THE UNIVERSITY OF TULSA

THE GRADUATE SCHOOL

OIL FIELD DEVELOPMENT OPTIMIZATION UNDER GEOLOGICAL UNCERTAINTY

by
Ranran Lu

A thesis submitted in partial fulfillment of
the requirements for the degree of Doctor of Philosophy
in the Discipline of Petroleum Engineering

The Graduate School
The University of Tulsa

2020

THE UNIVERSITY OF TULSA

THE GRADUATE SCHOOL

OIL FIELD DEVELOPMENT OPTIMIZATION UNDER GEOLOGICAL UNCERTAINTY

by
Ranran Lu

A THESIS

APPROVED FOR THE DISCIPLINE OF

PETROLEUM ENGINEERING

By Thesis Committee

Albert C. Reynolds Jr., Chair
Rami Mustafa Younis
Mustafa Onur
Siamack A. Shirazi

COPYRIGHT STATEMENT

ABSTRACT

Ranran Lu (Doctor of Philosophy in Petroleum Engineering)

Oil Field Development Optimization Under Geological Uncertainty

Directed by Albert C. Reynolds Jr.

227 pp., Chapter 6: Discussions and Conclusions

(451 words)

Field development optimization is a critical part of the decision making process for reservoir management. Many researchers have studied individual parts of this optimization process, where the two most typical problems are the well control optimization and well placement optimization where the number, types and drilling orders of the wells to be drilled are assumed to be known which is not the case most of the time; hence, there is a strong need to do research on the full field development optimization problem where the number of wells, their types, drilling sequences, locations and control settings are considered as optimization (design) variables. The full field development optimization problem is a mixed integer problem. Common mathematical optimization algorithms, e.g. branch and bound, for solving mixed integer problems are too computationally intensive to apply to field optimization because each iteration of an optimization requires at least one run of a reservoir simulation to evaluate the cost function, which in this work is represented by the net present value (NPV) of production over the life of the reservoir. The full field development problem becomes even more computationally intensive when one considers geological uncertainty where one generally performs robust optimization to maximize the average NPV of life-cycle production over a set of N_e plausible reservoir models representing geological uncertainty. For robust optimization, one needs to perform at least N_e reservoir simulation runs at each iteration of an optimization algorithm. Moreover, when geological uncertainty is considered, it may be desirable to conduct bi-objective optimization to minimize the downside risk.

This research focuses on proposing parameterizations for field optimization, and developing efficient methodologies to solve the field development optimization problem. The focus is a sequential optimization as an alternative to simultaneous optimization. The problems considered include (i) selecting a fixed number of wells from a given set of potential drilling paths and determining the well types (injectors or producers), drilling order and control settings, (ii) optimizing number of wells, types and locations followed by optimizing well controls, (iii) robust optimization of the second problem where a set of realizations is used to characterize the geological uncertainty, and (iv) bi-objective well placement optimization where the first objective is to maximize the average NPV and the second is to maximize the minimum NPV of a given set of realizations. The methodologies focus on hybridizing the Genetic Algorithm (GA), the Stochastic Simplex Approximate Gradient method (StoSAG) and the General Pattern Search (GPS) to handle different types of optimization variables and to combine global search ability and local search ability. The validity of the proposed methods are tested mainly on waterflooding examples, considering vertical wells, slanted wells and multi-segment wells subjected to the nonlinear well spacing constraints.

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to my advisor, Dr. Albert C. Reynolds, for his guidance, support and patience during my Ph.D. study. His integrity, hard work and dedication will always be an example in my life. I would also like to extend my appreciation to my committee members: Dr. Mustafa Onur, Dr. Rami Younis and Dr. Siamack Shirazi for their advice and comments on my research. I am very grateful to all the member companies of the TUPREP, Graduate School and the McDougall School of Petroleum Engineering for all kinds of support provided during my Ph.D. study. I also would like to show my thanks and appreciations to Judy, Ronda and Loreta for the administrative help.

I deeply appreciate all the help, support and encouragement from my friends and colleagues: Cintia, Sepide, Xin Li, Yuanshan, Bailian, Emilio, Guotong, Gosha, Javad, Reza, Shuaiwei, Wen, Xin Liu, Ying, Yu, Yang, Zhe and Zhenyu, et al..

I would like to show my greatest gratitude to my parents, my siblings and my husband, to whom I dedicate this dissertation, for their unconditional support and love. I am truly thankful for having them in my life.

TABLE OF CONTENTS

COPYRIGHT iii

ABSTRACT iv

ACKNOWLEDGEMENTS vi

TABLE OF CONTENTS ix

LIST OF TABLES xiii

LIST OF FIGURES xxix

CHAPTER 1: **INTRODUCTION** 1

1.1 **Literature Review** 2

1.2 **Research Objectives and Contextures** 21

1.3 **Dissertation Outline** 25

CHAPTER 2: **COMPUTATIONAL ALGORITHMS FOR FIELD DEVELOPMENT OPTIMIZATION** 26

2.1 **Problem Description** 26

2.2 **Basic Methodology Considered** 32

2.2.1 *Genetic Algorithm (GA)* 33

2.2.2 *Stochastic Simplex Approximate Gradient (StoSAG)* 35

2.2.3 *General Pattern Search (GPS)* 39

CHAPTER 3: **SELECTING WELLS FROM A GIVEN SET OF POTENTIAL PATHS** 43

3.1 **Problem Description and Methodology** 43

3.1.1 *Optimization of well locations, types and drilling order* 47

3.1.2 *Optimization of well controls* $\boldsymbol{u}$ 54

3.2 **Numerical Examples** 55

3.2.1 *Example 1: channelized reservoir* 56

3.2.2 *Example 2: Brugge model* 68

3.2.3 *Investigation on drilling order effect* 83

3.2.4 *Investigation on population size* 86

3.3 **Comparison with StoSAG Using Priority Parametrization** 89

3.3.1 *StoSAG using priority parametrization* 89

3.3.2 *Numerical examples* . . . 90
3.4 **Comparison with a Gradient-based Algorithm** . . . 94
3.4.1 *A gradient-based algorithm* . . . 94
3.4.2 *Optimization of* $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{u})$ *using GA* . . . 97
3.4.3 *Numerical example: channelized reservoir* . . . 100
3.5 **Summary and Discussion** . . . 102

CHAPTER 4: **JOINT OPTIMIZATION OF WELL NUMBER, TYPES AND LOCATIONS** 104
4.1 **Problem Description and Methodology** . . . 104
4.1.1 *Problem description* . . . 104
4.1.2 *Constraints handling* . . . 106
4.1.3 *Joint optimization of well status, types and locations* . . . 108
4.2 **Numerical Examples of Deterministic Optimization** . . . 115
4.2.1 *Example 1: PUNQ-S3 model* . . . 116
4.2.2 *Example 2: Brugge model* . . . 128
4.3 **Field Development Optimization Considering Geological Uncertainty** . . . 134
4.4 **Numerical Examples of Robust Optimization** . . . 142
4.4.1 *Example 1: PUNQ model* . . . 142
4.4.2 *Example 2: Brugge model* . . . 151
4.5 **Summary and Discussion** . . . 159

CHAPTER 5: **WELL PLACEMENT OPTIMIZATION WITH COMPLICATED WELL TRAJECTORIES** 161
5.1 **Well Trajectory Parameterization** . . . 161
5.1.1 *Slanted well trajectory parameterization* . . . 161
5.1.2 *Multi-segmented well trajectory parameterization* . . . 163
5.2 **Constrained Optimization using Augmented Lagrangian Method** . 168
5.3 **Bi-objective Well Placement Optimization** . . . 172
5.4 **Numerical Example: Channelized Reservoir** . . . 177
5.4.1 *Deterministic optimization using realization #1* . . . 180
5.4.2 *Bi-objective optimization using 10 realizations* . . . 187
5.5 **Numerical Example: Oseberg Reservoir** . . . 194
5.5.1 *Deterministic optimization using realization #1* . . . 197
5.5.2 *Bi-objective optimization using 10 realizations* . . . 199
5.6 **Summary and Discussion** . . . 202

CHAPTER 6: **DISCUSSION AND CONCLUSIONS** 203

NOMENCLATURE . . . 205

BIBLIOGRAPHY . . . 205

APPENDIX A: **MINIMUM WELL SPACING CONSTRAINT** . . . 216
APPENDIX B: **MULTI-SEGMENTED WELL TRAJECTORY CALCULATION** . . . 219

APPENDIX C: **EQUIVALENT WELL INDEX CALCULATION** 221

C.1 **Calculation of the Inputs for CMG** 221

C.2 **Two-segmented Perforation within a Gridblock** 225

LIST OF TABLES

3.1 Petrophysical parameters for each facies, Example 1, i.e., selecting $N_w = 8$ wells given $K_{max} = 30$ potential locations in a channelized reservoir. 59

3.2 Perforated layers of the horizontal wells, Example 1. 59

3.3 Fixed BHP's used in the optimization of $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ and initial guesses used in the subsequent control optimization, Bound A, Bound A→B and Bound B for Example 1. 60

3.4 Optimal solutions obtained at different number of generations for the joint optimization of well locations and types using seed 2, Bound A, Example 1. . 62

3.5 Optimal solutions obtained with sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$ using three seeds, Bound A, Example 1. For seed 1 and seed 3 results, wells that are different from the solution of seed 2 are marked in red. 65

3.6 Optimal solutions obtained with sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$ using three seeds, Bound B, Example 1. Injectors are in bold. 65

3.7 Effects of initial guesses and bounds on sequential optimization of well locations and types, drilling order (i.e., $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$) and controls using three seeds, Example 1. 66

3.8 Effects of initial guesses and bounds on simultaneous optimization of well locations and types, drilling order (i.e., $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$) and controls using three seeds, Example 1. 69

3.9 Optimal solutions obtained at different number of generations for the optimization of well locations and types using seed 3, Example 2-A. 73

3.10 Optimal solutions obtained with sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$ using three seeds, Example 2-A. Wells that are different from the solution of seed 3 are in red. 73

3.11 Comparison of the optimal NPV's obtained with optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$ and $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ followed by control optimization, Example 2-A. 75

3.12 Comparison of the optimal NPV's obtained with sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow (\boldsymbol{O})$ followed by control optimization, i.e., $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow (\boldsymbol{O}) \rightarrow (\boldsymbol{u})$, Example 2-A. 76

3.13 Optimal solutions obtained with sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$ using three seeds, Bound A, Example 2-A. Wells that are different from the solution of seed 3 in Table 3.10 are marked in red. 76

3.14 Optimal solutions obtained with sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$ using three seeds, Example 2-B. 80

3.15 Comparison of the optimal NPV's obtained with optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$ and $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ followed by control optimization, Example 2-B. 80

3.16 Comparison of the optimal NPV's obtained with simultaneous optimization of $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ followed by control optimization, i.e., $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O}) \rightarrow (u)$, Example 2-C. 82

3.17 Optimal drilling order within different reservoir lives using the optimal well locations and types $(\boldsymbol{x}, \boldsymbol{T})$ obtained with seed 1, Example 2-B. 84

3.18 Summary of NPV's obtained with drilling order optimization within different reservoir lives using the optimal well locations and types $(\boldsymbol{x}, \boldsymbol{T})$ obtained with seed 1, Example 2-B. 84

3.19 Summary of NPV's obtained with drilling order optimization within different drilling time T_d using the optimal well locations and types $(\boldsymbol{x}, \boldsymbol{T})$ obtained with seed 3, Example 2-A. 86

3.20 Optimal drilling order with different drilling time T_d using the optimal well locations and types $(\boldsymbol{x}, \boldsymbol{T})$ obtained with seed 3, Example 2-A. 86

3.21 Best NPV's obtained by optimizing well locations and types with about 2000 reservoir simulation runs using different population sizes, Example 2-A. . . . 88

4.1 Comparison of the optimal NPV's obtained with three different algorithms using three seeds, PUNQ model. 124

4.2 Comparison of simulation cost obtained with three different algorithms using three seeds, PUNQ model. 126

4.3 Comparison of optimal well numbers obtained with three different algorithms using three seeds, PUNQ model. 126

4.4 Comparison of optimal NPV's obtained with three different algorithms using three seeds, Brugge model. 132

4.5 Comparison of the simulation cost for the three different algorithms using three seeds, Brugge model. 133

4.6 Comparison of optimal well numbers obtained with three different algorithms using three seeds, Brugge model. 133

4.7 Summary of the optimal solutions and computational cost obtained with the three robust optimization procedures using three seeds, robust optimization of PUNQ model. 149

4.8 Summary of the optimal solutions and computational cost obtained with the three robust optimization procedures with GA and GPS, robust optimization of PUNQ model. 150

4.9 Summary of the optimal solutions and computational cost obtained with three robust optimization procedures, robust optimization of the Brugge model. . . 155

4.10 Summary of the optimal solutions and computational cost obtained with the three robust optimization procedures using the iterative sequential method with GA and GPS search steps, robust optimization of Brugge model. 159

5.1 Summary of the trajectories of all 8 wells obtained with the initial and optimal well placements after slanted well trajectory optimization followed by multi-segmented trajectory optimization using seed 3, deterministic optimization using realization #1, channelized model. 184

5.2 Summary of the trajectories of all 8 wells obtained for multi-segmented trajectory optimization using seed 1, deterministic optimization using realization #1, channelized model. 188

5.3 Petrophysical property parameters, Oseberg model. 195

LIST OF FIGURES

1.1 Schematic diagram of closed-loop field development management. 2

2.1 Schematic of GA. 34

2.2 Illustration of the StoSAG search direction. 39

2.3 Illustration of the General Pattern Search (GPS) search direction. 42

3.1 SUS selection for the GA algorithm where A, B, C, D, E, F, G, H represent the ranked individuals in a population with their lengths proportional to their ranks in descending order; F_p represents the sum of ranks for all individuals and N represents the number of parents to keep in the parent pool. 49

3.2 Illustration of the crossover operation for the mixed encoding chromosomes $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$. The red and blue colors represent $(\boldsymbol{x}, \boldsymbol{T})$ of the two parent chromosomes respectively. The green and orange colors represent $(\boldsymbol{O})$ of the two parent chromosomes, respectively. 50

3.3 Illustration of one-point crossover. The red and blue colors represent $\boldsymbol{x}$ of two parent chromosomes respectively. 51

3.4 Illustration of the original one-point crossover and the modified one-point crossover. The red and blue colors represent $\boldsymbol{x}$ of the two parent chromosomes respectively. The yellow color corresponds to a specific gene unit $\boldsymbol{x}_j^b, j = 1, 2, \cdots, N_w$. 51

3.5 The full set of potential well locations and trajectories projected to the horizontal permeability field, Example 1. 59

3.6 NPV versus generations obtained with joint optimization of well locations and types using three seeds, Bound A, Example 1. 62

3.7 NPV versus generations obtained with the optimization of drilling order using three seeds, Bound A, Example 1. 62

3.8 Optimal solutions obtained with sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$, using three seeds, plotted on the horizontal permeability field (upper row) and the oil saturation field (bottom row) of layer 7 at end of 4,000 days of production, Bound A, Example 1. 63

3.9 Optimal solutions obtained with sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$, using three seeds, plotted on the horizontal permeability field (upper row) and the oil saturation field (bottom row) of layer 7 at end of 4,000 days of production, Bound B, Example 1. 63

3.10 Optimal controls for the well locations, types and drilling order obtained with sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$ using seed 1, Bound A, Example 1. . . 66

3.11 Optimal controls for the well locations, types and drilling order obtained with sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$ using seed 2, Bound A→B, Example 1. 67

3.12 Optimal controls for the well locations, types and drilling order obtained with sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$ using seed 2, Bound B, Example 1. . . 67

3.13 Top structure of the Brugge model with the original 30 wells. Left: 3D structure plotted on the initial oil saturation field. Right: 2D depth contour of layer 1 where wells with label 21 to 30 are the locations of the original injectors and wells with label 1 to 20 are the locations of the original producers, Example 2. 69

3.14 The log-horizontal permeability ($\ln k_x$) field and porosity (ϕ) field of the Brugge model, Example 2. 70

3.15 NPV versus generations obtained with the optimization of well locations and types, i.e., $(\boldsymbol{x}, \boldsymbol{T})$, using three seeds, Example 2-A, i.e., selecting $N_w = 12$ wells given $K_{max} = 30$ potential locations. 73

3.16 NPV versus generations obtained with drilling order optimization using three seeds, Example 2-A. 73

3.17 Well configurations and oil saturation fields of layer 1. (a) shows the set of initial potential locations plotted on the oil saturation field; (b)-(d) show the optimal well configurations obtained with sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$ for three seeds, with the corresponding oil saturation fields after 20 years of production on the top layer, Example 2-A. ⊗ represents an injector and o represent a producer. 74

3.18 NPV versus generations obtained with the optimization of well locations and types using three seeds, Example 2-B, i.e., selecting $N_w = 12$ wells given $K_{max} = 64$ potential locations in a channelized reservoir. 79

3.19 NPV versus generations obtained with drilling order optimization using three seeds, Example 2-B. 79

3.20 Well configurations and oil saturation fields of layer 1. (a) shows the set of potential locations plotted on the initial oil saturation field; (b)-(d) show the optimal well configurations obtained with sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$ for three seeds, with the corresponding oil saturation fields after 20 years of production on the top layer, Example 2-B. ⊗ represents an injector and o represent a producer. 80

3.21 Well configurations and oil saturation fields of layer 1. (a) shows the set of potential locations plotted on the initial oil saturation field; (b)-(d) show the optimal well configurations obtained with simultaneous optimization of $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ for three seeds, with the corresponding oil saturation fields after 20 years of production on the top layer, Example 2-C. ⊗ represents an injector and o represent a producer. The first 12 wells drilled are marked in black color and the 8 infill wells are marked in blue color. 82

3.22 NPV versus generations obtained with drilling order optimization within different reservoir lives using the optimal well locations and types $(\boldsymbol{x}, \boldsymbol{T})$ obtained with seed 1, Example 2-B. 84

3.23 NPV versus generations obtained with drilling order optimization with different drilling time T_d using the optimal well locations and types $(\boldsymbol{x}, \boldsymbol{T})$ obtained with seed 3, Example 2-A. 85

3.24 Effect of population size on GA performance from Lobo et al. [51]. 87

3.25 NPV versus generations obtained for the optimization of well types and locations using three seeds, N_p=24, Example 2-A. 87

3.26 NPV versus generations obtained for the optimization of well types and locations using three seeds, N_p=72, Example 2-A. 88

3.27 NPV versus generations obtained for the optimization of well types and locations using three seeds, N_p=108, Example 2-A. 88

3.28 NPV versus iterations for the drilling order optimization process starting from four different initial guesses (Fig. 3 in [44]). 90

3.29 NPV versus iterations for the drilling order optimization of 4 injectors and 8 producers (Fig. 6 in [44].) Trial update using best performing perturbed priority sample is shown in red line where individual samples are shown as gray dots. Trial updates obtained with line search using StoSAG gradient estimated from normal distributed perturbations is shown in blue line. Trial updates obtained with line search using StoSAG gradient estimated from Bernoulli perturbations is shown in green line. 90

3.30 NPV versus generations obtained with simultaneous optimization of $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ using three seeds, Bound A, Example 1. 92

3.31 NPV versus iterations obtained with simultaneous optimization of $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ following the methodology of [44] using three seeds, Example 1. The green square represents a successful update of the best performing perturbation, the blue star represents a successful update along the StoSAG search direction, and the red circle represents a iteration where no successful updates is obtained. 93

3.32 NPV versus generations obtained with simultaneous optimization of $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ using three seeds, Example 2-A. 93

3.33 NPV versus iterations obtained with simultaneous optimization of $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ following the methodology of [44] using three seeds, Example 2-A. The green square represents a successful update of the best performing perturbation, the blue star represents a successful update along the StoSAG search direction, and the red circle represents a iteration where no successful updates is obtained. 93

3.34 Illustration of the logistic function and its derivative in Eq. 3.14. 96

3.35 NPV and number of wells versus simulation runs for the joint optimization of well locations, types and rates obtained with the gradient-based method, Example 1. 101

3.36 NPV versus generations for the joint optimization of well locations, types and rates obtained with GA, Example 1. 101

3.37 Optimal well configurations obtained with the gradient-based method and GA method respectively, plotted on the oil saturation fields of layer 7 at the end of 4,000 days, Example 1. 102

3.38 Optimal well injection/production rates obtained with the gradient-based method and GA, Example 1. 102

4.1 Constraint handling when a well is placed in an inactive gridblock. 107

4.2 The top layer of PUNQ model. Left: the depth of the top layer. Right: the initial pressure of the top layer. The black curve delineates the water-oil contact and the red curve delineates the gas-oil contact. 117

4.3 Locations of 20 wells in the reference scenario (left) and their locations after well placement optimization (right), plotted on the top layer of the PUNQ model. Injector locations are marked in brown color and labeled with ⊗; producer locations are marked in yellow color and labeled with o; the gas cap is marked in green color; the aquifer is marked in dark blue and the oil zone is marked in light blue color. 118

4.4 NPV versus the number of simulation runs in the well placement optimization of the 20 wells in the reference scenario of the PUNQ model. o's represent the StoSAG iterations and $*$'s represent the GPS iterations. 118

4.5 NPV versus number of simulations for the well placement optimization of the reference scenario, using GPS with three different initial mesh sizes, PUNQ model. 119

4.6 NPV versus number of simulation runs during the well control optimization using the optimal well locations of the reference scenario, PUNQ model. The red dashed line represents the optimal NPV's obtained after the well placement optimization. 120

4.7 Optimal controls obtained for the reference scenario with optimal well locations, PUNQ model. 121

4.8 The set of potential locations (marked in yellow and red colors) for the initial population, plotted on the top layer of the PUNQ model. 123

4.9 Comparison of the optimal well numbers obtained with three different algorithms using three seeds, PUNQ model. "W1-(s,T,x) and "W1-x" respectively represent the first-stage and the second-stage optimization of the iterative sequential algorithm, "W2-(s,T,x) and "W2-x" respectively represent the first-stage and the second-stage optimization of the mixed-encoded GA, "W3-(s,T,x) and "W3-x" respectively represent the first-stage and the second-stage optimization of the Genocop III method, and the black dashed line represents the NPV of the reference scenario. 125

4.10 NPV's versus the number of simulation runs obtained with the best seed of each of the three algorithms, PUNQ model. In the iterative sequential algorithm, $*$'s represent the NPV's of current population re-evaluated with the updated locations after a StoSAG search step, $*$'s represent the NPV's of current population re-evaluated with the updated locations after a GPS search step. 125

4.11 Comparison of optimal well numbers obtained with three different algorithms using three seeds, PUNQ model. “W1” represents the iterative sequential algorithm, “W2” represents the mixed-encoded GA, and “W3” represents the Genocop III method. 126

4.12 Optimal well locations obtained with the three algorithms using the best seed, plotted on the top layer of the PUNQ model. 127

4.13 Optimal NPV’s obtained with different drilling cost using the mixed-encoded GA algorithm, PUNQ model. 128

4.14 Optimal number of wells obtained with different drilling cost using the mixed-encoded GA algorithm, PUNQ model. 128

4.15 Locations of the 30 wells in the reference scenario (left) and their locations after well placement optimization (right), plotted on the top layer of the Brugge model. ⊗ represents an injector; o represents a producer. 130

4.16 NPV versus the number of simulation runs in the well placement optimization of the 30 wells in the reference scenario of the Brugge model. o’s represent the StoSAG iterations and $*$’s represent the GPS iterations. 130

4.17 Comparison of optimal well numbers obtained with three different algorithms using three seeds, Brugge model. “W1-(s,T,x) and “W1-x” respectively represent the first-stage and the second-stage optimization of the iterative sequential algorithm, “W2-(s,T,x) and “W2-x” respectively represent the first-stage and the second-stage optimization of the mixed-encoded GA, “W3-(s,T,x) and “W3-x” respectively represent the first-stage and the second-stage optimization of the Genocop III method, and the black dashed line represent the NPV of the reference scenario. 132

4.18 NPV’s versus the number of simulations obtained with the best seed of each of the three algorithms, Brugge model. In the iterative sequential algorithm, $*$’s and $*$’s respectively represent the NPV’s of current population re-evaluated with the updated locations after a StoSAG search step and a GPS search step. 133

4.19 Comparison of the optimal well numbers obtained with three different algorithms using three seeds, Brugge model. “W1” represents the iterative sequential algorithm, “W2” represents the mixed-encoded GA, and “W3” represents the Genocop III method. 134

4.20 Optimal well locations obtained with the three algorithms using the best seed, plotted on the oil saturation field of the top layer after 20 years of production, Brugge model. ⊗ represents an injector; o represents a producer. 134

4.21 Cdf’s of NPV obtained with the reference scenario (black curve) and the optimal well placements for the three robust procedures using three seeds (blue curve for seed 1, red curve for seed 2 and green curve for seed 3), robust optimization of the PUNQ model. The green, blue and red circles respectively represent the P2, P50 and P100 of the cdf of NPV obtained with the reference scenario. 143

4.22 Realizations of the log-horizontal permeability fields (upper row) and the log-vertical permeability fields (bottom row), corresponding to P2, P50 and P100 of the cdf of NPV obtained with the reference well placement scenario, PUNQ model. k_h and k_v represent the horizontal and vertical permeability respectively. 144

4.23 Expected NPV of the susbet ensemble versus number of simulation runs obtained with Robust-1 using three seeds (blue curve for seed 1, red curve for seed 2 and green curve for seed 3), robust optimization of the PUNQ model. The solid lines represent the expected NPV’s of the subset ensemble; the dashed lines in color represent the optimal expected NPV’s of the full ensemble; the black dashed line is the expected NPV of the full ensemble obtained with the reference scenario. 145

4.24 Expected NPV of the full ensemble versus number of simulations obtained for Robust-2 using three seeds, robust optimization of PUNQ model. o represents an update by GA, $*$ represents an update by StoSAG, and □ represents an update by GPS. 146

4.25 Expected NPV versus number of simulations obtained with Robust-2 using seed 2, robust optimization of PUNQ model. o, $*$, and □ respectively represent the expected NPV of the full ensemble obtained using GA, StoSAG and GPS; . and . respectively represent the expected NPV of the subset ensemble obtained using StoSAG and GPS. 146

4.26 Expected NPV of the full ensemble versus number of simulations obtained for Robust-3 using three seeds, robust optimization of PUNQ model. o represents an update by GA, $*$ represents an update by StoSAG, and □ represents an update by GPS. 147

4.27 Optimal types and locations of drilled wells obtained with Robust-1 using three seeds, robust optimization of the PUNQ model. Injector locations are marked in brown color and labeled with $\otimes$; producer locations are marked in yellow color and labeled with o; the gas cap is marked in green color; the aquifer is marked in dark blue and the oil zone is marked in light blue color. 148

4.28 Optimal types and locations of drilled wells obtained for Robust-2 using three seeds, robust optimization of PUNQ model. Injector locations are marked in brown color and labeled with $\otimes$; producer locations are marked in yellow color and labeled with o; the gas cap is marked in green color; the aquifer is marked in dark blue and the oil zone is marked in light blue color. 148

4.29 Optimal types and locations of drilled wells obtained for Robust-3 using three seeds, robust optimization of PUNQ model. Injector locations are marked in brown color and labeled with $\otimes$; producer locations are marked in yellow color and labeled with o; the gas cap is marked in green color; the aquifer is marked in dark blue and the oil zone is marked in light blue color. 149

4.30 Expected NPV of the full ensemble versus number of simulations obtained for Robust-2 using the iterative sequential method with GA and GPS steps, robust optimization of the PUNQ model. o's and □'s respectively represent updates by GA and GPS. 150

4.31 Expected NPV of the full-ensemble versus number of simulations obtained for Robust-3 using the iterative sequential method with GA and GPS steps, robust optimization of PUNQ model. o's and □'s respectively represent updates by GA and GPS. 151

4.32 Cdf's of NPV's obtained with the reference scenario (black curve) and the optimal well placement for three robust optimization procedures, robust optimization of Brugge model. The green, blue and red circles, respectively, represent the P2, P50 and P100 of the cdf of NPV obtained with the reference scenario. 152

4.33 Petrophysical parameters for the top layer of the realizations corresponding to P2, P50 and P100 of the cdf of NPV obtained with the reference well placement scenario, Brugge model. k_h and k_v represents the horizontal and vertical permeability respectively, ϕ is the porosity, NTG is the net to gross ratio and S_w^{con} is the connate water saturation. 153

4.34 Expected NPV of the subset ensemble obtained with the best candidate solution of a population versus number of simulation runs obtained with Robust-1, robust optimization of Brugge model. The solid line in blue represents the expected NPV's of the subset ensemble, the blue dashed line represents optimal expected NPV of the full ensemble, and the black dashed line represents the expected NPV of the full ensemble obtained with the reference scenario. . . 155

4.35 Expected NPV versus number of simulations obtained with Robust-2, robust optimization of the Brugge model. o, ∗ respectively represent the expected NPV of the full ensemble obtained using GA and StoSAG; . represents the expected NPV of the subset ensemble obtained using StoSAG. 156

4.36 Expected NPV of the full ensemble versus number of simulations obtained for Robust-3, robust optimization of the Brugge model. o represents an update by GA, $*$ represents an update by StoSAG, and □ represents an update by GPS. 156

4.37 Optimal types and locations of drilled wells obtained for Robust-1, plotted on the oil saturation field at the top layer of realization #14 after 20 years of production, robust optimization of the Brugge model. $\otimes$ represents an injector and o represents a producer. 157

4.38 Optimal types and locations of drilled wells obtained for Robust-2, plotted on the oil saturation field at the top layer of realization #14 after 20 years of production, robust optimization of the Brugge model. $\otimes$ represents an injector and o represents a producer. 157

4.39 Optimal types and locations of drilled wells obtained for Robust-3, plotted on the oil saturation field at the top layer of realization #14 after 20 years of production, robust optimization of the Brugge model. $\otimes$ represents an injector and o represents a producer. 158

4.40 Cdf's of NPV's obtained with the reference scenario (black curve) and optimal well placement for the three robust optimization procedures with GA and GPS search steps, robust optimization of Brugge model. The green, blue and red circles respectively represent the P2, P50 and P100 of the cdf of NPV obtained with the reference scenario. 158

4.41 The expected NPV of the full ensemble versus number of simulation runs obtained for Robust-2 using the iterative sequential method with GA and GPS steps, robust optimization of Brugge model. o's and □'s respectively represent updates by GA and GPS. 159

4.42 The expected NPV of the full ensemble versus number of simulation runs obtained for Robust-3 using the iterative sequential method with GA and GPS steps, robust optimization of Brugge model. o's and □'s respectively represent updates by GA and GPS. 160

5.1 Illustration of well trajectory parameterization for a slanted and a multi-segmented well. 165

5.2 Illustration of the smoothed trajectory of two consecutive well segments for a multi-segmented well. 166

5.3 Projections of the initial well trajectories on the horizontal permeability field (upper row) and the porosity field (bottom row) of layer 7 for three different realizations, channelized model. 179

5.4 NPV versus number of simulation runs obtained for slanted well trajectory optimization followed by multi-segmented trajectory optimization, deterministic optimization using realization #1, channelized model. The solid lines represent the slanted well trajectory optimization (StoSAG in blue and GPS in red) and the dashed lines represent the subsequent multi-segmented well trajectory optimization (StoSAG in blue and GPS in red). 182

5.5 Cumulative oil production, water production and water injection obtained with the initial and optimal well placements after slanted well trajectory optimization followed by multi-segmented trajectory optimization using seed 3, deterministic optimization using realization #1, channelized model. 183

5.6 Projections of the initial and optimal well placements after slanted well trajectory optimization followed by multi-segmented trajectory optimization on layer 7 using seed 3, deterministic optimization using realization #1, channelized model. 184

5.7 Oil saturation after 3,000 days of production plotted on layer 7, obtained with the initial and optimal well placements after slanted well trajectory optimization followed by multi-segmented trajectory optimization using seed 3, deterministic optimization using realization #1, channelized model. 185

5.8 Projections of the initial and optimal injector perforations obtained with slanted followed by multi-segmented trajectory optimization using seed 3, onto the $x - z$ plane of the horizontal permeability field, deterministic optimization using realization #1, channelized model. 186

5.9 NPV versus number of simulations obtained for multi-segmented trajectory optimization, deterministic optimization using realization #1, channelized reservoir. The blue lines represent the StoSAG algorithm and the red lines represent the GPS algorithm. 186

5.10 Cumulative oil production, water production and water injection obtained for the initial and optimal well placements after multi-segmented well trajectory optimization using seed 1, deterministic optimization using realization #1, channelized model. 188

5.11 Projections of the initial and optimal well placement after multi-segmented trajectory optimization onto layer 7 using seed 1, deterministic optimization using realization #1, channelized model. 188

5.12 Oil saturation after 3,000 days of production plotted on layer 7, obtained for the initial and optimal well placement after multi-segmented trajectory optimization using seed 1, deterministic optimization using Realization #1, channelized model. 189

5.13 Projections of the initial and optimal injector perforations obtained with multi-segmented well trajectory optimization using seed 1, onto the $x - z$ plane of the horizontal permeability field, deterministic optimization using Realization #1, channelized model. 189

5.14 Bi-objective well placement optimization using 10 realizations, channelized model. (a) Average NPV over 10 realizations versus number of simulation runs in primary optimization using StoSAG (blue) and GPS (red); (b) the augmented Lagrangian function, L_A, minimum NPV, J_R, and average NPV, J_E, versus number of simulation runs in secondary optimization using StoSAG (blue) and GPS (red) where L_A is represented by marker x, J_R is represented by the solid line, J_E is represented by the dashed line and $J_E(\boldsymbol{v}_E^*)$ is represented by the dotted line; (c) Cdf's of NPV's for the initial well placement and the optimal well placement after primary and secondary optimization. 191

5.15 Number of violated constraints and indicies of the risk realization during the secondary optimization of the bi-objective optimization of slanted well trajectories using 10 realizations, channelized model. 191

5.16 Cumulative oil production, water production and water injection obtained after the primary and secondary optimization of the bi-objective optimization, channelized model. 192

5.17 NPV versus number of simulation runs for the deterministic optimization using realization #8, channelized model. The blue line represents the StoSAG algorithm and the red line represents the GPS algorithm. 193

5.18 Bi-objective well placement optimization using 10 realizations using a different seed, channelized model. (a) Average NPV over 10 realizations versus number of simulation runs in primary optimization using StoSAG (blue) and GPS (red); (b) the augmented Lagrangian function, L_A, minimum NPV, J_R, and average NPV, J_E, versus number of simulation runs in secondary optimization using StoSAG (blue) and GPS (red) where L_A is represented by marker x, J_R is represented by the solid line, J_E is represented by the dashed line and $J_E(\boldsymbol{v}_E^*)$ is represented by the dotted line; (c) Cdf's of NPV for the initial well placement and the optimal well placement after primary and secondary optimization. . 193

5.19 Top structure and initial oil saturation of the Oseberg model. The tar mat is highlighted with a blue line. 195

5.20 Horizontal permeability (mD, upper row) and porosity field (bottom row) for realization #1, Oseberg. 197

5.21 Horizontal permeability (mD, upper row) and porosity field (bottom row) for realization #8, Oseberg. 198

5.22 NPV versus number of simulations obtained for deterministic optimization of slanted well trajectories using realization #1, Oseberg. The blue line represents the StoSAG algorithm and the red line represents the GPS algorithm. 198

5.23 Oil saturation after 3,900 days of production obtained with initial and optimal well locations, deterministic optimization using realization #1, Oseberg. . . . 199

5.24 Bi-objective well placement optimization using 10 realizations, Oseberg reservoir. (a) Average NPV over 10 realizations versus number of simulation runs in primary optimization using StoSAG (blue) and GPS (red); (b) the augmented Lagrangian function, L_A, minimum NPV, J_R, and average NPV, J_E, versus number of simulation runs in secondary optimization using StoSAG (blue) and GPS (red) where L_A is represented by marker x, J_R is represented by the solid line, J_E is represented by the dashed line and $J_E(\boldsymbol{v}_E^*)$ is represented by the dotted line; (c) Cdf's of NPV for the initial well placements and the optimal well placement after primary and secondary optimization. 201

5.25 Number of violated constraints and indicies of the risk realization during secondary optimization of the bi-objective optimization of slanted well trajectories, Oseberg. 201

5.26 Oil saturation in layer 10 after 3,900 days of production obtained with initial well locations, and optimal well locations after robust optimization and bi-objective optimization, for realization #1 and #7, Oseberg. 202

C.1 Numbering of the Cartesian grids. Numbers in parenthesis on the top surface of each gridblock indicate the (i, j, k) gridblock indicies. 223

C.2 Two-segmented perforation within one simulation gridblock. 225

CHAPTER 1

INTRODUCTION

Closed-loop reservoir management (CLRM) is a general reservoir management framework that consists of a loop of three sequentially performed inter-related tasks during the life of a reservoir: data collection, history matching and optimization [36]. Fig. 1.1 presents the schematic diagram of the CLRM framework. Relevant data, e.g., geological, seismic, well logs and production data can be gathered via different surveillance operations. These data can be assimilated to obtain a posterior reservoir model that better matches the data. The importance of this step is that the new model is expected to have more accurate predictive capability as well as reduce the uncertainty about the reservoir. Based on the calibrated reservoir model, the most beneficial operating strategies can be devised using an optimization algorithm. Typically, in CLRM, only the well control parameters of existing wells are optimized, such as bottomhole pressures, production rates and valve settings.

Our focus in this work is on the optimization step of CLRM where the cost (objective) function is the net present value (NPV) of production over the remaining life of the reservoir (life-cycle optimization) and the design (optimization) variables may include well controls (operating pressure or rates on specified time intervals or control step), well paths, well drilling order, well type (injector or producer) and the number of wells. When the optimization process includes some of the aforementioned key decision variables, it is referred to as field development optimization, rather than simply production optimization. This problem is essentially a mixed-integer problem determining high-dimensional variables which can be either numerical or categorical, continuous or discrete. This research focuses on developing suitable algorithms to solve the field development optimization and its sub-problems under geological uncertainty.

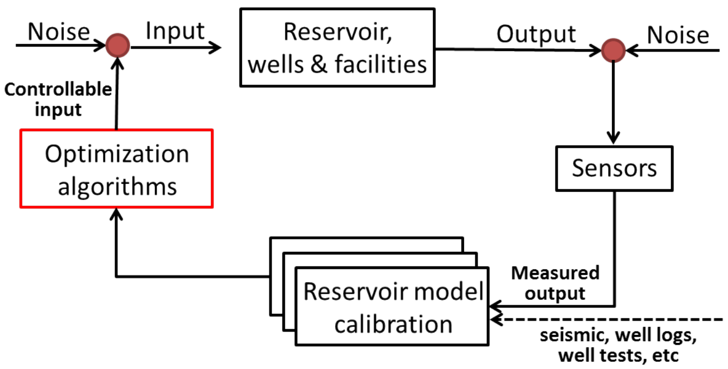

Noise
Input
Reservoir, wells & facilities
Output
Noise
Controllable input
Optimization algorithms
Sensors
Reservoir model calibration
Measured output
seismic, well logs, well tests, etc

Figure 1.1: Schematic diagram of closed-loop field development management.

When geological uncertainty is considered, instead of a single reservoir model, a set of models that are consistent with observed data, geological information and seismic interpretation are obtained in the data assimilation process. Correspondingly, the optimization process has to be adapted to improve the average reservoir performance such as the life-cycle NPV over a set of reservoir models selected to represent the geological uncertainty, which is commonly referred to as robust optimization. Note the average NPV of production over the reservoir life is the standard estimation of expected value and we sometimes see the average NPV referred to as the expected NPV and we may occasionally do so here. Although maximizing the average NPV of life-cycle production is important, the variance in the resulting NPV's of individual models at the computed optimal values of the design variables may be large and in this case, there is considerable downside risk. To mitigate downside risk, one can consider bi-objective optimization [22, 86, 20, 49, 50], or even multi-objective optimization [48]. One common bi-objective optimization formulation is to consider the two conflicting objectives of maximizing the expected value of the life-cycle NPV of production and maximizing the minimum value of the ensemble NPV's [49, 50], and this is the bi-objective optimization considered here.

