersos2 := itersos2-1;
  let {k in V_Nop[i]} eta_Nop[i,z,k].sosno := itersos2;
  let {k in V_Nop[i]} eta_Nop[i,z,k].ref := Nop_bar[i,1,k,1];
}
param itersos1;
let itersos1 := 0;
for{i in R, z in P} {
  let itersos1 := itersos1+1;
  let {l in V_RM[i]} eta_RM[i,z,l].sosno := itersos1;
  let {l in V_RM[i]} eta_RM[i,z,l].ref := RMpp_bar[i,1,1,l];
}
for{i in R} {
  let itersos1 := itersos1+1;
  let {m in {1,2}} tau_zNgi[i,m].sosno := itersos1;
  let {m in {1,2}} tau_zNgi[i,m].ref := zNgi_bar[m,1];

  let itersos1 := itersos1+1;
  let {n in {1,2}} tau_zNwi[i,n].sosno := itersos1;
  let {n in {1,2}} tau_zNwi[i,n].ref := zNwi_bar[1,n];
}

solve;

display qo_F_plateau >CASE-1_Results.out;
display RM >CASE-1_Results.out;
display Nop >CASE-1_Results.out;
.
.
.
close CASE-1_Results.out;
```

A.2 REF-CASE-1

A.2.1 Model File (REF-CASE-1_Model.mod)

```
# Part 0: Index Description
# i = reservoirs
# j = breakpoints for Np
# k = breakpoints for Nop
# l = breakpoints for RM
# z = points in time during the production period
#-------------------------------------------------------------------------------------

# Part 1: Declaration (var, set, param, etc)
param nr;
set R := 1..nr;
param np;
set P := 1..np;
param tup;
param qo_F_plateau;
param Nop {i in R, z in P};
param Nwi {i in R, z in P};
param Ngi {i in R, z in P};
param RM {i in R, z in P};
param max_qgi_pgi;
param max_qwi_pwi;
param nbp_Np;
param nbp_Nop {i in R};
param nbp_RM {i in R};
set V_Np := 1..nbp_Np;
set V_Nop {i in R} := 1..nbp_Nop[i];
set V_RM {i in R} := 1..nbp_RM[i];
param Np_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param Nop_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param RMpp_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param qopp_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param Gp_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param Wp_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param Gi_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param Wi_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};

var Nwt {i in R, z in P} integer >= 0;
var qopp {i in R, z in P} >= 0;
var qo {i in R, z in {2..np}} >= 0;
var qg {i in R, z in {2..np}} >= 0;
var qw {i in R, z in {2..np}} >= 0;
var qgi {i in R, z in {2..np}} >= 0;
var qwi {i in R, z in {2..np}} >= 0;
var Np {i in R, z in P} >= 0;
var Gp {i in R, z in P} >= 0;
var Wp {i in R, z in P} >= 0;
var Gi {i in R, z in P} >= 0;
var Wi {i in R, z in P} >= 0;
var Nop_F {z in P} integer >= 0;
var Ngi_F {z in P} integer >= 0;
var Nwi_F {z in P} integer >= 0;
var Nwt_F {z in P} integer >= 0;
var qopp_F {z in P} >= 0;
var qo_F {z in {2..np}} >= 0;
var qg_F {z in {2..np}} >= 0;
var qw_F {z in {2..np}} >= 0;
var qgi_F {z in {2..np}} >= 0;
var qwi_F {z in {2..np}} >= 0;
var Np_F {z in P} >= 0;
var Gp_F {z in P} >= 0;
var Wp_F {z in P} >= 0;
var Gi_F {z in P} >= 0;
var Wi_F {z in P} >= 0;
var lambda {i in R, z in P, j in V_Np, k in V_Nop[i]} >= 0;
var eta_Np {i in R, z in P, j in V_Np} >= 0;
var eta_Nop {i in R, z in P, k in V_Nop[i]} >= 0;
#-------------------------------------------------------------------------------

# Part 2: Objective Function
minimize 3_1: sum{z in {2..np}} (qo_F_plateau - qo_F[z]);
#-------------------------------------------------------------------------------

# Part 3: Constraints
subject to 3_2 {z in {2..np}}: qo_F[z] <= qo_F_plateau;
subject to 3_3a {i in R, z in {2..np}}: qo[i,z] <= qopp[i,z];
subject to 3_3b {i in R, z in {2..np}}: qo[i,z] <= qopp[i,z-1];
subject to 3_4a {i in R, z in {2..np}}: qgi[i,z] <= Ngi[i,z-1]*max_qgi_pgi;
subject to 3_4b {i in R, z in {2..np}}: qwi[i,z] <= Nwi[i,z-1]*max_qwi_pwi;
subject to 3_9 {i in R}: Np[i,1] = 0;
subject to 3_10a {i in R, z in {2..np}}: Np[i,z] = Np[i,z-1] +
     tup*qo[i,z]/1000000;
subject to 3_10b {i in R, z in {2..np}}: Gp[i,z] = Gp[i,z-1] + tup*qg[i,z]/1000;
subject to 3_10c {i in R, z in {2..np}}: Wp[i,z] = Wp[i,z-1] +
     tup*qw[i,z]/1000000;
subject to 3_10d {i in R, z in {2..np}}: Gi[i,z] = Gi[i,z-1] + tup*qgi[i,z]/1000;
subject to 3_10e {i in R, z in {2..np}}: Wi[i,z] = Wi[i,z-1] +
     tup*qwi[i,z]/1000000;
subject to 3_11 {i in R, z in P}: Nwt[i,z] = Nop[i,z] + Ngi[i,z] + Nwi[i,z];
subject to 3_12a {z in P}: Nop_F[z] = sum{i in R} Nop[i,z];
subject to 3_12b {z in P}: Ngi_F[z] = sum{i in R} Ngi[i,z];
subject to 3_12c {z in P}: Nwi_F[z] = sum{i in R} Nwi[i,z];
subject to 3_12d {z in P}: Nwt_F[z] = sum{i in R} Nwt[i,z];
subject to 3_12e {z in P}: qopp_F[z] = sum{i in R} qopp[i,z];
subject to 3_12f {z in {2..np}}: qo_F[z] = sum{i in R} qo[i,z];
subject to 3_12g {z in {2..np}}: qg_F[z] = sum{i in R} qg[i,z];
subject to 3_12h {z in {2..np}}: qw_F[z] = sum{i in R} qw[i,z];
subject to 3_12i {z in {2..np}}: qgi_F[z] = sum{i in R} qgi[i,z];
subject to 3_12j {z in {2..np}}: qwi_F[z] = sum{i in R} qwi[i,z];
subject to 3_12k {z in P}: Np_F[z] = sum{i in R} Np[i,z];
subject to 3_12l {z in P}: Gp_F[z] = sum{i in R} Gp[i,z];
subject to 3_12m {z in P}: Wp_F[z] = sum{i in R} Wp[i,z];
subject to 3_12n {z in P}: Gi_F[z] = sum{i in R} Gi[i,z];
subject to 3_12o {z in P}: Wi_F[z] = sum{i in R} Wi[i,z];
subject to 3_15a {i in R, z in P}: Np[i,z] = sum{j in V_Np, k in V_Nop[i]}
     lambda[i,z,j,k]*Np_bar[i,j,k,RM[i,z]];
subject to 3_15b {i in R, z in P}: Nop[i,z] = sum{j in V_Np, k in V_Nop[i]}
     lambda[i,z,j,k]*Nop_bar[i,j,k,RM[i,z]];
subject to 3_15d {i in R, z in P}: qopp[i,z] = sum{j in V_Np, k in V_Nop[i]}
     lambda[i,z,j,k]*qopp_bar[i,j,k,RM[i,z]];
subject to 3_15e {i in R, z in P}: 1 = sum{j in V_Np, k in V_Nop[i]}
     lambda[i,z,j,k];
```