1.1 Literature Review

Field development optimization is a critical part of the decision making process for a reservoir. Many researchers have studied several individual parts, where the two most typical problems are the well control optimization in which the injection/production rate and/or the bottomhole pressure of wells are optimized and well location optimization in which the well locations and trajectories are optimized. Recently, there has been a growing interest in solving the joint optimization problem of well locations and controls with both sequential and simultaneous approaches. Some researchers also investigated on the field development optimization problem using global search algorithms. Previous procedures in the literature are discussed in this section according to their design parameters and algorithms applied.

The well control optimization problem has been studied extensively in previous literature, using both adjoint gradient-based method [42, 73] and stochastic methods, e.g. PSO [84, 80], CMA-ES [25, 77], EnOpt [13], StoSAG [19, 24], SPSA [47], etc. The adjoint method provides the most accurate gradient and is computationally the most efficient method when the cost function is a differential function of the optimization variables which is the case when the optimization variables are well controls, but is not the case when the optimization variables are well types and number of wells. In addition to lack of differentiability issue, the use of a gradient-based optimization algorithm is precluded if the reservoir simulator used lacks the capability to compute all the gradient information needed which is often the case. When the simulator can compute the needed gradient, one should use adjoint gradient for well control optimization and to some extend for well placement optimization. Of these two types of design variables, the use of adjoint gradient of the NPV with respect to the well placement variables is more tenuous for two reasons. Firstly, unlike the well control variables which explicitly affect the source/sink terms in reservoir simulation equations, the well locations exert their effects on the well index which indirectly controls the value of the source/sink terms. Secondly, the NPV function is rough and becomes non-differentiable when part of a well path crosses a gridblock.

Despite the aforementioned obstacles, there has been attempts to use the adjoint based techniques in the well placement optimization. Sarma et al. [74] considered to optimizing the well locations for fully-penetrating vertical wells under bottomhole pressure control using the adjoint formulation. In their work, the discrete parameters $((i, j)$ well location indices) are replaced with their continuous counterparts in the spatial domain ((x, y) well coordinates) and the discontinuous Dirac-delta function (defining wells as point sources) in the underlying governing PDEs (reservoir simulation equations) is replaced with a continuous function (a bivariate Gaussian function) of the defined well coordinates (x, y). With this modification, the source/sink term becomes a continuous and differentiable function with respect to the continuous well location variables which makes the adjoint formulation possible. The discretization of the modified PDEs yields (additional) non-zero source/sink beyond the original well blocks. The extent and total number of these additional terms are determined by the size of the base of the approximating Gaussian function. In practice, the base is made small enough so that the non-zero source/sink term (pseudo-wells) for gridblocks other than the nearest ones are discarded. In their work, the NPV function is defined based on the well terms for both the original and the additional perforated gridblocks. However, there is an inconsistency when mapping the continuous source/sink back to point source/sink term since they define similar but different dynamics and only keeping the nearest pseudo-wells in the continuous source/sink term may underestimate the point source/sink term.

Forouzanfar and Reynolds [27] used an idea similar to that of Sarma et al. [74] to smooth the objective functions for the placement of vertical, horizontal and deviated wells operating at specified rates or bottomhole pressures. In their work, they perforate the surrounding gridblocks of the well trajectory as well as the gridblocks actually penetrated by the well. However, their representation is only an approximation since they assumed the surrounding gridblocks and the actually penetrated gridblocks have (i) same well indices (ii) same phase mobilities and (iii) same drawdown pressures. These assumptions are not valid if there are abrupt changes in permeabilities, pressures and saturations between the surrounding gridblocks and the actual penetrated gridblocks. They found that the NPV obtained with wells under rate control is better estimated than those under bottomhole pressure control. For rate-controlled wells, the total production rate remains constant while for the bottomhole pressure controlled wells, the total production rate may be overestimated due to the extra perforations in surrounding gridblocks. In their work, the well trajectories are parametrized as the coordinates of the perforation center (x_w, y_w, z_w), the perforation length (l_w), the inclination angle (φ) and the azimuth angle (θ). Instead of using the adjoint framework, they carried out the optimization using a derivative free method, i.e., Bound Optimization by Quadratic Algorithm (BOBYQA) which does not require the NPV function to be differentiable everywhere. One advantage of the BOBYQA is that it uses the reservoir simulator as a "black box," and hence can handle the economic and field operating constraints (e.g., maximum water cut constraint) as reactive constraints in the reservoir simulator.

Volkov and Bellout [82] proposed an efficient approach to approximate the gradient of the NPV term with respect to the well placement design parameters through the adjoint framework using one forward run where the reservoir simulation is performed and the state variables are saved and one backward run where the adjoint system is solved. At each step of the backward simulation run, the adjoint system is formulated with the state variables and solved with respect to the corresponding adjoint variables $\boldsymbol{\lambda}$. The adjoint gradient for the control variables are solved directly using Eq. 1.1.

$$\nabla_{\boldsymbol{u}} J = \frac{\partial J}{\partial \boldsymbol{u}} + (\frac{\partial g}{\partial \boldsymbol{u}})^T \boldsymbol{\lambda}, \tag{1.1}$$

where $\boldsymbol{u}$ is the vector of well control variables, J is the objective function (e.g. the cumulative oil production or the net present value of life-cycle production); g represents the nonlinear residual equation at each gridblock of the reservoir; λ is the adjoint vector. Denote the vector of well location variables as $\boldsymbol{x}$. The gradient components corresponding to the well configuration variables are obtained by a finite difference scheme by perturbing each element x_j individually and evaluating the residual and NPV terms of the perturbed x_j. Unlike analytical derivatives, finite difference may be able to capture the coarse-scale slope of the function in the case of function roughness or discontinuities. However, the suitable perturbation size is case-dependent, which means that parameter tuning has to be performed for each different reservoir model considered and that the obtained perturbation ranges are assumed valid for all feasible well configurations explored by the optimization procedure. In their work, they considered optimization of the placement of both vertical wells and slanted wells. Since the (approximated) adjoint gradient can be obtained for both the control variables and well configuration variables, this approach can handle simultaneous optimization of both terms. They tested the simultaneous optimization on a 2D reservoir with 15 vertical wells under rate control. They did not show the comparison between simultaneous optimization and sequential optimization using the adjoint framework. The well spacing constraints and signed well-to-boundary distance constraints are considered where the derivative of the constraints with respect to the well configuration variables are also obtained by finite difference.

Zandvliet et al. [87] placed eight "pseudo-wells" in the neighboring gridblocks around each actual well. A pseudo-well is a well with a very low rate with negligible effect on the reservoir's overall performance. They used an adjoint method to calculate the gradients of NPV with respect to the small flow rates prescribed for the pseudo-wells. The pseudo-well that gives the largest gradient determines the location of the well in next iteration. Vlemmix et al. [81] applied a similar concept to find the optimum well trajectory for a single horizontal/deviated well. Their method includes creating "pseudo-side-tracks" from each trajectory point. Each pseudo-side-track is assigned a very small perforation length that makes their effect on the overall reservoir performance negligible. These side tracks are then used to find the approximate gradient direction for each trajectory coordinate. A smoothing step is needed after every update of the trajectory coordinates, to ensure the drillability of the well (i.e., the curvature of the well trajectory has to stay below a maximum value allowed by the directional drilling technology and equipment). However, this search direction is not guaranteed to be uphill due to the following reasons: (i) the largest gradient of the NPV with respect to the control settings of the pseudo-wells (pseudo-tracks) is not necessarily positive (ii) the interactions between wells may counter the positive effect expected by increasing the flow rates of each pseudo-well (pseudo-track) individually (iii) the decrease in the flow rate of a main well may have a negative effect on the performance countering the positive effect by increasing the flow rate of its best pseudo-well (pseudo-track). In both of their works, only the well placement variables are optimized.

Some researchers rephrased the well placement optimization problem as the well control optimization problem. Wang et al. [83] proposed to put a well in every gridblock and optimize a modified objective function (NPV minus an approximation of the drilling cost as a function of the injection rate) using the adjoint gradient in order to find the optimal number of injectors, their locations and rates. In their work, for each well, when the injection rate is zero, the approximate drilling cost becomes zero, and this well is eliminated; when the injection rate is large enough, the approximated drilling cost equals the true drilling cost and this well needs to be drilled. They also imposed a total injection rate constraint which helps to eliminate unnecessary wells. However, it is challenging to find a suitable approximation function for the drilling cost and to decide the total injection rate. Zhang et al. [89] speeded up the algorithm proposed in [83] by eliminating more than one injectors at each iteration. Forouzanfar and Reynolds [26] experimented with different approximation functions of the drilling cost, proposed an heuristic initialization procedure to determine the total injection rate and total production rate, and extended the method to the placement of both injectors and producers. However, there is an inconsistency in their objective function and the gradient calculation. The method used in [83], [89] and [26], is only applicable to wells under rate control. Issues with the methodology are that (i) wells eliminated at the current iteration may not be brought back in future iterations; (ii) an unnecessarily large amount of wells may be obtained in the final solution which means the solution may be somewhat suboptimal. Compared to the previously mentioned algorithms which also use the adjoint framework, this method can optimize the number, type and control settings for wells under total liquid rate control, as well as the well locations. However, their method has only been applied to the placement of vertical wells with only one control step and it is difficult to include the well trajectory parameters in the optimization process.

As the adjoint gradient is not generally available, some researchers used the stochastic gradient to approximate the true gradient and thus calculate a search direction. Calculating stochastic gradients does not require the objective function to be differentiable. Hence, there is no need to smooth the objective function or to change the optimization variables to well control variables. However, the advantage of the stochastic gradient procedures comes with the disadvantage that an approximate gradient generally reduces the overall performance of the optimization algorithm. Common stochastic gradient-based methods applied in well placement optimization include SPSA, EnOpt and StoSAG. Bangerth et al. [5] applied an integer variant of the SPSA method to well placement optimization by simply rounding the perturbation size and the step size to the nearest integers. In their work, they considered the placement of 7 vertical wells for a 2D reservoir where the well locations are represented by the (i, j) gridblock indices and compared the optimum results obtained with SPSA, finite different method (FDM) and the very fast simulated annealing (VFSA). They concluded that SPSA was more computationally efficient in finding good locations than VFSA, while VFSA obtained good locations more reliably but with more function evaluations than SPSA; the FDM is less reliable and requires more simulation runs than both SPSA and FDM. Li et al. [47] combined the optimization of the placement of vertical wells and their corresponding rates into a joint problem using a modified SPSA algorithm. The total injection rate and total production rate are constrained to be equal to 1 pore volume per year (linear constraints). In the modified SPSA, the perturbations to the well location variables and well rates are randomly chosen from the full set of feasible directions. For the well placement variables, at each iteration, a random neighbor of the grid block is chosen and the NPV is evaluated for locating a vertical well in this gridblock as well as in the gridblock in the opposite direction, and the well is then moved to the grid block with the higher NPV. In this SPSA variant, the step size of the line search is limited to a single grid block. Li et al. investigated the two common approaches for the joint optimization of well locations and controls, which are namely the simultaneous and sequential approaches. In the simultaneous approach, the vector of optimization variables includes both the control variables and the well trajectory variables. In the sequential approach, the well placement optimization is carried out first with the controls fixed at their initial values, and then based on the optimal well locations obtained, the well control optimization is carried out. According to their paper, simultaneous optimization for vertical wells under rate control outperformed sequential optimization for both deterministic and robust optimization regarding the number of iterations and the optimum NPV's obtained. They showed that in robust optimization, using only a small random subset of model realizations resulted in optimization results similar to those obtained when the entire ensemble of models is used. Jesmani et al. [37] considered two well placement optimization problems where the location of a vertical well is optimized in one problem and the trajectory for a horizontal well is optimized in the other problem, using the original SPSA algorithm. They also investigated an approach to handle geological uncertainty in [47]. Their results suggested that the robust well placement optimization using randomly selected subset of realizations saves significant computational time, while the optimum of the expected NPV remains almost the same as in the case where all realizations are used. They also observed that the SPSA algorithm is a local search method in which the performance is highly dependent on the initial well configuration. Besides that, Jesmani et al. [37] considered well length constraints and the inter-well distance constraint by combining the SPSA algorithm with the decoder method and the projection method respectively; however, no conclusion of the performance of these constraint handling methods have been drawn due to the limited trials.

Chen et al. [12] proposed the ensemble based optimization (EnOpt) for the robust control optimization problem. In EnOpt, at each iteration, (i) for each realization, one perturbation vector is generated and the corresponding NPV is calculated for the perturbed control vector, and (ii) the cross-covariance between the perturbed controls and NPV's is used as the search direction. Fonseca et al. [24] proposed a variant for robust control optimization, i.e., StoSAG, in which the search direction is estimated as a stochastic approximation of the simplex gradient. Fonseca et al. [24] theoretically show that when there is a large variation in the ensemble of reservoir models, StoSAG provides a better approximation of the true gradient than EnOpt. They provide examples to illustrate this theoretical result, i.e., they show, via computational experiments, that StoSAG generally yields a significantly higher value of the life-cycle NPV for robust optimization than is achieved with the standard EnOpt algorithm. Though both StoSAG and EnOpt are designed for robust optimization, i.e., optimization over an ensemble of reservoir realizations characterizing the geological uncertainty, they can be readily adapted to deterministic optimization, see Do and Reynolds [19]. Leeuwenburgh et al. [45] applied EnOpt to deterministic optimization of vertical well placement in a 2D reservoir for up to 9 wells whose locations are represented by the (x, y) coordinates. They concluded that EnOpt works well for the tested deterministic well placement problems. Zhang et al. [90] applied StoSAG to the deterministic well placement and control optimization problem for a 3D channelized reservoir with 13 slanted wells with water-alternating-gas/surfactant-alternating-gas flooding process. The well locations are represented as the (x, y, z) coordinates of the heel point and toe point of a well in the reservoir. He concluded that StoSAG works well for both problems, however, simultaneous optimization did not outperform sequential optimization. Lu et al. [52] improved the StoSAG search direction and proposed updating the well placement variables and the control variables alternatively iteration by iteration, referred to as an iterative simultaneous procedure. Their results showed that the iterative simultaneous procedure outperforms the sequential procedure for deterministic optimization, but these two procedures perform similarly for robust optimization where 10 realizations are used to represent the geological uncertainty. They also developed an efficient StoSAG gradient estimation for bi-objective optimization where the two objectives are to maximize the expected NPV and minimize the minimum NPV among all the realizations representing the geological uncertainty.

Some researchers ([31, 44]) reparameterized the categorical variables (to-drill or not-to-drill, injector or producer, drilling order of wells) as continuous variables and carried out optimization using a search direction estimated by stochastic gradients. Hanea et al. [31] determined the well drilling order, types and times for converting a producer to an injector under geological uncertainty using the Stochastic Simplex Approximate Gradient (StoSAG) in a Quasi-Newton method. For the case of determining drilling order and well type for a given set of well paths, each well is assigned one continuous priority variable (for drilling order) and one continuous type variable. The larger the priority variable, the earlier the well is drilled. The well is an injector if the type variable is greater than 0.5 and otherwise, the well is a producer. However, this procedure can be problematic as perturbing a continuous priority or type variables may not change the actual drilling order or well type which then leads to a low quality stochastic gradient, i.e., a poor search direction. In the extreme case where the perturbation produces no change in drilling order and does not change the type of even one well, the stochastic gradient would be zero even when the current estimate of the optimization variables is far from optimum.

Evolutionary methods have also been tried for optimization problems like those considered in this work. The Genetic algorithm (GA) is one of the most popular methods in well placement optimization. GA can be thought of as a self-adapting system. Bittencourt and Horne [8] developed a hybrid GA and applied it to determine the best well locations and trajectories for a real case of an offshore oil reservoir development project. In their project, geological uncertainty is not considered; the total number and types of wells are pre-specified; wells trajectories can only be vertical or horizontal and if horizontal, wells can take any orientation in the oil zone; no restrictions concerning well spacing and injector/producer patterns are applied. Their work focuses on well placement optimization where the control settings are pre-fixed. The chromosome is formulated as a binary string of the head point coordinates (gridblock indicies), the trajectory type (vertical or horizontal) and the orientation (8 directions) for 33 wells. The length of the well is not optimized either. In the final solution, they found that about five wells were placed very close to injectors, faults or even to other producers which can provide heuristic guidance for reducing the number of wells. Montes et al. [59] studied the effect of several GA parameters such as initial population, population size, and mutation rate on optimizing the well locations for vertical wells. These are important GA parameters that can directly affect the efficiency and reliability of GA convergence to a result close to a global optimum. If these parameters are not selected carefully, GA may require an unnecessary large number of reservoir simulations (for fitness evaluation) or can get trapped around a local optimum. Montes et al. [59] tested a deterministic layercake reservoir with 5 vertical wells where the (i, j) indicies are used to describe well locations. Their results show that the initial population does not have a significant effect on the results. They also conclude that the general rule in GA literature for the optimum population size (twice the number of bits used in a chromosome) applies for their problem too. However, it is not clear that this conclusion applies when the number of variables becomes very large. Their results are not conclusive about the mutation rate. Yeten et al. [86] determined the optimal location and trajectory of a non-conventional well (heel and toe point of the main bore, number of junctions and number of laterals emanating from the same junction, lateral trajectory, the wellbore diameter and the total liquid production rate) using the genetic algorithm with a binary encoding. In their work, the trajectory of both the main bore and the laterals are described by the coordinates of the entry point (h_x, h_y, h_z), the length of the trajectory projected onto the x-y plane l_{xy}, the orientation of the well in the x-y plane θ, and the depth to the trajectory endpoint t_z. Both deterministic optimization and robust optimization are tested. In robust optimization, 10 realizations are used represent the geological uncertainty and the objective function is defined as a weighted sum of the average NPV and the standard deviation. In all their test cases, the maximum number of junctions on the main bore is set to be 4 and maximum number of laterals emanating from the same junction is set to 1. Emerick et al. [21] used the Genetic Algorithm for Numerical Optimization of Constrained Problems (Genocop III) to determine the number, types, locations and trajectories of slanted wells subject to various constraints for a deterministic reservoir. In their work, the chromosome is composed of both binary variables (well type, i.e., injector or producer, and well status, i.e., open or shut-in) and integer variables ((i, j, k) gridblock indices of the heels and toes for each well), but treated as a string of real numbers. In their work, an arithmetic crossover is used to make linear combinations of a randomly selected parent pair and the genes of the produced offsprings are truncated to the nearest integers. They also mutate the locations variables, the type variables and the status variables in different ways and at different mutation probabilities. Emerick et al. [21] considered both linear and nonlinear constraints where any individual in the population has to be repaired before its evaluation if it is infeasible. These constraints include maximum well length, minimum distance between wells and user-defined undesirable regions for well placement. In their work, they are able to test different cases with up to a maximum number of 16 wells within 5000 simulations. Based on three full-field reservoir models, (i) they have shown significant increases in the NPV and cumulative oil production using Genocop III over the well placement scenarios proposed by reservoir engineers; (ii) the optimization defining the whole initial population randomly achieved a value of NPV lower than the results using the engineer's suggested case in the initial population. Lee et al. [43] used GA to optimize the number, location and trajectory of horizontal wells using a 2D node-based configuration. In their method, a horizontal well can have multiple kick-off points. However, they only tested their methodology on two small 2D synthetic reservoirs (one with 10 × 10 gridblocks and one with 20 × 20 gridblocks) and considered the placement of three horizontal wells at maximum. Though their estimated optimal NPV's are larger compared to placing three vertical wells, increasing the number of kick-off points gives negligible increase in the estimated optimal NPV's. Morales et al. [60] used GA to optimize the heel and toe coordinates of a horizontal well in a reservoir with 30 × 30 × 11 grids considering 5 different realizations of the reservoir. In their method, the fitness value is calculated as the weighted sum of the NPV's of realizations whose NPV's are among the top p_s percent and they showed that the trajectories of the horizontal well obtained with different p_s values have significant changes. Salam et al. [72] proposed to calculate the fitness of each candidate solution using a proxy trained by artificial neural networks (ANN) and to evolve to next generation following the GA operators. They optimized the placement of 10 vertical producers and their conversion time from production into injection respectively and showed that the hybrid algorithm of GA and ANN only requires a few hundred simulation runs for the two problems they considered. Besides GA, some other methods have also been tried, PSO [65, 66], simulated annealing [6], CMA-ES [18, 25], and pattern search algorithms [7, 33, 32].

Bellout et al. [7] proposed a nested joint optimization algorithm where the upper level optimization problem is the well placement optimization using Pattern Search Algorithms, i.e., Generalized Pattern search (GPS), Hooke-Jeeves Direct Search (HJDS) or a hybrid optimization parallel search package (HOPSPACK, a distributed computing implementation of GPS, and the lower level is the well control optimization using the adjoint gradient. In the sequential optimization (well placement optimization with fixed controls followed by well control optimization using the optimal locations obtained), two control strategies are considered in the well placement optimization: a fixed control strategy where injectors operate at their upper bounds of the bottomhole pressure and producers operate at their lower bounds of the bottomhole pressure, and a "reactive" control strategy where injectors operate at their upper bounds of the bottomhole pressure and producers operate at their lower bounds of the bottomhole pressure until a prescribed limit to the maximum water cut is reached. In their work, they consider a 2D reservoir with 5 vertical wells and 10 control intervals, based on which the nested joint optimization algorithm outperformed sequential optimization with both aforementioned control strategies. From their results, GPS and HOPSPACK are slightly faster than HJDS, but they yield lower NPV values than HJDS. However, GPS and HOPSPACK can be accelerated, in terms of clock time, if a cluster is available. Li and Jafarpour [46] used an iterative procedure in which they alternated between a full well placement optimization and a full well control optimization. Again, different optimization approaches were used for the two problems, where a coordinate descent random search algorithm is used for the well placement optimization and the adjoint gradient-based method is used for the well control optimization. Their work pertains to vertical wells under total liquid rate control. Though the number of control steps is only provided for one case with five wells, they tested cases with up to 15 wells. For all cases, they were able to achieve significant improvements over a single well placement followed by well control optimization. As we mentioned previously, Lu et al. [52] used a modified iterative procedure based on StoSAG to optimize well trajectories and controls for 13 slanted wells with 12 control steps, and found that the iterative procedure also outperformed the procedure of a single well placement followed by well control optimization. The above mentioned works essentially all decompose the joint problem into two smaller subproblems. This enables the application of a specialized optimization method for each of the two subproblems.

Li et al. [47] combined the optimization of vertical well placements and their corresponding rates into a joint problem using a modified SPSA algorithm where the total injection rate and total production rate are constrained to be 1 pore volume per year (linear constraints). The results they obtained for the PUNQ model showed that simultaneous optimization outperformed sequential optimization. Isebor et al. [34] applied a hybrid technique that combined the PSO and the mesh adaptive direct search (MADS) for the joint optimization problem of the example considered by Bellout et al. [7], but with only 5 control steps. Besides that, they also developed a consistent filter-based treatment ([23]) of nonlinear constraints in both stand-alone PSO and the PSO-MADS hybrid. In their work, they handled three types of nonlinear constraints (minimum distance between wells, maximum field rate, and maximum well water cut) whereas in Bellout et al. [7] only bound constraints are considered. Bellout et al. [7] required about 4,000 simulations for each of their optimization runs for 10 location variables with 50 control variables, whereas Isebor et al. [34] used around 30,000 simulations for each run where there are 10 location variables and 25 optimization variables. Forouzanfar et al. [25] implemented the CMA-ES optimization algorithm to solve both the simultaneous and sequential joint optimization problem of slanted wells where they kept the parametrization of well trajectories and introduced a new parametrization strategy for the control variables. Their numerical results for PUNQ model showed that simultaneous optimization algorithm gave slightly better NPV's than sequential optimization framework. In summary, Li et al. [47], Isebor et al. [34] and Forouzanfar et al. [25] all show that the simultaneous approach outperforms the sequential approach. However, the alternate conclusion has been reached by others. Specifically, Humphries et al. [33] used a hybrid algorithm that combines PSO and GPS for simultaneous optimization, which is an approach similar to the one used by Isebor et al. [34]. In their sequential optimization implementation, PSO is used for the well placement optimization and GPS is used for the well control optimization. Humphries et al. [33] did not find the simultaneous approach to consistently provide better solutions than the sequential procedure. Humphries et al. [32] later applied their method to joint optimization for non-conventional wells with different complexity of trajectories and suggested that the sequential approach is better able to deal with increasingly complex well parameterizations than the simultaneous approach, and for their complex problems where there are 24 trajectory variables (4 slanted wells) and 20 control variables (5 control intervals), compared to simultaneous optimization, sequential optimization gives 5% higher NPV's under bound constraints and 9% higher NPV's under nonlinear constraints.

Most of the well control and placement optimization problems mentioned above assume the number, types and drilling orders of the wells to be drilled are known which is not the case most of the time. Hence, there is a strong need to research on the full field development optimization problem where the number of wells, their types, drilling sequences, locations and control settings are optimized altogether. The full field development optimization problem determines both categorical variables (drill-or-not, injector or producer, and well drilling sequence) and numerical variables (well locations and control settings). Hence, as a mixed-integer problem, it is far more complicated than the joint optimization of well locations and controls. Recall that Wang et al. [83], Zhang et al. [89], and Forouzanfar and Reynolds [26] rephrased the field development optimization problem as a well control optimization problem where the drilling cost is defined as a smooth function of the injection/production rate and optimize the number, type, locations and control settings for wells under total liquid rate control. Ciaurri et al. [14] proposed to solve the field development optimization problem by well pattern optimization followed by a second-stage well placement optimization which provides a minor change of the well locations within the patterns using PSO method. Although this method is more computationally tractable, it suffers from three drawbacks: firstly, it is only applicable to newly discovered reservoirs where no wells has been drilled; secondly, the control setting of the wells has to be pre-specified; thirdly, this method theoretically will lead to suboptimal results since wells are constrained to repeated patterns. Awotunde [2] optimized the well number, types, locations and controls using differential evolution (DE). He proposed to divide the search space of the control variables at the first control step (or the coefficient of a polynomial/trigonometric parametrization of the control variables) into three intervals (indicating an injector, no-well and a producer respectively), referred to as well control zonation method. Hence, the “drill-or-not” status and “injector/producer” type of a well can be determined from the control variables. The settings of the no-well zone has a large effect on the algorithm performance and its optimal settings are case dependent. Based on the well control zonation method, Awotunde [1] introduced two additional design variables, the project life and the fractional times at which the wells are brought online, and he is able to simultaneously optimize well types, locations, controls and well open time. However, the well controls over the reservoir life were assumed to follow a Cosine function of time and wells were brought online without considering the rig availability. Awotunde [3] compared the well control zonation method and the mixed-integer non-linear programming (MINLP) procedure using DE and his results showed the former method outperforms the latter in most cases provided that the no-well zone is set as large as or larger than the combined width of the two other zones. Khan et al. [41] extended the well control zonation method to include the vertical/horizontal well type in addition to the injector/producer type and used PSO as the optimizer. They found that the well control zonation method outperforms the MINLP method, which is consistent with the observations of Awotunde [3].

Isebor et al. [35] introduced a ternary categorical variable (where the three levels represents injector, do-not-drill and producer) in addition to the locations and control settings and then optimized these variables using PSO-MADS. To handle the integer and ternary variables, they simply round the variables to the nearest integer value before evaluation in the PSO search step and use a discrete mesh in the MADS poll step. They noted that both PSO and MADS perform satisfactorily for continuous problems, but with the inclusion of integer variables, they may converge to relatively poor local optima. What is more, the convergence of PSO-MADS with multi-level categorical variables (the ternary variable) is not clearly understood. Shirangi and Durlofsky [76] extended the work of [35] to closed loop field development optimization under geological uncertainty where the history matching process is solved with the Random Maximum Likelihood (RML, [64]) method using the adjoint gradient and the optimization process is carried out using the PSO-MADS method. In the work of Shirangi and Durlofsky [76], the well locations are represented by the (i, j, k) gridblock indices of their two end points and the number of control steps per well is set equal to the number of wells, N_w, with the length of the first $N_w - 1$ control steps equal to the specified constant drilling time. Based on a deterministic 2D reservoir with four vertical wells where each new well is drilled every 210 days, they demonstrated that simultaneous optimization (optimizing the well types, locations, time-varying controls of these four wells simultaneously) obtained 16% higher NPV than well-by-well optimization where the location and type of the first well is optimized first and given the optimal location and type of the first well, the location and type of the second well (drilled at 210 days) are optimized, and so on. This simple case took 200 equivalent simulation runs (number of times a batch of simulation runs is submitted to the compute cluster with a maximum of 400 available cores), which equals 80,000 actual simulation runs. They solved the full problem of closed-loop field development for a 2D reservoir with a maximum number of 8 vertical wells and 8 control steps with 50 realizations used to represent the geological uncertainty. However, due to the heavy computational cost of PSO-MADS, they proposed a sample validation procedure (optimizing over an representative subset, e.g. 10 realizations) in the field development optimization. This case took 1700 equivalent simulation runs using a cluster with a maximum of 400 cores, i.e., approximately 350,000 actual reservoir simulation runs. A 3D example with 6 wells are also presented where they fix the number, type and drilling sequences of wells and pre-specify whether a well is vertical or horizontal. Although the procedure using PSO-MADS may be computationally tractable for realistic field cases using parallelization with a large number of cores, there is a concern about whether simultaneous optimization will still outperform the well-by-well procedure with an increase of the problem complexity (e.g. considering geological uncertainty, optimizing number of wells and trajectories for complicated wells) and number of variables (optimizing over more wells and using more control steps), according to the findings in [32].

Yeten et al. [86], Essen et al. [22], Durlofsky et al. [20], Liu and Reynolds [49], Lu et al. [52] have all considered the optimization process using a set of model realizations representing the geological uncertainty. However, only a small ensemble is used in robust optimization, which may not be able to properly capture the subsurface uncertainty. The computational cost for a large ensemble of realizations may be prohibitive. Thus, it is significant to develop strategies to improve computational efficiency for large ensemble size.

Yang et al. [85] proposed conducting robust optimization on a small subset of representative realizations which is selected based on a reliable ranking of the calculated performance of all the realizations under a reference operating condition. However, the ranking of reservoir performance changes if the well operational controls are updated in the optimization process and in fact can change significantly so that the chosen subset no longer gives an adequate representation of the full ensemble [53]. To further reduce the computational cost, Yang et al. also replaced the reservoir simulation process with a trained response surface model. Li et al. [47] evaluated the objective function with a subset of randomly selected realizations at each iteration of the SPSA algorithm in order to achieve the desired computational efficiency. Although SPSA is able to handle a noisy objective function, there is no guarantee that selected subsets can represent the full set. Shirangi and Durlofsky [76] proposed reselecting the representative realizations based on the solutions obtained with a previous complete subset optimization, and repeat the subset optimization until a prespecified relative improvement ratio is reached, i.e., 0.5 in their work. In their work, the relative improvement ratio is defined as the increase in the expected NPV for the subset of realizations (based on the initial expected NPV for the subset) over the increase in the expected NPV for the full set of realizations (compared to the initial expected NPV in the full set). However solving a complete optimization problem for each subset is expensive, and the obtained solution is not guaranteed to be the optimal solution of the full-set.

Many researchers have worked on multi-objective optimizations of the well controls. For both deterministic and roust optimization, it is desirable to maximize the long-term NPV and the short-term NPV at the same time so that the invested money can be payed back sooner and hence can be invested on other projects without significantly harming the long-term benefit. For cases considering geological uncertainty, it is desirable to maximize the expected NPV and to minimize the risk (i.e., minimize the standard deviation of the NPV's of all realizations or maximize the minimum NPV). Essen et al. [22] proposed an adjoint-based hierarchical production optimization method in which the secondary objective (maximization of the short-term NPV) is optimized based on the null space of the Hessian matrix of the primary objective (maximization of the long-term NPV). They observed a significant increase in short-term objectives without significantly decreasing the life-cycle NPV. However, the authors point out that the convergence of the method is slow, due to infeasible solution steps. They also introduced an alternative method, referred to as switching method in which the long-term NPV is not treated as a strict constraint. Chen and Reynolds [11] carried out a robust short term control optimization after long-term control optimization subject to the constraint that the average long-term NPV should not be decreased using the Augmented Lagrangian method for waterflooding examples and showed that robust long-term alternating with short-term optimization is able to increase the short-term NPV without compromising the life-cycle NPV. Liu and Reynolds [49] built the Pareto front of maximizing the long-term NPV and maximizing the short-term NPV using an adjusted weighted sum method and an normal boundary intersection method with adjoint gradient for an optimal well control problem of a waterflood example. Liu and Reynolds [49] found that both methods gave similar estimates of the Pareto front where the solution points distribute fairly evenly. Yeten et al. [86] and Durlofsky et al. [20] used a simple treatment of uncertainty by incorporating the standard deviation of the NPV's for all realizations into the objective function with the framework of GA to maximize the expected NPV and to minimize the risk over an ensemble of realizations. However, Liu and Reynolds [49] observed the undesired feature that the reduction in the standard deviation of the NPV's is achieved mostly by reducing the highest plausible NPV's. Thus, they propose to maximize the minimum NPV instead of minimizing the standard deviation in bi-objective optimization where the primary objective is to maximize the average NPV. However, the computational cost of finding the Pareto front may be expensive. What is more, for a problem with rough objective function surface and multiple local solutions, e.g. the joint optimization of well control settings and well placement, our experience suggests that it may be difficult to obtain a good approximation of the Pareto front. Hence, it is much cheaper and practical to simply find a reasonable trade-off instead of the whole front. Liu and Reynolds [50] combined lexicographic method with adjoint based gradient to solve a bi-objective well control optimization problem where the two objectives are optimized in a sequence according to the order of importance in order to find a reasonable trade-off solution.

1.2 Research Objectives and Contextures

Formulating field development planning as an optimization problem is slowly becoming an acceptable approach in the petroleum industry. Many researchers have studied several individual parts, where the two most typical problems are the well control optimization which has been extensively studied in the literature and well placement optimization which is continuing to gain more and more attention.

As limited adjoint capability is available in commercial simulators, we focus our attention on a gradient-based algorithm where the true gradient is approximated by a stochastic gradient when optimizing well controls and well trajectories. Specifically we use the StoSAG (stochastic simplex approximate gradient) to replace the true gradient in a steepest ascent (or descent) algorithm. We consider both deterministic optimization based on a single reservoir model and robust optimization when we maximize the average net present value of productions over a set of reservoir models selected to represent the geological uncertainty in reservoir model parameters.

For general field development optimization, we will also need to incorporate some combinations of the following variables: the number of wells that will be drilled, the well types (injector or producer) and the drilling order. Generally, there is a strong need to conduct research on the complete field development problem where the number of wells, their types, drilling order, locations and control settings are optimized altogether. Since the complete field development optimization is in fact a mixed-integer programming problem, different algorithms (e.g. evolutionary algorithms), other than (stochastic) gradient-based algorithms, will likely need to be applied. Previous works solve the field development optimization problem in two ways. The first way re-parameterizes the categorical design variables as continuous variables (i.e., controls [83, 89, 26, 3]) or priorities [31, 44]), uses optimizers designed for continuous variables to estimate their optimal value and then map the estimated continuous variables back to the categorical variables. According to Lu and Reynolds [54], this type of methods not only enlarge the search space, but also give sub-optimal solutions. The second way parameterizes the categorical variables as they are and uses a hybrid of global and local search optimizers, i.e., PSO-MADS, where at each iteration, the categorical variables are again treated continuous and then truncated back to their nearest discrete values after an update, which does not seem ideal. As part of our research, we will attempt to develop suitable encodings for variables of different types and use the Genetic algorithm (GA), which handles the categorical variables more naturally, as the optimizer. Since it is well known that GA is computationally expensive, we also attempt to improve its efficiency by either decoupling the problem or combining GA with stochastic gradient methods.

Geological uncertainty, which leads to the uncertainty in the performance prediction, has a relevant effect on the decision-making process regarding reservoir development. The common method is to use multiple plausible geostatistical realizations when evaluating the objective function. Given a large ensemble of realizations which reasonably covers the uncertainty of the model parameters, the primary objective function of robust optimization will be to maximize the average NPV over all realizations efficiently. However, when the number of realizations of the reservoir model is large, computational feasibility requires that one reduces the number of models on which robust optimization will be preformed, i.e., to determine a representative subset of the the whole ensemble so that robust optimization on the subset will produce an expected value of NPV close to the one obtained via robust optimization on the full ensemble. Previous work on well placement and control optimization problems uses a subset of representative models equally spaced on the cumulative distribution function of NPV's of all realizations evaluated at a reference operating condition where the subset size varies from 3 [85] to 50 [16] depending on the user preference. Most of the previous work on the field development optimization problem, due to its complexity, uses only one reservoir model [83, 89, 26, 14, 2, 1, 3, 41, 35], except Shirangi and Durlofsky who used a subset of 10 realizations of a 2D synthetic reservoir with 56 optimization variables at the cost of around 350,000 simulations which is hardly affordable without parallel computing. However, it is difficult to specify the reference operating condition and it seems obvious that it is highly unlikely that the selected realizations will remain representative during the whole optimization process. As part of our research, we will attempt to develop a procedure to select an initial representative subset of reservoir models from the full set that represents geological uncertainty and to update this subset during the optimization so that at the end of robust optimization, the optimal parameter for the subset will also be at least weakly optimal for the complete set of reservoir models. Robust optimization maximizes the average NPV of production over a plausible set of reservoir models. However, the set of optimal design variables obtained by maximizing average NPV may produce widely varying values of NPV when their optimization variables are applied to each reservoir model. Hence, we also consider to achieve a secondary objective to reduce the risk which is to maximize the minimum NPV or minimize the standard deviation of the NPV's over the given ensemble.

The objectives of this research and our associated contributions are as follows:

- Our first objective was to explore the applicability of StoSAG algorithm in well placement optimization of slanted wells and multi-segmented wells considering minimum well spacing, maximum well length and drillability constraints. We firstly proposed suitable parameterizations for slanted wells and multi-segmented wells and then coupled the augmented Lagrangian method with StoSAG to solve the constrained well placement optimization problem. Though StoSAG has been shown to be efficient for well placement optimization problems [52, 30], we found the solutions obtained by StoSAG are only sub-optimal and added a subsequent General Pattern Search (GPS) method to fine tune the well locations and trajectories.

- A second objective was to develop improved methods for the field development optimization problem. Here, we consider two problems. The first one has not been considered previously in any journal publications where we want to select a fixed number of wells from a given set of potential drilling paths and to determine their types, drilling order and control settings. The second is a more general problem where we optimize well number, types, locations and control settings. We decouple the control optimization from both problems and optimize well control settings using StoSAG at the end to reduce problem complexity and to be computationally feasible. For the first problem, we developed different encodings for well locations, types and drilling order, and used GA with mixed operators which is shown to outperform the methods of [83, 89, 26] and [31, 44] in this work. For the second problem, we develop two different hybrids of GA, StoSAG and GPS to efficiently optimize well number, types and locations, which are able to consider a much larger problem compared to Isebor et al. [35] and are shown to significantly outperform the Genocop III method by [21].

- A third objective was to improve computational efficiency for robust optimization when a large number of realizations are used to characterize the uncertainty. Here, we developed a robust procedure to adaptively choose representative subsets from the full ensemble of reservoir models during the optimization process. The adaptive procedure outperforms the case of using a fixed subset throughout the optimization process, and only requires a few thousand simulations in contrast to a few hundred thousand simulations required for Shirangi and Durlofsky [76].

- In addition, we consider bi-objective optimization within the context of robust optimization. Here, we consider the bi-objective well placement optimization problem where the two objectives are to maximize the average NPV and maximize the minimum NPV. Following the lexicographic method, the two objectives are optimized in a sequence where the average NPV is optimized first, and starting from the optimal solution obtained, the minimum NPV is optimized subject to an additional constraint that the average NPV is not decreased by 1%. The constrained optimization problem is solved using augmented Lagrangian method where an efficient implementation of StoSAG and GPS are developed to solve the inner loop problem. Compared to previous work [86, 22, 11, 49, 50], this piece of work extends the bi-objective optimization from the optimal control problem to the optimal well placement problem whose objective function surface is rougher.

- Our final contribution was to embed the optimization methodology in usable software so that it can be applied to field problems.