```
subject to 3_15f {i in R, z in P, j in V_Np}: eta_Np[i,z,j] = sum{k in V_Nop[i]}
     lambda[i,z,j,k];
subject to 3_15g {i in R, z in P, k in V_Nop[i]}: eta_Nop[i,z,k] = sum{j in
     V_Np} lambda[i,z,j,k];
subject to 3_16a {i in R, z in P}: Gp[i,z] = sum{j in V_Np, k in V_Nop[i]}
     lambda[i,z,j,k]*Gp_bar[i,j,k,RM[i,z]];
subject to 3_16b {i in R, z in P}: Wp[i,z] = sum{j in V_Np, k in V_Nop[i]}
     lambda[i,z,j,k]*Wp_bar[i,j,k,RM[i,z]];
subject to 3_16c {i in R, z in P}: Gi[i,z] = sum{j in V_Np, k in V_Nop[i]}
     lambda[i,z,j,k]*Gi_bar[i,j,k,RM[i,z]];
subject to 3_16d {i in R, z in P}: Wi[i,z] = sum{j in V_Np, k in V_Nop[i]}
     lambda[i,z,j,k]*Wi_bar[i,j,k,RM[i,z]];
```

A.2.2 Data File (REF-CASE-1_Data.dat)

```
param nr := 3;
param np := 18;
param tup := 365;
param qo_F_plateau := 20000;
param: Nop Ngi Nwi RM :=
#i z   Nop Ngi Nwi RM
1  1   4   1    0   3
1  2   7   1    1   4
1  3   10  1    2   4
.
.
.
;
param max_qgi_pgi := 4000;
param max_qwi_pwi := 6360;
```

A.2.3 Run File (REF-CASE-1_Run.run)

```
model REF-CASE-1_Model.mod;
data REF-CASE-1_Data.dat;
data PWL_Table.dat;

option solver gurobi;
option gurobi_options "threads=8 timelim=43200 outlev=1 mipgap=0.0001
     nodefilestart=0.5 bestbound=1";

suffix sosno;
suffix ref;
param itersos2;
let itersos2 := 0;
for{i in R, z in P} {
  let itersos2 := itersos2-1;
  let {j in V_Np} eta_Np[i,z,j].sosno := itersos2;
  let {j in V_Np} eta_Np[i,z,j].ref := Np_bar[i,j,1,1];

  let itersos2 := itersos2-1;
  let {k in V_Nop[i]} eta_Nop[i,z,k].sosno := itersos2;
  let {k in V_Nop[i]} eta_Nop[i,z,k].ref := Nop_bar[i,1,k,1];
}

solve;

display qo_F_plateau >REF-CASE-1_Results.out;
display RM >REF-CASE-1_Results.out;
display Nop >REF-CASE-1_Results.out;
.
.
.
close REF-CASE-1_Results.out;
```

Appendix **B**

AMPL Code for Maximizing the NPV

B.1 CASE-2

B.1.1 Model File (CASE-2_Model.mod)

```
# Part 0: Index Description
# i = reservoirs
# j = breakpoints for Np
# k = breakpoints for Nop
# l = breakpoints for RM
# m = breakpoints for zNgi
# n = breakpoints for zNwi
# p = points in time prior to the production period
# z = points in time during the production period
#--------------------------------------------------------------------------------

# Part 1: Declaration (var, set, param, etc)
param nr;
set R := 1..nr;
param np;
set P := 1..np;
param tup;
param max_Nop_pd;
param max_Ni_pd;
param max_Nwt_py;
param max_Ngi {i in R};
param max_Nwi {i in R};
param zNgi_bar {m in {1,2}, n in {1,2}};
param zNwi_bar {m in {1,2}, n in {1,2}};
param RMds_bar {m in {1,2}, n in {1,2}};
param max_qgi_pgi;
param max_qwi_pwi;
param nbp_Np;
param nbp_Nop {i in R};
param nbp_RM {i in R};
```

```
set V_Np := 1..nbp_Np;
set V_Nop {i in R} := 1..nbp_Nop[i];
set V_RM {i in R} := 1..nbp_RM[i];
param Np_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param Nop_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param RMpp_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param qopp_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param Gp_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param Wp_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param Gi_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param Wi_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param Po;
param XR;
param VC;
param d;
param A {p in {1,2,3,4}};
param B;
param C {p in {1,2,3,4}};
param D {q in {1,2}};
param E {r in {1,2}};
param F {s in {1,2,3}};
param G {t in {1,2,3,4,5}};

var RM {i in R} integer >= 0;
var zNgi {i in R} binary >= 0;
var zNwi {i in R} binary >= 0;
var Nop {i in R, z in P} integer >= 0;
var Ngi {i in R, z in P} integer >= 0;
var Nwi {i in R, z in P} integer >= 0;
var Nwt {i in R, z in P} integer >= 0;
var qopp {i in R, z in P} >= 0;
var qo {i in R, z in {2..np}} >= 0;
var qg {i in R, z in {2..np}} >= 0;
var qw {i in R, z in {2..np}} >= 0;
var qgi {i in R, z in {2..np}} >= 0;
var qwi {i in R, z in {2..np}} >= 0;
var Np {i in R, z in P} >= 0;
var Gp {i in R, z in P} >= 0;
var Wp {i in R, z in P} >= 0;
var Gi {i in R, z in P} >= 0;
var Wi {i in R, z in P} >= 0;
var Nop_F {z in P} integer >= 0;
var Ngi_F {z in P} integer >= 0;
var Nwi_F {z in P} integer >= 0;
var Nwt_F {z in P} integer >= 0;
var qopp_F {z in P} >= 0;
var qo_F {z in {2..np}} >= 0;
var qg_F {z in {2..np}} >= 0;
var qw_F {z in {2..np}} >= 0;
var qgi_F {z in {2..np}} >= 0;
var qwi_F {z in {2..np}} >= 0;
var Np_F {z in P} >= 0;
var Gp_F {z in P} >= 0;
var Wp_F {z in P} >= 0;
var Gi_F {z in P} >= 0;
var Wi_F {z in P} >= 0;
var lambda {i in R, z in P, j in V_Np, k in V_Nop[i], l in V_RM[i]} >= 0;
var eta_Np {i in R, z in P, j in V_Np} >= 0;
var eta_Nop {i in R, z in P, k in V_Nop[i]} >= 0;
var eta_RM {i in R, z in P, l in V_RM[i]} >= 0;
var omega {i in R, m in {1,2}, n in {1,2}} >= 0;
var tau_zNgi {i in R, m in {1,2}} >= 0;
var tau_zNwi {i in R, n in {1,2}} >= 0;
var PVe >= 0;
var PVdp >= 0;
var PVa >= 0;
var Nt {i in R} integer >= 0;
var Nt_F integer >= 0;
var Nx_F integer >= 0;
var PVsc >= 0;
var max_qo_F >= 0;
var max_qg_F >= 0;
var max_qw_F >= 0;
var PVtc >= 0;
var PVap >= 0;
var PVd {z in {1..(np-1)}} >= 0;
var PVo {z in {2..np}} >= 0;
var PVr {z in {2..np}} >= 0;
var DCF {z in P};
var NPV;
#---------------------------------------------------------------------------------