1.3 Dissertation Outline

There are six chapters in this dissertation. Chapter 1 gives a brief review of the literature pertaining to various optimization problems in oil field development, our initial research objectives and our research contribution. In Chapter 2, the generalized field development optimization problem is described and the methodologies considered in this dissertation are discussed, including Genetic Algorithm (GA), Stochastic Simplex Approximate Gradient method (StoSAG) and Generalized Pattern Search Algorithm (GPS). In Chapter 3, we present the first major contribution of our work, a viable solution to select a fixed number of wells from a given set of potential drilling paths and to determine the well types, drilling order and control settings based on the Brugge model and a channelized reservoir model. In Chapter 4, we present the second major contribution of our work, a viable solution to simultaneous optimization of well number, types and locations under assumed well controls for both the Brugge model and the PUNQ model. Here, we also present an efficient robust optimization procedure given a large number of realizations to represent the geological uncertainty. In Chapter 5, we present an advanced parameterization of slanted and multi-segmented well trajectories and a bi-objective well placement optimization process to maximize the average NPV and minimize risk given multiple realizations of the reservoir using lexicographic method. Chapter 6 summarizes our work and presents conclusions.

CHAPTER 2

COMPUTATIONAL ALGORITHMS FOR FIELD DEVELOPMENT OPTIMIZATION

In this chapter, we firstly present the mathematical description of a general field development optimization problem, including the definition of the objective function, the parameterization of the design variables and the related constraints. In this dissertation, the field development optimization is solved using hybrids of three base algorithms, i.e., Genetic Algorithm (GA), Stochastic Simplex Approximate Gradient (StoSAG) and General Pattern Search (GPS). GA is introduced to handle the categorical variables naturally, StoSAG is mainly used to optimize well locations and controls, and GPS is mainly used to fine tune the well locations obtained by StoSAG. The three computational algorithms are described respectively in the second part of this chapter while their hybrids are presented in the following three chapters according to the problems considered.

2.1 Problem Description

The field development optimization considers maximizing the net present value (NPV) over the presumed life of the reservoir to find the optimal number of wells, the corresponding drilling order, types, locations and control settings. In this section, we present a general field development optimization problem but this dissertation focuses on solving various subsets of this general problem. Here, the general field development optimization problem is defined by

$$\max \quad J\left(\boldsymbol{s}, \boldsymbol{T}, \boldsymbol{O}, \boldsymbol{x}, \boldsymbol{u}\right) \tag{2.1a}$$

$$\text{s.t.} \quad c(\boldsymbol{s}, \boldsymbol{T}, \boldsymbol{O}, \boldsymbol{x}, \boldsymbol{u}) \leq= 0, \tag{2.1b}$$

$$s_j \in \{0,1\} \quad \forall\, j = 1, \cdots, K_{max}, \tag{2.1c}$$

$$T_j \in \{0,1\} \quad \forall\, j = 1, \cdots, K_{max}, \tag{2.1d}$$

$$\boldsymbol{x} \in \Omega, \tag{2.1e}$$

$$\boldsymbol{u}^{\text{low}} \leq \boldsymbol{u} \leq \boldsymbol{u}^{\text{up}}, \tag{2.1f}$$

where K_{max} defines the maximum number of wells to be drilled; $\boldsymbol{s}$ is a K_{max}-dimensional categorical variable vector where the jth element, s_j, is a binary variable defining whether well j is to be drilled or not, i.e., $s_j = 0$ corresponds to "not-to-drill" and $s_j = 1$ corresponds to "to-drill"; $\boldsymbol{T}$ is a K_{max}-dimensional categorical variable vector where its jth element, T_j, is a binary variable defining the type of well j, i.e., $T_j = 0$ corresponds to "injector" and $T_j = 1$ corresponds to "producer"; $\boldsymbol{O}$ is a sequence of ordinal number, i.e., a permutation of $\{1, 2, \cdots, K_{max}\}$, where the jth element, O_j, represents well j is drilled as the O_jth well; $\boldsymbol{x}$ is a $bpl * K_{max}$-dimensional vector of the well location parameters, where bpl is the number of parameters representing the location of each well; Ω represents the potential set of locations and/or paths that wells can take; $\boldsymbol{u}$ is a $N_c * K_{max}$-dimensional vector of the well control parameters where N_c is the number of control steps (here N_c is assumed to be the same for all wells), where $\boldsymbol{u}$ includes the injection/production rates of wells under rate control or bottom-hole pressure of wells under pressure control at each control step; $\boldsymbol{u}^{\text{low}}$ and $\boldsymbol{u}^{\text{up}}$ are the lower and upper bounds of the control settings respectively; $c(\boldsymbol{s}, \boldsymbol{T}, \boldsymbol{O}, \boldsymbol{x}, \boldsymbol{u})$ represents the linear/non-linear constraints, where the notation $\leq=$ means the constraints could be either of the form $c \leq 0$ or $c = 0$. Possible constraints include bounds on field production or injection rates, balanced injection and production constraint, maximum water cut constraint, or minimum well spacing constraints, etc.

The NPV for a given vector $\boldsymbol{m}$ of the reservoir model parameter is denoted by $J\left(\boldsymbol{m}, \boldsymbol{s}, \boldsymbol{T}, \boldsymbol{O}, \boldsymbol{x}, \boldsymbol{u}\right)$ and defined by,

$$J\left(\boldsymbol{m},\boldsymbol{s},\boldsymbol{T},\boldsymbol{O},\boldsymbol{x},\boldsymbol{u}\right)=\sum_{n=1}^{N_t}\left\{\frac{\Delta t^n}{(1+b)^{\frac{t^n}{365}}}\left[\begin{array}{c}\sum_{j=1}^{N_P}\left(r_o^n\cdot\overline{q}_{o,j}^n-c_w^n\cdot\overline{q}_{w,j}^n-c_g^n\cdot\overline{q}_{g,j}^n\right)\\-\sum_{k=1}^{N_I}\left(c_{wi}^n\cdot\overline{q}_{wi,k}^n+c_{gi}^n\cdot\overline{q}_{gi,k}^n\right)\end{array}\right]\right\}-\sum_{j=1}^{K_{max}}s_jC_{w,j}, \tag{2.2}$$

where n denotes the nth time step of the reservoir simulation; N_t is the total number of time steps; t^n is the simulation time at the end of the nth time step; Δt^n is the nth time step size; N_P and N_I denote the number of producers and injectors, respectively, which depend on the "drill or not" variables represented by the components of $\boldsymbol{s}$ and the components of the type indicator vector $\boldsymbol{T}$; $\overline{q}_{o,j}^n$, $\overline{q}_{w,j}^n$, $\overline{q}_{g,j}^n$, respectively, denote the average oil production rate (STB/day), the average water production rate (STB/day) and the average gas production rate (Mscf/day) at the jth producer over the nth simulation time step; $\overline{q}_{wi,k}^n$ and $\overline{q}_{gi,k}^n$ denote the average water injection rate (STB/day) and the average gas injection rate (Mscf/day) over the kth injector for the nth time step; r_o^n is the oil price (\$/STB); c_w^n is the water disposal cost (\$/STB); c_g^n is the gas disposal cost (\$/Mscf) (negative if gas is sold); c_{wi}^n and c_{gi}^n are the water injection cost (\$/STB) and gas injection cost (\$/Mscf), respectively; $C_{w,j}$ denotes the drilling cost for the jth well which depends on the trajectory variables defining this well; and b is the annual discount rate. In this work, we use a commercial reservoir simulator to simulate the dynamic performance of the flooding process. The NPV function of Eq. 2.2 is computed using the output of a reservoir simulation run based on $(\boldsymbol{s},\boldsymbol{T},\boldsymbol{O},\boldsymbol{x},\boldsymbol{u})$ and a specific plausible reservoir model $\boldsymbol{m}$.

The problem described in Eq. 2.1 can be extremely complicated when $(\boldsymbol{s},\boldsymbol{T},\boldsymbol{O},\boldsymbol{x},\boldsymbol{u})$ are all considered as design variables even for the deterministic case where the reservoir model $\boldsymbol{m}$ is assumed known. Previous work in the literature focuses on solving subsets of Eq. 2.1 where some design variables are assumed known. Only a few works (e.g., Hanea et al. [31] and Leeuwenburgh et al. [44]) have considered drilling order optimization. When $(\boldsymbol{s},\boldsymbol{T},\boldsymbol{O},\boldsymbol{x})$ is known, Eq. 2.1 describes an optimal control problem with a smooth objective function and continuous optimization variables which can be solved efficiently using gradient-based algorithms, given the availability of a reservoir simulator that provides the relevant adjoint solutions needed to calculate the gradient of $J(\boldsymbol{m}, \boldsymbol{u})$ efficiently [42, 73, 10]; otherwise, one can use methods based on stochastic gradients, e.g. ensemble optimization (EnOpt) [13], stochastic simplex approximate gradient (StoSAG) method [24], simultaneous perturbation stochastic approximation (SPSA) [47]. Another possibility is to use a derivative free algorithm such as particle swarm optimization (PSO) [84, 80], covariance matrix adaptation-evolutionary strategy (CMA-ES) [25, 77] and Genetic Algorithm (GA) [86]. When $(\boldsymbol{s}, \boldsymbol{T}, \boldsymbol{O}, \boldsymbol{u})$ is known, Eq. 2.1 describes an optimal well placement problem where the objective function may not be differentiable for all value of $\boldsymbol{x}$; in fact, the optimization variables can be either continuous or discrete depending on how the well trajectory is parameterized. For vertical wells, the well location is usually defined by the discrete (i, j, k) indices of the gridblocks penetrated by a well [5, 7, 47] where in most papers the well is assumed to be completely vertically penetrating. Emerick et al. [21] used the (i, j, k) gridblock indices of the starting point and end point to describe slanted well trajectories. In this case, the design parameters are again discrete variables since gridblock indices can only take integer values. However, for slanted wells, the well trajectory can also be parametrized as continuous variables, for example using the (x, y, z) coordinates of the center point of each well, the perforation length l, the inclination angle φ and the azimuth angle θ [66, 27]. However, this parametrization requires a special scaling treatment for the optimization process if the (stochastic) gradient-based method is used (Forouzanfar and Reynolds [27]). Vlemmix et al. [81] parameterized a single multi-segmented well as the (x, y, z) coordinates for a set of trajectory points where the number of trajectory points is equal to the number of gridblocks through which the trajectory passes. However, a smoothing step is needed after every update of the coordinates of the trajectory points, to ensure the drillability of the well. Yeten et al. [86] optimized the well trajectory for a multilateral unconventional well where the main bore is described by coordinates of the entry point (h_x, h_y, h_z), the length of the trajectory projected onto the x-y plane l_{xy}, the azimuth angle θ, and the depth to the trajectory endpoint t_z; and the lateral is described by the coordinates of the junction point (there can be multiple laterals from one junction, the number of junctions and the number of laterals emanating from each junction are to be determined). In this case, the optimization variables are mixed discrete and continuous variables. There currently exists no adjoint code to calculate the gradient of the NPV function with respect to the well location variables. Gradient-based algorithms using stochastic gradient and derivative free algorithms are widely used. When $(\boldsymbol{s}, \boldsymbol{T}, \boldsymbol{O})$ is known, Eq. 2.1 reduces to the problem of jointly optimizing well controls and well locations. The problem of optimizing well locations leads to a rougher objective function surface (Forouzanfar and Reynolds [28]) than the one for the problem of optimizing only well controls. For problems involving rough objective functions, gradient-based algorithms generally work poorly although a gradient-based algorithm tends to be highly efficient for the pure well control optimization problem. Thus, it may be advantageous to use a sequential algorithm to jointly optimize well controls and trajectories where a gradient-based algorithm is used for the well controls and an algorithm with a better capability for finding global optima of a noisy objective function is employed for the well trajectory optimization. Another reason for decoupling the joint optimization problem into two subproblems, i.e., well control optimization and well placement optimization is that, in practice, well control optimization is not conducted until all wells have been drilled. A third reason is that when all the optimization variables $(\boldsymbol{s}, \boldsymbol{T}, \boldsymbol{O}, \boldsymbol{x}, \boldsymbol{u})$ need to be determined, Eq. 2.1 is in fact a mixed-integer problem with both continuous and categorical variables, referred to as the full field development optimization problem. In this case, the gradient (or stochastic gradient) based method is no longer suitable since the gradient with respect to the categorical variables is not defined. As discussed below, some authors have tried to replace the categorical variables as continuous variables, but it is not clear that such a procedure can yield optimal results.

Only a few papers have included well status, types and drilling order in addition to well paths as design variables [35, 76, 2, 31, 44]. The first two papers (Isebor et al. [35] and Shirangi et al. [76]) assign each of the K_{max} wells a label, preset their drilling order and opening time, and optimize the well status and type (-1 represents a producer, 0 represents do-not-drill and 1 represents an injector), locations and controls for these K_{max} wells. Even though the well status and types are considered as ternary variables, at each iteration of the Particle Swarm Optimization (PSO), the ternary variable are treated as continuous variables and then truncated back to the nearest integer within the set $\{-1, 0, 1\}$. What is more, to switch the drilling order of well i and well j $(i \neq j)$, the locations and controls for well i and j have to be switched which is quite difficult for PSO or MADS when well i and well j are far away from each other. Moreover, even if well i is not drilled (i.e., well status of well i corresponds to do-not-drill), it still holds (reserves) T_d days of drilling time (i.e., the drilling rig does not work for the next T_d days) and because of the "rig down time," the solution obtained must be at least somewhat suboptimal. In the last two of these papers, well status is not optimized while well type and drilling order were determined using continuous variables which they refer to as priority variables (see section 1.1 for an explanation of priority variables). As well status, types and drilling order are discrete variables, treating them as continuous variables and then truncating back to discrete values at each iteration of an optimization algorithm, does not seem ideal. Instead, we believe that it makes more sense to treat them directly as discrete categorical variables which enables the use of the genetic algorithm.

Probably one of the most important concerns in the decision-making process based on the optimization results would be the uncertainty in the reservoir geological description. The most common approach for handling geological uncertainty is to generate multiple reservoir descriptions and do robust optimization in which, the expected value of NPV is optimized. As noted previously, we use the terminology expected value whereas the expected value is actually replaced by its standard estimator, i.e., the average of the NPV where the average is over the set of reservoir models (vectors of reservoir model parameters) $\boldsymbol{m}_k, k = 1, 2, \cdots, N_e$ which represent the uncertainty. Thus, the expectation of NPV is defined by

$$J_E\left(\boldsymbol{s}, \boldsymbol{T}, \boldsymbol{O}, \boldsymbol{x}, \boldsymbol{u}\right) = \frac{1}{N_e} \sum_{k=1}^{N_e} J(\boldsymbol{m}_k, \boldsymbol{s}, \boldsymbol{T}, \boldsymbol{O}, \boldsymbol{x}, \boldsymbol{u}) \tag{2.3}$$

Robust optimization is much more expensive than deterministic optimization in Eq. 2.1 since each objective function evaluation requires N_e simulation runs where N_e generally ranges from ten to one, or two hundreds. There are at least three major concerns with robust optimization, one is that maximizing the expected objective function over the given ensemble still may incur considerable downside risk [49], and a second is that optimization over a large set of realizations of $\boldsymbol{m}$ is generally not computationally feasible for field development optimization problems and a third concern is that life-cyle robust optimization may yield unacceptably low short-term production. The last issue is not dealt with here, but is generally addressed by bi-objective optimization (Essen et al. [22], Chen and Reynolds [11] and Liu and Reynolds [49]).

2.2 Basic Methodology Considered

The general problem stated here is a mixed-integer problem with a high-dimensional search space which includes binary variables ("drill-or-not" status and types), ordinal variables (drilling order), numerical discrete variables (locations for vertical wells) and numerical continuous variables (well control settings). Considering that (i) GA handles categorical variables naturally; (ii) the Stochastic Simplex Approximate Gradient (StoSAG) based method has been proven to be efficient in optimizing well locations and controls, but searches relatively poorly when close to a local optimum due to gradient estimation error, and (iii) General Pattern search method (GPS) is a good local search method once we are close to an optimum, but is computationally expensive for high-dimensional optimization problems, we focus on developing hybrids of these three main algorithms when solving different aspects of the optimization problem defined in Eq. 2.1.

The performance of an algorithm (e.g. steepest descent and general pattern search) may depend crucially on scaling. For example, if the producers are under bottomhole pressure control and the injectors are under rate control, changes to BHP may produce much larger variations in the NPV than do similar changes to injection rate; same thing may happen to the coordinates of the well end points along the (x, y) directions and the z direction. Hence, if the optimization problem is poorly scaled, we introduce a new variable vector which has similar magnitude in all its components and then solve the optimization problem in terms of the new variable vector. In this dissertation, for all the optimal well control problem and the well placement optimization of slanted wells and multi-segmented wells, scaling is adopted; see Chapter 5 for detailed explanation.

2.2.1 Genetic Algorithm (GA)

The genetic algorithm (GA) is a metaheuristic algorithm inspired by the process of natural selection which is often used to solve global optimization problems. GA evolves iteratively starting from a population of randomly generated individuals through a set of operators: selection, crossover, mutation and replacement which are designed to move the population away from a local optima where a traditional hill climbing algorithm might get stuck. Each individual in the population corresponds to a set of decision variables (or scaled variables), which in GA terminology is referred to as a chromosome or genotype, denoted as $\boldsymbol{c}$. The parameterization of the decision variables is referred to as encoding. Traditionally, solutions are represented as strings of binary variables, but other encodings are possible, i.e., permutation encoding and real value encoding. Different encodings, crossover and mutation strategies are used to account for different types of variables (e.g. well "on/off status," well types, drilling order, locations and controls), depending on the specific problem considered.

In each generation, the fitness of every individual in the population is evaluated and a parent pool is selected from the current population by a stochastic procedure in which the more fit individuals are likely to be selected. Next, a new generation is formed based on recombining and mutating chromosomes of the individuals from the parent pool. Furthermore, the best chromosome will randomly replace one individual in this new generation. The new generation of candidate solutions forms the population from which the parent pool will be selected at the next iteration of the algorithm. This generational process is repeated until a termination condition is reached. Common termination conditions include specifying a fixed number of generations based on available computational resources or terminating GA when iterations (generations) no longer produce better results. A flow chart of GA is shown in Fig. 2.1 and a general description of GA using binary encoding as an example is provided in Algorithm 1.

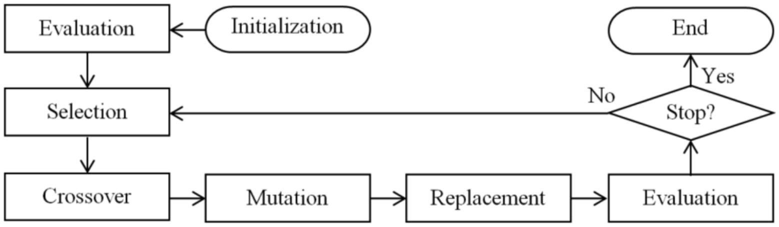


Figure 2.1: Schematic of GA.

Algorithm 1 Algorithm outline for GA with binary encoding

1. Preset population size N_p, maximum number of generations N_g, crossover rate p_c and mutation rate p_m.
2. Initialization: generate N_p feasible individuals with chromosome $\boldsymbol{c}$ randomly sampled from $\{0, 1\}$.
3. Evaluation: evaluate the objective function of Eq. 2.2 for each individual in the current population.
4. Selection: select N_p individuals from current population to form a parent pool.
5. Crossover: recombine the candidate solutions from the parents pool with a probability of p_c using a modified one-point crossover to produce offsprings.
6. Mutation: the produced offsprings invert bits at positions randomly chosen with probability p_m.
7. Termination criteria check. If number of generations exceeds N_g, terminate; otherwise, go to step 3.

Usually in Genetic Algorithms (GA), the crossover probability is set fairly high, 0.6 to 0.9, and mutation probability of a gene is kept far smaller, 0.001 to 0.01. A higher crossover probability helps to explore the solution space more globally while a small crossover probability may slow down the evolution process. A very small mutation rate may lead to genetic drift (which is non-ergodic in nature) while a high mutation rate may lead to a loss of good solutions, unless elitist selection is employed. Finding an appropriate population size is a difficult task. If it is too small, GA may not be able to reach high quality solutions. If it is too large, GA converges so slowly that it mat require an infeasible number of forward model runs (reservoir simulation runs in this work) in order to obtained good optimum. It has been shown, both theoretically and empirically, that the optimal size differs from problem to problem. A somewhat widely-accepted rule of thumb on selecting the population size is that it should be set proportionally to the problem's size and difficulty. However, problem difficulty is very hard to estimate for real-world problems. The general rule in GA literature for the optimum population size is to set it equal to two times the number of bits used in a chromosome [59]. However, this rule of thumb is computationally expensive for a high-dimensional optimization problem, e.g. problems considered in this work, and we find that good results can be generated with a considerably smaller population size.

2.2.2 Stochastic Simplex Approximate Gradient (StoSAG)

In this section, we simply use $\boldsymbol{u}$ to denote the optimization variables. For the well control optimization problem, $\boldsymbol{u}$ represents the (scaled) well control settings of all wells at all control steps (e.g well bottomhole pressure, well injection/production rate); for the well placement optimization problem, $\boldsymbol{u}$ refers to the (scaled) well trajectory parameters for all wells; for the simultaneous optimization problem, $\boldsymbol{u}$ includes both the (scaled) well control settings and the (scaled) well placement variables. Following the formulation of the gradient of Eq. 2.3, the gradient of J_E is equal to the sum of the gradient of NPV calculated for each model, i.e.,

$$\nabla_{\boldsymbol{u}} J_E(\boldsymbol{u}) = \frac{1}{N_e} \sum_{k=1}^{N_e} \nabla_{\boldsymbol{u}} J(\boldsymbol{u}, \boldsymbol{m}_k). \tag{2.4}$$

In Eq. 2.4, the $\nabla_{\boldsymbol{u}}$ is used to represent the true gradient and $\nabla_{\boldsymbol{u}} J_E(\boldsymbol{u})$ represents an uphill direction, i.e., if $\boldsymbol{u}^\ell$ is the estimate of the optimal value $\boldsymbol{u}^*$ obtained at iteration ℓ, $J_E(\boldsymbol{u}^\ell)$ increases in the direction $\nabla_{\boldsymbol{u}} J_E(\boldsymbol{u}^\ell)$. Thus the steepest ascent algorithm is given by

$$\boldsymbol{u}^{l+1} = \boldsymbol{u}^\ell + \alpha_\ell \boldsymbol{\nabla}_{\boldsymbol{u}} J_E(\boldsymbol{u}^\ell), \quad \text{for } \ell = 0, 1, \cdots, \tag{2.5}$$

where $\boldsymbol{u}^0$ is the initial guess and α_ℓ is the step size which can be chosen by a line search procedure. In the StoSAG algorithm, $\nabla_{\boldsymbol{u}} J(\boldsymbol{u}^\ell, \boldsymbol{m}_k)$ is approximated by a stochastic gradient denoted by $\boldsymbol{d}_k^\ell$ and $\nabla_{\boldsymbol{u}} J_E(\boldsymbol{u}^\ell)$ in Eq. 2.5 is replaced by

$$\boldsymbol{d}^\ell = \frac{1}{N_e} \sum_{k=1}^{N_e} \boldsymbol{d}_k^\ell, \tag{2.6}$$

so the steepest ascent algorithm of Eq. 2.5 is replaced by

$$\boldsymbol{u}^{l+1} = \boldsymbol{u}^\ell + \alpha_\ell \frac{\boldsymbol{d}^\ell}{\|\boldsymbol{d}^\ell\|_\infty}, \tag{2.7}$$

for $\ell = 0, 1, \cdots$, until convergence. α_ℓ is commonly chosen by backtracking where a maximum allowable step size, α_0, is usually set between $0.1[\boldsymbol{u}^{\text{up}} - \boldsymbol{u}^{\text{low}}]$ and $0.2[\boldsymbol{u}^{\text{up}} - \boldsymbol{u}^{\text{low}}]$ to enable a large initial trial update, and a maximum allowable number of step size cuts is usually set as 5.

To generate a simplex gradient of $J(\boldsymbol{u}^{\boldsymbol{\ell}}, \boldsymbol{m}_k)$, we need to generate different perturbations of $\boldsymbol{u}^\ell$ for each $\boldsymbol{m}_k$. We define the perturbed vector by $\hat{\boldsymbol{u}}_{j,k}^\ell$, $j = 1, 2, \cdots, N_{pert}$ where the hat denotes a perturbed vector and the subscript j denotes the jth perturbation vector for the realization $\boldsymbol{m}_k$. Next, we define the $N_u \times N_{pert}$ matrix $\Delta \boldsymbol{U}_k^\ell$ and the N_{pert}-dimensional column vector $\Delta \boldsymbol{J}_k^\ell$ respectively by

$$\Delta \boldsymbol{U}_k^\ell = [\hat{\boldsymbol{u}}_{1,k}^\ell - \boldsymbol{u}^\ell, \hat{\boldsymbol{u}}_{2,k}^\ell - \boldsymbol{u}^\ell, \cdots, \hat{\boldsymbol{u}}_{N_p,k}^\ell - \boldsymbol{u}^\ell],$$

and

$$\Delta \boldsymbol{J}_k^\ell = [J(\hat{\boldsymbol{u}}_{1,k}^\ell, \boldsymbol{m_k}) - J(\boldsymbol{u}^\ell, \boldsymbol{m_k}\boldsymbol{)}, J(\hat{\boldsymbol{u}}_{2,k}^\ell, \boldsymbol{m_k}) - J(\boldsymbol{u}^\ell, \boldsymbol{m_k}\boldsymbol{)}, \cdots, J(\hat{\boldsymbol{u}}_{N_p,k}^\ell, \boldsymbol{m_k}) - J(\boldsymbol{u}^\ell, \boldsymbol{m_k}\boldsymbol{)}]^T.$$

Then the approximate simplex gradient, $\boldsymbol{d}_k^\ell$, in Eq. 2.6 is given by

$$\boldsymbol{d}_k^\ell = (\Delta \boldsymbol{U}_k^{\ell,+})^T \Delta \boldsymbol{J}_k^\ell, \tag{2.8}$$

where throughout the superscript + denotes the Moore Penrose pseudo-inverse which can be obtained using singular value decomposition, N_{pert} represents the total number of perturbations for each realization $\boldsymbol{m}_k$, and N_u represents the dimension of the design variables vector. The perturbations can be generated by any method, for example, we can use

$$\hat{\boldsymbol{u}}^{\ell}_{j,k} = \boldsymbol{u}^{\ell} + L\boldsymbol{Z}^{\ell}_{j,k} = \boldsymbol{u}^{\ell} + \boldsymbol{C}_U^{1/2}\boldsymbol{Z}^{\ell}_{j,k}, \tag{2.9}$$

where, $\boldsymbol{Z}^{\ell}_{j,k}$ is a N_u-dimensional vector of independent standard random normal deviates and $\boldsymbol{L}$ is the lower triangular matrix of the Cholesky decomposition of a specified covariance matrix $\boldsymbol{C}_U$ which controls the perturbation sizes and correlations between the components of the perturbed vectors [9]. $\boldsymbol{C}_U$ is a block diagonal matrix with N_w blocks and each block, $\boldsymbol{C}_U^w$, corresponds to the correlation matrix over the N_c control steps of well w where each element, $C_U^{i,j,w}$ is given by

$$C_U^{i,j,w} = \sigma^2 \left[1 - \frac{3}{2}\left(\frac{|t_i - t_j|}{N_s}\right) + \frac{1}{2}\left(\frac{|t_i - t_j|}{N_s}\right)^3 \right], \tag{2.10}$$

where, σ is the standard deviation of the perturbations, e.g., 0.03 for the normalized variables in the scale of [0,1], i and j represent the ith and jth control steps respectively, $i = 1, 2, \cdots, N_c$, $j = 1, 2, \cdots, N_c$, t_i and t_j represent the times corresponding to the middle of ith and jth control step interval respectively, and N_s is the length of time over which we wish the controls of well w to be temporally correlated.

When $N_e = 1$, the optimization process is referred to as deterministic optimization where N_{pert} is usually set to a number ranges from 5 to 50. Do and Reynolds [19] used only 5 to 10 for the deterministic case where there is only a single reservoir model, but Fonseca et al. [24] found that we may need far more than 50 perturbations in order to obtain a stochastic gradient which has a direction fairly close to the true gradient. When we consider robust optimization where the number, N_e, is such that $N_e \geq 10$, following Fonseca et al. [24], we use only one perturbation per reservoir model. Because the stochastic gradient of $J_E(\boldsymbol{u}^{\ell})$

is only an approximation of the true gradient, the search direction $\boldsymbol{d}^l$ may not point uphill from $\boldsymbol{u}^\ell$, and if not, it may not be possible to obtain a higher value of $J_E(\boldsymbol{u}^\ell)$ by searching in the direction $\boldsymbol{d}^l$. When this occurs, we may try to find an uphill direction by generating new perturbations and computing new stochastic gradient, $\boldsymbol{d}^l$. In our implementation, if we cannot improve the expected NPV after five trials, the StoSAG algorithm is terminated. The StoSAG algorithm is summarized in Algorithm 2 and illustrated in Fig. 2.2. In Fig. 2.2, we suppose there are only two wells (one injector under rate control and one producer under bottomhole pressure control) and only one control step, the $x-$axis represents the injection rate and the $y-$axis represents the production BHP, the 2D contour map represents the NPV (for illustration only). In Fig. 2.2, lines in red and blue respectively represent the gradient direction and the StoSAG direction at different estimates of the design variables, where in this schematic, the angle between the gradient direction and the StoSAG search direction is less than 90°, i.e., the StoSAG direction is in an uphill search direction.

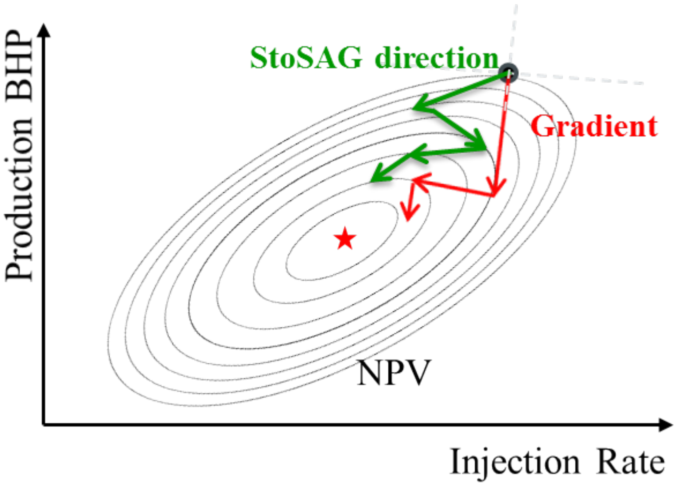


Figure 2.2: Illustration of the StoSAG search direction.

For the case of a single reservoir model, the expectation of the StoSAG search direction $\boldsymbol{d}^\ell$ is the true gradient $\nabla_{\boldsymbol{u}} J$ with an error term $O\left(\max\{\|\delta \boldsymbol{u}^i\|_2^2\}_{i=1}^{i=N_{pert}*N_e}\right)$ [24]. Hence, the larger is the perturbation size, the larger is the error term. When optimizing the well place-

Algorithm 2 Outline for the StoSAG Algorithm

```
Initialize the parameters: $n_{sc},\ n_{cuts} = 0,\ N_{res},\ is = 0,\ \alpha_\ell = \alpha_0$, where $n_{sc}$ is the maximum
    allowed number of step size cuts, $n_{cuts}$ is the number of step size cuts, $N_{res}$ is the
    maximum number of the consecutive search direction re-computation, $is$ is the number
    of search direction re-computation, $\alpha_0$ is the maximum step size.

1. Generate $N_{pert} * N_e$ samples of perturbation at current estimate $\boldsymbol{u}^\ell$ and compute the
    search direction with Eq. 2.6. Set $is = is + 1$.

2. Compute a trial update $\boldsymbol{u}^{trial,\ell+1}$ using Eq. 2.5 (truncate each element of $\boldsymbol{u}^{trial,\ell+1}$ to its
    nearest integer for discrete variables).

3. Check if the trial update is acceptable.
    • If $J_E(\boldsymbol{u}^{trial,\ell+1}) > J_E(\boldsymbol{u}^\ell)$,
        – set $\boldsymbol{u}^{\ell+1} = \boldsymbol{u}^{trial,\ell+1}$, $\ell = \ell + 1, is = 0, n_{cuts} = 0, \alpha_{\ell+1} = \alpha_0$ and go to step 1.
    • Otherwise,
        – set $n_{cuts} = n_{cuts} + 1$,
        – if $n_{cuts} > n_{sc}$ and $is > N_{res}$, terminate the optimization algorithm,
        – if $n_{cuts} > n_{sc}$ and $is < N_{res}$, then go to step 1,
        – if $n_{cuts} \leq n_{sc}$, set $\alpha_\ell = \alpha_\ell/2$, then go to step 2.

4. Terminate if the maximum number of simulations or iterations are used.
```

ment of vertical wells where the location variables ((i, j) gridblock indicies) are discrete, the ±1 Bernoulli distribution is often used to generate random perturbations of the current estimate of the well locations. In this circumstance, it is possible for $O\left(\max\{\|\delta \boldsymbol{u}^i\|_2^2\}_{i=1}^{i=N_{pert}*N_e}\right)$ to be quite large. Meanwhile, the objective function surface of the well placement problem is quite rough compared to the well control optimization problem, and this can lead to a poor StoSAG performance. In this sense, we may not be able to obtain a result close to optimum.

2.2.3 General Pattern Search (GPS)

General Pattern Search (GPS) [15] is a class of direct search methods, originally introduced by Torczon [79]. Each iteration consists of an optional search step and a poll step. Both steps evaluate the objective function at a finite number of trial points on a mesh M_ℓ in order to find an improved mesh point. In this work, M_ℓ is given by

$$M_\ell = \left\{ \boldsymbol{u}^\ell \pm \Delta_\ell \boldsymbol{e}_i : i \in \{1, 2, \cdots, N_x\} \right\}, \tag{2.11}$$

where, $\boldsymbol{u}^\ell$ is the incumbent point at the lth iteration, Δ_ℓ is the mesh size at lth iteration, N_x is the number of elements in $\boldsymbol{u}^\ell$, and $\boldsymbol{e}_i$ represents the unit vector where the ith element of $\boldsymbol{e}_i$ is 1. The difference between search and poll steps is in the way the trial points are selected. Any strategy, including none, can be used in the search step, and it is often tailored to the particular application to obtain specific efficient algorithms. Since GPS is used to fine tune the well locations obtained by StoSAG, we only consider the poll step in this work. Our implementation of GPS is described below.

There are N_x coordinate directions in total and each coordinate direction is associated with two objective function evaluations, $J(\boldsymbol{u}^\ell + \Delta_\ell \boldsymbol{e}_i)$ and $J(\boldsymbol{u}^\ell - \Delta_\ell \boldsymbol{e}_i)$. If both $J(\boldsymbol{u}^\ell + \Delta_\ell \boldsymbol{e}_i)$ and $J(\boldsymbol{u}^\ell - \Delta_\ell \boldsymbol{e}_i)$ are smaller than $J(\boldsymbol{u}^\ell)$, then we move on to the $(i+1)$th coordinate direction. If $J(\boldsymbol{u}^\ell + \Delta_\ell \boldsymbol{e}_i) > J(\boldsymbol{u}^\ell)$ and $J(\boldsymbol{u}^\ell + \Delta_\ell \boldsymbol{e}_i) > J(\boldsymbol{u}^\ell - \Delta_\ell \boldsymbol{e}_i)$, then incumbent point $\boldsymbol{u}^\ell$ is updated, i.e., $\boldsymbol{u}^{\ell+1} = \boldsymbol{u}^\ell + \Delta_\ell \boldsymbol{e}_i$ and consecutive trial updates $\boldsymbol{u}^{\ell+1} = \boldsymbol{u}^{\ell+1} + k\Delta_\ell \boldsymbol{e}_i,\ k = 1, 2, \cdots, 5$ on the mesh M_ℓ along $\boldsymbol{e}_i$ direction are taken until objective function stops increasing. Similarly, if $J(\boldsymbol{u}^\ell - \Delta_\ell \boldsymbol{e}_i) > J(\boldsymbol{u}^\ell)$ and $J(\boldsymbol{u}^\ell - \Delta_\ell \boldsymbol{e}_i) > J(\boldsymbol{u}^\ell + \Delta_\ell \boldsymbol{e}_i)$, then incumbent point $\boldsymbol{u}^\ell$ is updated, i.e., $\boldsymbol{u}^{\ell+1} = \boldsymbol{u}^\ell - \Delta_\ell \boldsymbol{e}_i$ and consecutive trial updates $\boldsymbol{u}^{\ell+1} = \boldsymbol{u}^{\ell+1} - k\Delta_\ell \boldsymbol{e}_i,\ k = 1, 2, \cdots, 5$ on the mesh M_ℓ along $-\boldsymbol{e}_i$ direction are taken until objective function stops increasing. One should note, in the original GPS algorithm, Δ_ℓ is initially set large to search globally and Δ_ℓ shrinks size to search locally whenever a local optimizer on M_ℓ is found, i.e., when no trial update that improves the objective function is found after searching along N_x coordinate directions. GPS algorithm is only considered to converge when Δ_ℓ is smaller than a user-defined minimum allowable mesh size. The value of the initial mesh size Δ_0 depends on the actual problem to be solved. For example, in the well placement optimization of vertical wells, when well locations are represented by the (i, j) gridblock indicies, we can start from a small mesh size (e.g., $\Delta_0 = 1$ gridblock) if the GPS method is used to fine tune the well locations after the StoSAG optimization or start from a larger mesh size (e.g., $\Delta_0 = 5$) if

GPS method is used as the standalone optimizer. The way to shrink the mesh size Δ_ℓ is also problem dependent. For example, for the case where the mesh size has to be a positive integer, one can use $\Delta_\ell = \Delta_\ell - 1$; otherwise if the mesh size is a continuous variable, one can use $\Delta_\ell = \Delta_\ell/2$. For detailed implementation of GPS in this work, please see Algorithm 3. Fig. 2.3 shows an illustration of the GPS algorithm for the well control optimization where there is one injector under liquid rate control and one producer under BHP control considering one control step.

Algorithm 3 General Pattern Search Algorithm

```
Set $\ell = 0$, $idx = 0$, $FL_{GPS} = 0$ and $\Delta_0$. Initialize the design variable $\boldsymbol{u}^\ell$, set the initial
    mesh size $\Delta_\ell = \Delta_0$ and the minimum allowable mesh size $\Delta_{\min}$. Set the maximum
    allowable consecutive trial updates along a coordinate direction $\alpha_0 = 5$, create the
    mesh $M_\ell = \{\boldsymbol{u}^\ell \pm \Delta_\ell \boldsymbol{e}_i : i \in \{1, 2, \cdots, N_x\}\}$.
Do
    • Set $idx = idx + 1$, set $i$ equal to the modulus of $idx$ divided by $N_x$.
    • Evaluate $J(\boldsymbol{u}^\ell \pm \Delta_\ell \boldsymbol{e}_i)$.
      If $J(\boldsymbol{u}^\ell + \Delta_\ell \boldsymbol{e}_i) > J(\boldsymbol{u}^\ell)$ or $J(\boldsymbol{u}^\ell - \Delta_\ell \boldsymbol{e}_i) > J(\boldsymbol{u}^\ell)$,
        – set $FL_{GPS} = 0$ and $k = 0$.
        – If $J(\boldsymbol{u}^\ell + \Delta_\ell \boldsymbol{e}_i) > J(\boldsymbol{u}^\ell - \Delta_\ell \boldsymbol{e}_i)$, set $\boldsymbol{u}^{\ell,k} = \boldsymbol{u}^\ell + \Delta_\ell \boldsymbol{e}_i$, $\boldsymbol{d}_\ell = \boldsymbol{e}_i$;
        – otherwise, set $\boldsymbol{u}^{\ell,k} = \boldsymbol{u}^\ell - \Delta_\ell \boldsymbol{e}_i$, $\boldsymbol{d}_\ell = -\boldsymbol{e}_i$.
        – For $k = 1, \cdots, \alpha_0$
            * If $J(\boldsymbol{u}^{\ell,k-1} + \Delta_\ell \boldsymbol{d}^\ell) > J(\boldsymbol{u}^{\ell,k-1})$, set $\boldsymbol{u}^{\ell,k} = \boldsymbol{u}^{\ell,k-1} + \Delta_\ell \boldsymbol{d}^\ell$;
            * else, break the for-loop.
        – EndFor
        – Set $\boldsymbol{u}^{\ell+1} = \boldsymbol{u}^{\ell,k}$ and $\ell = \ell + 1$.
      Otherwise,
        – set $FL_{GPS} = FL_{GPS} + 1$.
    • If $FL_{GPS} = N_x$,
        – $\boldsymbol{u}^\ell$ is a local optimizer of the current mesh, then reduce current mesh size $\Delta_\ell$,
          e.g. $\Delta_\ell = \Delta_\ell/2$.
        – If $\Delta_l \geq \Delta_{\min}$, then re-create the mesh $M_\ell = \{\boldsymbol{u}^\ell \pm \Delta_\ell \boldsymbol{e}_i : i \in \{1, 2, \cdots, N_x\}\}$,
          and set $FL_{GPS} = 0$;
        – otherwise, terminate the algorithm.
EndDo
```

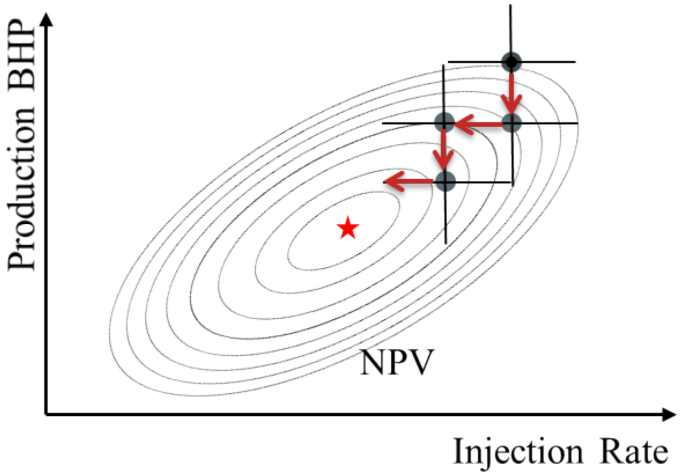


Figure 2.3: Illustration of the General Pattern Search (GPS) search direction.

CHAPTER 3

SELECTING WELLS FROM A GIVEN SET OF POTENTIAL PATHS

Reservoir engineers are often asked to select "the best" N_w well paths from a set of K_{max} potential paths where $K_{max} > N_w$, determine whether each of the N_w wells should be an injector or a producer and perhaps determine the optimal drilling order. The number of wells, N_w, that will be drilled is determined by the drilling budget. Often optimizing operating well controls is not done until the well paths, types and drilling order have been selected. In this chapter, our focus is solely on the problem: given $K_{\mathrm{m}ax}$ plausible well paths, select the optimal N_w wells, the type of each of these wells, the drilling order and find the optimal wells controls where the objective function we wish to maximize is the net present value (NPV) of production over the life of the reservoir. That is to say, we consider the optimization problem given in Eq. 2.1 with the design variables as $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$. Well controls are optimized only after $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ is optimized for a given set of well controls. The set of K_{max} potentially good well paths is determined by drilling engineers working with geoscientists and production engineers focused on drilling sweet zones with good flow capacity. As an aid to finding good potential well paths, one may use a reservoir quality map generated from geological information and static data to identify high permeable regions with significant mobile oil; see [56, 39, 40, 17].

3.1 Problem Description and Methodology

In this chapter, each path is described by its perforations where the perforation segments can be pre-specified with different shapes, i.e., vertical, horizontal, slanted or curved. As the objective is to select the N_w "best" well paths out of K_{max} possibilities rather than to optimize the trajectories of K_{max} paths, we simply map each potential path to its label.

Hence the set of potential paths is represented as a set of positive integers, denoted as Ω, where the numbers are used for labeling or identification only, and they do not indicate quantity, rank, or any other measurement. Thus,

$$\Omega = \{i | i = 1, 2, \cdots, K_{max}\}$$

where, each i in Ω corresponds to a specific path. Let $\boldsymbol{x} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_{N_w}]$ denote any N_w paths out of K_{max} possibilities. The entries $\boldsymbol{x}_j, j = 1, 2, \cdots, N_w$ must all be distinct but can correspond to any value in Ω encoded as its binary representation. One realization of $\boldsymbol{x}$ can be obtained by making N_w selections without replacement from Ω, so there are $\frac{K_{max}!}{N_w!(K_{max}-N_w)!}$ possibilities for $\boldsymbol{x}$. In this chapter, $\boldsymbol{x}_j$ is denoted as a $\boldsymbol{x}_j^b$, where the superscript b is used to emphasize it is a binary vector. It is important to emphasize that $\boldsymbol{x}_j^b$ is actually the binary representation of one of the numbers in Ω where the number of labels in Ω determines the length of $\boldsymbol{x}_j^b$. If for example, there are 32 possible paths, then we need a binary string of length 5 in order to define a unique binary string for each of the 32 (2^5) potential paths. If there are only 30 potential paths, we still require a binary string of length 5 to uniquely describe the 30 potential paths but in this case, there will be two coded strings which will pertain to none of the potential paths. In general, we need a string of length bpl to uniquely define K_{max} potential paths, where bpl is the integer such that

$$2^{bpl-1} < K_{max} \leq 2^{bpl}. \tag{3.1}$$

Denote the vector $\boldsymbol{T}$ as a N_w-dimensional binary vector of the form $\boldsymbol{T} = [T_1, T_2, \cdots, T_{N_w}]$ where each entry $T_j, j = 1, 2, \cdots, N_w$ determines whether the well placed at the potential path corresponding to $\boldsymbol{x}_j^b$ is an injector or a producer. T_j is a two-level categorical variable, where $T_j = 0$ indicates a producer and $T_j = 1$ indicates an injector. The vector $\boldsymbol{T}$ has 2^{N_w} possibilities. Similarly, denote $\boldsymbol{O} = [O_1, O_2. \cdots, O_{N_w}]$ as an N_w-dimensional vector where each entry O_j determines the drilling sequence of the well with its path corresponding to $\boldsymbol{x}_j^b$. For example, $O_j = 5$ (when $N_w \geq 5$) means the well represented by $\boldsymbol{x}_j^b$ is the 5th well drilled. As O_j is an ordinal number which tells the drilling sequence of the well represented by $\boldsymbol{x}_j^b$ and we are going to select N_w wells, $\boldsymbol{O}$ always represents a permutation of the numbers from 1, 2,$\cdots$, N_w. We assume there is only one rig and wells are drilled one by one which means that all entries of $\boldsymbol{O}$ have to be distinct. Hence a realization of $\boldsymbol{O}$ can be obtained by making N_w selections without replacement from a list of numbers from 1 to N_w, where there are $N_w!$ possibilities for $\boldsymbol{O}$. We assume each well is drilled in T_d days which means that the well with drilling sequence O_j will be drilled at time $T_d(O_j - 1)$ days. We let $\boldsymbol{u} = [\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}, \cdots , \boldsymbol{u}^{(N_w)}]^T$ denote the vector of well controls for N_w wells where each entry $\boldsymbol{u}^{(j)}$ denotes the control settings of the well placed at the location represented by $\boldsymbol{x}_j^b$. Since there can be N_c control steps, $\boldsymbol{u}^{(j)} = [u_1^{(j)}, u_2^{(j)}, \cdots , u_{N_c}^{(j)}]$ where N_c is the total number of control steps, $\boldsymbol{u}$ is a $N_w * N_c$-dimensional vector of continuous variables, determining the injection/production rates of the wells under rate control or the BHP's of wells under bottomhole pressure control. Note before a well is drilled, its controls are immaterial and this well is fixed to shut-in status. Therefore, the number and the length of the active control steps for each well may be different.

The problem considered in this chapter is a mixed-integer problem with a high-dimensional design variable vector $\boldsymbol{c} = (\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O}, \boldsymbol{u})$. The genetic algorithm (GA) is adopted as the main methodology to optimize $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$. Even though it is possible to simultaneously optimize the control variables $\boldsymbol{u}$ together with the well locations, types and drilling order $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$, we consider a sequential optimization workflow where $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ is firstly optimized with fixed $\boldsymbol{u}$ and then $\boldsymbol{u}$ is optimized based on the optimal $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ obtained, for the reasons discussed below. Note if we choose $N_w = 12$ wells out of $K_{max} = 64$ locations as in the second numerical example presented later, then $bpl = 6$, $\dim(\boldsymbol{x}) = N_w * bpl = 72$, $\dim(\boldsymbol{T}) = 12$, $\dim(\boldsymbol{O}) = 12$, which sums up to 96. If each of the 12 wells had 20 control steps, them $\dim(\boldsymbol{u}) \approx 240$. If $\boldsymbol{u}$ is included in the design variable, i.e., $\boldsymbol{c} = (\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O}, \boldsymbol{u})$, then $\boldsymbol{c}$ would contain 336 components which would require a population size of 772, based on common rule of thumb for GA that the population size be twice the length of $\boldsymbol{c}$. If 50 generations were produced using GA with this population size, over 30,000 reservoir simulation runs would be required which would likely make GA practically infeasible. The computational cost would of course be reduced by using only one to two control steps per well, but this would in many cases yield a suboptimal result (Oliveira and Reynolds [63]). A second reason for sequential optimization is that the NPV function is generally a fairly smooth function of the controls [42, 73], so the gradient-based methods using gradients computed from the adjoint method or from a stochastic approximation are applicable and relatively effective for optimizing controls but is relatively inefficient for optimizing categorical variables. Hence, a sequential optimization procedure can break a high-dimensional optimization problem into two smaller problems, and enables us to use different algorithms suitable for each specific sub-problem, i.e in the first stage of optimization, the optimization variables $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ are all categorical variables, hence GA, an algorithm with global search ability, can be employed whereas in the second stage of optimization, the vector of well controls, $\boldsymbol{u}$, is optimized using the Stochastic Simplex Approximate Gradient (StoSAG) [24]. A third reason for not including $\boldsymbol{u}$ directly in $\boldsymbol{c}$ is that in industry, control optimization is often done after the wells have been drilled. Thus our general procedure is to maximize $J(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ with each component of $\boldsymbol{u}$ set equal to its upper bound for a BHP-controlled injector and set equal to its lower bound for a BHP-controlled producing well, and then maximize $J(\boldsymbol{u})$ based on the optimal $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ obtained. Note that there is no guarantee that sequential algorithms converge. Isebor et al. [34] stated that simultaneous optimization of well locations and controls gave a higher NPV than sequential optimization. Different than [34] who focused on optimizing vertical wells, Humphries and Haynes [32] optimized the well placement and control settings for non-conventional wells and found that sequential approaches gave a higher NPV than simultaneous optimization for complex well parameterizations. After providing results of our workflow for two examples, we comment for sequential versus simultaneous optimization.

In this chapter, we used mixed encodings to account for different types of variables in order to fully characterize the variable set $\boldsymbol{c} = [\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O}]$. The location variables $\boldsymbol{x}$ and the type variables $\boldsymbol{T}$ are encoded as binary variables, and $\boldsymbol{O}$ is parameterized as ordinal numbers using permutation encoding. Note $\boldsymbol{x}$ is encoded with binary strings instead of ordinal numbers to enable the generation of new potential locations which do not belong to any of the parents in the crossover operation. The commonly used crossover operation for ordinal numbers, Partially Mapped Crossover (PMX), can only recombine the existing potential locations in the parent pair. If well locations are parameterized as ordinal variables and PMX is applied, when $K_{max} >> N_w$, it is likely that the initial population of candidate solutions (randomly generated) will not include some of the potential paths and it is highly unlikely that these paths will be brought back to the gene pool in following iterations by mutation alone.

3.1.1 Optimization of well locations, types and drilling order

In this section, we give a detailed description of the five GA operators: initialization, selection, crossover, mutation and replacement, adapted to solve for the specific problem of this chapter. Note that the three parts $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ of design variable vector $\boldsymbol{c}$ have different encodings and each encoding requires a different crossover and mutation procedure. Hence, GA operators suitable for the mixed encodings are designed to fit for the purpose of simultaneous optimization of $\boldsymbol{c} = (\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$.

1. Initialization

In the implementation of GA, we need to preset the population size N_p, the maximum number of generations N_g, the crossover rate p_c and the mutation rate p_m. Initially, N_p individuals will be generated to form the initial population in which each entry of chromosome $\boldsymbol{c}$ has to be sampled randomly and independently.

As both $\boldsymbol{x}$ and $\boldsymbol{T}$ are binary strings, each of their entry is sampled randomly from $\{0, 1\}$. Note, there are two types of constraints in the parameterization of $\boldsymbol{x}$. Firstly, when K_{max} is strictly less than 2^{bpl}, binary encodings can be generated that do not correspond to any of the potential locations. Secondly, we do not allow two wells to be drilled at the same location, or two potential locations cannot be selected twice in a single chromosome, i.e., $\boldsymbol{x}_i^b \neq \boldsymbol{x}_j^b$ for $i \neq j$. Any infeasible trial chromosomes (chromosomes violating any of these two constraints) are abandoned and only feasible chromosomes are accepted in the initial population. Due to the ordinal feature of $\boldsymbol{O}$, the drilling sequence is encoded as a permutation of the sequence $[1, 2, \cdots, j, j+1, \cdots, N_w]$ where the jth entry represents the drilling order of the well placed at the path represented by $\boldsymbol{x}_j^b$. For example, if $N_w = 8$, a possible chromosome segment for the drilling order is

$$\boldsymbol{O} = [6\ 4\ 5\ 3\ 7\ 8\ 1\ 2]. \tag{3.2}$$

According to Eq. 3.2, well $\boldsymbol{x}_1^b$ is drilled the 6th (i.e., $5T_d$ days), well $\boldsymbol{x}_2^b$ is drilled 4th (i.e., $3T_d$ days), well $\boldsymbol{x}_3^b$ is drilled 5th (i.e., $4T_d$ days) and so on. For each individual of the initial population, $\boldsymbol{O}$ is sampled as a random permutation. It is worth mentioning that in the initialization process, we force the first well drilled to be a producer as it generally would make no sense to open an injector first. Then a random sample of $(\boldsymbol{x}, \boldsymbol{T})$ are concatenated with a random sample of $\boldsymbol{O}$ to form a random chromosome $\boldsymbol{c}$. After all N_p individuals are generated, the objective function defined in Eq. 2.2 for each individual is evaluated.

2. Selection

After the current generation has been evaluated, a portion of this generation is selected in order to breed a new generation where the more fit individuals are likely to be selected. Stochastic Universal Sampling (SUS) is a development of fitness proportionate selection where fitness can be defined as any function positively correlated to the objective function J. To apply SUS, we first order and rank the N_p chromosomes according to their NPV in ascending order so that the chromosome with the highest NPV has a ranking of N_p, the chromosome corresponding to the second highest NPV is given a ranking of $N_p - 1$ and so on. The ranking/ordering of two individuals of equal NPV's is determined randomly. The ranking is used as the fitness function. We let F_j denote the fitness of chromosome j where the chromosomes $j = 1,\ 2,\ \cdots,\ N_p$ are ordered according to their fitness i.e., $F_1 \geq F2 \geq F_3 \geq \cdots \geq F_{N_p}$. The cumulative fitness of individual j is denoted by $F_p(j)$ and defined by $F_p(j) = \sum_{k=1}^{j} F_k$, so the total fitness is $F_p(N_p) = \sum_{k=1}^{N_p} F_k$. SUS then uses a single random value r to sample the ranked individuals by choosing them at evenly spaced intervals. This gives a weaker member of the population (according to its fitness) a chance to be chosen and thus avoids having the candidate space saturated by the fittest members. For illustration purpose, consider selecting 8 parents from a population of 8, i.e., $N_p = 8$. In Fig. 3.1, A, B, C, D, E, F, G and H represent the eight individuals with descending fitness (values 8, 7, 6, $\cdots$, 1, respectively). Note that the total fitness is $F_p(N_p) = 36$ for this case. First, a random number $r \sim U\left(0, \frac{F_p(N_p)}{N_p}\right)$ is generated. For $i = 1, 2, \cdots, N_p$, the ith selected individual is the one with fitness F_j where $j \in \{1, 2, \cdots, N_p\}$ is the index satisfying $F_p(j-1) < r + \frac{(i-1)F_p(N_p)}{N_p} \leq F_p(j), j \in \{1, 2 \cdots, N_p\}$ where $F_p(0) = 0$. The N_p selected individuals compose the parent pool to breed the next generation. The selected parents are then shuffled and divided into $\frac{N_p}{2}$ parent pairs. For the r selected in Fig. 3.1, the individuals selected as parents are A, A, B, C, C, D, F and G.

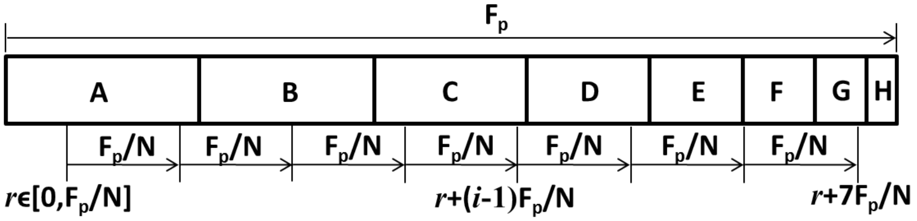
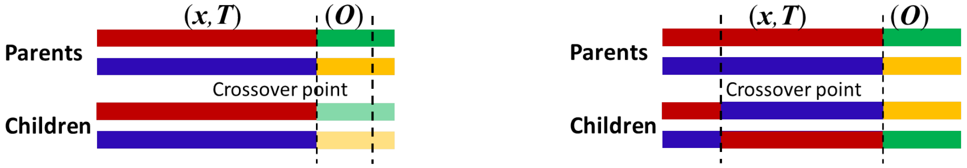


Figure 3.1: SUS selection for the GA algorithm where A, B, C, D, E, F, G, H represent the ranked individuals in a population with their lengths proportional to their ranks in descending order; F_p represents the sum of ranks for all individuals and N represents the number of parents to keep in the parent pool.

3. Crossover

The crossover operation produces offsprings by recombining a selected pair of candidate solutions from the parent pool at the crossover rate p_c, which is the probability of performing a crossover operation. For each pair of selected parents, a random number $r_i, i = 1, 2 \cdots, N_p/2$ is generated from the uniform distribution $\mathcal{U}[0, 1]$ and compared with the crossover rate p_c. If $r_i > p_c$, the pair of offsprings are kept the same as the parent pair; otherwise, a random position is selected on chromosome $\boldsymbol{c}$. If the randomly generated crossover point falls in the part of $\boldsymbol{c}$ corresponding to the drilling order $\boldsymbol{O}$ as in Fig. 3.2(a), then Partially Mapped Crossover (PMX), which is discussed later, is conducted for $\boldsymbol{O}$. If the crossover point falls in between two genes corresponding to $(\boldsymbol{x}, \boldsymbol{T})$ as in Fig. 3.2(b), then a modified version of one-point crossover is applied as discussed later.



(a) Crossover point in $\boldsymbol{O}$

(b) Crossover point in $(\boldsymbol{x}, \boldsymbol{T})$

Figure 3.2: Illustration of the crossover operation for the mixed encoding chromosomes $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$. The red and blue colors represent $(\boldsymbol{x}, \boldsymbol{T})$ of the two parent chromosomes respectively. The green and orange colors represent $(\boldsymbol{O})$ of the two parent chromosomes, respectively.

One-point crossover is the most common crossover mechanism used in GA, and is applied wherever the random position selected is in the $(\boldsymbol{x}, \boldsymbol{T})$ part of the chromosome $\boldsymbol{c}$. In this case, the genes on the right of selected position are exchanged between the pair of parents (see Fig. 3.3). However, if the crossover point falls in the $\boldsymbol{x}$ part, the produced offsprings may possibly violate one of the two types of constraints mentioned previously, i.e., (i) $\boldsymbol{x}_j^b$ may correspond to a chromosome which is not one of the potential paths in Ω or (ii) $\boldsymbol{c}$ may be a chromosome such that $\boldsymbol{x}_j^b = \boldsymbol{x}_i^b$ for $i \neq j$. Whenever (i) is violated, we abandon the infeasible trial offsprings and repeat the crossover operation where the crossover point is selected in the $\boldsymbol{x}$ part of $\boldsymbol{c}$ until feasible offsprings are formed. Quite often the selected pair of parents have wells in common at different locations in the two chromosomes, see the yellow part of the chromosomes in Fig. 3.4(a). If the selected crossover point is in between two positions represented by the yellow genes as is the case for the two parents of Fig. 3.4(a), then the produced offsprings violate constraint (ii) since one potential location (represented by the yellow genes) appears twice in one single chromosome, see Child 1 in Fig. 3.4(a). However, abandoning the infeasible offspring and repeating the crossover operation until feasible offsprings are obtained greatly reduces the diversity. Hence, instead, we propose to do crossover on the remaining part of chromosomes that excludes the common wells, and then replace the common wells back in their original place, see Fig. 3.4(b). The crossover process is repeated until N_p offsprings are generated.

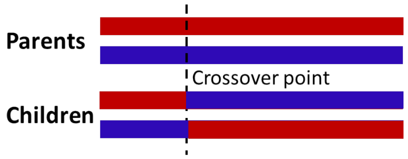
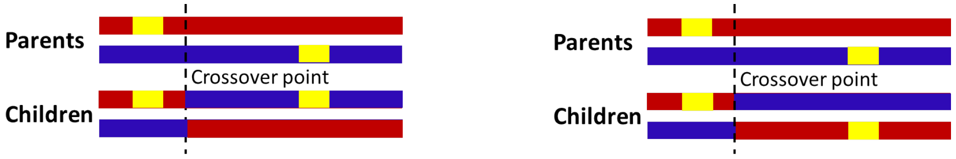


Figure 3.3: Illustration of one-point crossover. The red and blue colors represent $\boldsymbol{x}$ of two parent chromosomes respectively.



(a) One-point crossover

(b) Modified one-point crossover

Figure 3.4: Illustration of the original one-point crossover and the modified one-point crossover. The red and blue colors represent $\boldsymbol{x}$ of the two parent chromosomes respectively. The yellow color corresponds to a specific gene unit $\boldsymbol{x}_j^b, j = 1, 2, \cdots, N_w$.

Adding more crossover sites (i.e., N-point crossover) allows the head and tail section of a chromosome to be accepted together in the offspring and can give offsprings of a larger diversity compared to one-point crossover. However, it can destroy good gene segments and can sometimes reduce the performance of the genetic algorithm. We started with the one-point crossover and did not see a loss of diversity in the offsprings. Hence, one-point crossover are used throughout our work for the binary encoded vectors $(\boldsymbol{x}, \boldsymbol{T})$. The classical one-point (or N-point) crossover operator is no longer suitable for the ordinal vector $\boldsymbol{O}$ since they may generate offspring chromosomes where two or more wells are drilled at the same time, hence, we used PMX crossover instead.

The partially mapped crossover (PMX) was proposed by Goldberg and Lingle [29] for ordinal numbers. Firstly, two random crossover points are chosen along the chromosome string, then the substrings between these two crossover points are exchanged between the selected pair of parents, and then the remaining information of the offsprings are determined according to the mapping relationship in the selected substrings. Consider, for example, two parents (denoted as $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$) with one random crossover point between the 3rd and 4th bits and the other crossover point between the 6th and 7th bits where, with the two crossover points (shown below) marked with |, $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ are given by:

$$\begin{aligned} \boldsymbol{P}_1 &= [8 \quad 4 \quad 3 \quad | \, 5 \quad 7 \quad 6 \, | \quad 1 \quad 2], \\ \boldsymbol{P}_2 &= [4 \quad 6 \quad 5 \quad | \, 3 \quad 7 \quad 8 \, | \quad 1 \quad 2]. \end{aligned} \tag{3.3}$$

The mapping sections that are interchanged are between these two crossover points. In this example, the mapping systems are 5 → 3, 7 → 7 and 6 → 8. Now these two mapping sections are exchanged with each other so that the two offsprings have the following form,

$$\begin{aligned} \boldsymbol{C}_1 &= [\times \quad \times \quad \times \quad | \, 3 \quad 7 \quad 8 \, | \quad \times \quad \times], \\ \boldsymbol{C}_2 &= [\times \quad \times \quad \times \quad | \, 5 \quad 7 \quad 6 \, | \quad \times \quad \times], \end{aligned} \tag{3.4}$$

where the entries of genes marked × are still to be determined. Then, determine the remaining bits using information from the original parents that produce no conflicts, i.e., do not generate two equal bits in one chromosome, to obtain

$$\begin{aligned} \boldsymbol{C}_1 &= [\times \quad 4 \quad \times \quad | \, 3 \quad 7 \quad 8 \, | \quad 1 \quad 2], \\ \boldsymbol{C}_2 &= [4 \quad \times \quad \times \quad | \, 5 \quad 7 \quad 6 \, | \quad 1 \quad 2]. \end{aligned} \tag{3.5}$$

If it did not cause a conflict, the first × in the first offspring $\boldsymbol{C}_1$ would be 8 which comes from the first parent $\boldsymbol{P}_1$. However, 8 is already in $\boldsymbol{C}_1$ so it cannot be used again. According to the mapping sections, 8 → 6, and the fact that 6 is absent from $\boldsymbol{C}_1$ in Eq. 3.5, 6 should then occupy the first × in $\boldsymbol{C}_1$. The second × in $\boldsymbol{C}_1$ is 3 according to Eq. 3.3, but 3 is already in this offspring, $\boldsymbol{C}_1$ of Eq. 3.5. Thus, according to mapping 3 → 5, 5 should replace the second × since 5 has no conflict with the existing elements. Thus the first offspring $\boldsymbol{C}_1$ is given by

$$\boldsymbol{C}_1 = [6 \quad 4 \quad 5 \quad | \, 3 \quad 7 \quad 8 \, | \quad 1 \quad 2]. \tag{3.6}$$

Analogously, the second offspring C_2 is given by

$$\boldsymbol{C}_2 = [4 \quad 8 \quad 3 \quad | \, 5 \quad 7 \quad 6 \, | \quad 1 \quad 2]. \tag{3.7}$$

4. Mutation

All N_p offsprings are mutated before they are evaluated. Again, we used different mutation operators for the binary encoded variables $(\boldsymbol{x}, \boldsymbol{T})$ and the permutation encoded variables $\boldsymbol{O}$.

For each gene of $(\boldsymbol{x}, \boldsymbol{T})$, a random number r_i is generated from the uniform distribution $\mathcal{U}[0, 1]$ and compared with the mutation rate p_m, which represents the probability of mutation operation. If $r_i < p_m$, the corresponding gene will be flipped, i.e., we change 0 to 1 or vice versa. If either constraint (i) or (ii) is violated after the mutation operation, the chromosome will be repaired by replacing the infeasible chromosome, $\boldsymbol{x}$, with a feasible chromosome following three steps: (i) find the potential paths that do not appear in $\boldsymbol{x}$ and their corresponding binary representations; (ii) calculate the Euclidean distance between $\boldsymbol{x}_j^b$ and each binary representation; (iii) replace $\boldsymbol{x}_j^b$ (the well that cause a conflict) with one of these binary representations which has the shortest Euclidean distance to $\boldsymbol{x}_j^b$. Note that mutation of a type variables gene T_j will not cause infeasibility.

Since each element of the drilling order $\boldsymbol{O}$ is related to each other, i.e., $\boldsymbol{O}$ is a sequence, we cannot perform the classical gene by gene mutation as we do for $(\boldsymbol{x}, \boldsymbol{T})$. Instead, we do mutations, chromosome by chromosome, with a larger mutation probability, $\hat{p}_m$, than is used when mutating genes of binary variables $(\boldsymbol{x}, \boldsymbol{T})$. In particular, for each individual in the current population, we generate one random number $r_i \sim \mathcal{U}(0, 1), i = 1, 2, \cdots, N_p$. If $r_i < \hat{p}_m$, then mutate the drilling order $\boldsymbol{O}$ of ith individual, otherwise, leave $\boldsymbol{O}$ of the ith chromosome unchanged. Three mutation operations are used with equal probability. We use $\boldsymbol{O}$ in Eq. 3.2 as an example to demonstrate the procedure used when $r_i < p_m$.

- Two-point swap. Randomly select two points in the $\boldsymbol{O}$ segment of a chromosome, and swap the ordinal numbers at these two points. For example, select the 3rd and 6th locations in Eq. 3.2, then the mutated drilling order is obtained as

$$[6 \quad 4 \quad |8| \quad 3 \quad 7 \quad |5| \quad 1 \quad 2]. \tag{3.8}$$

- Flipping of genes. Randomly select two points out of the $\boldsymbol{O}$ segment of a chromosome, and flip the ordinal numbers in between these two points, i.e., reverse the order in which they appear. For example, select the 3rd and 6th locations, then the mutated drilling order of Eq. 3.2 is obtained as

$$[6 \quad 4 \quad |8 \quad 7 \quad 3 \quad 5| \quad 1 \quad 2]. \tag{3.9}$$

- Re-combine genes. Randomly select one point out of the $\boldsymbol{O}$ segment of a chromosome, split it into two segments and re-combine the two segments by putting the first segment of genes after the second segment of genes. For example, select one point between the 6th and 7th locations and then the mutated drilling order of Eq. 3.2 is obtained as

$$[\quad 1 \quad 2 \quad |6 \quad 4 \quad 5 \quad 3 \quad 7 \quad 8]. \tag{3.10}$$

5. Replacement

In the replacement operation, the best candidate solution from the current generation (with highest NPV or fitness) will randomly replace one of the newly generated offsprings so that the best chromosome will be carried over to the next generation.

In this chapter, besides simultaneous optimization of $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$, we also considered a two-stage sequential optimization procedure. In the first stage, the locations of N_w wells from K_{max} potential locations, $\boldsymbol{x}$, and the well types, $\boldsymbol{T}$, are optimized where all N_w wells operate from time zero since $\boldsymbol{x}$ and $\boldsymbol{T}$ are always represented as binary vectors. In the second stage, the drilling order of N_w wells $\boldsymbol{O}$ is optimized based on the optimal $(\boldsymbol{x}, \boldsymbol{T})$ obtained in the first stage. In this way the drilling order $\boldsymbol{O}$ is considered as a second order effort. The performance of the sequential procedure and the simultaneous procedure are compared based on two numerical examples.

3.1.2 Optimization of well controls $\boldsymbol{u}$

As mentioned earlier, StoSAG combined with steepest ascent is used to optimize $\boldsymbol{u}$

based on the optimal well locations, types and drilling orders, $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$, obtained for fixed values of the well controls. In the implementation of StoSAG, at each iteration, based on the current estimate of control settings $\boldsymbol{u}^{\ell}$, 10 perturbations are generated from a normal distribution where the ith diagonal of $\boldsymbol{C}_u$ is $\left(0.03 * (u_i^{\mathrm{up}} - u_i^{\mathrm{low}})\right)^2$ and $\boldsymbol{C}_u$ is selected to enforce a temporal correlation of the controls of each well; see Chen and Reynolds [9]. The maximum step size, along the normalized stochastic gradient direction (gradient divided by its infinity norm) is set as $0.2 * [\boldsymbol{u}^{\mathrm{up}} - \boldsymbol{u}^{\mathrm{low}}]$. If the NPV is increased, then update $\boldsymbol{u}^{\ell}$ and move to the next iteration. If the NPV is not increased, then the step size is cutback by 50%. The maximum number of step-size cuts is set as 5. If after 5 step-size cuts, the NPV cannot be improved, 10 new perturbations around $\boldsymbol{u}^{\ell}$ are re-sampled to estimate a new stochastic gradient. The maximum number of re-samples is set as 5, i.e., after 5 re-samples, the algorithm is terminated.

3.2 Numerical Examples

In this section, the proposed methodology is tested on two waterflooding examples, a channelized reservoir with three different facies (Example 1) and the Brugge model (Example 2). In Example 1, we select 8 best wells from 30 potential paths (either vertical or horizontal). In Example 2, we consider three cases: in case A, we try to determine the best 12 well locations out of 30 potential locations; in case B, we still select 12 best wells, but the number of potential paths is enlarged to 64. In case C, we consider simultaneously selecting 12 wells starting from time 0 and 8 infill wells starting from the fourth year, out of 64 potential locations. For both examples, we first determine the optimal well locations, types, drilling order using GA with BHP's for producers and injectors, respectively, fixed at their minimum and maximum values, and then the optimal control settings are determined using the StoSAG algorithm while we fix the well locations, types and drilling order at the optimal values obtained in the first optimization. We also compare simultaneous optimization of $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ and sequential optimization where $(\boldsymbol{x}, \boldsymbol{T})$ are optimized first assuming all wells start operation at 0 days and then drilling order $(\boldsymbol{O})$ is optimized with $(\boldsymbol{x}, \boldsymbol{T})$ fixed at the optimal values obtained, referred to as $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$.

In the implementation of GA, the crossover probability p_c is set to 0.8 for both $(\boldsymbol{x}, \boldsymbol{T})$ and $\boldsymbol{O}$. However, for the binary encoded parameters $(\boldsymbol{x}, \boldsymbol{T})$, the mutation probability $\hat{p}_m$ is set to 0.01. For the ordinal parameters $\boldsymbol{O}$, p_m is defined by

$$\hat{p}_m = \min\{0.03 * i_g, 0.3\}, \tag{3.11}$$

where i_g represents the i_gth generation. As discussed previously, we do mutations gene by gene for $(\boldsymbol{x}, \boldsymbol{T})$ and chromosome by chromosome for $\boldsymbol{O}$. The population size N_p is set to be 50; the maximum number of generations N_g are set differently for Example 1 and Example 2. Based on our numerical experiments, we found that the rule of thumb in the GA literature that N_p should be set equal to twice the number of bits in a chromosome gives a larger value of N_p than what is needed. $N_p = 50$ gives a reasonable trade-off between fast convergence and the good quality of the solution.

3.2.1 Example 1: channelized reservoir

The first example considered is a channelized reservoir model defined on a 50 × 50 × 14 reservoir simulation grid where each grid block is 200 ft × 200 ft × 10 ft. The reservoir is composed of three geologic zones: zone 1 is composed of the first four reservoir simulation layers, zone 2 contains the middle six simulation layers, zone 3 is composed of the bottom four simulation layers. There are three facies, i.e., the channel facies, the levee facies and the shale facies generated using the object-based modeling. When generating the petrophycical parameters, the azimuth angle is set to 0° and the correlation length is set to 1,000 ft in both the major and minor correlation directions. We assume no correlation in the vertical direction. Both the horizontal permeability field and the porosity field are generated following normal distribution where the mean and deviation of the petro-physical parameters for each individual facies are given in Table 3.1. The vertical permeability field is set equal to 1/10 of the horizontal permeability field. The top of the reservoir is at the depth of 4,800 ft. The initial reservoir pressure is 3,800 psi and the initial reservoir oil saturation is 0.9. Note the upper bound on the production BHP's and the lower bound on the injection BHP's are set equal to the initial reservoir pressure, i.e., 3,800 psi.

Wells can only be drilled at the 30 possible locations shown in Fig. 3.5 and each location corresponds to one prescribed well trajectory (a fully penetrating vertical well or a horizontal well). Each potential well location corresponds to a label. The potential horizontal wells are completed in different layers and the prescribed well trajectories at different potential locations do not intersect with each other. Note that two potential well locations can intersect or overlap with each other areally when their paths are projected. For example, the label "24/28" in Fig. 3.5 means that path 24 and path 28 project to the same line in the x-y plane but path 24 is a horizontal well located in layer 12 whereas path 28 is a horizontal well located in layer 14. The square symbol represents the starting point of a well, the circle symbol represents the end point of a well and the bold line represents the perforation in the plotted layer. This configuration contains 11 vertical wells fully penetrating all 14 layers (with labels from 1 to 11) and 19 horizontal wells completed in different layers (with labels from 12 to 30), see Table 3.2. The perforation lengths of the horizontal wells are equal to 2,000 ft which are much longer than those of the vertical paths which have a perforation length equal to the reservoir thickness, 140 ft.

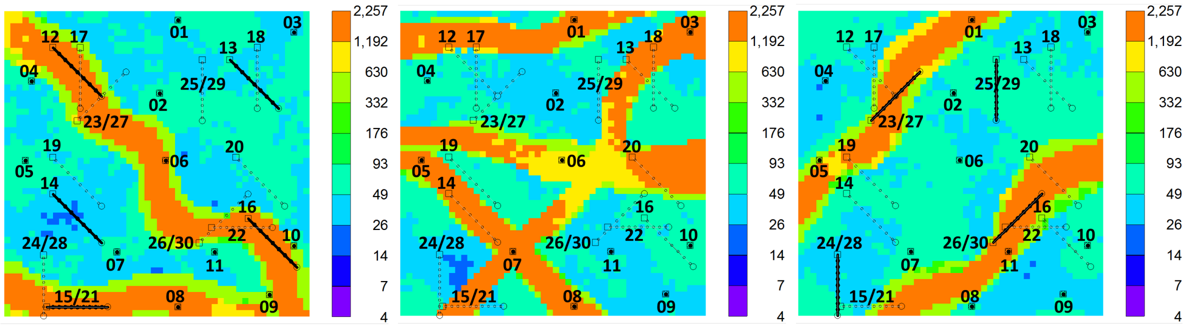
The problem is to select $N_w = 8$ best wells from $K_{max} = 30$ potential locations and determine their optimal locations $\boldsymbol{x}$, types $\boldsymbol{T}$, drilling order $\boldsymbol{O}$ and control settings $\boldsymbol{u}$. All wells are under BHP control. We consider the effect of the value of the upper bound on BHP's at injectors and the lower bound on BHP's at producers and the effect of the values of fixed BHP's when we optimize $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ with the value of BHP's held fixed. Two sets of bounds on the control settings are considered, where in bound A the upper bound for injectors is 4,600 psi and the lower bound for producers is 3,000 psi and in bound B the upper bound for injectors is 5,000 psi and the lower bound for producers is 1,000 psi. A producer will be shut-in if its water cut goes above 0.98. The production life, which is set to be 4,000 days, is divided into 16 control steps where the first 7 control steps are all 90 days in length, the subsequent 8 control steps are all 365 days in length and the last control step is 450 days. We assume it takes three months to drill a well, $T_d = 90$ days. Accordingly, the first well has 16 control steps, the second well has 15 control steps, and so on. The eighth well drilled has 9 control steps. Hence, there are 100 control variables in total, i.e., the vector of the continuous control variables, $\boldsymbol{u}$, is 100-dimensional. In this case, the drilling cost of a vertical well is $C_{w,j} = \$8 \times 10^6$ and the drilling cost of each horizontal well is $C_{w,j} = \$16 \times 10^6$. The economic parameters for the computation of NPV are set as: $r_o^n = \$50.0/\text{STB}$, $c_w^n = \$5.0/\text{STB}$, $c_{wi}^n = \$5.0/\text{STB}$, b=10.0%. Only oil and water flow in the reservoir.

Given $K_{max} = 30$, a $bpl = 5$ bit binary vector is used to describe the possibility of each of the 30 potential locations. Hence, with N_w=8, $\boldsymbol{x}$ is a $N_w * bpl = 40$-dimensional binary variable vector, $\boldsymbol{T}$ is a 8-dimensional binary variable vector and $\boldsymbol{O}$ is a 8-dimensional ordinal vector, so there are 56 categorical optimization variables. Hence, a sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ and $\boldsymbol{u}$ is adopted where $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ are determined first fixing the BHP's of injectors and producers at their upper bound and lower bound respectively and then $\boldsymbol{u}$ is optimized based on the optimal $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ obtained. In simultaneous optimization of $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ with GA using mixed encoding, the maximum number of generations is set equal to 80. In sequential optimization of $(\boldsymbol{x}, \boldsymbol{T})$ using binary encoding followed by optimization of $\boldsymbol{O}$ using permutation encoding, the maximum number of generations are set as 50 and 30, respectively, so that the computational effort of simultaneous and sequential optimization are comparable in that 80 generations are used in both cases.

In this example, we consider the determination of optimal $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O}, \boldsymbol{u})$ under bound A and bound B, where we denote the optimization processes as “Bound A” and “Bound B” respectively. For both Bound A and Bound B, we set the BHP's at the bounds when determining the optimal $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ and then determine the optimal $\boldsymbol{u}$ using the optimal $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ obtained. To investigate the effects of controls used to determine well locations, types and drilling order, we also considered a case where we set the controls at bound A to determine the optimal $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ and then optimize well controls within bound B, referred to as “Bound A→B”. Note that the bounds in Bound A are closer to the initial reservoir pressure than those of Bound B. Bound A→B is a special case of Bound B where the controls are not set at the bounds (1,000 psi for production BHP and 5,000 psi for injection BHP) when determining $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$, but are fixed equal to the initial guesses used for the subsequent optimization of well controls (3,000 psi for production BHP and 4,600 psi for injection BHP).