# Part 2: Objective Function
maximize 4_5: NPV;
#---------------------------------------------------------------------------------

# Part 3: Constraints
subject to 3_3a {i in R, z in {2..np}}: qo[i,z] <= qopp[i,z];
subject to 3_3b {i in R, z in {2..np}}: qo[i,z] <= qopp[i,z-1];
subject to 3_4a {i in R, z in {2..np}}: qgi[i,z] <= Ngi[i,z-1]*max_qgi_pgi;
subject to 3_4b {i in R, z in {2..np}}: qwi[i,z] <= Nwi[i,z-1]*max_qwi_pwi;
subject to 3_5a {i in R, z in {2..np}}: Nop[i,z] - Nop[i,z-1] >= 0;
subject to 3_5b {i in R, z in {2..np}}: Ngi[i,z] - Ngi[i,z-1] >= 0;
subject to 3_5c {i in R, z in {2..np}}: Nwi[i,z] - Nwi[i,z-1] >= 0;
subject to 3_6a {i in R, z in P}: Ngi[i,z] <= zNgi[i]*max_Ngi[i];
subject to 3_6b {i in R, z in P}: Nwi[i,z] <= zNwi[i]*max_Nwi[i];
subject to 3_7a: Nop_F[1] <= max_Nop_pd;
subject to 3_7b: Ngi_F[1] + Nwi_F[1] <= max_Ni_pd;
subject to 3_8 {z in {2..np}}: Nwt_F[z] - Nwt_F[z-1] <= max_Nwt_py;
subject to 3_9 {i in R}: Np[i,1] = 0;
subject to 3_10a {i in R, z in {2..np}}: Np[i,z] = Np[i,z-1] +
     tup*qo[i,z]/1000000;
subject to 3_10b {i in R, z in {2..np}}: Gp[i,z] = Gp[i,z-1] + tup*qg[i,z]/1000;
subject to 3_10c {i in R, z in {2..np}}: Wp[i,z] = Wp[i,z-1] +
     tup*qw[i,z]/1000000;
subject to 3_10d {i in R, z in {2..np}}: Gi[i,z] = Gi[i,z-1] + tup*qgi[i,z]/1000;
subject to 3_10e {i in R, z in {2..np}}: Wi[i,z] = Wi[i,z-1] +
     tup*qwi[i,z]/1000000;
subject to 3_11 {i in R, z in P}: Nwt[i,z] = Nop[i,z] + Ngi[i,z] + Nwi[i,z];
subject to 3_12a {z in P}: Nop_F[z] = sum{i in R} Nop[i,z];
subject to 3_12b {z in P}: Ngi_F[z] = sum{i in R} Ngi[i,z];
subject to 3_12c {z in P}: Nwi_F[z] = sum{i in R} Nwi[i,z];
subject to 3_12d {z in P}: Nwt_F[z] = sum{i in R} Nwt[i,z];
subject to 3_12e {z in P}: qopp_F[z] = sum{i in R} qopp[i,z];
subject to 3_12f {z in {2..np}}: qo_F[z] = sum{i in R} qo[i,z];
subject to 3_12g {z in {2..np}}: qg_F[z] = sum{i in R} qg[i,z];
subject to 3_12h {z in {2..np}}: qw_F[z] = sum{i in R} qw[i,z];
subject to 3_12i {z in {2..np}}: qgi_F[z] = sum{i in R} qgi[i,z];
subject to 3_12j {z in {2..np}}: qwi_F[z] = sum{i in R} qwi[i,z];
subject to 3_12k {z in P}: Np_F[z] = sum{i in R} Np[i,z];
subject to 3_12l {z in P}: Gp_F[z] = sum{i in R} Gp[i,z];
subject to 3_12m {z in P}: Wp_F[z] = sum{i in R} Wp[i,z];
subject to 3_12n {z in P}: Gi_F[z] = sum{i in R} Gi[i,z];
subject to 3_12o {z in P}: Wi_F[z] = sum{i in R} Wi[i,z];
subject to 3_15a {i in R, z in P}: Np[i,z] = sum{j in V_Np, k in V_Nop[i], l in
     V_RM[i]} lambda[i,z,j,k,l]*Np_bar[i,j,k,l];
subject to 3_15b {i in R, z in P}: Nop[i,z] = sum{j in V_Np, k in V_Nop[i], l in
     V_RM[i]} lambda[i,z,j,k,l]*Nop_bar[i,j,k,l];
subject to 3_15c {i in R, z in P}: RM[i] = sum{j in V_Np, k in V_Nop[i], l in
     V_RM[i]} lambda[i,z,j,k,l]*RMpp_bar[i,j,k,l];
subject to 3_15d {i in R, z in P}: qopp[i,z] = sum{j in V_Np, k in V_Nop[i], l
     in V_RM[i]} lambda[i,z,j,k,l]*qopp_bar[i,j,k,l];
subject to 3_15e {i in R, z in P}: 1 = sum{j in V_Np, k in V_Nop[i], l in
     V_RM[i]} lambda[i,z,j,k,l];
subject to 3_15f {i in R, z in P, j in V_Np}: eta_Np[i,z,j] = sum{k in V_Nop[i],
     l in V_RM[i]} lambda[i,z,j,k,l];
```