Table 3.1: Petrophysical parameters for each facies, Example 1, i.e., selecting $N_w = 8$ wells given $K_{max} = 30$ potential locations in a channelized reservoir.

Facies	Porosity		Horizontal Permeability, mD	
	Mean	Variance	Mean	Variance
Channel	0.25	0.03	1500	200
Levee	0.20	0.03	500	100
Shale	0.15	0.03	50	10



(a) Layer 1 (b) Layer 7 (c) Layer 12

Figure 3.5: The full set of potential well locations and trajectories projected to the horizontal permeability field, Example 1.

Table 3.2: Perforated layers of the horizontal wells, Example 1.

Path Label	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Completed Layer	1	1	1	1	1	3	3	3	3	3	3	12	12	12	12	14	14	14	14

1. Optimal solution of $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$

In this section, we use Bound A as the base case to illustrate the performance of the proposed methodology. In Bound A, the BHP controls at all producers are fixed at their lower bound (3,000 psi), the BHP controls at all injectors are fixed at their upper bound (4,600 psi) when optimizing $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$. Fig. 3.6 shows the NPV versus number of generations

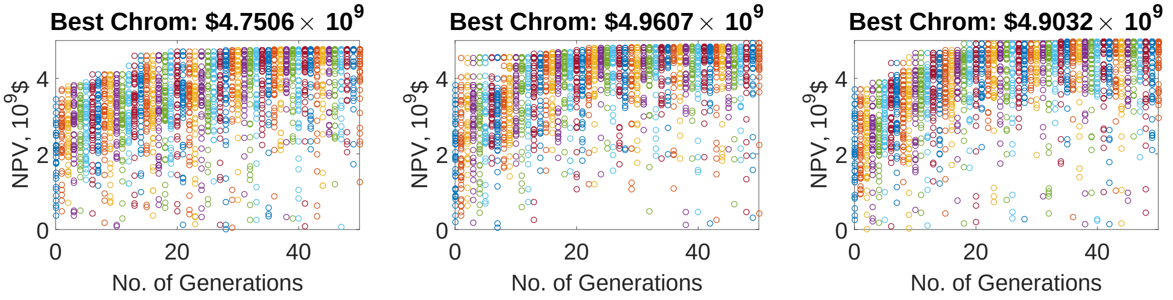
Table 3.3: Fixed BHP's used in the optimization of $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ and initial guesses used in the subsequent control optimization, Bound A, Bound A→B and Bound B for Example 1.

	Case Bound A	Case Bound A→B	Case Bound B
Upper bound on injection BHP, psi	4600	5000	5000
Lower bound on production BHP, psi	3000	1000	1000
Injection BHP for $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ optimization, psi	4600	4600	5000
Production BHP for $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ optimization, psi	3000	3000	1000
Initial injection BHP for control optimization, psi	4500	4600	4800
Initial production BHP for control optimization, psi	3100	3000	1200

for the joint optimization of only well locations and types (no optimization of $\boldsymbol{O}$) obtained with three different initial random seeds. The NPV's for the candidate solutions in the initial generation range widely, approximately from $\$0 \times 10^9$ to $\$4 \times 10^9$, i.e., the well locations and types significantly impact the NPV. The best NPV's obtained with seed 2 and seed 3 are somewhat higher than the best NPV obtained with the run using seed 1, but seed 1 gives only a 4% lower optimal NPV than the highest optimal NPV so the method is fairly robust to the choice of the initial population. The best NPV among the $N_p = 50$ candidate solutions increases significantly during the first 20 generations but the rate of improvement slows down dramatically during the following 30 generations and is essentially stable for the last few generations for all three seeds. Table 3.4 shows the best candidate solution obtained up to 10, 20, 30, 40 and 50 generations using seed 2. In Table 3.4 and similar tables presented later, "P" corresponds to a producer and "I" corresponds to an injector, e.g., "I04" represent an injector at the well path represented by label 4 in Fig. 3.5; $N_{inj}, N_{pro}, N_V, N_H$ respectively represent the optimal number of injectors, producers, vertical wells and horizontal wells. As we can see, the optimal well locations and types change greatly in the first 20 generations (specifically, generations 10 and 20 have only one well in common), but generations 40 and 50 have identical optimal solutions.

Based on the optimal well locations and types, a second stage of drilling order optimization is carried out and the resulting NPV versus the number of generations is shown in Fig. 3.7. Compared to the optimal NPV in Fig. 3.6, the initial NPV's in Fig. 3.7 are much smaller. This is because in the first optimization stage where $(\boldsymbol{x}, \boldsymbol{T})$ is optimized, all wells are assumed to begin operating at time 0 while in the second optimization stage, wells are brought online sequentially (every $T_d = 90$ days) which corresponds to a shorter total operation time and hence gives smaller NPV's. The NPV's for the candidate solutions in the initial generation in Fig. 3.7 have a relatively small range, about $\$0.2 \times 10^9$ which indicates the well drilling order has a much smaller impact on the NPV compared to the well locations and types. The best NPV obtained with each seed increases only slightly (on average 0.7%) by optimizing the drilling order and stabilized at around 30 generations for all three seeds. More precisely, if we had chosen the "optimal" drilling order based on the best NPV of the initial population of Fig. 3.7(b), we would have obtained an "optimal" NPV equal to $\$4.67 \times 10^9$ whereas as drilling order optimization increased this to $\$4.72 \times 10^9$, an absolute increase of 50 million dollars at the cost of 1500 reservoir simulation runs. However, if we simply ignored any evaluation of NPV for a set of possible drilling order, i.e., did not even generate the initial population of Fig. 3.7(b), we could have picked a drilling order that resulted in an NPV of $\$4.44 \times 10^9$ which is about 230 million lower than the best NPV from the initial population. Table 3.5 shows the optimal solutions including the well locations, types and drilling order for all three seeds. Each well is represented in the form "type-label-order," e.g., "P05-6" represents drilling the producer with potential path labeled 5 starting from $(6 - 1) * T_d = 300$ days. "V" represent a vertical well and "H" represents a horizontal well. If a well is horizontal, the completed layer of that well is also given, e.g. "H-1" represents a horizontal well completed in layer 1. As we can see from Table 3.5, the number of injectors is 3 or 4 for all 3 seeds. The variation in the optimal NPV is also small. Comparing the optimal solutions obtained with seed 1 and seed 2, they have 6 similar wells, including four wells in common {P05, P16, P23, I29}, one pair of wells at same areal location but completed in different layers {I21 and I15}, and one pair of wells close to each other {I13 and I18}. Comparing the optimal solutions obtained with seed 2 and seed 3, they have 6 wells in common {I04, P05, I15, P16, P20, P23} and one pair of wells close to each other {I13 and I18}. For all three seeds, the common wells have the same relative drilling order and the first four wells drilled follow the sequence "producer-injector-producer-injector". Fig. 3.8

shows the optimal well placement obtained with three initial random seeds plotted on the horizontal permeability field and the remaining oil saturation field of layer 7 after 4,000 days of production. Note that the distribution of the remaining oil saturation is fairly similar for the three estimates of optimal solutions. Hence, the optimization gives us a clear idea about good locations, their types, drilling order, the optimal NPV and the sweep efficiency.

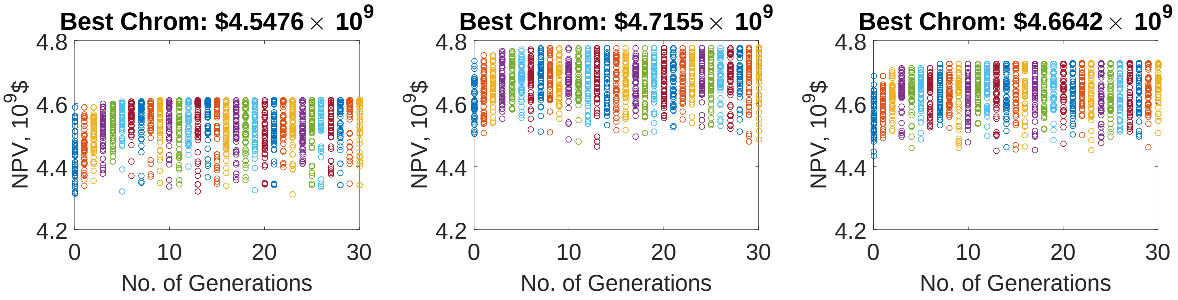


(a) Seed 1 (b) Seed 2 (c) Seed 3

Figure 3.6: NPV versus generations obtained with joint optimization of well locations and types using three seeds, Bound A, Example 1.

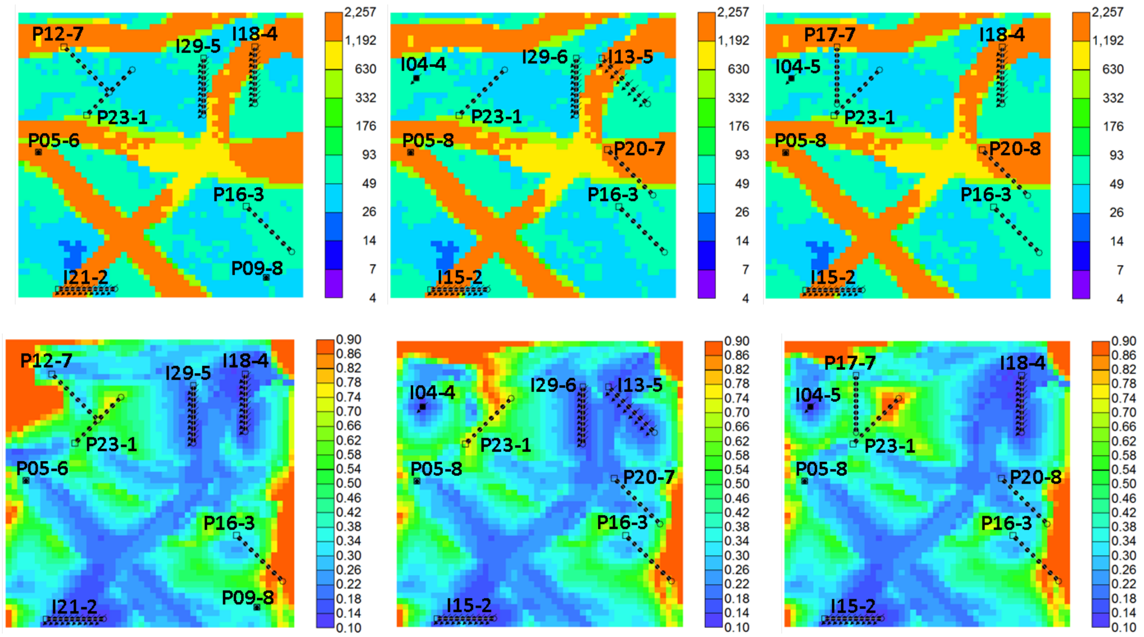
Table 3.4: Optimal solutions obtained at different number of generations for the joint optimization of well locations and types using seed 2, Bound A, Example 1.

Generations	NPV, \$ $\times 10^9$	N_{inj}	N_{pro}	N_V	N_H				Wells				
10	4.49	4	4	3	5	I04	I08	P10	I13	I14	P16	P23	P28
20	4.68	4	4	2	6	I04	P09	P12	I13	I15	P16	P23	I29
30	4.75	4	4	2	6	P01	I04	I13	I15	P16	P20	P23	I29
40	4.96	4	4	2	6	I04	P05	I13	I15	P16	P20	P23	I29
50	4.96	4	4	2	6	I04	P05	I13	I15	P16	P20	P23	I29



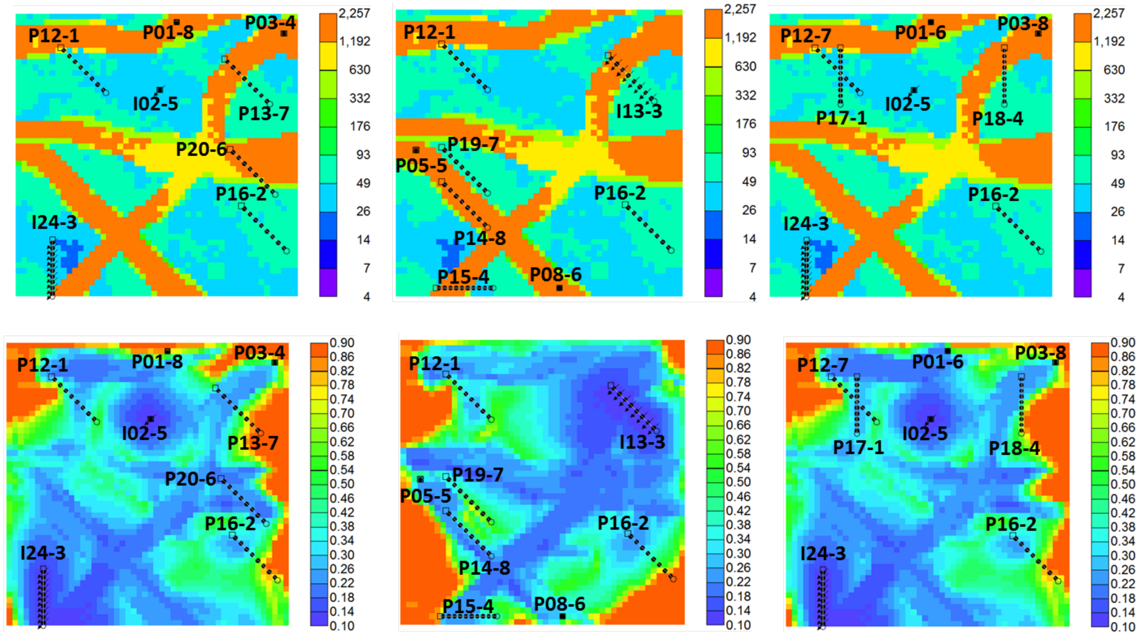
(a) Seed 1 (b) Seed 2 (c) Seed 3

Figure 3.7: NPV versus generations obtained with the optimization of drilling order using three seeds, Bound A, Example 1.



(a) Seed 1 (b) Seed 2 (c) Seed 3

Figure 3.8: Optimal solutions obtained with sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$, using three seeds, plotted on the horizontal permeability field (upper row) and the oil saturation field (bottom row) of layer 7 at end of 4,000 days of production, Bound A, Example 1.



(a) Seed 1 (b) Seed 2 (c) Seed 3

Figure 3.9: Optimal solutions obtained with sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$, using three seeds, plotted on the horizontal permeability field (upper row) and the oil saturation field (bottom row) of layer 7 at end of 4,000 days of production, Bound B, Example 1.

Fig. 3.9 presents the optimal wells obtained with sequential optimization using three seeds for Bound B, plotted on the horizontal permeability field (upper row) and the oil saturation field (bottom row) of layer 7 at end of 4,000 days of production. In this case, the BHP's are fixed at their bounds (5,000 psi for injectors, 1,000 psi for the producers) at all control steps when optimizing $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$. From Fig. 3.9, with seed 1 and seed 3, we obtain similar flooding patterns (with 6 wells in common) with two injection wells where the injected water at I24 flows from the southwest corner to the northeast direction along the channels and the producers are gathered at the north and east sides of the reservoir. On the other hand, for seed 2, there is only one injector (i.e., I13) and the injected water flows from the northeast to the southwest with most producers at the south and west sides of the reservoir. Despite the difference in flow paths and well paths obtained for seed 1 and 3 compared to those generated with seed 2, the estimated optimal NPV's are almost identical for the 3 seeds as shown in Table 3.6. Table 3.6 shows a summary of the optimal solutions obtained with sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$ for Bound B. Compared to Bound A where 3 to 4 injectors are obtained, only one (horizontal) to two injectors (one horizontal, one vertical) are obtained in case Bound B, due to the higher injection BHP's. Meanwhile, for all three seeds of Bound B, (i) the first injector drilled is horizontal and is brought online $T_d = 90$ days later than that in Bound A; (ii) the first four wells drilled follow the sequence "producer-producer-injector-producer". Hence, the optimal results of Bound B also illustrate that the proposed optimization procedure can provide a clear guidance on the optimal well combinations and drilling sequence, indicating GA algorithm is quite robust.

Based on the optimal well locations, types and drilling order obtained with sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$, control optimization is carried out using StoSAG. For both Bound A and Bound B, the initial guesses of BHP's for the control optimization are set close to the bounds and are given in Table 3.3. For Bound A→B, the initial guesses of BHP's are set at the upper bound on injection BHP's (4,600 psi) and lower bound on production BHP's (3,000 psi) of bound A, which are far from the bounds of case Bound B (upper bound on injection BHP's at 5,000 psi and lower bound on production BHP's at 1,000 psi).

Table 3.5: Optimal solutions obtained with sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$ using three seeds, Bound A, Example 1. For seed 1 and seed 3 results, wells that are different from the solution of seed 2 are marked in red.

Seeds	NPV, \$ $\times 10^9$	N_{inj}	N_{pro}	N_V	N_H	Wells							
Seed 1	4.55	3	5	2	6	P05-6 V	P09-8 V	P12-7 H-1	P16-3 H-1	I18-4 H-3	I21-2 H-3	P23-1 H-12	I29-5 H-14
Seed 2	4.72	4	4	2	6	I04-4 V	P05-8 V	I13-5 H-1	I15-2 H-1	P16-3 H-1	P20-7 H-3	P23-1 H-12	I29-6 H-14
Seed 3	4.66	3	5	2	6	I04-5 V	P05-6 V	I15-2 H-1	P16-3 H-1	P17-7 H-3	I18-4 H-3	P20-8 H-3	P23-1 H-12

Table 3.6: Optimal solutions obtained with sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$ using three seeds, Bound B, Example 1. Injectors are in bold.

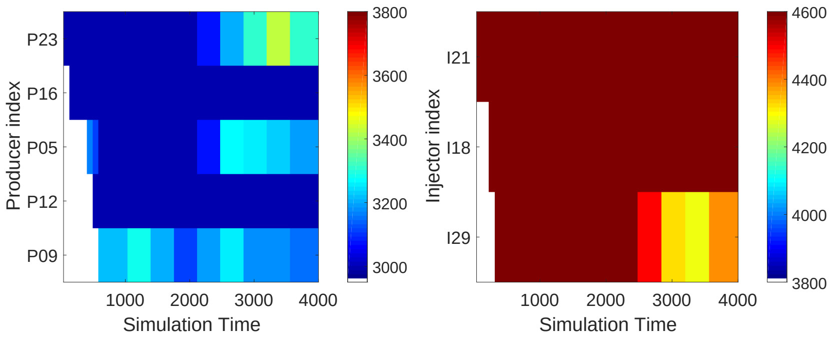
Seeds	NPV, \$ $\times 10^9$	N_{inj}	N_{pro}	N_V	N_H	Wells							
Seed 1	6.39	2	6	3	5	P01-8 V	**I02-5** V	P03-4 V	P12-1 H-1	P13-7 H-1	P16-2 H-1	P20-6 H-3	**I24-3** H-12
Seed 2	6.36	1	7	2	6	P05-5 V	P08-6 V	P12-1 H-1	**I13-3** H-1	P14-8 H-1	P15-4 H-1	P16-2 H-1	P19-7 H-3
Seed 3	6.40	2	6	3	5	P01-6 V	**I02-5** V	P03-8 V	P12-7 H-1	P16-2 H-1	P17-1 H-3	P18-4 H-3	**I24-3** H-12

2. Investigation of the effect of initial guesses and bounds

The estimated NPV's obtained for case Bound A, Bound A→B as well as Bound B obtained for the optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$ followed by well control optimization, are shown in Table 3.7. Figs. 3.10 and 3.12 show the optimal controls obtained for Bound A using seed 1 and for Bound B using seed 2. Note in both figures, wells tend to operate at or very close to the bounds for the first 2,000 days which contributes more than 80% of the NPV. Fig. 3.11 shows the optimal controls obtained for Bound A→B using seed 2. Note unlike cases Bound A and Bound B where the well control optimization did not increase NPV significantly beyond the NPV obtained by optimization of $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ with fixed BHP controls, in case Bound A→B, optimizing well controls resulted in about a 30% increase in NPV. However, the final optimal NPV results for Bound A→B are on the order of 5% lower than those obtained for the Bound B case where during $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ optimization well controls were fixed at bounds; see Table 3.3. It is important to note the bounds on BHP's are the same for case Bound B and case Bound A→B, the only differences are the fixed values of BHP's used when optimizing $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ and the initial guesses used in the subsequent optimization step where we optimize only controls. Specifically, if we put the controls at the bounds of Bound B (i.e., injecting at 5,000 psi and producing at 1,000 psi) for the optimal $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ obtained in Bound A→B (i.e., injecting at 4,600 psi and producing at 3,000 psi), we obtain an average NPV which is 21% lower than that in Bound A→B, and 26% lower than that obtained in Bound B, indicating that the well controls used when optimizing $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ strongly affect the optimal well locations, type, drilling order (see Tables 3.5 and 3.6) which in turn affect the optimal well controls (see Figs. 3.11 and 3.12). The results presented to this point and other results not presented, indicate that with our workflow when the optimization of $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ and the optimization of well controls are done in two separate steps, the fixed values of BHP used in the optimization of $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ may strongly influence the result and lead to at least a somewhat suboptimal final estimate of the optimal NPV.

Table 3.7: Effects of initial guesses and bounds on sequential optimization of well locations and types, drilling order (i.e., $(\boldsymbol{x}, \boldsymbol{T}) \to \boldsymbol{O}$) and controls using three seeds, Example 1.

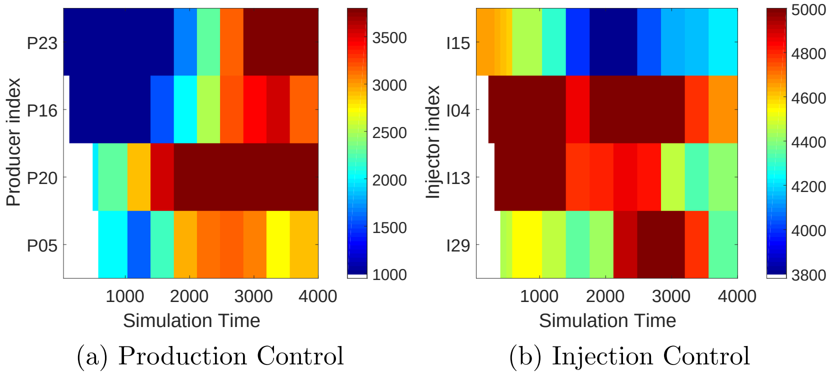
Seeds	Optimal NPV, $\$ \times 10^9$					
	Bound A		Bound A→B		Bound B	
	$(\boldsymbol{x}, \boldsymbol{T}) \to \boldsymbol{O}$	$(\boldsymbol{u})$	$(\boldsymbol{x}, \boldsymbol{T}) \to \boldsymbol{O}$	$(\boldsymbol{u})$	$(\boldsymbol{x}, \boldsymbol{T}) \to \boldsymbol{O}$	$(\boldsymbol{u})$
Seed 1	4.55	4.62	4.55	5.96	6.39	6.39
Seed 2	4.72	4.72	4.72	6.09	6.36	6.43
Seed 3	4.66	4.70	4.66	6.07	6.40	6.43
Average	4.64	4.68	4.64	6.04	6.38	6.42



(a) Production Control

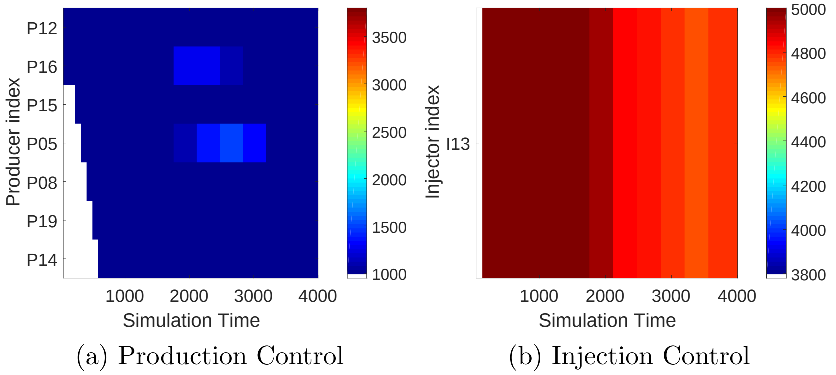
(b) Injection Control

Figure 3.10: Optimal controls for the well locations, types and drilling order obtained with sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}) \to \boldsymbol{O}$ using seed 1, Bound A, Example 1.



(a) Production Control (b) Injection Control

Figure 3.11: Optimal controls for the well locations, types and drilling order obtained with sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$ using seed 2, Bound A→B, Example 1.



(a) Production Control (b) Injection Control

Figure 3.12: Optimal controls for the well locations, types and drilling order obtained with sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$ using seed 2, Bound B, Example 1.

3. Investigation of simultaneous $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ versus sequential $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O})$ optimization

Simultaneous optimization of the well locations, types and drilling order using GA with mixed encodings with three initial random seeds is also carried out and the optimal NPV's obtained for the three cases are presented in Table 3.8. Comparing Tables 3.7 and 3.8, sequential optimization slightly outperformed simultaneous optimization with same computation cost based on the average optimal NPV's obtained for three different runs. However, it is impossible to argue this is a general conclusion and based purely on mathematical intuition, one would guess that simultaneous optimization would generally perform as well as or better than sequential optimization as simultaneous optimization ensures the best possible solution exists somewhere in the search space. Isebor et al. [34] compare simultaneous versus sequential optimization for only one simple problem where the reservoir model is 2D, and the number of wells is fixed at five with three producers and two injectors. Moreover, the only optimization variables are the well controls based on five control steps and the locations of the five vertical wells, i.e., unlike our work, well types and drilling order are not considered. The problem they consider involves only 35 total design variables. In addition, they only compare sequential and simultaneous optimization using PSO-MADS. For their case, based on five distinct optimizations, simultaneous optimization gives an NPV about 4% to 8% higher than sequential optimization which is in conflict with our conclusion on this issue. On the other hand, Humphries and Haynes [32] compare sequential and simultaneous optimization of locations and controls for a set of problems using PSO-MADS. They find that as the problem become more complex, e.g., as allowed well paths become more "complex," i.e., slanted well trajectories are allowed as opposed to requiring all wells be vertical, sequential optimization results in 5% to 9% higher NPV than simultaneous optimization. Moreover, as we consider well locations, types and drilling order while [34] and [32] consider only well locations and controls, when comparing sequential versus simultaneous optimization, there is no way to know a priori whether the conclusions of either paper on sequential versus simultaneous optimization would apply to the problem we consider here. Given the expected highly rough optimization problem where two types of categorical design variables are used and the large search space, we are not surprised to find that, for the problems we consider, sequential optimization performs slightly better than simultaneous optimization which agrees with the conclusion of [32].

3.2.2 Example 2: Brugge model

The Brugge field was developed by TNO in conjunction with the SPE Applied Technology Workshop held in Brugge in June 2008 to benchmark the technology available at that time for closed loop (real-time) reservoir management, see Peters et al. [71]. The original model was constructed with approximately 20 million gridblocks and then upscaled to a

Table 3.8: Effects of initial guesses and bounds on simultaneous optimization of well locations and types, drilling order (i.e., $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$) and controls using three seeds, Example 1.

Seeds	Optimal NPV, $\$ \times 10^9$					
	Bound A		Bound A→B		Bound B	
	$(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$	$(\boldsymbol{u})$	$(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$	$(\boldsymbol{u})$	$(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$	$(\boldsymbol{u})$
Seed 1	4.61	4.64	4.61	5.93	6.16	6.36
Seed 2	4.53	4.54	4.53	6.12	6.33	6.39
Seed 3	4.65	4.70	4.65	5.95	6.38	6.43
Average	4.60	4.63	4.60	6.00	6.29	6.39

60,048 (139 × 48 × 9) gridblock model. The stratigraphy of the Brugge field is modeled after a typical North Sea Brent field and is an elongated half-dome with one internal fault and one boundary fault. Originally, 20 vertical producers are placed inside the oil zone while 10 vertical injectors are placed close to the water oil contact. The top structure map with the original well locations is shown in Fig. 3.13. Fig. 3.14 shows the log horizontal permeability field and the porosity field of layer 1 and layer 9 respectively.

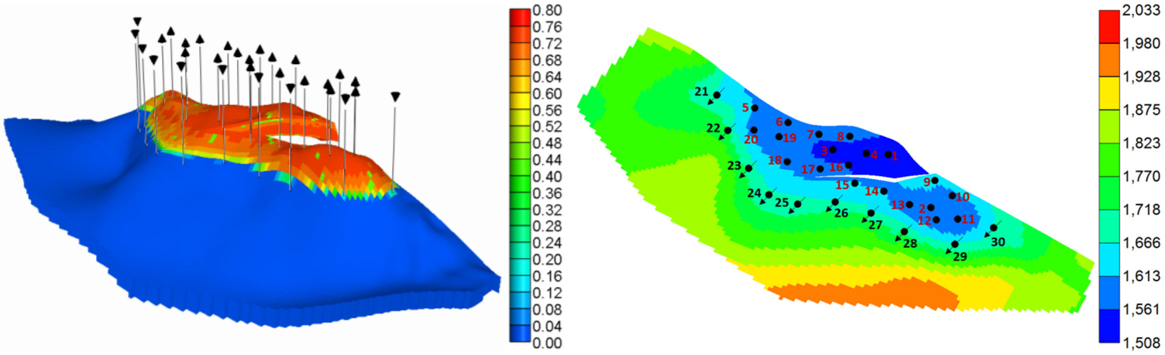
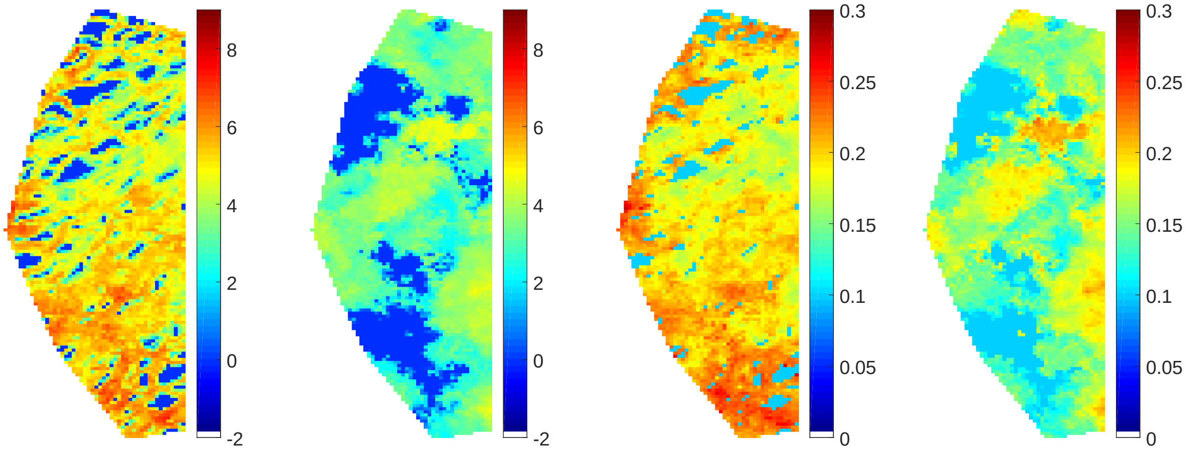


Figure 3.13: Top structure of the Brugge model with the original 30 wells. Left: 3D structure plotted on the initial oil saturation field. Right: 2D depth contour of layer 1 where wells with label 21 to 30 are the locations of the original injectors and wells with label 1 to 20 are the locations of the original producers, Example 2.

In this example, we considered three cases. In case A, we use the original 30 well locations as 30 potential locations and select 12 wells out of these 30 potential locations. In case B, we enlarge the number of potential locations to 64 but still select 12 wells. For both cases, we determined the optimal locations, types, drilling order and control settings for 12 wells. In case C, we considered simultaneously selecting 12 wells starting from time



(a) $\ln k_x$-1st layer (b) $\ln k_x$-9th layer (c) ϕ-1st layer (d) ϕ-9th layer

Figure 3.14: The log-horizontal permeability ($\ln k_x$) field and porosity (ϕ) field of the Brugge model, Example 2.

0 and 8 infill wells starting from the fourth year, from 64 potential locations. The initial average reservoir pressure is 2,514 psi. All wells operate under BHP control. The upper bound on the production BHP's and the lower bound on the injection BHP's are set the same as the initial reservoir pressure, i.e., 2,514 psi. The lower bound on the production BHP's is set equal to 725 psi and the upper bound on injection BHP's is set equal to 2,662 psi. In this example, producers only penetrate the first 8 layers and injectors penetrate all 9 layers. Either producers or injectors can be drilled at any potential location in the oil zone. We force a well drilled at potential locations in the aquifer to be an injector. A producer will be shut-in if its water cut goes above 0.94. The reservoir life is set as 20 years. The economic parameters are set as: r_o^n = \$80.0/STB, c_w^n = \$5.0/STB, c_{wi}^n = \$5.0/STB, b=10.0%. The drilling cost for each well is set as $C_{w,j} = \$8 \times 10^6$. We assume it takes one month to drill a well, i.e., $T_d = 30$ days.

1. Case A of Example 2: $N_w = 12$, $K_{max} = 30$

In this case, the locations of the 30 wells used in the original Brugge model are used as the 30 potential locations. Given $K_{max} = 30$ and $N_w = 12$, a $bpl = 5$ bit binary vector is used to describe the possibility of each potential location. Hence, $\boldsymbol{x}$ is a $N_w * bpl = 60$-

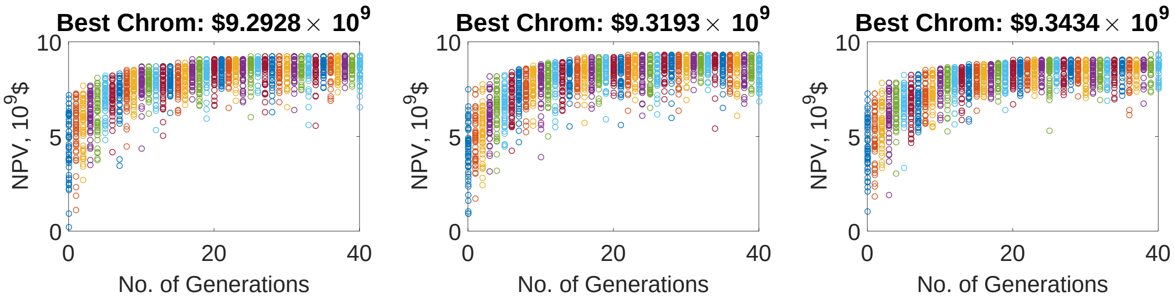
dimensional binary variable vector, $\boldsymbol{T}$ is a 12-dimensional binary variable vector and $\boldsymbol{O}$ is a 12-dimensional ordinal vector. The reservoir life is divided into 20 control steps where each control step contains 365 days. Since it takes 30 days to drill a well, all wells are drilled within the first year, i.e., all wells have 20 control steps. However, the first control step is 365 days for the first well (from days 0 to 365 days), 335 days for the second well (from 30 days to 365 days), 305 days for the third well (from 60 days to 365 days), etc. Hence, $\boldsymbol{u}$ is a $N_w * N_c = 240$-dimensional vector composed of continuous variables.

A sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ and $\boldsymbol{u}$ is adopted where $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ is determined first with the BHP's of injectors and producers fixed at their upper bound and lower bound respectively and then $\boldsymbol{u}$ is optimized based on the optimal $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ obtained. In simultaneous optimization of $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ with GA using mixed encoding, the maximum number of generations are set as 60. In sequential optimization of $(\boldsymbol{x}, \boldsymbol{T})$ using binary encoding and $\boldsymbol{O}$ using permutation encoding, the maximum number of generations are set as 40 and 20 respectively.

Fig. 3.15 shows the NPV's versus number of generations for the joint optimization of well locations and types with three seeds. In this case, the NPV's for the candidate solutions in the initial generation range approximately from $\$0 \times 10^9$ to $\$7.4 \times 10^9$, again indicating that well locations and types have a big impact on the NPV. The highest optimal NPV is obtained with seed 3, but the worst optimal NPV estimate is only 0.5% worse than the best optimal NPV of $\$9.34 \times 10^9$ which indicates that GA algorithm is quite robust. The average best NPV in the initial population is $\$7.32 \times 10^9$ and the average best NPV in the final solution is $\$9.34 \times 10^9$, i.e., GA algorithm improved the NPV by 21.6% on average. Table 3.9 shows the best solutions obtained at different generations using seed 3. From Table 3.9, the optimal well locations and types change substantially during the first 20 generations while in the following 20 generations, only two wells changed locations and types. The difference between the maximum and the minimum of the optimal NPV's for all three seeds ($0.025 \times 10^9\$$) is quite small compared to their absolute average value ($9.32 \times 10^9\$$). Based on the optimal well locations and types, a second-stage drilling order optimization is carried out and NPV versus the number of generations is shown in Fig. 3.16. Similar to Example 1, the optimal NPV's obtained are lower than that in Fig. 3.15 due to a shorter operational time when drilling order is considered. The NPV's for the candidate solutions in the initial generation have a relatively small range, about $\$0.2 \times 10^9$. If we had chosen the "optimal" drilling order based on the best NPV of the initial population of Fig. 3.16(c), we would have obtained an "optimal" NPV equal to $\$9.04 \times 10^9$ whereas as drilling order optimization increased this to $\$9.08 \times 10^9$, an absolute increase of 40 million dollars at the cost of 1000 reservoir simulation runs. However, if we simply ignored any evaluation of NPV for a set of possible drilling order, i.e., did not even generate the initial population of Fig. 3.16(c), we could have picked a drilling order that resulted in an NPV of $\$8.89 \times 10^9$ which is about 150 million lower than the best NPV from the initial population.

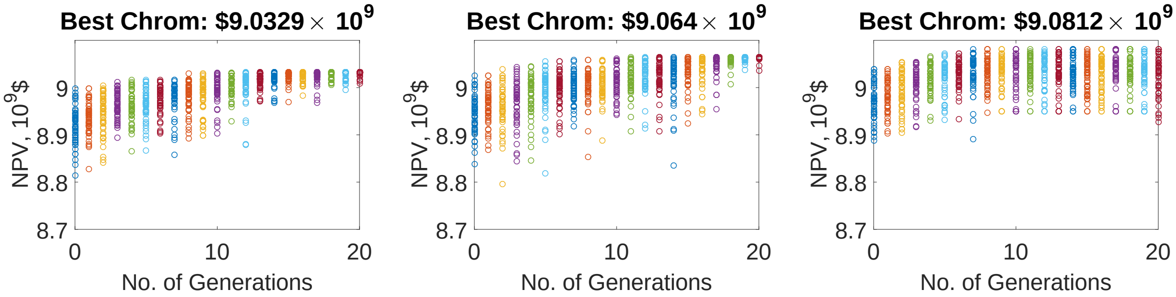
Table 3.10 shows the optimal solutions obtained with sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$. From Table 3.10, all three seeds resulted in 4 injectors and 8 producers. Comparing the optimal well configurations obtained with seed 2 and seed 3, they have 9 out of 12 wells in common. Comparing the optimal solutions obtained with seed 1 and seed 3, they again have 9 out of 12 wells in common. For all three seeds, the first injector drilled is I01 and is drilled as the 6th or 7th well; the first producer drilled is P19. Fig. 3.17 shows the optimal well configurations obtained with all three seeds and the corresponding oil saturation fields of layer 1 after 20 years of production. From Fig. 3.17, the well locations and the oil saturation field obtained for the three seeds are quite similar to each other. Note that all three seeds obtained four injectors where two injectors are placed in the aquifer and two injectors are placed in the oil zone. Among the two injectors placed in the oil zone, all three seeds chose I01 as the first injector. Hence, as in Example 1 (the channelized reservoir), the optimization gives us a clear indication about good potential locations, their corresponding types and when to drill the injectors.