```
subject to 3_15g {i in R, z in P, k in V_Nop[i]}: eta_Nop[i,z,k] = sum{j in
     V_Np, l in V_RM[i]} lambda[i,z,j,k,l];
subject to 3_15h {i in R, z in P, l in V_RM[i]}: eta_RM[i,z,l] = sum{j in V_Np,
     k in V_Nop[i]} lambda[i,z,j,k,l];
subject to 3_16a {i in R, z in P}: Gp[i,z] = sum{j in V_Np, k in V_Nop[i], l in
     V_RM[i]} lambda[i,z,j,k,l]*Gp_bar[i,j,k,l];
subject to 3_16b {i in R, z in P}: Wp[i,z] = sum{j in V_Np, k in V_Nop[i], l in
     V_RM[i]} lambda[i,z,j,k,l]*Wp_bar[i,j,k,l];
subject to 3_16c {i in R, z in P}: Gi[i,z] = sum{j in V_Np, k in V_Nop[i], l in
     V_RM[i]} lambda[i,z,j,k,l]*Gi_bar[i,j,k,l];
subject to 3_16d {i in R, z in P}: Wi[i,z] = sum{j in V_Np, k in V_Nop[i], l in
     V_RM[i]} lambda[i,z,j,k,l]*Wi_bar[i,j,k,l];
subject to 3_17a {i in R}: zNgi[i] = sum{m in {1,2}, n in {1,2}}
     omega[i,m,n]*zNgi_bar[m,n];
subject to 3_17b {i in R}: zNwi[i] = sum{m in {1,2}, n in {1,2}}
     omega[i,m,n]*zNwi_bar[m,n];
subject to 3_17c {i in R}: RM[i] = sum{m in {1,2}, n in {1,2}}
     omega[i,m,n]*RMds_bar[m,n];
subject to 3_17d {i in R}: 1 = sum{m in {1,2}, n in {1,2}} omega[i,m,n];
subject to 3_17e {i in R, m in {1,2}}: tau_zNgi[i,m] = sum{n in {1,2}}
     omega[i,m,n];
subject to 3_17f {i in R, n in {1,2}}: tau_zNwi[i,n] = sum{m in {1,2}}
     omega[i,m,n];
subject to 4_6: PVe = sum{p in {1,2,3,4}} A[p]/(1+d)^(p-1);
subject to 4_7: PVdp = B*Nwt_F[1]/(1+d)^3;
subject to 4_8 {z in {1..(np-1)}}: PVd[z] = B*(Nwt_F[z+1] -
     Nwt_F[z])/(1+d)^(z+3);
subject to 4_9: PVa = (sum{p in {1,2,3,4}} C[p]/(1+d)^(p-1)) + D[1]*Nop_F[np] +
     D[2]*(Ngi_F[np] + Nwi_F[np]);
subject to 4_10a {i in R}: Nop[i,np] <= 4*Nt[i];
subject to 4_10b: Nt_F = sum{i in R} Nt[i];
subject to 4_11: Nx_F = Ngi_F[np] + Nwi_F[np];
subject to 4_12: PVsc = sum{p in {1,2,3,4}} (E[1]*Nt_F +
     E[2]*Nx_F)/(4*(1+d)^(p-1));
subject to 4_13a {z in {2..np}}: qo_F[z] <= max_qo_F;
subject to 4_13b {z in {2..np}}: qg_F[z] <= max_qg_F;
subject to 4_13c {z in {2..np}}: qw_F[z] <= max_qw_F;
subject to 4_14: PVtc = sum{p in {1,2,3,4}} (F[1]*max_qo_F + F[2]*max_qg_F +
     F[3]*max_qw_F)/(4*(1+d)^(p-1));
subject to 4_15: PVap = PVe + PVdp + PVa + PVsc + PVtc;
subject to 4_16 {z in {2..np}}: PVo[z] = (G[1] + G[2]*Nop_F[z-1] + G[3]*qo_F[z]
     + G[4]*qg_F[z] + G[5]*qw_F[z])/(1+d)^(z+3);
subject to 4_17 {z in {2..np}}: PVr[z] = Po*XR*VC*(Np_F[z] -
     Np_F[z-1])/(1+d)^(z+3);
subject to 4_18a: DCF[1] = -PVd[1];
subject to 4_18b {z in {2..(np-1)}}: DCF[z] = PVr[z] - PVo[z] - PVd[z];
subject to 4_18c: DCF[np] = PVr[np] - PVo[np];
subject to 4_19: NPV = -PVap + (sum{z in P} DCF[z]);
```

B.1.2 Data File (CASE-2_Data.dat)

```
param nr := 3;
param np := 18;
param tup := 365;
param max_Nop_pd := 4;
param max_Ni_pd := 1;
param max_Nwt_py := 4;
param: max_Ngi max_Nwi :=
#i      max_Ngi max_Nwi
1       1       3
2       0       2
3       0       1
;
```

```
param: zNgi_bar zNwi_bar RMds_bar :=
#m n   zNgi_bar zNwi_bar RMds_bar
1  1   0        0        1
1  2   0        1        2
2  1   1        0        3
2  2   1        1        4
;
param max_qgi_pgi := 4000;
param max_qwi_pwi := 6360;
```

B.1.3 Data File (CASE-2_Econ.dat)

```
param Po := 60;
param XR := 8.5;
param VC := 6.29;
param d := 0.12;
param: A :=
1      665
2      240
3      380
4      366
;
param B := 480;
param: C :=
1      192
2      449
3      449
4      192
;
param: D :=
1      12
2      5
;
param: E :=
1      484
2      50
;
param: F :=
1      0.10914
2      0.23848
3      0.22074
;
param: G :=
1      400
2      4
3      0.007388
4      0.018289
5      0.022829
;
```

B.1.4 Run File (CASE-2_Run.run)

```
model CASE-2_Model.mod;
data CASE-2_Data.dat;
data CASE-2_Econ.dat;
data PWL_Table.dat;

option solver cplex;
option cplex_options "threads=8 timelimit=43200 mipdisplay=2 mipgap=0.0001
     nodefile=3 bestbound return_mipgap=1";

suffix sosno;
suffix ref;
param itersos2;
let itersos2 := 0;
for{i in R, z in P} {
  let itersos2 := itersos2-1;
  let {j in V_Np} eta_Np[i,z,j].sosno := itersos2;
  let {j in V_Np} eta_Np[i,z,j].ref := Np_bar[i,j,1,1];

  let itersos2 := itersos2-1;
  let {k in V_Nop[i]} eta_Nop[i,z,k].sosno := itersos2;
  let {k in V_Nop[i]} eta_Nop[i,z,k].ref := Nop_bar[i,1,k,1];
}
param itersos1;
let itersos1 := 0;
for{i in R, z in P} {
  let itersos1 := itersos1+1;
  let {l in V_RM[i]} eta_RM[i,z,l].sosno := itersos1;
  let {l in V_RM[i]} eta_RM[i,z,l].ref := RMpp_bar[i,1,1,l];
}
for{i in R} {
  let itersos1 := itersos1+1;
  let {m in {1,2}} tau_zNgi[i,m].sosno := itersos1;
  let {m in {1,2}} tau_zNgi[i,m].ref := zNgi_bar[m,1];

  let itersos1 := itersos1+1;
  let {n in {1,2}} tau_zNwi[i,n].sosno := itersos1;
  let {n in {1,2}} tau_zNwi[i,n].ref := zNwi_bar[1,n];
}

solve;

display RM >CASE-2_Results.out;
display Nop >CASE-2_Results.out;
display Ngi >CASE-2_Results.out;
.
.
.
close CASE-2_Results.out;
```

B.2 REF-CASE-2

B.2.1 Model File (REF-CASE-2_Model.mod)

```
# Part 0: Index Description
# i = reservoirs
# j = breakpoints for Np
# k = breakpoints for Nop
# l = breakpoints for RM
# p = points in time prior to the production period
# z = points in time during the production period
#-----------------------------------------------------------------------------

# Part 1: Declaration (var, set, param, etc)
param nr;
set R := 1..nr;
param np;
set P := 1..np;
```

```
param tup;
param Nop {i in R, z in P};
param Nwi {i in R, z in P};
param Ngi {i in R, z in P};
param RM {i in R, z in P};
param max_qgi_pgi;
param max_qwi_pwi;
param nbp_Np;
param nbp_Nop {i in R};
param nbp_RM {i in R};
set V_Np := 1..nbp_Np;
set V_Nop {i in R} := 1..nbp_Nop[i];
set V_RM {i in R} := 1..nbp_RM[i];
param Np_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param Nop_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param RMpp_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param qopp_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param Gp_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param Wp_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param Gi_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param Wi_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param Po;
param XR;
param VC;
param d;
param A {p in {1,2,3,4}};
param B;
param C {p in {1,2,3,4}};
param D {q in {1,2}};
param E {r in {1,2}};
param F {s in {1,2,3}};
param G {t in {1,2,3,4,5}};

var Nwt {i in R, z in P} integer >= 0;
var qopp {i in R, z in P} >= 0;
var qo {i in R, z in {2..np}} >= 0;
var qg {i in R, z in {2..np}} >= 0;
var qw {i in R, z in {2..np}} >= 0;
var qgi {i in R, z in {2..np}} >= 0;
var qwi {i in R, z in {2..np}} >= 0;
var Np {i in R, z in P} >= 0;
var Gp {i in R, z in P} >= 0;
var Wp {i in R, z in P} >= 0;
var Gi {i in R, z in P} >= 0;
var Wi {i in R, z in P} >= 0;
var Nop_F {z in P} integer >= 0;
var Ngi_F {z in P} integer >= 0;
var Nwi_F {z in P} integer >= 0;
var Nwt_F {z in P} integer >= 0;
var qopp_F {z in P} >= 0;
var qo_F {z in {2..np}} >= 0;
var qg_F {z in {2..np}} >= 0;
var qw_F {z in {2..np}} >= 0;
var qgi_F {z in {2..np}} >= 0;
var qwi_F {z in {2..np}} >= 0;
var Np_F {z in P} >= 0;
var Gp_F {z in P} >= 0;
var Wp_F {z in P} >= 0;
var Gi_F {z in P} >= 0;
var Wi_F {z in P} >= 0;
var lambda {i in R, z in P, j in V_Np, k in V_Nop[i]} >= 0;
var eta_Np {i in R, z in P, j in V_Np} >= 0;
var eta_Nop {i in R, z in P, k in V_Nop[i]} >= 0;
var PVe >= 0;
var PVdp >= 0;
var PVa >= 0;
var Nt {i in R} integer >= 0;
var Nt_F integer >= 0;
var Nx_F integer >= 0;
var PVsc >= 0;
var max_qo_F >= 0;
var max_qg_F >= 0;
var max_qw_F >= 0;
var PVtc >= 0;
var PVap >= 0;
var PVd {z in {1..(np-1)}} >= 0;
var PVo {z in {2..np}} >= 0;
var PVr {z in {2..np}} >= 0;
var DCF {z in P};
var NPV;
#--------------------------------------------------------------------------------

# Part 2: Objective Function
maximize 4_5: NPV;
#--------------------------------------------------------------------------------

# Part 3: Constraints
subject to 3_3a {i in R, z in {2..np}}: qo[i,z] <= qopp[i,z];
subject to 3_3b {i in R, z in {2..np}}: qo[i,z] <= qopp[i,z-1];
subject to 3_4a {i in R, z in {2..np}}: qgi[i,z] <= Ngi[i,z-1]*max_qgi_pgi;
subject to 3_4b {i in R, z in {2..np}}: qwi[i,z] <= Nwi[i,z-1]*max_qwi_pwi;
subject to 3_9 {i in R}: Np[i,1] = 0;
subject to 3_10a {i in R, z in {2..np}}: Np[i,z] = Np[i,z-1] +
     tup*qo[i,z]/1000000;
subject to 3_10b {i in R, z in {2..np}}: Gp[i,z] = Gp[i,z-1] + tup*qg[i,z]/1000;
subject to 3_10c {i in R, z in {2..np}}: Wp[i,z] = Wp[i,z-1] +
     tup*qw[i,z]/1000000;
subject to 3_10d {i in R, z in {2..np}}: Gi[i,z] = Gi[i,z-1] + tup*qgi[i,z]/1000;
subject to 3_10e {i in R, z in {2..np}}: Wi[i,z] = Wi[i,z-1] +
     tup*qwi[i,z]/1000000;
subject to 3_11 {i in R, z in P}: Nwt[i,z] = Nop[i,z] + Ngi[i,z] + Nwi[i,z];
subject to 3_12a {z in P}: Nop_F[z] = sum{i in R} Nop[i,z];
subject to 3_12b {z in P}: Ngi_F[z] = sum{i in R} Ngi[i,z];
subject to 3_12c {z in P}: Nwi_F[z] = sum{i in R} Nwi[i,z];
subject to 3_12d {z in P}: Nwt_F[z] = sum{i in R} Nwt[i,z];
subject to 3_12e {z in P}: qopp_F[z] = sum{i in R} qopp[i,z];
subject to 3_12f {z in {2..np}}: qo_F[z] = sum{i in R} qo[i,z];
subject to 3_12g {z in {2..np}}: qg_F[z] = sum{i in R} qg[i,z];
subject to 3_12h {z in {2..np}}: qw_F[z] = sum{i in R} qw[i,z];
subject to 3_12i {z in {2..np}}: qgi_F[z] = sum{i in R} qgi[i,z];
subject to 3_12j {z in {2..np}}: qwi_F[z] = sum{i in R} qwi[i,z];
subject to 3_12k {z in P}: Np_F[z] = sum{i in R} Np[i,z];
subject to 3_12l {z in P}: Gp_F[z] = sum{i in R} Gp[i,z];
subject to 3_12m {z in P}: Wp_F[z] = sum{i in R} Wp[i,z];
subject to 3_12n {z in P}: Gi_F[z] = sum{i in R} Gi[i,z];
subject to 3_12o {z in P}: Wi_F[z] = sum{i in R} Wi[i,z];
subject to 3_15a {i in R, z in P}: Np[i,z] = sum{j in V_Np, k in V_Nop[i]}
     lambda[i,z,j,k]*Np_bar[i,j,k,RM[i,z]];
subject to 3_15b {i in R, z in P}: Nop[i,z] = sum{j in V_Np, k in V_Nop[i]}
     lambda[i,z,j,k]*Nop_bar[i,j,k,RM[i,z]];
subject to 3_15d {i in R, z in P}: qopp[i,z] = sum{j in V_Np, k in V_Nop[i]}
     lambda[i,z,j,k]*qopp_bar[i,j,k,RM[i,z]];
subject to 3_15e {i in R, z in P}: 1 = sum{j in V_Np, k in V_Nop[i]}
     lambda[i,z,j,k];
subject to 3_15f {i in R, z in P, j in V_Np}: eta_Np[i,z,j] = sum{k in V_Nop[i]}
     lambda[i,z,j,k];
subject to 3_15g {i in R, z in P, k in V_Nop[i]}: eta_Nop[i,z,k] = sum{j in
     V_Np} lambda[i,z,j,k];
subject to 3_16a {i in R, z in P}: Gp[i,z] = sum{j in V_Np, k in V_Nop[i]}
     lambda[i,z,j,k]*Gp_bar[i,j,k,RM[i,z]];
subject to 3_16b {i in R, z in P}: Wp[i,z] = sum{j in V_Np, k in V_Nop[i]}
     lambda[i,z,j,k]*Wp_bar[i,j,k,RM[i,z]];
subject to 3_16c {i in R, z in P}: Gi[i,z] = sum{j in V_Np, k in V_Nop[i]}
     lambda[i,z,j,k]*Gi_bar[i,j,k,RM[i,z]];
subject to 3_16d {i in R, z in P}: Wi[i,z] = sum{j in V_Np, k in V_Nop[i]}
     lambda[i,z,j,k]*Wi_bar[i,j,k,RM[i,z]];
```