Based on the optimal well locations, types and drilling order obtained with sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$, we carried out control optimization for the selected 12 best wells using StoSAG starting from the initial guesses where at all control steps, the BHP's



(a) Seed 1 (b) Seed 2 (c) Seed 3

Figure 3.15: NPV versus generations obtained with the optimization of well locations and types, i.e., $(\boldsymbol{x}, \boldsymbol{T})$, using three seeds, Example 2-A, i.e., selecting $N_w = 12$ wells given $K_{max} = 30$ potential locations.



(a) Seed 1 (b) Seed 2 (c) Seed 3

Figure 3.16: NPV versus generations obtained with drilling order optimization using three seeds, Example 2-A.

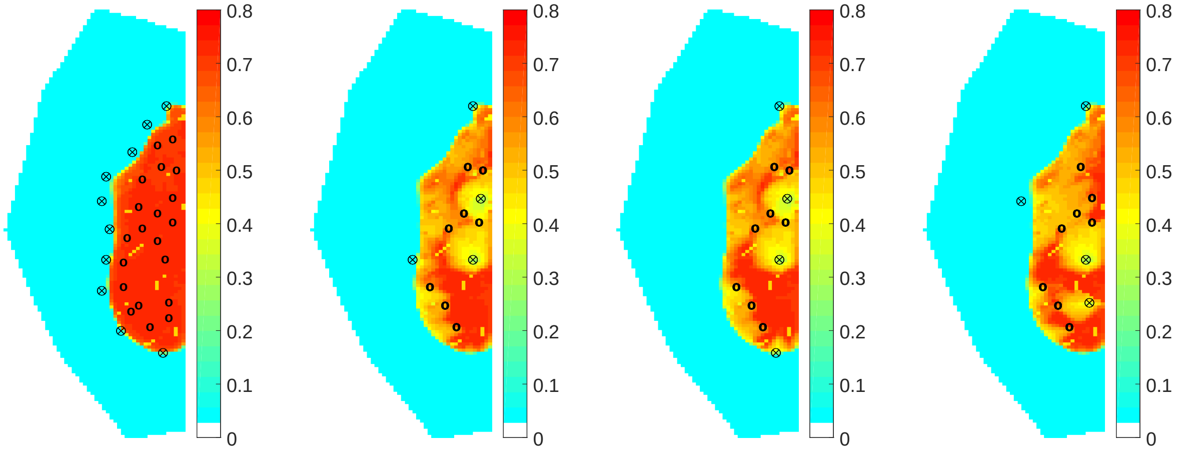
Table 3.9: Optimal solutions obtained at different number of generations for the optimization of well locations and types using seed 3, Example 2-A.

Generations	NPV, $ × 10^9	N_{inj}	N_{pro}	Wells											
10	8.58	3	9	P01	P02	P07	P08	P11	P12	P13	I16	P17	P19	I21	I30
20	9.02	3	9	I01	P02	P03	P07	P08	P11	P13	P16	P17	P19	I21	I25
30	9.04	3	9	I01	P02	P03	P07	P08	P11	P13	P16	P17	P19	I21	I26
40	9.34	4	8	I01	P02	P03	P07	P08	I09	P11	P13	P16	P19	I21	I25

Table 3.10: Optimal solutions obtained with sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$ using three seeds, Example 2-A. Wells that are different from the solution of seed 3 are in red.

Seeds	NPV, $ × 10^9	N_{inj}	N_{pro}	Wells											
Seed 1	9.03	4	8	I01-6	P02-4	P03-3	P06-5	I07-12	P08-11	P11-8	P13-9	P16-2	P19-1	I21-7	I27-10
Seed 2	9.06	4	8	I01-6	P02-5	P03-2	P06-5	I07-12	P08-10	P11-8	P13-11	P16-3	P19-1	I21-7	I30-9
Seed 3	9.08	4	8	I01-7	P02-3	P03-2	P07-6	P08-12	I09-9	P11-5	P13-10	P16-4	P19-1	I21-11	I25-8

at producers are 870 psi (150 psi more than the fixed production BHP during the $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ optimization) and the BHP's at injectors are 2,640 psi (22 psi less than the fixed injection



(a) S_o-initial (b) S_o^{opt}-seed 1 (c) S_o^{opt}-seed 2 (d) S_o^{opt}-seed 3

Figure 3.17: Well configurations and oil saturation fields of layer 1. (a) shows the set of initial potential locations plotted on the oil saturation field; (b)-(d) show the optimal well configurations obtained with sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$ for three seeds, with the corresponding oil saturation fields after 20 years of production on the top layer, Example 2-A. $\otimes$ represents an injector and o represent a producer.

BHP during the $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ optimization). Note different wells have different starting times since they are drilled one by one. However, well control optimization did not increase the value of NPV obtained by optimizing only $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ with well controls fixed at bounds, see Table 3.11. We did observe that, the control settings of the injectors increased to their upper bounds and those of the producers decreased to their lower bounds in a few iterations, but caused an increase in NPV smaller than $\$0.01 \times 10^9$ (10 million dollar), which suggests that the upper bound on the injection pressure and the lower bound of the producing pressure are the optimal control settings for this example. The fact that the optimal controls found were at bounds, for this example, is likely due to the following: we are placing a relatively small number of wells for the lengths of reservoir life specified, compared with 30 wells specified in the original Brugge example.

Table 3.11 presents a summary of the optimal NPV's obtained with both sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$ and simultaneous optimization of $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$. Based on the average optimal NPV, the sequential process performed as well or better than the simultaneous process, which is consistent with the results of Example 1.

Table 3.11: Comparison of the optimal NPV's obtained with optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$ and $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ followed by control optimization, Example 2-A.

Seeds	Optimal NPV, \$ $\times 10^9$				
	Sequential			Simultaneous	
	$(\boldsymbol{x}, \boldsymbol{T})$	$(\boldsymbol{O})$	$(\boldsymbol{u})$	$(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$	$(\boldsymbol{u})$
Seed 1	9.29	9.03	9.03	8.96	8.96
Seed 2	9.32	9.06	9.06	9.11	9.11
Seed 3	9.34	9.08	9.08	9.11	9.11
Average	9.32	9.06	9.06	9.06	9.06

We denote the minimum production BHP's of 725 psi and the maximum injection BHP's of 2,662 psi as Bound B which is the case we just considered. Now, we also consider a set of bounds (referred to as Bound A) where the lower bound on producer BHP's is 1,619.5 psi and the upper bound on injection BHP's is 2,588 psi. Note the bounds of Bound A are midway between the value of Bound B and the initial reservoir pressure. Similar to Example 1, the determination of the optimal $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O}, \boldsymbol{u})$ using Bound A is denoted as "Bound A." The case where we set the controls at Bound A to determine the optimal $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ and then optimize well controls within Bound B, is referred to as "Bound A→B". Table 3.12 shows the optimal NPV's obtained for the optimization process of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow (\boldsymbol{O}) \rightarrow (\boldsymbol{u})$ for Bound A, Bound B and Bound A→B.

From Table 3.12, the control optimization improves the NPV's only slightly for Bound A, but significantly for Bound A→B, indicating that the bounds on BHP's affect the optimal controls and that increasing the upper bound for injection pressure and reducing the lower bound on production BHP's during the control optimization can further improve the NPV's. Compared to Bound B, slightly less NPV's are obtained for Bound A→B where the optimal $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ are obtained with BHP's fixed at the middle of Bound B, and controls are optimized within Bound B. If we obtain a global optimizer for the optimization of $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ with well control fixed, it is optimal for the controls used. However, it may be possible to increase the optimal NPV by well control optimization using the optimal $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ obtained. We find that for all the examples considered, that if we fix controls at bounds (upper bound of the injection

BHP and lower bound for the production BHP) during $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ optimization, then the well control optimization have a negligible effect on the optimal NPV. However, if we fix well controls at non-bounds during the optimization of $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$, then well control optimization with the $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ fixed, tends to move pressure closer the bounds but ultimately produce a lower final NPV than is obtained by optimization of $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ with controls fixed at the bounds. Similar to Example 1, this fact recommends that it is preferable to operate the injectors at their maximum BHP and producers at their minimum BHP to determine the optimal $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$. Table 3.13 shows the optimal $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ obtained for Bound A where wells that are different from the solution of seed 3 in Table 3.10 are marked in red. Different from Example 1 where different flooding patterns for Bound A→B and Bound B are obtained, similar wells are obtained in this case. From Table 3.13, Bound A also gives 4 injectors and 8 producers which is consistent with the result of the base case (Bound B) given in Table 3.10. What is more, 8 out of 12 wells selected in Table 3.13 are also selected in Table 3.10. Hence, it is not surprising that similar optimal NPV's after control optimization are obtained for Bound A→B and Bound B.

Table 3.12: Comparison of the optimal NPV's obtained with sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow (\boldsymbol{O})$ followed by control optimization, i.e., $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow (\boldsymbol{O}) \rightarrow (\boldsymbol{u})$, Example 2-A.

Seeds	Optimal NPV, $ × 10^9					
	Bound A		Bound A→B		Bound B	
	$(\boldsymbol{x}, \boldsymbol{T}) \rightarrow (\boldsymbol{O})$	$(\boldsymbol{u})$	$(\boldsymbol{x}, \boldsymbol{T}) \rightarrow (\boldsymbol{O})$	$(\boldsymbol{u})$	$(\boldsymbol{x}, \boldsymbol{T}) \rightarrow (\boldsymbol{O})$	$(\boldsymbol{u})$
Seed 1	6.29	6.29	6.29	8.73	9.03	9.03
Seed 2	6.40	6.29	6.29	8.96	9.06	9.06
Seed 3	6.29	6.29	6.29	8.87	9.08	9.08
Average	6.33	6.29	6.29	8.85	9.06	9.06

Table 3.13: Optimal solutions obtained with sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$ using three seeds, Bound A, Example 2-A. Wells that are different from the solution of seed 3 in Table 3.10 are marked in red.

Seeds	NPV, $ × 10^9	N_{inj}	N_{pro}	Wells											
Seed 1	6.29	4	8	I01-5	P02-3	I03-12	P06-10	P08-9	P11-6	P13-11	P16-2	P17-4	P19-1	I21-8	I30-7
Seed 2	6.40	4	8	I01-7	P02-3	P06-8	I07-10	P08-12	P11-4	P13-6	P16-1	P17-11	P19-2	I21-9	I30-5
Seed 3	6.29	4	8	I01-6	P02-4	P03-1	P06-10	I07-9	P08-12	P11-5	P16-3	P18-11	P19-2	I21-8	I30-7

2. Case B of Example 2: $N_w = 12$, $K_{max} = 64$

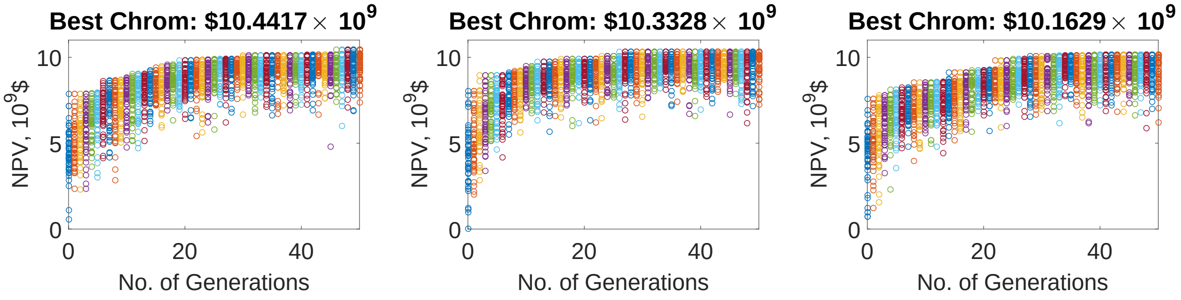
In this case, the number of potential locations is increased to 64 by adding 34 additional potential paths to the 30 locations used in case A, including 4 more locations in the aquifer near the water-oil contact and 30 more locations in the oil zone, see Fig. 3.20(a). Given $K_{max} = 64$ and $N_w = 12$, a $bpl = 6$ bit binary vector is used to provide a unique representation of each of the 64 potential well locations. Hence, $\boldsymbol{x}$ is a $N_w * bpl = 72$-dimensional binary variable vector, $\boldsymbol{T}$ is a 12-dimensional binary variable vector and $\boldsymbol{O}$ is a 12-dimensional ordinal vector. The reservoir life is divided into 40 control steps where each control step contains 182.5 days. As it takes 30 days to drill a well, the first 7 wells are drilled within the 182.5 days. That is to say, the first 7 wells all have 20 control steps, but with a different length for the first control step which is 182.5 days for the first well (from days 0 to 182.5 days), 152.5 days for the second well (from 30 days to 182.5 days), and so on. The remaining 5 wells are drilled in the second half year, i.e., they have 39 control steps where the first control step length is 155 days for the eighth well (from 210 days to 365 days), 125 days for the ninth well (from 240 days to 365 days), and so on. Hence, $\boldsymbol{u}$ is a 475-dimensional vector composed of continuous variables. A sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ and $\boldsymbol{u}$ is adopted where $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ are determined by first fixing the BHP's of injectors and producers at their upper bound and lower bound, respectively, and then $\boldsymbol{u}$ is optimized based on the optimal $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ obtained. In sequential optimization of $(\boldsymbol{x}, \boldsymbol{T})$ using binary encoding and $\boldsymbol{O}$ using permutation encoding, we still use the population size equal to 50, however, the maximum number of generations are set as 50 and 20 respectively. Note GA is allowed to run for 10 more generations compared to case A when optimizing $(\boldsymbol{x}, \boldsymbol{T})$ since the length of the chromosome is 12 bits longer. One may also choose to increase the population size N_p.

Fig. 3.18 shows the NPV's versus the number of generations for the joint optimization of well locations and types with three seeds. Similar to case A, the NPV's for the candidate solutions in the initial generation range approximately from $\$0 \times 10^9$ to $\$8.0 \times 10^9$. The highest NPV is obtained with seed 1, but the worst optimal NPV estimate is only 2.5% less than the best optimal NPV of $\$10.44 \times 10^9$, so again the GA algorithm is quite robust. The average best NPV in the initial population is \$7.82 × 10^9 and the average best NPV in the final population is \$10.31 × 10^9. Hence, the GA algorithm improved the NPV by 24% on average. Compared to case A, the average optimal NPV is 10% larger due to more options of well locations.

Based on the optimal well locations and types, a second stage of drilling order optimization is carried out and NPV versus the number of generations is given in Fig. 3.19. Similar to case A, the optimal NPV's are lower than all the initial NPV's in Fig. 3.18 due to a shorter operational time when drilling order is considered. The NPV's for the candidate solutions in the initial generation have a relatively small range, about \$0.2 × 10^9, which is consistent with case A where we select 12 wells out of 30 potential locations. If we had chosen the "optimal" drilling order based on the best NPV of the initial population of Fig. 3.19(a), we would have obtained an "optimal" NPV equal to \$10.12 × 10^9 whereas as drilling order optimization increased this to \$10.13 × 10^9, an absolute increase of 10 million dollars at the cost of 1000 reservoir simulation runs. However, if we simply ignored any evaluation of NPV for a set of possible drilling order, i.e., did not even generate the initial population of Fig. 3.19(a), we could have picked a drilling order that resulted in an NPV of \$9.94 × 10^9 which is about 180 million lower than the best NPV from the initial population.

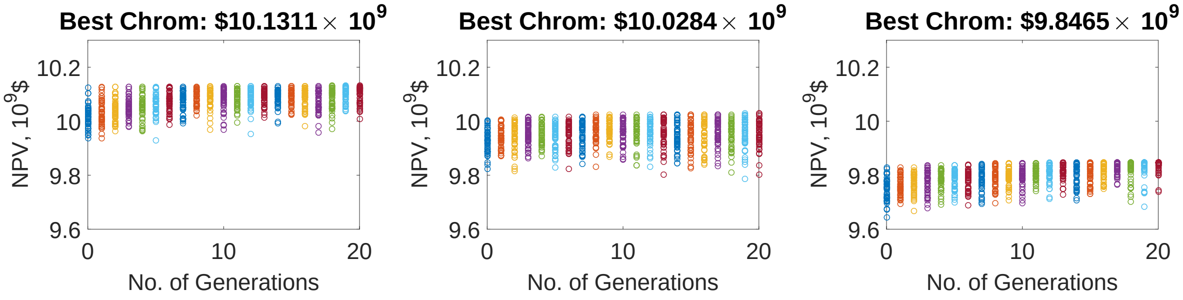
Table 3.14 shows the optimal solutions obtained with sequential optimization of $(\boldsymbol{x}, \boldsymbol{T})$ and $\boldsymbol{O}$. In this case, the number of possible selections of 12 wells among 64 potential locations is $\frac{64!}{12!52!} \approx 6 \times 10^{10}$. Hence it is difficult to comment on the quality of the optimal solutions by comparing the labels of the selected locations. Instead, the locations of optimal wells obtained with three seeds are shown in Fig. 3.20. From Fig. 3.20, the three runs using different seeds lead to consistent solutions in that they all lead to (i) 8 producers and 4 injectors, (ii) similar locations of the producers, and (iii) 2 injectors in the oil zone and 2 injectors in the aquifer. For all three seeds, the first injector is drilled as the 6th or 7th well. Hence, the optimization gives us a clear indication about the good potential locations, their corresponding types and when to drill the injectors.

Based on the optimal well locations, types and drilling order obtained with well controls fixed at bounds (upper bound of the injection BHP and lower bound of the production BHP), we carried out control optimization for the selected 12 best wells using StoSAG starting from the initial guesses where at all control steps, the BHP's at producers are 870 psi (145 psi higher than the lower bound of production BHP) and the BHP's at injectors are 2,640 psi (22 psi lower than the upper bound of injection BHP). Similar to case A, NPV is not improved by control optimization. Table 3.15 presents a summary of the optimal NPV's obtained with both sequential optimization $((\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O})$ and simultaneous optimization $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$. As in case A, sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$ outperformed simultaneous optimization of $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ for all three seeds.



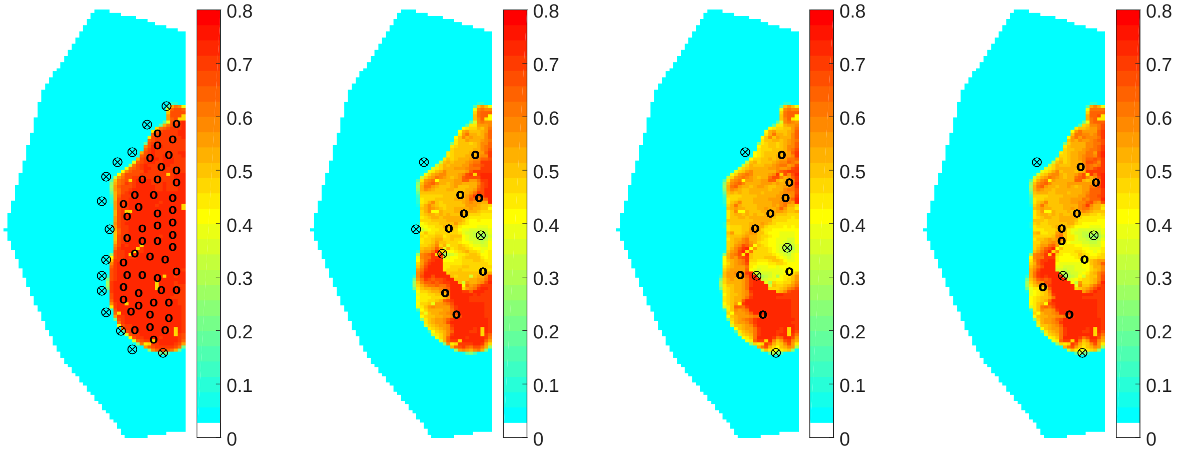
(a) Seed 1 (b) Seed 2 (c) Seed 3

Figure 3.18: NPV versus generations obtained with the optimization of well locations and types using three seeds, Example 2-B, i.e., selecting $N_w = 12$ wells given $K_{max} = 64$ potential locations in a channelized reservoir.



(a) Seed 1 (b) Seed 2 (c) Seed 3

Figure 3.19: NPV versus generations obtained with drilling order optimization using three seeds, Example 2-B.



(a) S_o-initial (b) S_o^{opt}-seed 1 (c) S_o^{opt}-seed 2 (d) S_o^{opt}-seed 3

Figure 3.20: Well configurations and oil saturation fields of layer 1. (a) shows the set of potential locations plotted on the initial oil saturation field; (b)-(d) show the optimal well configurations obtained with sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$ for three seeds, with the corresponding oil saturation fields after 20 years of production on the top layer, Example 2-B. $\otimes$ represents an injector and o represent a producer.

Table 3.14: Optimal solutions obtained with sequential optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$ using three seeds, Example 2-B.

Seeds	NPV, \$ $\times 10^9$	N_{inj}	N_{pro}	Wells											
Seed 1	10.13	4	8	P03-6	P07-12	P16-4	P22-9	P24-2	P27-1	P31-3	P40-5	I42-11	I48-7	I56-10	I63-8
Seed 2	10.03	4	8	P03-1	P07-11	P16-5	P24-2	I25-12	P27-3	P29-4	P35-10	P40-7	I43-8	I53-6	I60-9
Seed 3	9.85	4	8	P01-6	P03-4	P13-10	P16-5	P19-1	I25-9	P27-3	P29-2	P33-12	I48-7	I60-11	I63-8

Table 3.15: Comparison of the optimal NPV's obtained with optimization of $(\boldsymbol{x}, \boldsymbol{T}) \rightarrow \boldsymbol{O}$ and $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ followed by control optimization, Example 2-B.

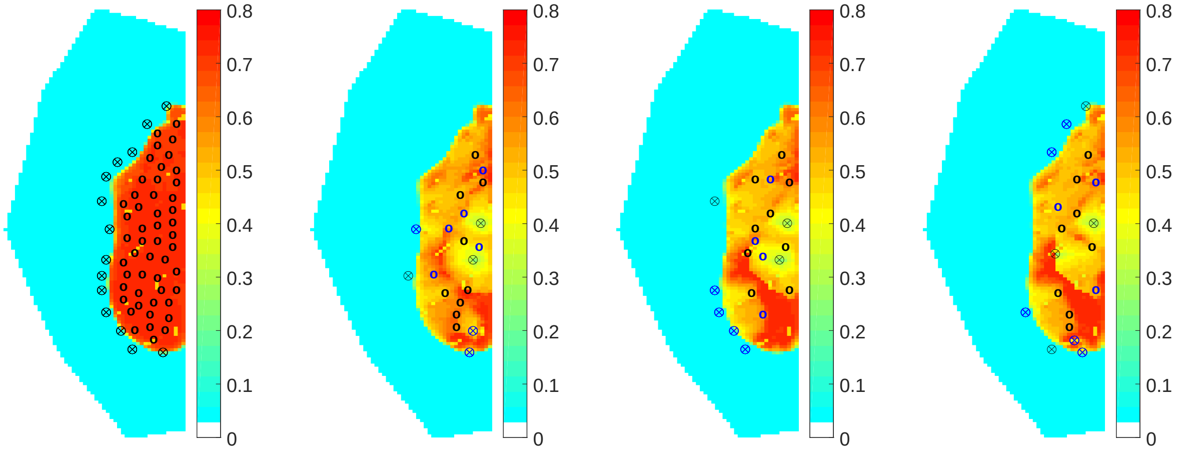
Seeds	Optimal NPV, \$ $\times 10^9$				
	Sequential			Simultaneous	
	$(\boldsymbol{x}, \boldsymbol{T})$	$(\boldsymbol{O})$	$(\boldsymbol{u})$	$(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$	$(\boldsymbol{u})$
Seed 1	10.44	10.13	10.13	10.07	10.07
Seed 2	10.33	10.03	10.03	9.42	9.42
Seed 3	10.16	9.85	9.85	9.51	9.51
Average	10.31	10.00	10.00	9.67	9.67

3. Case C of Example 2

Based on the 64 potential locations presented in case B, we carried out a study to select 12 wells starting from time 0 in addition to selecting 8 infill wells starting from the fourth year, i.e., we select in total $N_w = 20$ wells. Due to the three year's gap between the original wells and the infill wells, it is risky to adopt a sequential procedure where the well locations and types are determined assuming all 20 wells operate from time 0 and then optimize drilling order based on the optimal locations and types obtained. Hence, in this example, we only consider simultaneous optimization of well locations, types and drilling order. In this case, the drilling order is still parameterized as a permutation of the sequence $[1, 2, \cdots, j, j+1, \cdots, N_w]$. However, wells with $O_j \leq 12$ are drilled at $(O_j - 1)T_d$ days, while wells with $O_j > 12$ are drilled at $(O_j - 13)T_d + T_{infill}$ days, where T_{infill} is the time at which we drill the first infill well, i.e., the 13th well. In this example, $T_{infill} = 1460$ days. Given $N_w = 20$ and $bpl = 6$, $\boldsymbol{x}$ is a 120-dimensional vector, $\boldsymbol{T}$ is a 20-dimensional vector, $\boldsymbol{O}$ is a 20-dimensional vector, which sums up to 160. In this case, we used a larger population size $N_p = 80$ and run for 70 generations.

Table 3.16 shows a summary of the results obtained with simultaneous optimization for case C. As one can see, based on the average NPV of three runs, the infill case obtained a 29% increase compared to the initial NPV and 3.4% higher NPV compared to the simultaneous result of Example 2-B (Table 3.15), before the subsequent control optimization. However, simultaneous optimization only obtained a suboptimal solution, i.e., the best NPV obtained with seed 3 (\$$10.07 \times 10^9$) is slightly less than the best NPV of Example 2-B (Table 3.15) obtained by sequential optimization (\$$10.13 \times 10^9$). Similar to previous cases, the subsequent control optimization only lead to slight increase in the optimal NPV; see the result of $\boldsymbol{u}$ in Table 3.16. This is because the optimal control settings for the optimal $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ happen to be at the bounds. Similar to Example 1, we extended the lower bound of production pressure from 725 psi to 500 psi and the upper bound of injection pressure from 2,662 psi to 2,800 psi and re-did the well control optimization to obtain the optimal NPV's presented in the "$\hat{\boldsymbol{u}}$" column of Table 3.16. As one can see, after extending the bounds on pressure, the NPV is improved by 4.5% on average. The 12 wells scheduled to drill within the first year include 3 injectors and 9 producers on average which is similar to case B, the 8 infill wells starting at year 4 include 4 injectors and four producers on average. Fig. 3.21

shows the optimal well locations plotted on the remaining oil saturation field after 20 years of production. From Fig. 3.21, generally, we still obtain two injectors in the oil zone where the two injectors are scheduled to drill within the first year after 120 days (i.e., after the 5th drilling sequence).



(a) S_o-initial (b) S_o^{opt}-seed 1 (c) S_o^{opt}-seed 2 (d) S_o^{opt}-seed 3

Figure 3.21: Well configurations and oil saturation fields of layer 1. (a) shows the set of potential locations plotted on the initial oil saturation field; (b)-(d) show the optimal well configurations obtained with simultaneous optimization of $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ for three seeds, with the corresponding oil saturation fields after 20 years of production on the top layer, Example 2-C. $\otimes$ represents an injector and o represent a producer. The first 12 wells drilled are marked in black color and the 8 infill wells are marked in blue color.

Table 3.16: Comparison of the optimal NPV's obtained with simultaneous optimization of $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ followed by control optimization, i.e., $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O}) \rightarrow (u)$, Example 2-C.

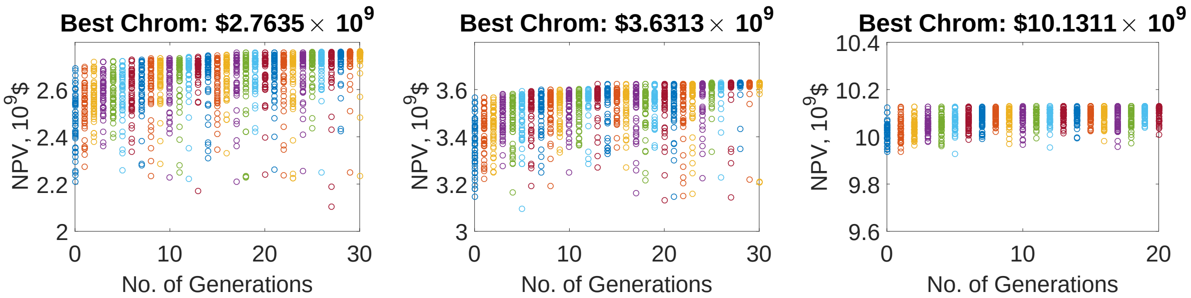
Seeds	Optimal NPV, \$ $\times 10^9$				Number of wells			
	Initial	$(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$	$\boldsymbol{u}$	$\hat{\boldsymbol{u}}$	first 12 wells		8 infill wells	
					N_{Inj}	N_{Pro}	N_{Inj}	N_{Pro}
Seed 1	7.33	10.03	10.05	10.52	3	9	3	5
Seed 2	7.80	9.88	9.90	10.31	3	9	4	4
Seed 3	8.05	10.07	10.10	10.58	4	8	5	3
Average	7.73	10.00	10.02	10.47	3	9	4	4

3.2.3 Investigation on drilling order effect

In both examples (the channelized reservoir and the Brugge model), a sequential optimization of $(\boldsymbol{x}, \boldsymbol{T})$ followed by $\boldsymbol{O}$ was carried out. In the optimization of $(\boldsymbol{x}, \boldsymbol{T})$, all wells are assumed to start operating from time 0 and the various candidate solutions during all generations have a large range, i.e., approximately \$4 × 10^9 for the channelized reservoir model and \$7 × 10^9 for the Brugge model. Based on the optimal $(\boldsymbol{x}, \boldsymbol{T})$ obtained, the candidate solutions in the subsequent drilling order optimization only have a small range which is approximately \$280 × 10^6 (6% of the optimal NPV) for the channelized reservoir example and \$200 × 10^6 (2% of the optimal NPV) for the Brugge example. Nevertheless, Leeuwenburgh et al. [44] find that the drilling order has a much larger impact on the NPV's than in our cases. In [44], different drilling orders can lead to a change which is approximately 90% of the optimal NPV in one case and 17% of the optimal NPV in another case. We believe, this is because that [44] used a very short reservoir life, i.e., 645 days which is only 120 days after drilling 16 wells at an interval of 35 days, while in our work the reservoir life is much longer, i.e., 20 years. Hence, the impact of drilling order depends on the reservoir life, T_{res}, and the time required to drill a well, T_d.

1. Investigation of the effect of T_{res} on drilling order

In Example 2-B, 12 paths are selected out of 64 potential paths. Based on the optimal $(\boldsymbol{x}, \boldsymbol{T})$ obtained, a secondary drilling order optimization considering different reservoir lives is carried out. Besides the reservoir life of 20 years, two shorter reservoir lives are considered to resemble the cases in [44], one is 1.5 years which is 217.5 days after all 12 wells are drilled and the other one is 2 years which is 390 days after all 12 wells are drilled. Fig. 3.22 shows the NPV's versus number of generations during the drilling order optimization for the three different reservoir lives. Table 3.17 shows the optimal drilling order obtained. The optimal drilling orders obtained for reservoir lives of 1.5 years and 2.0 years are very similar to each other while for the reservoir life of 20 years, the first injector is drilled 30 days earlier and the drilling order of I56 and P03 are almost switched. Table 3.18 shows a summary of the NPV's, including the NPV of the worst drilling order encountered, the range of NPV's in the first generation, the optimal NPV and the optimal NPV minus the worst NPV. The results of Table 3.18 indicate that the shorter the reservoir life, the larger the ranges of the NPV's in the initial generation and the larger the improvements in NPV's using drilling order optimization. The results of Table 3.18 partially explain why Leeuweeburg et al. [44] found drilling order has a larger impact when the reservoir life is short. In industry, the reservoir life is often a few decades and the time required to drill a well is usually quite short (e.g. 1 to 3 months).



(a) 1.5 years

(b) 2 years

(c) 20 years

Figure 3.22: NPV versus generations obtained with drilling order optimization within different reservoir lives using the optimal well locations and types $(\boldsymbol{x}, \boldsymbol{T})$ obtained with seed 1, Example 2-B.

Table 3.17: Optimal drilling order within different reservoir lives using the optimal well locations and types $(\boldsymbol{x}, \boldsymbol{T})$ obtained with seed 1, Example 2-B.

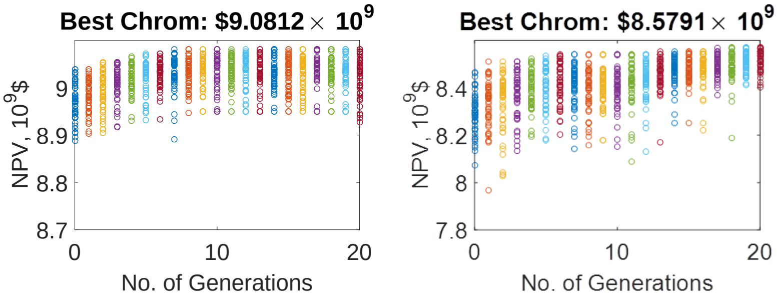
T_{res} (days)	Wells											
547.5	P03-10	P07-12	P16-5	P22-4	P24-1	P27-2	P31-3	P40-8	I42-11	I48-9	I56-6	I63-7
730.0	P03-11	P07-12	P16-4	P22-5	P24-1	P27-2	P31-3	P40-8	I42-10	I48-9	I56-6	I63-7
7300	P03-6	P07-12	P16-4	P22-9	P24-2	P27-1	P31-3	P40-5	I42-11	I48-7	I56-10	I63-8

Table 3.18: Summary of NPV's obtained with drilling order optimization within different reservoir lives using the optimal well locations and types $(\boldsymbol{x}, \boldsymbol{T})$ obtained with seed 1, Example 2-B.

T_{res}	NPV, \$ × 10^9			
years	Initial Worst	Initial Best - Initial Worst	Optimal	Optimal - Initial Worst
1.5	2.21	0.48	2.76	0.55
2.0	3.17	0.42	3.63	0.46
20	9.94	0.19	10.13	0.19

2. Investigation of the effect of T_d on drilling order

In Example 2-A, 12 paths are selected out of 30 potential paths. Based on the optimal $(\boldsymbol{x}, \boldsymbol{T})$ obtained, a secondary drilling order optimization considering different drilling time T_d is carried out. Besides $T_d = 30$ days, we also consider $T_d = 90$ days. Fig. 3.23 shows the NPV's versus number of generations during the drilling order optimization. Table 3.20 shows the optimal NPV's and drilling order obtained. Candidate solutions for $T_d = 90$ days have lower NPV's (approximately by 5.5%) than those for $T_d = 30$ days. This is because for $T_d = 90$ days, the total operation time for 12 wells are less than that for $T_d = 30$ days. For both scenarios, the first injector being drilled is I01. However, for $T_d = 30$ days, I01 is drilled as the 7th well at 180 days while for $T_d = 90$ days, I01 is drilled as the 5th well at 360 days. Applying the optimal drilling order obtained for $T_d = 30$ days to $T_d = 90$ days gives a $\$94 \times 10^6$ lower NPV, indicating that both the optimal drilling order and the optimal NPV's are dependent on the time required to drill a well, T_d. Table 3.19 shows a summary of the NPV's, including the NPV of the worst drilling order encountered, the range of NPV's in the first generation, the optimal NPV and the optimal NPV minus the worst NPV. The results of Table 3.19 indicate that the longer the drilling time, the larger the ranges of the NPV's in the initial generation and the larger the improvements in NPV's using drilling order optimization.



(a) $T_d = 30$ days (b) $T_d = 90$ days

Figure 3.23: NPV versus generations obtained with drilling order optimization with different drilling time T_d using the optimal well locations and types $(\boldsymbol{x}, \boldsymbol{T})$ obtained with seed 3, Example 2-A.

Table 3.19: Summary of NPV's obtained with drilling order optimization within different drilling time T_d using the optimal well locations and types $(\boldsymbol{x}, \boldsymbol{T})$ obtained with seed 3, Example 2-A.

T_d	NPV, \$ $\times 10^9$			
days	Initial Worst	Initial Best - Initial Worst	Optimal	Optimal - Initial Worst
30	8.89	0.15	9.08	0.20
90	8.07	0.39	8.58	0.51

Table 3.20: Optimal drilling order with different drilling time T_d using the optimal well locations and types $(\boldsymbol{x}, \boldsymbol{T})$ obtained with seed 3, Example 2-A.

T_d (days)	NPV, \$ $\times 10^9$	Wells											
30	9.08	I01-7	P02-3	P03-2	P07-6	P08-12	I09-9	P11-5	P13-10	P16-4	P19-1	I21-11	I25-8
90	8.58	I01-5	P02-3	P03-2	P07-9	P08-12	I09-10	P11-8	P13-11	P16-4	P19-1	I21-7	I25-6

3.2.4 Investigation on population size

In a traditional genetic algorithm (GA), the population size is set by the user to a fixed value at the beginning of the search and remains constant through the entire run. Specification of the population size is problematic in many ways. If it is too small, then GA may not be able to reach high quality solutions. If it is too large, then GA spends unnecessary computational resources, and, in fact, may require more computational resources than can be allocated. The scenarios are illustrated in Fig. 3.24.

Finding an adequate population size is a difficult task. It has been shown, both theoretically and empirically, that the optimal population size is something that differs from problem to problem. A somewhat widely accepted intuition towards population size is that it should be set proportionally to the problem's size and difficulty. However, problem difficulty is very hard to define for real-world problems. Faced with such difficulties, many users end up either using a so-called "standard setting" (50-100 population size) or rule of thumb that the population size be twice the length of the chromosome [59], guessing a number, or doing experimentation with a number of different sizes to see which one works best. In this section, we investigate the effect of population size on GA performance based on Example 2-A.

In addition to $N_p = 50$ used previously, three more different population sizes are tested for the joint optimization of well locations and types in case A of Example 2. The

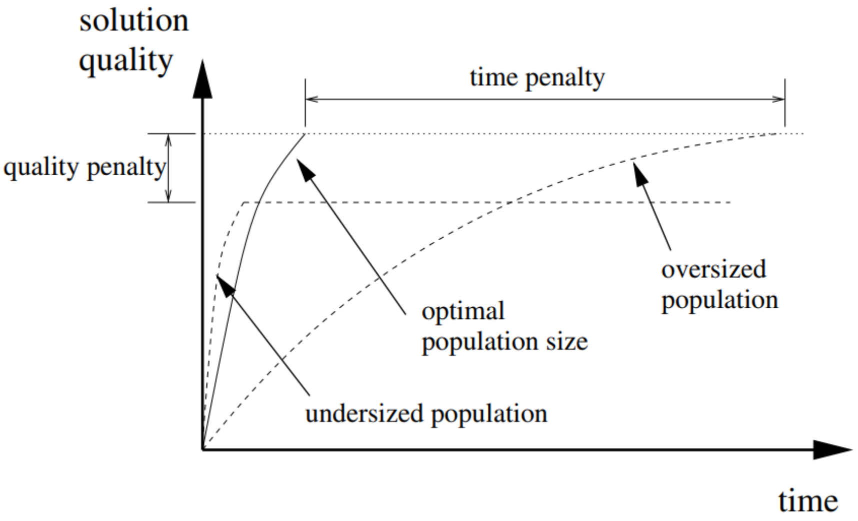
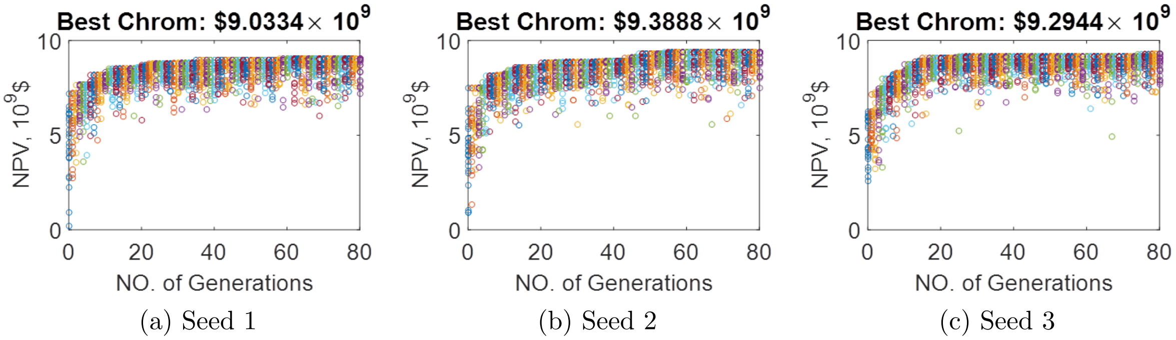


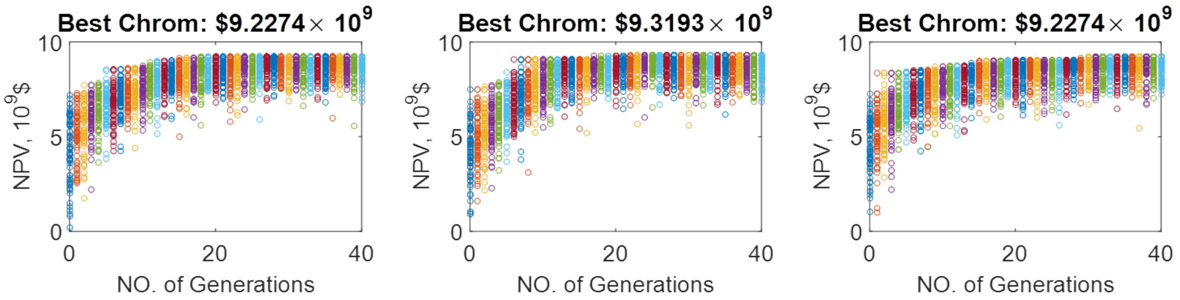
Figure 3.24: Effect of population size on GA performance from Lobo et al. [51].

sizes tested are $N_p = 24, 72$, and 108 which are equivalent to $\frac{1}{3}$, $\frac{1}{1}$ and $1\frac{1}{2}$ of the chromosome length respectively. Figs. 3.25, 3.26 and 3.27 respectively show the NPV versus generations obtained with three seeds using $N_p = 24, 72$, and 108. The corresponding results for $N_p = 50$ are shown in Fig. 3.15. As one can see, seed 1 got stuck at a sub-optimal solution using $N_p = 24$ and the optimal results obtained using $N_p = 50,\ 72,\ 108$ do not vary significantly from seed to seed. Most importantly, with an increase in N_p, better optimal NPV's are obtained. Table 3.21 shows a summary of the NPV's obtained including the one using $N_p = 50$. From Table 3.21, $N_p = 50$ is the best choice given 2000 simulation runs. The general rule in GA literature for the optimum population size (twice the number of bits used in a chromosome) is not necessary for this problem.