```
subject to 4_6: PVe = sum{p in {1,2,3,4}} A[p]/(1+d)^(p-1);
subject to 4_7: PVdp = B*Nwt_F[1]/(1+d)^3;
subject to 4_8 {z in {1..(np-1)}}: PVd[z] = B*(Nwt_F[z+1] -
     Nwt_F[z])/(1+d)^(z+3);
subject to 4_9: PVa = (sum{p in {1,2,3,4}} C[p]/(1+d)^(p-1)) + D[1]*Nop_F[np] +
     D[2]*(Ngi_F[np] + Nwi_F[np]);
subject to 4_10a {i in R}: Nop[i,np] <= 4*Nt[i];
subject to 4_10b: Nt_F = sum{i in R} Nt[i];
subject to 4_11: Nx_F = Ngi_F[np] + Nwi_F[np];
subject to 4_12: PVsc = sum{p in {1,2,3,4}} (E[1]*Nt_F +
     E[2]*Nx_F)/(4*(1+d)^(p-1));
subject to 4_13a {z in {2..np}}: qo_F[z] <= max_qo_F;
subject to 4_13b {z in {2..np}}: qg_F[z] <= max_qg_F;
subject to 4_13c {z in {2..np}}: qw_F[z] <= max_qw_F;
subject to 4_14: PVtc = sum{p in {1,2,3,4}} (F[1]*max_qo_F + F[2]*max_qg_F +
     F[3]*max_qw_F)/(4*(1+d)^(p-1));
subject to 4_15: PVap = PVe + PVdp + PVa + PVsc + PVtc;
subject to 4_16 {z in {2..np}}: PVo[z] = (G[1] + G[2]*Nop_F[z-1] + G[3]*qo_F[z]
     + G[4]*qg_F[z] + G[5]*qw_F[z])/(1+d)^(z+3);
subject to 4_17 {z in {2..np}}: PVr[z] = Po*XR*VC*(Np_F[z] -
     Np_F[z-1])/(1+d)^(z+3);
subject to 4_18a: DCF[1] = -PVd[1];
subject to 4_18b {z in {2..(np-1)}}: DCF[z] = PVr[z] - PVo[z] - PVd[z];
subject to 4_18c: DCF[np] = PVr[np] - PVo[np];
subject to 4_19: NPV = -PVap + (sum{z in P} DCF[z]);
```

B.2.2 Data File (REF-CASE-2_Data.dat)

```
param nr := 3;
param np := 18;
param tup := 365;
param: Nop Ngi Nwi RM :=
#i z   Nop Ngi Nwi RM
1  1   4   1    0   3
1  2   7   1    1   4
1  3   10  1    2   4
.
.
.
;
param max_qgi_pgi := 4000;
param max_qwi_pwi := 6360;
```

B.2.3 Data File (REF-CASE-2_Econ.dat)

```
param Po := 60;
param XR := 8.5;
param VC := 6.29;
param d := 0.12;
param: A :=
1      665
2      240
3      380
4      366
;
param B := 480;
param: C :=
1      192
```

```
2      449
3      449
4      192
;
param: D :=
1      12
2      5
;
param: E :=
1      484
2      50
;
param: F :=
1      0.10914
2      0.23848
3      0.22074
;
param: G :=
1      400
2      4
3      0.007388
4      0.018289
5      0.022829
;
```

B.2.4 Run File (REF-CASE-2_Run.run)

```
model REF-CASE-2_Model.mod;
data REF-CASE-2_Data.dat;
data REF-CASE-2_Econ.dat;
data PWL_Table.dat;

option solver cplex;
option cplex_options "threads=8 timelimit=43200 mipdisplay=2 mipgap=0.0001
     nodefile=3 bestbound return_mipgap=1";

suffix sosno;
suffix ref;
param itersos2;
let itersos2 := 0;
for{i in R, z in P} {
  let itersos2 := itersos2-1;
  let {j in V_Np} eta_Np[i,z,j].sosno := itersos2;
  let {j in V_Np} eta_Np[i,z,j].ref := Np_bar[i,j,1,1];

  let itersos2 := itersos2-1;
  let {k in V_Nop[i]} eta_Nop[i,z,k].sosno := itersos2;
  let {k in V_Nop[i]} eta_Nop[i,z,k].ref := Nop_bar[i,1,k,1];
}

solve;

display RM >REF-CASE-2_Results.out;
display Nop >REF-CASE-2_Results.out;
display Ngi >REF-CASE-2_Results.out;
.
.
.
close REF-CASE-2_Results.out;
```

Appendix C

AMPL Code for Uncertainty Analysis

C.1 Model File (CASE-1xx_Model.mod)

```
# Part 0: Index Description
# i = reservoirs
# j = breakpoints for Np
# k = breakpoints for Nop
# l = breakpoints for RM
# p = points in time prior to the production period
# z = points in time during the production period
#--------------------------------------------------------------------------------