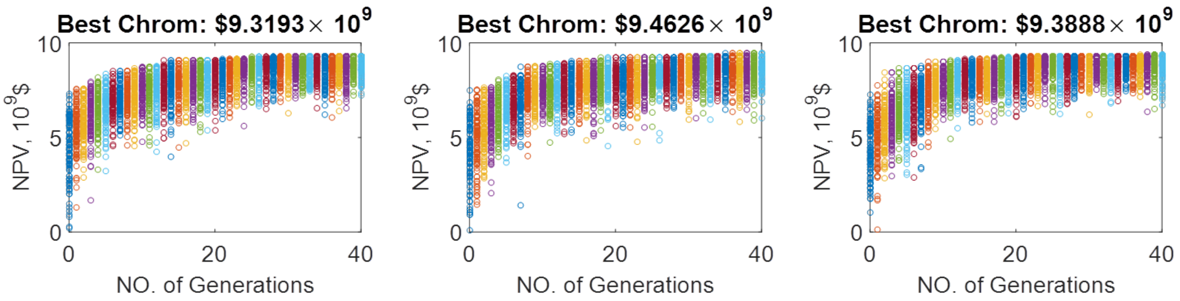
(a) Seed 1 (b) Seed 2 (c) Seed 3

Figure 3.25: NPV versus generations obtained for the optimization of well types and locations using three seeds, N_p=24, Example 2-A.



(a) Seed 1 (b) Seed 2 (c) Seed 3

Figure 3.26: NPV versus generations obtained for the optimization of well types and locations using three seeds, N_p=72, Example 2-A.



(a) Seed 1 (b) Seed 2 (c) Seed 3

Figure 3.27: NPV versus generations obtained for the optimization of well types and locations using three seeds, N_p=108, Example 2-A.

Table 3.21: Best NPV's obtained by optimizing well locations and types with about 2000 reservoir simulation runs using different population sizes, Example 2-A.

Item	Optimal NPV, $\times 10^9$\$							
	N_p=24		N_p=50		N_p=72		N_p=108	
Iterations	40	80	20	40	20	40	20	40
Simulations	960	1920	1000	2000	1440	2880	2160	4320
Seed 1	8.96	9.03	9.07	9.29	9.23	9.23	9.09	9.32
Seed 2	8.91	9.39	9.20	9.32	9.27	9.32	9.23	9.46
Seed 3	9.15	9.29	8.88	9.34	9.05	9.23	9.27	9.39
Average	9.01	9.24	9.05	9.32	9.18	9.26	9.20	9.39

3.3 Comparison with StoSAG Using Priority Parametrization

3.3.1 StoSAG using priority parametrization

Somewhat similar to our work, Leeuwenburgh et al. [44] selected N_w wells from a given set of K_{max} potential wells with $K_{max} > N_w$ and determined the drilling order of the N_w wells using the priority parameterization. Recall that in the priority parametrization, (i) each well is assigned one continuous priority variable for drilling order, and (ii) the larger the priority variable, the earlier the well is drilled. Unlike our work, they did not consider the well type as a distinct optimization variable. Instead, for each of the K_{max} potential well paths, they assigned a fixed well type. Then, they fix the well controls and optimize only the set of priority variables where each priority variable corresponds to one specific well out of the K_{max}, and each priority variable is restricted to be in the interval [0, 2]. At each iteration of the StoSAG algorithm, the N_w wells with the N_w highest priority variables are selected to be drilled and the drilling order is based on the ordering of the priority variables where the well with the highest value of its priority variable is drilled first. The paths of these N_w wells and their drilling order along with the values of the fixed well controls are input into the reservoir simulator to evaluate the NPV of production. In order to avoid having multiple priority variables reach the same bound, [44] introduced an inequality constraint on the sum of the priority variables and used the interior point method to solve the constrained optimization problem. However, their numerical results show that, this methodology may lead to lower NPV's than the initial guess which is feasible but suboptimal; see the cyan curve in Fig. 3.28 (or equivalently Fig. 3 of [44]).

To avoid solving a constrained optimization problem, they proposed to reset the updated priority variables (after each iteration that improves NPV) to pre-set values with equal intervals, such that only the ordering of values of priority variables changes at each iteration. However, their numerical results showed that this method underperformed the one in which the best performing perturbed vector of priorities is selected as the trial update as opposed to using StoSAG which underscores our concern that it may be difficult to obtain

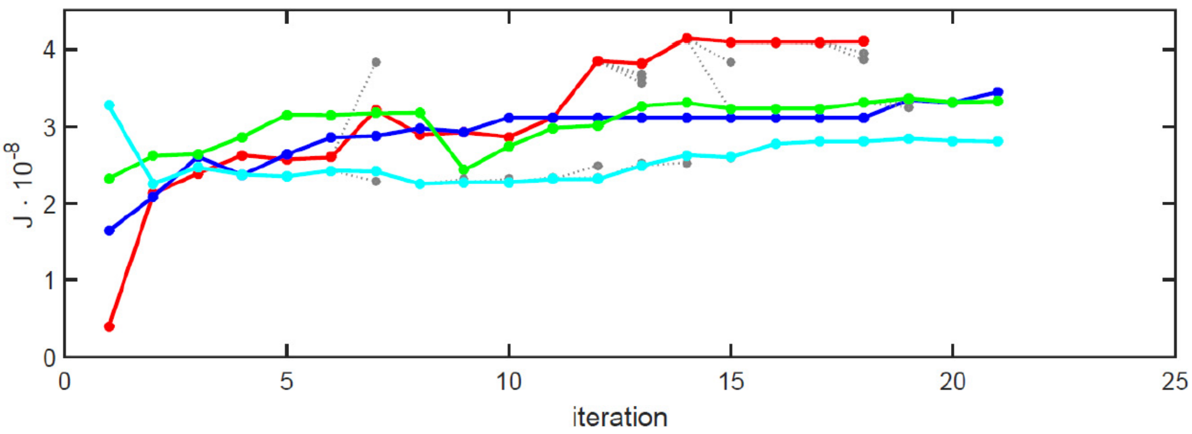


Figure 3.28: NPV versus iterations for the drilling order optimization process starting from four different initial guesses (Fig. 3 in [44]).

good stochastic derivatives of the objective function when drilling order is parameterized using continuous priority variables; see Fig. 3.29 or equivalently Fig. 6 of [44].

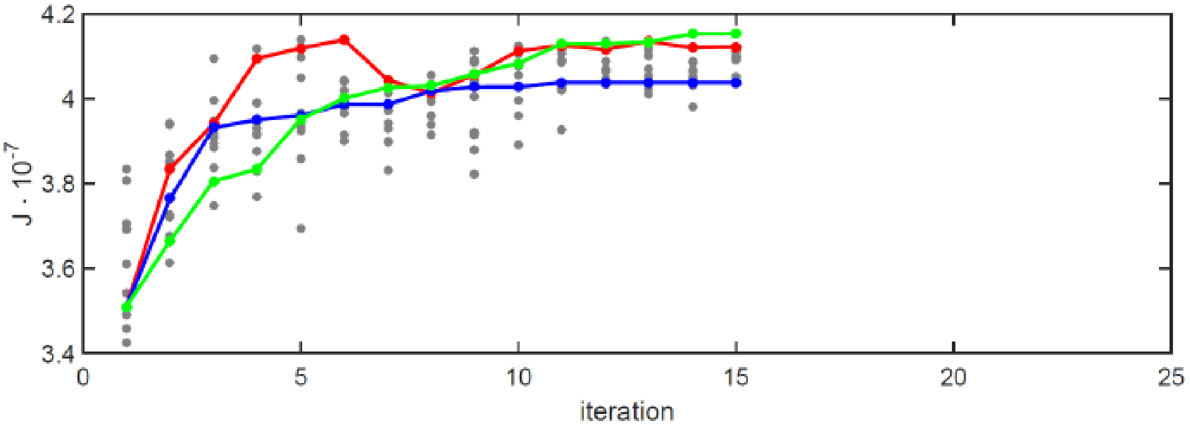


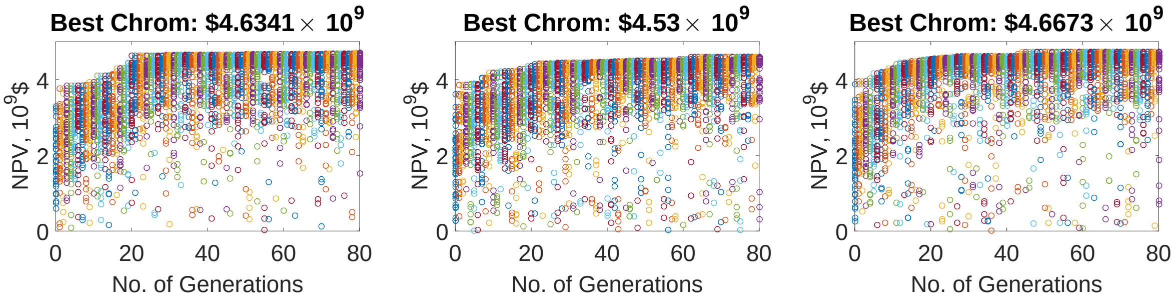
Figure 3.29: NPV versus iterations for the drilling order optimization of 4 injectors and 8 producers (Fig. 6 in [44].) Trial update using best performing perturbed priority sample is shown in red line where individual samples are shown as gray dots. Trial updates obtained with line search using StoSAG gradient estimated from normal distributed perturbations is shown in blue line. Trial updates obtained with line search using StoSAG gradient estimated from Bernoulli perturbations is shown in green line.

3.3.2 Numerical examples

Next, we apply a modified procedure of [44] to the two examples in last section. We do not consider one of their examples because they do not provide enough information to reproduce their example results and, even if they did, it would be preferable to consider a more complicated problem. Leeuwenburgh et al. [44] consider two procedures for updating the vector of priority variables at each iteration. In the first procedure, they generate a set of perturbations of the current estimate of this priority vector and use the stochastic gradient (StoSAG). Then they consider potential updates obtained by taking steps in the direction of StoSAG to try to find an updated vector that increases NPV. In the second procedure, they simply consider the perturbed vector of priorities and select as the update the one that gives the largest NPV. In our modification, we include in the total set of trial updates both the perturbed priorities and the updates generated by taking steps in the StoSAG direction. From this total set, we select the one that gives the highest NPV. Since this modification makes use of both the random search ability and the search ability along the StoSAG gradient, it seems clear that it should outperform both individual update methods considered in [44].

For both Example I (the channelized reservoir) and case A of Example 2 (the Brugge model), there are 30 potential locations. Following [44], each of the 30 potential locations is assigned one priority variable, based on which, the N_w well locations and drilling order can be determined. Besides the 30 priority variables, we also added the type variables. For the channelized reservoir, 30 well type variables are added. For case A of the Brugge model, only 20 well types variables are added since 10 out of 30 potential locations are in the aquifer which can only have "injector" type. The priority variables are preset as values from 0.1 to 3.0 with an interval of 0.1; the type variables are preset as either 0.25 or 0.75. If a type variable is greater than 0.5, an injector is placed; otherwise, a producer is placed. At each iteration, the perturbation is sampled from Gaussian distribution $\mathcal{N}(0, 0.2^2)$ for priority variables and $\mathcal{N}(0, 0.5^2)$ for type variables. The initial step size in the normalized StoSAG direction is set as 0.5, the maximum number of step-size cuts is set as 3, and the maximum number of perturbation resamples is set as 5.

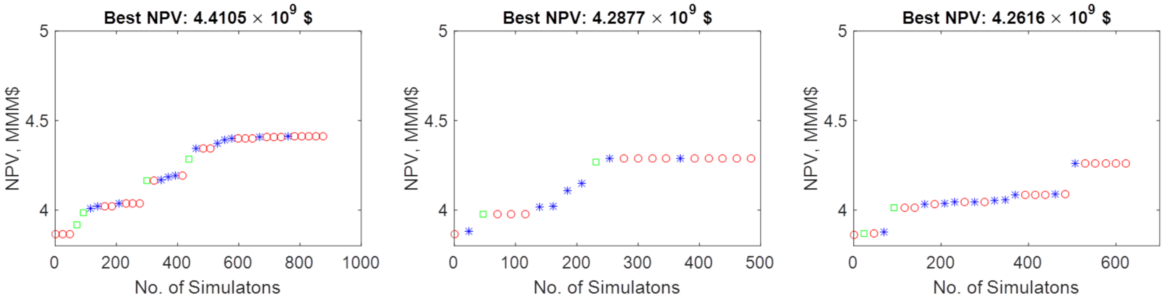
For the channelized reservoir, Fig. 3.30 shows the NPV's versus generations obtained with simultaneous optimization of well locations and types using GA for three seeds. The initial guess of the priority method is chosen as the best individual of the 50 random samples of the well locations, types and drilling order generated at the first iteration of GA using seed 3 in Fig. 3.30. Fig. 3.31 shows the NPV versus the number of iterations using three seeds following the modified methodology of [44]. The green square represents the case where the update is the best performing perturbation, the blue star represents the scenario where the update is along the StoSAG search direction, and the red circle represents an iteration where no update that increases the NPV is obtained. When this occurs, new perturbations are generated. If at one iteration, we generate five sets of perturbations without improving NPV, the algorithm is terminated. As one can see, almost half of the iterations fail to improve the NPV's by searching along the StoSAG gradient direction. Moreover, the average of the optimal NPV's obtained in Fig. 3.31 is approximately 6% less than that generated with GA in Fig. 3.30.



(a) Seed 1 (b) Seed 2 (c) Seed 3

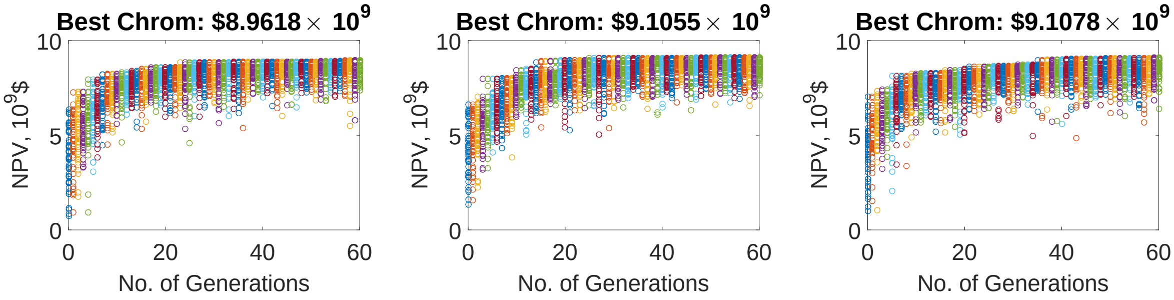
Figure 3.30: NPV versus generations obtained with simultaneous optimization of $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ using three seeds, Bound A, Example 1.

For case A of Example 2, Fig. 3.32 shows the NPV versus generations obtained with simultaneous optimization of well locations, types and drilling order using GA with three seeds. The initial guess of the priority method is chosen as the best individual of the 50 random samples generated at the first iteration of GA using seed 2 in Fig. 3.32. Fig. 3.33 shows the NPV versus the number of iterations using three seeds following the modified methodology of [44]. As one can see, most of the NPV gain is obtained by random perturbations while searching along the StoSAG direction only yields a slight increase. Moreover, the optimal NPV's obtained in Fig. 3.33 are approximately 8% less than the NPV's generated with GA in Fig. 3.32.



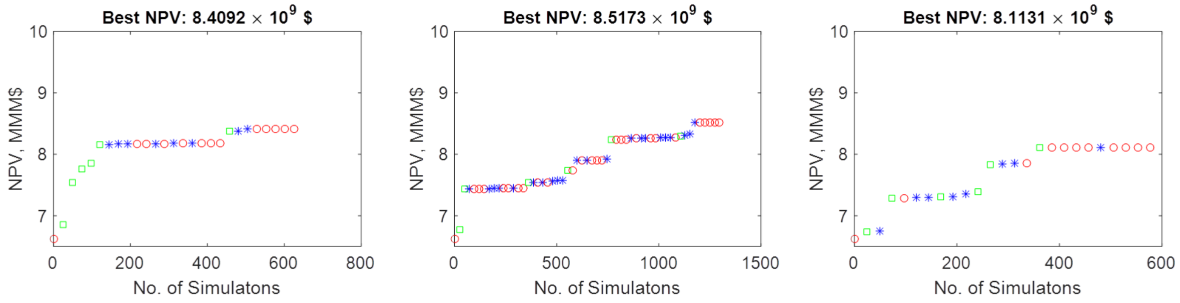
(a) Seed 1 (b) Seed 2 (c) Seed 3

Figure 3.31: NPV versus iterations obtained with simultaneous optimization of $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ following the methodology of [44] using three seeds, Example 1. The green square represents a successful update of the best performing perturbation, the blue star represents a successful update along the StoSAG search direction, and the red circle represents a iteration where no successful updates is obtained.



(a) Seed 1 (b) Seed 2 (c) Seed 3

Figure 3.32: NPV versus generations obtained with simultaneous optimization of $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ using three seeds, Example 2-A.



(a) Seed 1 (b) Seed 2 (c) Seed 3

Figure 3.33: NPV versus iterations obtained with simultaneous optimization of $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ following the methodology of [44] using three seeds, Example 2-A. The green square represents a successful update of the best performing perturbation, the blue star represents a successful update along the StoSAG search direction, and the red circle represents a iteration where no successful updates is obtained.

3.4 Comparison with a Gradient-based Algorithm

As mentioned in Chapter 1, [83, 89, 26] used a gradient-based method to optimize the number, types and control settings of wells given K_{max} potential locations, a problem similar to the one considered in this chapter. Hence, in this section, we provide a comparison between GA and a gradient-based algorithm.

3.4.1 A gradient-based algorithm

In the gradient-based method in [83, 89, 26], a well is placed at each potential location with a pre-specified type (injector or producer) and a liquid rate $q_j, j = 1, 2, \cdots, K_{max}$ which applies during the whole reservoir life. The liquid rates are the optimization variables and if $q_j \to 0$, well j is not drilled. The cost of drilling a well in location j is then approximated with a continuous and differentiable function of the well rate, $f(q_j)$, and the actual drilling cost, $C_{w,j}$. The total drilling cost is $\sum_{j=1}^{K_{max}} f(q_j)C_{w,j}$. It is desirable to have $f(q_j) = 0$ when $q_j = 0$ and $f(q_j) = 1$ when q_j is large. By optimizing the control settings of K_{max} wells and by driving the rates to zero one by one to reduce the drilling cost, an approximation of the optimal well number, types, locations and rates can be eventually obtained. If the algorithm is terminated when the number of wells is equal to N_w, the same problem of interest is solved, which is to select N_w wells from K_{max} potential locations. Major disadvantages of this procedure are that (i) it is only applicable to wells under rate control, (ii) only one control step can be used and (iii) it cannot optimize the drilling order although that could be obtained by a sequential optimization. Since only one control step is considered, this problem can also be efficiently solved using GA by optimizing the well rates, types and locations simultaneously. It is interesting to compare the optimal results obtained with the gradient-based method and GA.

Wang et al. [83] proposed to define $f(q_j^\ell)$ using the following function

$$f(q_j^\ell) = \frac{q_j^\ell}{q_j^\ell + \beta}, \tag{3.12}$$

where, β is a small constant number, 10^{-10}; ℓ represents the ℓth iteration; q_j is the injection or production rate of well j at ℓth iteration. Hence, $\frac{\partial f(q_j^\ell)}{\partial q_j^\ell} = \frac{\beta}{(q_j^\ell+\beta)^2} \approx \frac{\beta}{\left(q_j^\ell\right)^2}$. When β is small, Eq. 3.12 tends to be insensitive to the change of controls which is not desirable. Thus, Forouzanfar and Reynolds [26] proposed to use the following function for $f(q_j^\ell)$

$$f(q_j^\ell) = \left(\frac{q_j^\ell}{q_j^{\ell-1}}\right)^{0.25}, \tag{3.13}$$

where, $q_j^{\ell-1}$ represents the control variable (flow rate) of well j at $(\ell-1)$th iteration. However, the above function can take values significantly different than 1.0 for $q_j^\ell \neq 0$. To avoid this difficulty, Forouzanfar and Reynolds [26] simply set $f(q_j^\ell) = 1.0$ if $q_j^\ell \neq 0$ and $f(q_j^\ell) = 0.0$ if $q_j^\ell = 0$. Hence there is an inconsistency in the evaluation of $f(q_j^\ell)$ and $\frac{\partial f}{\partial q_j^\ell}$. To ensure enough sensitivities and keep the consistency, we propose to use a logistic form for $f(q_j^\ell)$ given by,

$$f(q_j^\ell) = \frac{1 - e^{-\omega q_j^\ell}}{1 + e^{-\omega q_j^\ell}}, \tag{3.14}$$

where, ω is a weighting factor to control the magnitude of $\frac{\partial f}{\partial q_j^\ell}$. The logistic function is composed of one steep part for small values of ωq_j^ℓ where $\omega q_j^\ell \ll 5$ and one flat part where $f(q_j^\ell) \approx 1$ for $\omega q_j^\ell > 5$. Here, based on a computational experiment, ω is chosen to satisfy $\omega q_j^{crit} \approx 5$ (see Fig. 3.34) where q_j^{crit} is an estimate of the minimum liquid rate necessary for well j to be profitable. q_j^{crit} is obtained by solving

$$q_j^{crit} * (r_o - r_w)T_{res} - C_{w,j} = 0, \tag{3.15}$$

where r_o is the oil price, r_w is the water injection cost, and T_{res} is the reservoir life. Assuming balanced injection and production, the left hand side of Eq. 3.15 represents the maximum net revenue (non-discounted) for a well with rate q_j^{crit}, i.e., the revenue of the oil production (for a producer) or the displaced oil (for an injector) deducted by the water injection cost.

Besides the new form for $f(q_j^\ell)$, two more modifications are made based on [26] in the implementation of the gradient-based method. The first modification is that each feasible

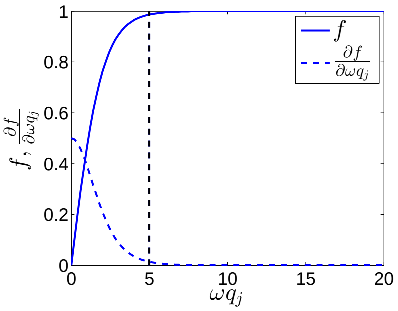


Figure 3.34: Illustration of the logistic function and its derivative in Eq. 3.14.

potential location is assigned two wells of different types (one injector and one producer) with same initial rates to avoid bias due to pre-specifying the type of each well as was done in [26]. Hence, the optimization variable set is $\boldsymbol{u} = [q_1^I, q_1^P, q_2^I, q_2^P, \cdots, q_{K_{max}}^I, q_{K_{max}}^P]$, where the superscript "I" represents an injector and the superscript "P" represents a producer. The objective is to eliminate wells one by one until only N_w wells are left while maximizing the NPV function using the adjoint gradient $\nabla_{\boldsymbol{u}} J(\boldsymbol{u}^\ell)$ where $J(\boldsymbol{u}^\ell)$ is given by

$$\begin{aligned} J\left(\boldsymbol{u}^\ell\right) = & \sum_{n=1}^{N_t} \left\{ \frac{\Delta t^n}{(1+b)^{\frac{t^n}{365}}} \left[\sum_{j=1}^{N_P} \left(r_o^n \cdot \overline{q}_{o,j}^n - c_w^n \cdot \overline{q}_{w,j}^n - c_g^n \cdot \overline{q}_{g,j}^n \right) - \sum_{k=1}^{N_I} \left(c_{wi}^n \cdot \overline{q}_{wi,k}^n + c_{gi}^n \cdot \overline{q}_{gi,k}^n \right) \right] \right\} \\ & - \sum_{j=1}^{2K_{max}} f(u_j^\ell) C_{w,j}. \end{aligned} \tag{3.16}$$

The second modification made to the algorithm used in [26] is that we removed the equality constraint on the total injection (or production) rate so that the well control settings can be adjusted more flexibly, i.e., only bound constraints are enforced. The lower bound, $\boldsymbol{u}^{\text{low}}$, is a $2K_{max}$-dimensional vector whose elements are zero and the upper bound, $\boldsymbol{u}^{\text{up}}$, is a $2K_{max}$-dimensional vector whose elements are the maximum allowable rate for each well. The bound constraints are enforced using the gradient projection method discussed below. Rewrite the active constraints in the following form

$$A\boldsymbol{u}^\ell = \boldsymbol{b}, \tag{3.17}$$

where, A is the $N_{active} \times 2K_{max}$-dimensional sensitivity matrix of the active bound constraints,

N_{active} is the number of active bound constraints and $\boldsymbol{b}$ is an N_{active}-dimensional vector whose element can be either 0 if a lower bound constraint is active or u_j^{up} if an upper bound constraint is active. The projected search direction can be obtained as

$$\boldsymbol{d}^\ell = (I - A^T(AA^T)^{-1}A)\nabla_{\boldsymbol{u}} J(\boldsymbol{u}^\ell). \tag{3.18}$$

At each iteration, $\nabla_{\boldsymbol{u}} J(\boldsymbol{u}^\ell)$ can be obtained by subtracting $\nabla_{\boldsymbol{u}} \left\{\sum_{j=1}^{2*K_{max}} f(q_j^\ell)C_{w,j}\right\}$ from the adjoint gradient outputted from the reservoir simulator. $\nabla_{\boldsymbol{u}} J(\boldsymbol{u}^\ell)$ is then projected onto the active bound constraints using Eq. 3.18 and an aggressive initial step size α_{max}^ℓ along $\boldsymbol{d}^\ell$ is then tried to bring at least one component of $\boldsymbol{u}$ to either its lower or upper bound; see Eqs. 3.19 and 3.20. If taking a step $\alpha_{max}^\ell \boldsymbol{d}^\ell$ fails to improve the objective function, backtracking is implemented with a maximum number of step-size cuts equal to 5. If after 5 cuts, the trial update still fails in improving the objective function, we accept the initial trial step $\alpha_{max}^\ell \boldsymbol{d}^\ell$ even though this decreases the NPV. We continue the iterations until there are only N_w wells left or $\boldsymbol{d}^\ell$ is empty. The detailed algorithm is shown in Algorithm 4.

$$\alpha_{max}^\ell = \min\{\alpha_{max,j}^\ell\}_{j=1}^{N_u}. \tag{3.19}$$

$$\alpha_{max,j}^\ell = \begin{cases} \frac{u_j^{\text{low}} - u_j^\ell}{d_j^\ell}, & \text{if } d_j^\ell < 0, \\ \frac{u_j^{\text{up}} - u_j^\ell}{d_j^\ell}, & \text{if } d_j^\ell > 0, \end{cases} \quad \text{for } j = 1, 2, \cdots, N_u. \tag{3.20}$$

3.4.2 Optimization of $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{u})$ using GA

It is computationally feasible to optimize the well control settings using GA when the number of control steps is small. As the well control variables are continuous variables, it is suitable to use a real value encoding. However, for the joint optimization of well locations, types and controls using GA, well types change from generation to generation and the bounds on controls for injectors and producers are usually different. For example, if injectors are under rate control and producers are under BHP control where the upper bound on the

Algorithm 4 Pseudo-code for the gradient-based optimization

```
1. Place one injector and one producer at each of the $K_{max}$ potential paths. Set $q_j^{crit}$, $j = 1, 2, \cdots, 2 * K_{max}$ and $\omega$. Formulate the vector of the initial control variables, $\boldsymbol{u}^0$. Set the number of existing wells as $2 * K_{max}$, the maximum number of step-size cuts $n_{sc} = 5$ and the number of step-size cuts $n_{cuts} = 0$. Set $\ell = 0$.
2. Calculate the gradient $\nabla_{\boldsymbol{u}} J(\boldsymbol{u}^\ell)$ where $J(\boldsymbol{u}^\ell)$ is given in Eq. 3.16.
3. Find the active constraints at the estimate of $\boldsymbol{u}^\ell$, form $A$ of Eq. 3.17 and then calculate $\boldsymbol{d}^\ell$ using Eq. 3.18.
4. If the number of existing wells is equal to $N_w$ or $\boldsymbol{d}^\ell$ is empty, terminate the algorithm.
5. Calculate $\alpha_{max}^\ell$ using Eqs. 3.19 and 3.20. Set $\alpha^\ell = \alpha_{max}^\ell$.
6. Take a trial step, $\boldsymbol{u}^{\ell+1,trial} = \boldsymbol{u}^\ell + \alpha^\ell \boldsymbol{d}^\ell$, run reservoir simulation and evaluate $J(\boldsymbol{u}^{\ell,trial})$.
7. If $J(\boldsymbol{u}^{\ell,trial}) > J(\boldsymbol{u}^\ell)$, set $\boldsymbol{u}^\ell = \boldsymbol{u}^{\ell,trial}, \ell = \ell + 1$, $n_{cuts} = 0$.
8. Otherwise, check if $n_{cuts} < n_{sc}$. If yes, set $\alpha^\ell = \alpha^\ell/2$ and $n_{cuts} = n_{cuts} + 1$, go to step 6; otherwise, set $\alpha^\ell = \alpha_{max}^\ell$ and update $\boldsymbol{u}^{\ell+1} = \boldsymbol{u}^\ell + \alpha^\ell \boldsymbol{d}^\ell$, set $\ell = \ell + 1$ and $n_{cuts} = 0$.
9. Calculate the number of existing wells as number of potential wells with non-zero rate. Go to step 2.
```

injection rate is much larger than the upper bound on the production BHP's, it is possible that after crossover and mutation, one producer is assigned a BHP value which is greater than its upper bound, which thus may shut-in this producer; if both injectors and producers are under BHP control, it is quite often that one producer may be shut-in if it switches to an injector without changing the BHP's in the next generation. To avoid these potential problems, choking factors $\hat{\boldsymbol{u}}$ between 0 to 1 are used as the design variables instead of well controls $\boldsymbol{u}$. If a well is an injector ($T_j = 1$), the actual control variable $u_i^{(j)}$ is then given by

$$u_i^{(j)} = u_I^{\text{low}} + \hat{u}_i^{(j)}(u_I^{\text{up}} - u_I^{\text{low}}), \tag{3.21}$$

where, u_I^{low} and u_I^{up} are the lower bound and upper bound specified for the control variables of injectors, $i = 1, 2, \cdots, N_w$, $j = 1, 2, \cdots, N_c$. If a well is a producer ($T_j = 0$), the actual control variable $u_i^{(j)}$ is given by

$$u_i^{(j)} = u_P^{\text{low}} + \hat{u}_i^{(j)}(u_P^{\text{up}} - u_P^{\text{low}}), \tag{3.22}$$

where, u_P^{low} and u_P^{up} are the lower bound and upper bound specified for the control variables of producers.

For real value encoded chromosomes, linear recombination is commonly used as the crossover operation where the variables of the offsprings are chosen as the linear combinations of variables of the parents ($\boldsymbol{P}_1$ and $\boldsymbol{P}_2$) using

$$\boldsymbol{C}_1 = a\boldsymbol{P}_1 + (1-a)\boldsymbol{P}_2. \tag{3.23}$$

Here, a is a scaling factor randomly sampled from the uniform distribution $\mathcal{U}[-d, 1+d]$. A value of $d = 0.25$ ensures (statistically) that the variable range of the offspring is the same as the variable range spanned by the variables of the parents [61]. In the mutation, if $\hat{u}_i^{(j)}$ is to be mutated, "controlled mutation" is applied, i.e., it will be replaced by a random value sampled from the uniform distribution given by $\mathcal{U}[0.9\hat{u}_i^{(j)}, 1.1\hat{u}_i^{(j)}]$. If the variables of the produced offsprings are outside the specified bounds, we simply truncate them to the nearest bounds.

To jointly optimize N_w well locations, types and control settings, we consider mixed encodings of real values and binary values of

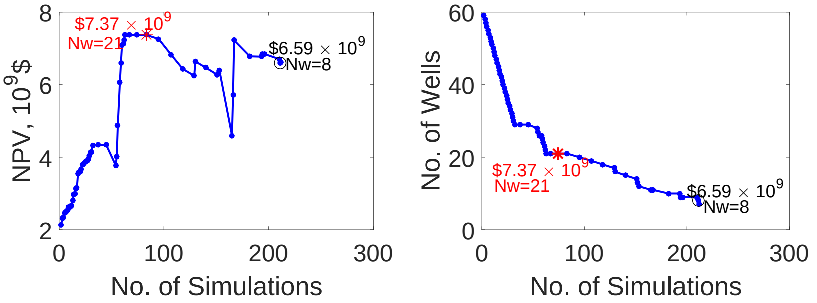
$$\boldsymbol{c} = [\boldsymbol{x}, \boldsymbol{T}, \hat{\boldsymbol{u}}]. \tag{3.24}$$

For the mixed encoding in Eq. 3.24, a similar procedure discussed for the crossover/mutation operations for the chromosome of $(\boldsymbol{x}, \boldsymbol{T}, \boldsymbol{O})$ is applied. If the randomly generated crossover point falls in the part of $\boldsymbol{c}$ corresponding to the well control settings $\hat{\boldsymbol{u}}$, then the linear recombination crossover is carried out for $\hat{\boldsymbol{u}}$; if the crossover point falls in the gene parts corresponding to $(\boldsymbol{x}, \boldsymbol{T})$, then chromosomes after the crossover point are exchanged using the modified one-point crossover scheme discussed earlier. Similarly, the mutation operation starts with the common gene-by-gene mutation. If any binary gene describing $(\boldsymbol{x}, \boldsymbol{T})$ is to be mutated, it will be flipped, i.e., 0 is changed to 1 or vice versa; if any of the gene describing the well control settings is to be mutated, “controlled mutation” will be applied.

3.4.3 Numerical example: channelized reservoir

Based on the channelized reservoir example introduced in Section 3.2.1, we now compare the performance of the gradient-based method [26] and GA. We consider the case that both injectors and producers are under liquid rate control. The lower and upper bounds for the injection and production rates for all wells are set as 0 STB/day and 40,000 STB/day respectively. In this example, the critical rate q_j^{crit} (see Eq. 3.15) is approximately 50 STB/day for a vertical path and 100 STB/day for a horizontal path, the weighting factor ω is set as 0.1 and initial rate for each well at all potential locations is set as 1,200 STB/day. The optimal rates, locations and types of $N_w = 8$ wells are determined. Fig. 3.35 shows the NPV versus simulation runs (on the left) and the number of remaining wells versus simulation runs (on the right) obtained with the gradient-based method. Note that the NPV does not increase monotonically with respect to the simulation runs. When all well rates are far greater than q_j^{crit}, this problem reduces to a pure control optimization problem. Whenever the back tracking line search method cannot find a step size that improves the NPV and the number of wells is still greater than N_w, an aggressive step size is taken to bring at least one component of $\boldsymbol{u}$ to either its lower or upper bound. Taking this aggressive step size may decrease the NPV. From Fig. 3.35, we see that this NPV decrease happens frequently after about 100 simulation runs. The highest NPV obtained is \$7.37 × 10^9 with 21 wells. When the number of wells is reduced to 8, the NPV is reduced to \$6.59 × 10^9.

Fig. 3.36 shows the NPV’s versus the number of generations obtained with GA. With GA, the optimization variables are the locations of the 8 wells selected out of the 30 paths, $\boldsymbol{x}$, the type of the 8 wells, $\boldsymbol{T}$, and the vectors of the scaled well controls, $\hat{\boldsymbol{u}}$. A chromosome thus has the form $(\boldsymbol{x}, \boldsymbol{T}, \hat{\boldsymbol{u}})$. Binary encoding is used for $(\boldsymbol{x}, \boldsymbol{T})$. For $\hat{\boldsymbol{u}}$, real value encoding is used with a linear recombination crossover and controlled mutation. The optimal NPV obtained with GA is \$7.69 × 10^9 which is 14% higher the NPV obtained using the gradient-based method. Moreover, the optimal NPV of GA is also higher than the best NPV that the gradient-based algorithm achieved (\$7.37 × 10^9 when the number of well is 21). Fig. 3.37 shows the optimal well placement and the remaining oil saturation field of layer 7. Fig. 3.38 shows the optimal well injection/production rates obtained with the gradient-based method (left) and GA (right) respectively. The gradient-based algorithm finds one injector and seven producers while the optimal result obtained with GA indicates that we should select 2 injectors and 6 producers.



(a) NPV versus simulation runs (b) Number of wells versus simulation runs

Figure 3.35: NPV and number of wells versus simulation runs for the joint optimization of well locations, types and rates obtained with the gradient-based method, Example 1.

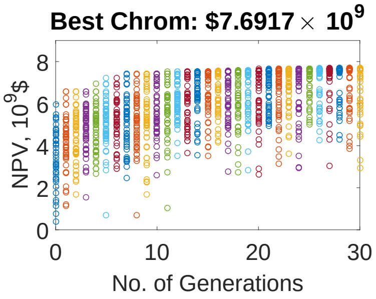
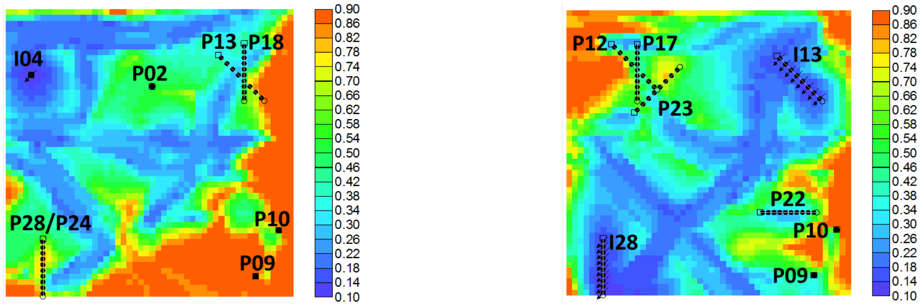


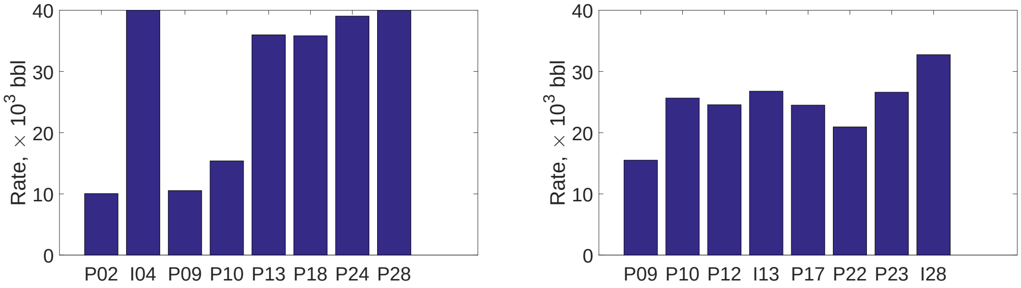
Figure 3.36: NPV versus generations for the joint optimization of well locations, types and rates obtained with GA, Example 1.