# Part 1: Declaration (var, set, param, etc)
param nr;
set R := 1..nr;
param np;
set P := 1..np;
param tup;
param max_Nop_pd;
param max_Ni_pd;
param max_Nwt_py;
param max_Ngi {i in R};
param max_Nwi {i in R};
param RM {i in R};
param zNgi {i in R};
param zNwi {i in R};
param max_qgi_pgi;
param max_qwi_pwi;
param nbp_Np;
param nbp_Nop {i in R};
param nbp_RM {i in R};
set V_Np := 1..nbp_Np;
set V_Nop {i in R} := 1..nbp_Nop[i];
set V_RM {i in R} := 1..nbp_RM[i];
param Np_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param Nop_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param RMpp_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param qopp_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param Gp_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param Wp_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param Gi_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param Wi_bar {i in R, j in V_Np, k in V_Nop[i], l in V_RM[i]};
param Po;
param XR;
param VC;
param d;
param A {p in {1,2,3,4}};
param B;
param C {p in {1,2,3,4}};
param D {q in {1,2}};
param E {r in {1,2}};
param F {s in {1,2,3}};
param G {t in {1,2,3,4,5}};
param u_N;
param u_C;
param u_Po;

var Nop {i in R, z in P} integer >= 0;
var Ngi {i in R, z in P} integer >= 0;
var Nwi {i in R, z in P} integer >= 0;
var Nwt {i in R, z in P} integer >= 0;
var qopp {i in R, z in P} >= 0;
var qo {i in R, z in {2..np}} >= 0;
var qg {i in R, z in {2..np}} >= 0;
var qw {i in R, z in {2..np}} >= 0;
var qgi {i in R, z in {2..np}} >= 0;
var qwi {i in R, z in {2..np}} >= 0;
var Np {i in R, z in P} >= 0;
var Gp {i in R, z in P} >= 0;
var Wp {i in R, z in P} >= 0;
var Gi {i in R, z in P} >= 0;
var Wi {i in R, z in P} >= 0;
var Nop_F {z in P} integer >= 0;
var Ngi_F {z in P} integer >= 0;
var Nwi_F {z in P} integer >= 0;
var Nwt_F {z in P} integer >= 0;
var qopp_F {z in P} >= 0;
var qo_F {z in {2..np}} >= 0;
var qg_F {z in {2..np}} >= 0;
var qw_F {z in {2..np}} >= 0;
var qgi_F {z in {2..np}} >= 0;
var qwi_F {z in {2..np}} >= 0;
var Np_F {z in P} >= 0;
var Gp_F {z in P} >= 0;
var Wp_F {z in P} >= 0;
var Gi_F {z in P} >= 0;
var Wi_F {z in P} >= 0;
var lambda {i in R, z in P, j in V_Np, k in V_Nop[i]} >= 0;
var eta_Np {i in R, z in P, j in V_Np} >= 0;
var eta_Nop {i in R, z in P, k in V_Nop[i]} >= 0;
var PVe >= 0;
var PVdp >= 0;
var PVa >= 0;
var Nt {i in R} integer >= 0;
var Nt_F integer >= 0;
var Nx_F integer >= 0;
var PVsc >= 0;
var max_qo_F >= 0;
var max_qg_F >= 0;
var max_qw_F >= 0;
var PVtc >= 0;
var PVap >= 0;
var PVd {z in {1..(np-1)}} >= 0;
var PVo {z in {2..np}} >= 0;
var PVr {z in {2..np}} >= 0;
var DCF {z in P};
var NPV;
#-----------------------------------------------------------------------------------

# Part 2: Objective Function
maximize 4_5: NPV;
#-----------------------------------------------------------------------------------

# Part 3: Constraints
subject to 3_3a {i in R, z in {2..np}}: qo[i,z] <= qopp[i,z];
subject to 3_3b {i in R, z in {2..np}}: qo[i,z] <= qopp[i,z-1];
subject to 3_4a {i in R, z in {2..np}}: qgi[i,z] <= Ngi[i,z-1]*max_qgi_pgi;
subject to 3_4b {i in R, z in {2..np}}: qwi[i,z] <= Nwi[i,z-1]*max_qwi_pwi;
subject to 3_5a {i in R, z in {2..np}}: Nop[i,z] - Nop[i,z-1] >= 0;
subject to 3_5b {i in R, z in {2..np}}: Ngi[i,z] - Ngi[i,z-1] >= 0;
subject to 3_5c {i in R, z in {2..np}}: Nwi[i,z] - Nwi[i,z-1] >= 0;
subject to 3_6a {i in R, z in P}: Ngi[i,z] <= zNgi[i]*max_Ngi[i];
subject to 3_6b {i in R, z in P}: Nwi[i,z] <= zNwi[i]*max_Nwi[i];
subject to 3_7a: Nop_F[1] <= max_Nop_pd;
subject to 3_7b: Ngi_F[1] + Nwi_F[1] <= max_Ni_pd;
subject to 3_8 {z in {2..np}}: Nwt_F[z] - Nwt_F[z-1] <= max_Nwt_py;
subject to 3_9 {i in R}: Np[i,1] = 0;
subject to 3_10a {i in R, z in {2..np}}: Np[i,z] = Np[i,z-1] +
     tup*qo[i,z]/1000000;
subject to 3_10b {i in R, z in {2..np}}: Gp[i,z] = Gp[i,z-1] + tup*qg[i,z]/1000;
subject to 3_10c {i in R, z in {2..np}}: Wp[i,z] = Wp[i,z-1] +
     tup*qw[i,z]/1000000;
subject to 3_10d {i in R, z in {2..np}}: Gi[i,z] = Gi[i,z-1] + tup*qgi[i,z]/1000;
subject to 3_10e {i in R, z in {2..np}}: Wi[i,z] = Wi[i,z-1] +
     tup*qwi[i,z]/1000000;
subject to 3_11 {i in R, z in P}: Nwt[i,z] = Nop[i,z] + Ngi[i,z] + Nwi[i,z];
subject to 3_12a {z in P}: Nop_F[z] = sum{i in R} Nop[i,z];
subject to 3_12b {z in P}: Ngi_F[z] = sum{i in R} Ngi[i,z];
subject to 3_12c {z in P}: Nwi_F[z] = sum{i in R} Nwi[i,z];
subject to 3_12d {z in P}: Nwt_F[z] = sum{i in R} Nwt[i,z];
subject to 3_12e {z in P}: qopp_F[z] = sum{i in R} qopp[i,z];
subject to 3_12f {z in {2..np}}: qo_F[z] = sum{i in R} qo[i,z];
subject to 3_12g {z in {2..np}}: qg_F[z] = sum{i in R} qg[i,z];
subject to 3_12h {z in {2..np}}: qw_F[z] = sum{i in R} qw[i,z];
subject to 3_12i {z in {2..np}}: qgi_F[z] = sum{i in R} qgi[i,z];
subject to 3_12j {z in {2..np}}: qwi_F[z] = sum{i in R} qwi[i,z];
subject to 3_12k {z in P}: Np_F[z] = sum{i in R} Np[i,z];
subject to 3_12l {z in P}: Gp_F[z] = sum{i in R} Gp[i,z];
subject to 3_12m {z in P}: Wp_F[z] = sum{i in R} Wp[i,z];
subject to 3_12n {z in P}: Gi_F[z] = sum{i in R} Gi[i,z];
subject to 3_12o {z in P}: Wi_F[z] = sum{i in R} Wi[i,z];
subject to 6_10 {i in R, z in P}: Np[i,z] = u_N*(sum{j in V_Np, k in V_Nop[i]}
     lambda[i,z,j,k]*Np_bar[i,j,k,RM[i]]);
subject to 3_15b {i in R, z in P}: Nop[i,z] = sum{j in V_Np, k in V_Nop[i]}
     lambda[i,z,j,k]*Nop_bar[i,j,k,RM[i]];
subject to 3_15d {i in R, z in P}: qopp[i,z] = sum{j in V_Np, k in V_Nop[i]}
     lambda[i,z,j,k]*qopp_bar[i,j,k,RM[i]];
subject to 3_15e {i in R, z in P}: 1 = sum{j in V_Np, k in V_Nop[i]}
     lambda[i,z,j,k];
subject to 3_15f {i in R, z in P, j in V_Np}: eta_Np[i,z,j] = sum{k in V_Nop[i]}
     lambda[i,z,j,k];
subject to 3_15g {i in R, z in P, k in V_Nop[i]}: eta_Nop[i,z,k] = sum{j in
     V_Np} lambda[i,z,j,k];
subject to 3_16a {i in R, z in P}: Gp[i,z] = sum{j in V_Np, k in V_Nop[i]}
     lambda[i,z,j,k]*Gp_bar[i,j,k,RM[i]];
subject to 3_16b {i in R, z in P}: Wp[i,z] = sum{j in V_Np, k in V_Nop[i]}
     lambda[i,z,j,k]*Wp_bar[i,j,k,RM[i]];
subject to 3_16c {i in R, z in P}: Gi[i,z] = sum{j in V_Np, k in V_Nop[i]}
     lambda[i,z,j,k]*Gi_bar[i,j,k,RM[i]];
subject to 3_16d {i in R, z in P}: Wi[i,z] = sum{j in V_Np, k in V_Nop[i]}
     lambda[i,z,j,k]*Wi_bar[i,j,k,RM[i]];
subject to 6_11a: PVe = u_C*(sum{p in {1,2,3,4}} A[p]/(1+d)^(p-1));
subject to 6_11b: PVdp = u_C*(B*Nwt_F[1]/(1+d)^3);
subject to 6_11c {z in {1..(np-1)}}: PVd[z] = u_C*(B*(Nwt_F[z+1] -
     Nwt_F[z])/(1+d)^(z+3));
subject to 6_11d: PVa = u_C*((sum{p in {1,2,3,4}} C[p]/(1+d)^(p-1)) +
     D[1]*Nop_F[np] + D[2]*(Ngi_F[np] + Nwi_F[np]));
subject to 4_10a {i in R}: Nop[i,np] <= 4*Nt[i];
subject to 4_10b: Nt_F = sum{i in R} Nt[i];
subject to 4_11: Nx_F = Ngi_F[np] + Nwi_F[np];
subject to 6_11e: PVsc = u_C*(sum{p in {1,2,3,4}} (E[1]*Nt_F +
     E[2]*Nx_F)/(4*(1+d)^(p-1)));
subject to 4_13a {z in {2..np}}: qo_F[z] <= max_qo_F;
subject to 4_13b {z in {2..np}}: qg_F[z] <= max_qg_F;
subject to 4_13c {z in {2..np}}: qw_F[z] <= max_qw_F;
subject to 6_11f: PVtc = u_C*(sum{p in {1,2,3,4}} (F[1]*max_qo_F + F[2]*max_qg_F
     + F[3]*max_qw_F)/(4*(1+d)^(p-1)));
subject to 4_15: PVap = PVe + PVdp + PVa + PVsc + PVtc;
subject to 6_11g {z in {2..np}}: PVo[z] = u_C*((G[1] + G[2]*Nop_F[z-1] +
     G[3]*qo_F[z] + G[4]*qg_F[z] + G[5]*qw_F[z])/(1+d)^(z+3));
subject to 6_12 {z in {2..np}}: PVr[z] = u_Po*(Po*XR*VC*(Np_F[z] -
     Np_F[z-1])/(1+d)^(z+3));
subject to 4_18a: DCF[1] = -PVd[1];
subject to 4_18b {z in {2..(np-1)}}: DCF[z] = PVr[z] - PVo[z] - PVd[z];
subject to 4_18c: DCF[np] = PVr[np] - PVo[np];
subject to 4_19: NPV = -PVap + (sum{z in P} DCF[z]);
```

C.2 Data File (CASE-1xx_Data.dat)

```
param nr := 3;
param np := 18;
param tup := 365;
param max_Nop_pd := 4;
param max_Ni_pd := 1;
param max_Nwt_py := 4;
param: max_Ngi max_Nwi :=
#i     max_Ngi max_Nwi
1      1       3
2      0       2
3      0       1
;
param: RM zNgi zNwi:=
#i     RM zNgi zNwi
1      4  1    1
2      2  0    1
3      2  0    1
;
param max_qgi_pgi := 4000;
param max_qwi_pwi := 6360;
```

C.3 Data File (CASE-1xx_Econ.dat)

```
param Po := 60;
param XR := 8.5;
param VC := 6.29;
param d := 0.12;
param: A :=
1      665
```

```
2      240
3      380
4      366
;
param B := 480;
param: C :=
1      192
2      449
3      449
4      192
;
param: D :=
1      12
2      5
;
param: E :=
1      484
2      50
;
param: F :=
1      0.10914
2      0.23848
3      0.22074
;
param: G :=
1      400
2      4
3      0.007388
4      0.018289
5      0.022829
;
```

C.4 Files Related to CASE-101

C.4.1 Data File (CASE-101_Uncertainty.dat)

```
param u_N := 0.855240;
param u_C := 1.118941;
param u_Po := 0.924041;
```

C.4.2 Run File (CASE-101_Run.run)

```
model CASE-1xx_Model.mod;
data CASE-1xx_Data.dat;
data CASE-1xx_Econ.dat;
data CASE-101_Uncertainty.dat;
data PWL_Table.dat;

option solver cplex;
option cplex_options "threads=8 timelimit=3600 mipdisplay=2 mipgap=0.0001
     nodefile=3 bestbound return_mipgap=1";

suffix sosno;
suffix ref;
param itersos2;
let itersos2 := 0;
```

```
for{i in R, z in P} {
  let itersos2 := itersos2-1;
  let {j in V_Np} eta_Np[i,z,j].sosno := itersos2;
  let {j in V_Np} eta_Np[i,z,j].ref := Np_bar[i,j,1,1];

  let itersos2 := itersos2-1;
  let {k in V_Nop[i]} eta_Nop[i,z,k].sosno := itersos2;
  let {k in V_Nop[i]} eta_Nop[i,z,k].ref := Nop_bar[i,1,k,1];
}

solve;

display u_N, u_C, u_Po >CASE-101_Results.out;
display RM >CASE-101_Results.out;
display Nop >CASE-101_Results.out;
.
.
.
close CASE-101_Results.out;
```

C.5 Run File (MASTER_Run.run)

```
include CASE-101_Run.run;
reset;

include CASE-102_Run.run;
reset;

.
.
.

include CASE-200_Run.run;
reset;
```