(a) Gradient-based

(b) GA

Figure 3.37: Optimal well configurations obtained with the gradient-based method and GA method respectively, plotted on the oil saturation fields of layer 7 at the end of 4,000 days, Example 1.



(a) Gradient-based

(b) GA result

Figure 3.38: Optimal well injection/production rates obtained with the gradient-based method and GA, Example 1.

3.5 Summary and Discussion

In this chapter, we considered the problem of maximizing the NPV by optimizing the well locations, types and drilling order given a set of potential drilling paths and the number of wells that are authorized to be drilled. After the categorical variables are optimized, we optimize well controls. It is important to note the approach presented here would likely become too computationally expensive to be practical if the potential drilling paths are not restricted to a reasonably small number. Up to now, we have considered a maximum of 64 potential drilling paths. Increasing the number to 128 or more would likely require a computational cluster to be computationally feasible. Because we only consider selecting N_w well paths from K_{max} potential well paths, the optimization of the paths, types and drilling order can be accomplished with the Genetic Algorithm (GA). If, however, the well paths are allowed to be anywhere in the reservoir, it would be more natural to parametrize well trajectories (paths) as continuous variables and use StoSAG as the optimization algorithm [52, 90]. This is discussed in Chapter 5.

In implementations of GA, mixed encodings are applied where the well locations and types are parameterized using binary encoding and the drilling sequence is parameterized using ordinal numbers. Suitable mutation and crossover operations for the mixed encodings are proposed and applied. The sets of encoded variables are optimized both simultaneously and sequentially. Results indicate that GA gives good solutions in the following sense: (i) the NPV produced is significantly larger than the NPV of any member of a set of initial guesses; (ii) different runs of GA produce a variety of choices of optimal well paths, but the variation in the estimated optimal NPV's is relatively small, and (iii) GA outperformed the method proposed by Leeuwenburgh et al. [44] for the simultaneous optimization of well locations, types and drilling order, and GA also outperformed a gradient-based method by [83, 89, 26] for the simultaneous optimization of well locations, types and rates.

.

CHAPTER 4

JOINT OPTIMIZATION OF WELL NUMBER, TYPES AND LOCATIONS

In addition to the problem considered in last chapter, here, we consider a more general problem where no prior information on potential well paths is given and the number of wells to be drilled is not specified. Thus, the problem becomes as follows: given a maximum of $K_{\max}$ wells authorized to be drilled, determine how many wells actually should be drilled, the path of each well and the type of each well (i.e., an injector or a producer). Again the cost function to be maximized is the net present value (NPV) of production over the life the reservoir. We firstly consider the joint optimization of well number, types and locations for a deterministic reservoir model. Then we develop an efficient robust optimization procedure when a large number of realizations are used to characterize the reservoir uncertainty.

4.1 Problem Description and Methodology

4.1.1 Problem description

Optimizing the number of wells explicitly may change the size of design variables (well types, locations, control settings and so on) from iteration to iteration of the optimization algorithm. To fix the size of the design variables, we specify the maximum number of wells authorized to be drilled as K_{max}, and then assign each well a "drill-or-not" status variable, $s_j,\ j = 1, 2 \cdots , K_{max}$. We use $s_j = 1$ to denote "to-drill" well j and $s_j = 0$ to denote "not-to-drill" well j. Hence, the number of actual drilled wells can be obtained as $N_w = \sum_{j=1}^{K_{max}} s_j$ by determining s_j for K_{max} wells. Similarly, assign each of the K_{max} wells a type indicator, T_j. As a well can only be an injector or a producer, T_j is also parameterized as a binary variable, where, $T_j = 1$ represents an injector and $T_j = 0$ represents a producer. Let $\boldsymbol{x}_j$ denote the location of a well where the size of $\boldsymbol{x}_j$ depends on the trajectory of well j. In this chapter, we only consider placement of fully-penetrating vertical wells. Hence, $\boldsymbol{x}_j$ is simply the (i, j) indicies of the gridblock that well j is placed in. Given pre-specified operating conditions for injectors and producers, well j can be characterized by $\boldsymbol{w}_j = (s_j, T_j, \boldsymbol{x}_j)$. Note that, $(T_j, \boldsymbol{x}_j)$ has an impact on the development plan only if well j is drilled, i.e., $s_j = 1$; otherwise, the $(T_j, \boldsymbol{x}_j)$'s only act like place holders in the design variable set.

With this parametrization, it is difficult to consider simultaneous optimization of drilling order, $\boldsymbol{O}$, and $(\boldsymbol{s}, \boldsymbol{T}, \boldsymbol{x})$. Denote the time required to drill a well as T_d days, then following the last chapter, $\boldsymbol{O}$ could be parameterized as a K_{max}-dimensional ordinal number, i.e., a permutation of $\{1, 2, 3, \cdots, K_{max}\}$, and well j would be put into operation at time $(O_j - 1)T_d$ days. However, even if well j is not drilled (i.e., well status of well j corresponds to do-not-drill), it still holds (reserves) T_d days of drilling time (i.e., the drilling rig does not work for the next T_d days) and because of the "rig down time," the solution obtained will be at least somewhat suboptimal. [35, 76] assign each of the K_{max} wells a label, preset their drilling order and opening time, and optimize the well status ("an injector," "do-not-drill" and "a producer"), locations and controls for these K_{max} wells. In their work, to switch the drilling order of well i and well j $(i \neq j)$, the locations and controls for well i and j have to be switched which is difficult for Mesh Adaptive Direct Search (MADS) or Particle Swarm Optimization (PSO) (optimizers used in [35, 76]) when well i and well j are far away from each other. Moreover, in their parameterization, the total number of control steps should be no less than $\frac{K_{max}(K_{max}+1)}{2}$ which can be computationally intensive for stochastic algorithms (e.g., PSO and GA). Considering that in practice, well control optimization is usually not conducted until all wells have been drilled and that it is difficult to find the optimal solution of a high-dimensional optimization problem, we only consider optimizing the drilling order and controls after the optimal $(\boldsymbol{s}, \boldsymbol{T}, \boldsymbol{x})$ has been determined. In this way, the dimension of the drilling order and control variables can be reduced to account for only $N_w = \sum_{j=1}^{K_{max}} s_j$ wells. Since in last chapter, we have discussed the optimization of drilling order and well controls, in this chapter, we only focus on the joint optimization of well number, types and locations.

4.1.2 Constraints handling

Constrained optimization problems contain two classes of constraints: hard constraints (sometimes referred to as mandatory constraints) that must be satisfied by any solution, and soft constraints, that may or may not be satisfied (Jiang et al. [38]). In the optimization problem considered in this chapter, we consider four types of constraints, including one hard constraint and three soft constraints.

1. All K_{max} wells must be placed in the active gridblocks of a reservoir.

   Denoting the set of active gridblocks as Ω_{active}, this constraint is given by

   $$\boldsymbol{x}_j \in \Omega_{active}, \quad j = 1, 2, \cdots, K_{max}. \tag{4.1}$$

   This is a hard constraint that traditional constrained optimization methods (e.g., penalty method, augmented Lagrangian method, barrier method and so on) cannot handle. Violation of this type of constraints may result in failed simulation runs where there is no way to evaluate the NPV function. In this work, whenever a well is placed in an inactive gridblock after a trial update or a perturbation (see the case of $\boldsymbol{x}_{k+1}^{j,trial}$ in Fig. 4.1), it is projected to the nearest bound of the active gridblocks (see the case of $\boldsymbol{x}_{k+1}^{j}$ in Fig. 4.1). Note that since $\boldsymbol{x}_{k+1}^{j}$ represents the (i, j) gridblock indices which are discrete, $\boldsymbol{x}_{k+1}^{j}$ has to be truncated to the nearest integer values.

2. The inter-well distance between any pair of wells has to be larger than a predefined value of the minimum allowable well spacing, which is denoted by $R_{\min}$. Let $c_{i,j}\left(\boldsymbol{s}, \boldsymbol{T}, \boldsymbol{x}\right)$ denote the well spacing constraint corresponding to well i and well j, where $c_{i,j}$ is defined by

   $$c_{i,j}\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}\right) = \max\left\{0, -(r_{i,j} - R_{\min})\right\} \leq 0, \tag{4.2}$$

   where $r_{i,j}$ is the areal distance between well i and well j. In this chapter, $r_{i,j}$ is measured in terms of number of gridblocks. There are $\frac{K_{\max}(K_{\max}-1)}{2}$ well spacing constraints at maximum.

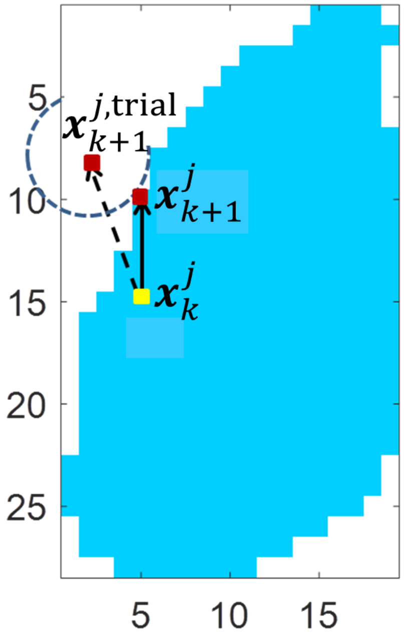


Figure 4.1: Constraint handling when a well is placed in an inactive gridblock.

Even though the $c_{i,j}$'s are soft constraints, it is difficult to couple the traditional constrained optimization methods designed for continuous variables into the mixed-integer problem in an efficient manner. In this work, we consider two methods to enforce the constraints in Eq. 4.2. One is the repair method used in Genocop III [57], which repairs infeasible solutions by combining them with feasible solutions. As an alternative, we propose a method based on a heuristic mapping. The idea is simple: whenever a pair of wells violate the well spacing constraints, they are moved away from each other along the line through the two locations until the distance between them is greater or equal to $R_{\min}$ while both wells still fall in active gridblocks. If either well hits the boundary of the active gridblocks, stop moving it and continue to move the other well until the distance is greater or equal to $R_{\min}$. Note that this operation may lead to constraint violations of other well pairs containing either of these two wells. Repeating this operation many times (in the extreme case, it can be even an infinity number of times) does not guarantee the well spacing constraints can be honored. Hence, we set the maximum number of repeats as $20\frac{K_{max}(K_{max}-1)}{2}$ where $\frac{K_{max}(K_{max}-1)}{2}$ is the maximum number of well pairs. Afterwards, we classify the K_{max} wells into two sets, one feasible set whose element does not cause any violation of the minimum well spacing constraints and one infeasible set whose element cause at least one violation of the minimum well spacing constraints. The active gridblocks within $R_{\min}$ gridblocks away from each well in the feasible set are marked as occupied gridblocks (the rest of the gridblocks are marked as unoccupied gridblocks). The well with the maximum number of violations, well w, whose location is $\boldsymbol{x}^{(w)}$, is then moved to the unoccupied active gridblock nearest to $\boldsymbol{x}^{(w)}$, denoted as $\boldsymbol{x}_{\text{new}}^{(w)}$, and well w is removed from the infeasible set and added to the feasible set. The gridblocks within $R_{\min}$ gridblocks away from the current location of well w, $\boldsymbol{x}_{\text{new}}^{(w)}$, are also marked as occupied and removed from the unoccupied gridblocks. If the infeasible set is not empty, i.e., there exist wells that cause violations, we continue to move the well with the maximum number of violations to the unoccupied active gridblock nearest to its original location following the same procedure. A detailed description of this procedure is given in Algorithm 5.

3. A producer cannot be placed in the aquifer. In this work, we explicitly convert a producer in the aquifer to an injector.

4. The gas-oil-ratio and water cut of a producer cannot exceed prescribed upper limits. These are two nonlinear state constraints and are enforced through the embedded well management procedure in commercial simulators, e.g., a producer is shut-in if its water cut exceeds a prescribed upper limit, or the oil rate is cutback by a predefined factor if the gas-oil-ratio is higher than the predefined maximum value.

4.1.3 Joint optimization of well status, types and locations

We propose two algorithms to simultaneously optimize well status, types and locations. Note that it is possible that the aforementioned constraints are violated after each update of the design variable (e.g., crossover, mutation, StoSAG perturbations, backtracking, GPS update and so on). Hence, the constraints on the design variables are enforced

Algorithm 5 Algorithm to enforce well spacing constraints

```
Set $N_{move} = 0, feasible = false$.
Do while ($N_{move} < 20\frac{K_{max}(K_{max}-1)}{2}$ and $feasible = false$ )
    – $N_{move} = N_{move} + 1$;
    – Find the infeasible well pair with minimum distance among all well pairs, move this pair of wells apart along the line through the two locations; stop moving a well if it hits the boundary of the active gridblocks and continue to move the other well until the distance is greater or equal to $R_{\min}$.
    – Check the well spacing constraint violations for all well pairs. If there is no violation, set $feasible = true$; otherwise, set $feasible = false$.
EndDo
If $feasible = false$,
    – Find all the infeasible pair of wells and put them in an infeasible set, $W^I$, and put the rest of the $K_{max}$ wells in a feasible set, $W^F$.
    – Do while ($feasible = false$)
        ∗ Put the active gridblocks within $R_{\min}$ gridblocks away from each well in $W^F$ as $\Omega^{\text{occupied}}$ and the rest of the active gridblocks as $\Omega^{\text{unoccupied}}$.
        ∗ Find the well with the largest number of constraint violations in $W^I$ and denote it as well $w$. Move well $w$ to a gridblock in $\Omega^{\text{unoccupied}}$ nearest to its original location and move well $w$ from the infeasible set $W^I$ to the feasible set $W^F$.
        ∗ If $W^I$ is empty, set $feasible = true$; otherwise, set $feasible = false$.
        ∗ If $feasible = false$ and $\Omega^{\text{unoccupied}}$ is empty, report warning message "$R_{\min}$ is specified too large" and break the loop.
    – EndDo
EndIf
```

before each evaluation of the objective function using the strategies presented in last section. Both algorithms we develop are compared with the Genocop III method used in [21].

The first algorithm is an iterative sequential optimization where we alternate optimizing well status and types using GA with pure binary encoding and optimizing locations of wells drilled in the chromosome with the highest objective function, $\boldsymbol{x}$, using StoSAG or GPS if a StoSAG step fails to improve the NPV. This algorithm is designed to improve the convergence speed of GA and is presented in Algorithm 6. Since we start with the GA step, a reasonable set of locations for all K_{max} wells have to be provided as a prior. In the GA step, we use the Stochastic Universal Sampling method [4] (discussed in Section 3.1.1) to select the parent pool. Since the chromosome only consists of binary variables, we simply use the one-point crossover operator and invert the bit of a gene if it is to be mutated (i.e., change 1 to 0 or vice versa). To apply this method, we need to pre-specify (i) the GA related parameters including population size N_p, maximum number of generations N_g, crossover rate p_c and mutation rate p_m and number of consecutive GA generations N_{GA}; (ii) StoSAG related parameters including number of perturbations N_{pert}, initial step size α_0, maximum number of step-size cuts is n_{sc}, maximum number of resampling N_{res} and the number of consecutive StoSAG iterations N_{Sto}. Although it is not clear how to set N_{GA} and N_{Sto}, our heuristic settings of $N_{GA} = 4$ and $N_{Sto} = 5$ give a significant increase in the convergence speed. We intentionally avoid using $N_{GA} = 1$ because it is not likely that the best chromosome gets updated after only one generation of GA evolution and we want to balance the contribution of optimizing (s_j, T_j) and $\boldsymbol{x}_j$. As the computational cost for the GPS method can increase dramatically with respect to the dimensions of the optimization variables, we limit the number of coordinate directions traversed in each GPS step to N_x. Since GPS is used after the StoSAG algorithm converges to fine-tune the well locations, a fixed and small mesh size is used, i.e., $\Delta_0 = 1$ gridblock.

A second algorithm which simultaneously optimizes well status, types and locations using GA with mixed encoding of binary variables (for well status and types) and discrete variables (for well locations) is proposed. Different crossover and mutation operations are required for different encodings. When doing crossover, for each pair of parents $(\boldsymbol{P}_1,\ \boldsymbol{P}_2)$, sample a random number from the uniform distribution, $r \sim U[0, 1]$. If $r < p_c$, then (i) carry out one-pint crossover for genes corresponding to well status and types; (ii) obtain the location information of an offspring using arithmetic crossover. When doing mutation, for each gene, sample a random number r from $U[0, 1]$. If $r < p_m$, invert bit for genes corresponding to well status and types or move wells to one of its neighboring gridblocks for genes corresponding to locations. A detailed description is provided in Algorithm 7.

Both the iterative sequential method and mixed-encoded GA are compared with the Genocop III method used in Emerick et al. [21], see Algorithm 8. Similar to the mixed-

Algorithm 6 Iterative sequential algorithm to jointly optimize well number, types and locations

```
1. Preset $N_p$, $N_g$, $p_c$, $p_m$, $N_{GA}$, $N_{Sto}$, $\Delta_0 = 1$ and the maximum number of simulation
      runs, $N_{sim}$. Set $FL_{Sto} = false$ and $FL_{GPS} = false$. In general, $FL_{Sto} = true$
      indicates that StoSAG has failed to improve the NPV in $N_{res}$ consecutive iterations
      and $FL_{GPS} = true$ indicates that GPS has failed to improve the NPV after searching
      along $N_x$ coordinate directions. Generate the initial locations for all $K_{max}$ wells based
      on engineering judgment and generate the initial population where each chromosome
      has the form $\boldsymbol{c} = [s_1, T_1, s_2, T_2, \cdots, s_{K_{max}}, T_{K_{max}}]$.

2. Continue for $N_{GA}$ generations of GA to optimize well status and types.

3. If $FL_{Sto} = true$ (the StoSAG step failed), $FL_{GPS} = true$ (the GPS step failed) and the
      best chromosome $\boldsymbol{c}$ remains the same during the last $N_{GA}$ iterations, go to step 4.

   Otherwise,

      • if $FL_{Sto} = false$ (the StoSAG step did not fail), optimize the locations of wells
        being drilled in the chromosome which gives the highest NPV among current
        population, $\boldsymbol{x}$, using StoSAG for $N_{Sto}$ iterations. If StoSAG fails to improve the
        NPV for $N_{res}$ iterations, set $FL_{Sto} = true$ (otherwise, $FL_{Sto}$ remains $false$).

      • else, optimize $\boldsymbol{x}$ using GPS for $N_x$ coordinate directions. If GPS fails to improve
        the NPV for $N_x$ coordinate directions, set $FL_{GPS} = true$ (otherwise, $FL_{GPS}$
        remains $false$).

      • If NPV is increased, evaluate each chromosome of the current population with
        the updated well locations $\boldsymbol{x}$. Go to Step 4.

   EndIf

4. Termination criteria check. If the number of generations is greater than $N_g$, or the number
      of simulations is greater than $N_{sim}$, terminate; otherwise, go to step 2.
```

encoded GA, the Genocop III method is designed for simultaneous optimization of well status, types and locations. However, it is different from the mixed-encoded GA in two aspects. Firstly, the Genocop III defines two populations, one search population where each chromosome is only required to satisfy the hard constraints and one reference population in which all chromosomes are required to satisfy all constraints considered. The population size of the search population N_{sp} and the population size of the reference population N_{rp} can be different, though we use $N_{sp} = N_{rp}$ in the numerical examples of this chapter. At each generation, a new search population is generated based on the existing search population using GA operators (i.e., selection, crossover, mutation and replacement). However, it is possible that some of the newly generated chromosomes are infeasible. Different from the mixed-encoded GA which simply uses the heuristic mapping discussed earlier, the Genocop III method repairs an infeasible chromosome S in the search population by combining it with a feasible chromosome R which is randomly selected from the reference population. Once a feasible chromosome Z is obtained after the repair of S, we evaluate the objective function $J(Z)$ and set $J(S) = J(Z)$. If Z gives a larger objective function than R for a maximization problem, R is then replaced by Z, i.e., Z is added to the reference population. Hence, the best feasible solution is always given in the reference population. Moreover, Z replaces S with the replacement ratio of p_r. Whether to replace an individual S with its repaired version Z is related to Lamarckian inheritance [78]: that is, can an organism pass on traits that it acquires during its lifetime to its offspring? Some researchers never replace individuals with their repaired versions ($p_r = 0$), others always replace individuals with their repaired versions ($p_r = 1$), and others recommend that value of p_r between 5% and 20% gives good results (Schoenauer and Michalewicz [75], Orvosh and Davis [67]). In this work, we use the same value for p_r as Emerick et al. [21], which is 25%. Secondly, Genocop III uses the arithmetic crossover (crossover operators designed for real values) while mixed-encoded GA use different crossover operations for different types of variables (arithmetic crossover for well locations and one-point crossover for binary variables).

In this work, the convergence criterion for the iterative sequential method is set as two consecutive failures in the GA step followed by the GPS step or the GPS step followed by the GA step; the convergence criterion for both the mixed-encoded GA method and the Genocop III method is set as failure to improve the NPV in 10 consecutive generations. The iterative sequential method combines the global search ability of GA and the local search ability of StoSAG or GPS and is more likely to converge faster than either the mixed-encode GA or the Genocop III method where only GA is utilized. The three algorithms are terminated either at convergence or when the number of simulation runs exceeds a pre-defined value (2,500 for deterministic optimization in this work).

Algorithm 7 Simultaneous optimization of well status, types and locations using mixed-encoded GA

1. Preset N_p, N_g, p_c, p_m and the maximum number of simulation runs N_{sim}. Generate the initial population where each chromosome has the form $\boldsymbol{c} = [s_1, T_1, \boldsymbol{x}_1, s_2, T_2, \boldsymbol{x}_2, \cdots, s_{K_{max}}, T_{K_{max}}, \boldsymbol{x}_{K_{max}}]$.
2. Evaluate the NPV for each of the chromosomes in the current population.
3. Termination criteria check. If the number of generations is greater than N_g, or the number of simulations is greater than N_{sim}, terminate.
4. Selection. A portion of current population is selected to form the parent pool using the Stochastic Universal Sampling (SUS) method.
5. Crossover. Select random pairs of parents from the parent pool without replacement. For each pair of parents ($\boldsymbol{P}_1$, $\boldsymbol{P}_2$), sample a random number from the uniform distribution, $r \sim U[0, 1]$. If $r < p_c$, then
   - randomly select a crossover point and exchange (s_j, T_j) on the right of the crossover point between $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$.
   - Sample a random number a from $U[-0.25, 1.25]$, and obtain the location information of an offspring by $\boldsymbol{x}_j^{(C_1)} = a\boldsymbol{x}_j^{(P_1)} + (1-a)\boldsymbol{x}_j^{(P_2)}, j = 1, 2, \cdots, K_{max}$ where $\boldsymbol{x}^{(P_1)}$ and $\boldsymbol{x}^{(P_2)}$ represent the location information of $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ respectively; repeat this procedure to obtain $\boldsymbol{x}_j^{(C_2)}$. Note, $\boldsymbol{x}_j^{(C_1)}$ and $\boldsymbol{x}_j^{(C_2)}$ need to truncated to the nearest integer value and bounded inside the active gridblocks. If well spacing constraints are violated, apply heuristic mapping given in Section 4.1.2.
   - Combine $(s_j^{C_1}, T_j^{C_1})$ and $\boldsymbol{x}_j^{C_1}$, $(s_j^{C_2}, T_j^{C_2})$ and $\boldsymbol{x}_j^{C_2}$ to obtain two offspring chromosomes, $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$.
6. Mutation. For each gene, sample a random number r from $U[0, 1]$, if $r < p_m$, invert bit for genes corresponding to (s_j, T_j) and move well $\boldsymbol{x}_j$ to neighboring gridblock for genes corresponding to $\boldsymbol{x}_j$. Goto step 2.

Following all three methods given in Algorithms 6, 7 and 8, we perform a second-stage optimization which fine tunes the locations of wells that are drilled using StoSAG followed by GPS. For simplicity of notation, the three two-stage optimization processes are still referred to as the iterative sequential optimization method, mixed-encoded GA and the Genocop III method respectively. In the iterative sequential method, if the first-stage optimization has converged, then the second-stage optimization is not required. In the mixed-encoded GA and the Genocop III method, even if the first-stage optimization has converged, a second-stage

Algorithm 8 Genocop III method

1. Preset the replacement probability p_r, the search population size N_{sp}, the reference population size N_{rp}. Preset N_g, p_c, p_m and the maximum number of simulation runs, N_{sim}. Generate the initial search population which contains N_{sp} individuals that satisfy the hard constraints (given in Eq. 4.1 in our case). Chromosomes in the search population are not required to satisfy the soft constraints (given in Eq. 4.2 in our case). Generate the initial reference population which contains N_{rp} individuals that satisfy all the constraints (in our case, they are Eqs. 4.1 and 4.2). Each chromosome has the form $\boldsymbol{c} = [s_1, T_1, \boldsymbol{x}_1, s_2, T_2, \boldsymbol{x}_2, \cdots, s_{K_{max}}, T_{K_{max}}, \boldsymbol{x}_{K_{max}}]$. Evaluate the objective function for each individual in the initial reference population.

2. Evaluate the objective function for each individual in the search population. Note, each individual, if infeasible, has to be repaired according to Algorithm 9.

3. Termination criteria check. If the number of generations is greater than N_g, or the number of simulations is greater than N_{sim}, terminate.

4. Selection. A portion of the current search population is selected to form the parent pool using the Stochastic Universal Sampling (SUS) method.

5. Crossover. Select random pairs of parents from the parent pool without replacement. For each pair of parents,

    - sample a random number r from the uniform distribution $U[0, 1]$;
    - if $r < p_c$, then do simple arithmetic crossover to generate a pair of offsprings, e.g.,

    $$\boldsymbol{c}_1 = a\boldsymbol{P}_1 + (1 - a)\boldsymbol{P}_2, \tag{4.3}$$

    where, $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ are the selected pair of parents, $\boldsymbol{c}_1$ is one of the generated offsprings, a is a random variable generated from a uniform distribution $U[-0.25, 1.25]$. Truncate the genes corresponding to (s_j, T_j) in $\boldsymbol{c}_1$ to the nearest integer in $\{0, 1\}$. Truncate the genes corresponding to $\boldsymbol{x}_j$ in $\boldsymbol{c}_1$ to the nearest integer and bound $\boldsymbol{x}_j$ inside the active gridblocks using the procedure given in Section 4.1.2. $\boldsymbol{c}_2$ is generated in the same way as for $\boldsymbol{c}_1$.

7. Mutation. For each gene, sample a random number r from $U[0, 1]$, if $r < p_m$, invert bits for genes corresponding to (s_j, T_j) and move well j to a random grid among the four neighboring gridblocks for genes corresponding to $\boldsymbol{x}_j$. Goto step 2.

optimization is still necessary considering that it is difficult for GA to find the optimal well location variables.

Algorithm 9 Pseudo-code for evaluation of the search population in Genocop III

```
For $i = 1 : N_{sp}$
  • Denote $S$ as the $i$th individual in the search population.
  • If $S$ does not satisfy constraints defined in Eq. 4.2,
    – Set $Z = S$;
    – Choose an individual, $R$, from the reference population;
    – Do while ($Z$ is not feasible)
      ∗ generate a random variable $a$ from $U[0, 1]$;
      ∗ $Z = aZ + (1 - a)R$;
    – EndDo
    – Evaluate the objective function of the repaired chromosome, $J(Z)$;
    – Set $J(S) = J(Z)$;
    – If $J(Z) > J(R)$ for a maximization problem, set $R = Z$;
    – Sample $r$ from $U[0, 1]$. If $r < p_r$, replace $S$ by $Z$.
  • EndIf
EndFor
```

4.2 Numerical Examples of Deterministic Optimization

In this section, the three proposed algorithms are tested on two numerical examples, the PUNQ model and the Brugge model, where the maximum number of wells is set equal to 20 and 30 respectively. For both examples, we determine the optimal well status, types and locations with well controls fixed at their minimum BHP's for producers and maximum BHP's (or rates) for injectors, respectively. For both examples, we generate a reference scenario of the well status, types and locations based on engineering judgment and this reference scenario is included as one of the candidate solutions in the initial population, i.e., the best NPV of the initial population is no less than the NPV obtained with the reference scenario. For both examples, the optimal results obtained by the proposed three algorithms are also compared with a pure well placement optimization of the reference scenario using StoSAG followed by GPS.

In this work, when using GA, we set population size $N_p = 50$, maximum number of generations $N_g = 50$, maximum number of simulation runs in each stage of the optimization process $N_{sim} = 2,500$, the crossover probability $p_c = 0.9$ and the mutation probability $p_m = 0.01$; when using StoSAG, we set the number of perturbations for each gradient estimation $N_{pert} = 10$, the initial step size $\alpha_0 = 5$ gridblocks, the maximum number of step-size cuts is $n_{sc} = 4$ and the maximum allowable number of consecutive failures in improving the NPV, N_{res}, as 2. In the iterative sequential algorithm, the number of generations in each GA step is set as $N_{GA} = 4$ and the number of iterations in each StoSAG step is set as $N_{Sto} = 5$. In both examples, the economic parameters are set as: $r_o^n = \$80.0$/STB, $c_w^n = \$5.0$/STB, $c_{wi}^n = \$5.0$/STB, b=10.0%. In the PUNQ model, we set the drilling cost of a well $C_{w,j} = \$30 \times 10^6$ while in the Brugge model, we set $C_{w,j} = \$6 \times 10^6$. Moreover, in both examples, the minimum well spacing is set equal to $R_{\min} = 3$ gridblocks.

4.2.1 Example 1: PUNQ-S3 model

The PUNQ-S3 reservoir is a synthetic model based on an actual North Sea reservoir. The original simulation grid for the PUNQ-S3 problem consists of 19 × 28 × 5 grids with $\Delta X = \Delta Y = 590.6$ ft and 1,761 active gridblocks. The top structure of the field in Fig. 4.2, shows that the field is bounded to the east and south by a fault, and links the north and west to a fairly strong aquifer. A small gas cap is located in the center of the dome shaped structure. The production life is set as 20 years. In this example, we consider the case that all wells are vertical and a well can only be placed at the center of active gridblocks. Producers operate under specified bottomhole pressure control at 1,740.45 psi. However, if the gas-oil-ratio of a producer is greater than 1,121.92 SCF/STB, the oil rate will be cut back to 75% of the oil flow rate at last time step and the well control will switch to oil rate control for this time step. Water injection wells operate under rate control at 314.5 STB/day. In this model, we preset the maximum number of wells authorized to be drilled, K_{max}, equal to 20. The reference scenario contains 10 injectors and 10 producers, as shown in Fig. 4.3(a).

1. Well placement optimization of the reference scenario

We carried out a well placement optimization of the 20 wells given in the reference scenario to illustrate the efficacy of StoSAG followed by GPS search and to benchmark the

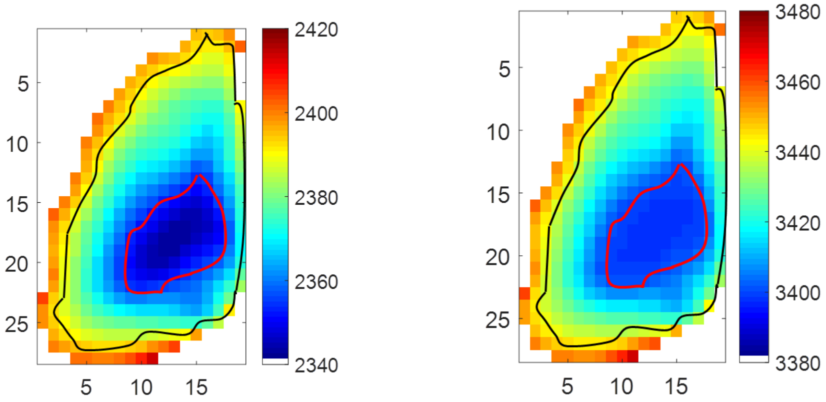
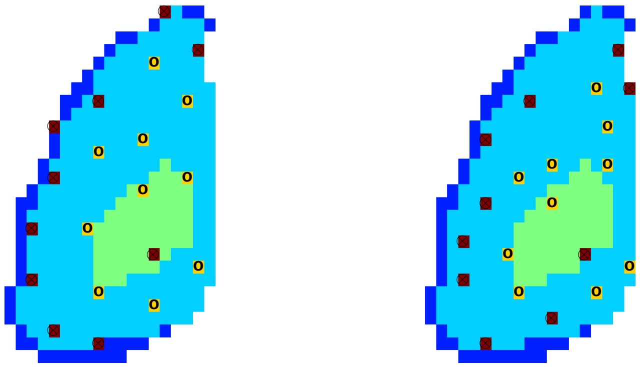


Figure 4.2: The top layer of PUNQ model. Left: the depth of the top layer. Right: the initial pressure of the top layer. The black curve delineates the water-oil contact and the red curve delineates the gas-oil contact.

three proposed algorithms. Note in the following GPS search step, we set the initial mesh size Δ_0 equal to one gridblock since GPS is used to fine-tune the solutions obtained by the StoSAG. Fig. 4.4 shows NPV versus the number of simulation runs where o's represent the StoSAG iterations and $*$'s represent the GPS iterations. As one can see, the StoSAG algorithm increased the NPV from \$1.88 × 10^9 to \$2.07 × 10^9 in 44 simulations, and after two consecutive failures in improving the NPV with StoSAG, the optimization switched to the GPS algorithm after 75 simulation runs. The GPS algorithm obtained a similar increase in the NPV, from \$2.07 × 10^9 to \$2.28 × 10^9, in 375 simulation runs. This confirms our previous discussion that StoSAG is efficient in optimizing numerical discrete variables, but cannot do a good local search. Thus, a secondary optimization using GPS can give another boost in the optimal NPV. Fig. 4.3(b) shows the optimal locations for the 20 wells after the well placement optimization. Compared with the initial locations in Fig. 4.3(a), only 7 wells have moderate changes in the locations while the rest of the wells only have minor changes. This also indicates that the locations given in the reference scenario are somewhat reasonable.

Considering that there are only 20 vertical wells (40 location variables) and the PUNQ model only contains $19 \times 28 \times 5$ active gridblocks (i.e., the search domain is relatively small), it is computationally feasible to use GPS alone for the well placement optimization following



(a) Reference (b) Optimal

Figure 4.3: Locations of 20 wells in the reference scenario (left) and their locations after well placement optimization (right), plotted on the top layer of the PUNQ model. Injector locations are marked in brown color and labeled with ⊗; producer locations are marked in yellow color and labeled with o; the gas cap is marked in green color; the aquifer is marked in dark blue and the oil zone is marked in light blue color.

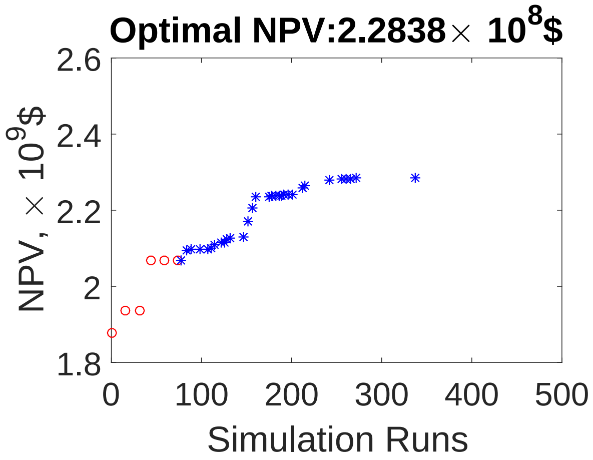


Figure 4.4: NPV versus the number of simulation runs in the well placement optimization of the 20 wells in the reference scenario of the PUNQ model. o's represent the StoSAG iterations and ∗'s represent the GPS iterations.

Algorithm 3. Hence, besides using StoSAG followed by GPS, we also consider the well placement optimization of the reference scenario using only GPS. Three different initial mesh sizes are used, i.e., $\Delta_0 = 1$ gridblock, $\Delta_0 = 2$ gridblocks and $\Delta_0 = 3$ gridblocks. In the application of GPS, the mesh size is reduced by one gridblock when the NPV's cannot be improved after searching all coordinate directions and GPS is terminated only when the minimum mesh size (i.e., 0 gridblock) is achieved. For each element of the variable vector at

ℓth iteration $\boldsymbol{x}^\ell$, e.g., the ith element, if either point of $\boldsymbol{x}^\ell + \Delta_\ell \boldsymbol{e}_i$ and $\boldsymbol{x}^\ell - \Delta_\ell \boldsymbol{e}_i$ gives higher NPV than $\boldsymbol{x}^\ell$, then 5 follow-up trial updates along the direction that improves the NPV are taken until the NPV stops improving where the size of each trial update equals Δ_ℓ. Fig. 4.5 shows the NPV versus number of simulation runs using GPS with three different Δ_0's. From Fig. 4.5, with a small initial mesh size, GPS tends to take small steps which may cause the algorithm to be trapped at a local solution which is far from the global optimum, see the case of $\Delta_0 = 1$. An optimal NPV of $\$2.24 \times 10^9$ is obtained using GPS with $\Delta_0 = 1$, which is 2% less than the NPV obtained using StoSAG follow by GPS with $\Delta_0 = 1$. Increasing the initial mesh size can produce higher optimal NPV's. For example, when Δ_0 is increased from one gridblock to two gridblocks, the optimal NPV is increased by 7% to $\$2.39 \times 10^9$. A further increase in Δ_0 to 3 gridblokcs gives an optimal NPV of $\$2.38 \times 10^9$ which is slightly less than that obtained with $\Delta_0 = 2$ gridblocks because a different local solution is obtained. Increasing the initial mesh size would inevitably increase the computational cost. For $\Delta_0 = 1,\ 2,\ 3$ gridblocks, the optimization processes cost approximately 400, 550, 850 simulation runs, respectively.

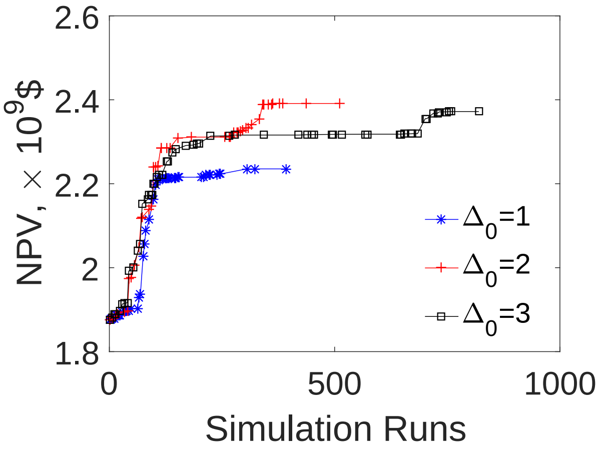


Figure 4.5: NPV versus number of simulations for the well placement optimization of the reference scenario, using GPS with three different initial mesh sizes, PUNQ model.

To investigate the effect of well controls, a subsequent control optimization is carried out based on the optimal well locations presented in Fig. 4.3(b). The upper and lower bounds of the water injection rate is set as 314.5 STB/day and 0 STB/day respectively.

The upper and lower bound of the bottomhole pressure for a producer is set as 3,379 psi and 1,740.45 psi respectively. The reservoir life is divided into 20 equal control steps where each control step has the length of one year. The temporal correlation length is set to be 730 days. During all control steps, the initial guesses for the injection rates are set as 283 STB/day and the initial guesses for the production BHP's are set as 1,885 psi. i.e., the initial guesses for controls are somewhat different than but close to the fixed values used in the well placement optimization. Fig. 4.6 shows the NPV versus the number of simulation runs during the control optimization where the NPV is increased by approximately 2% compared to setting the controls at the pre-fixed controls used to optimize the well locations, i.e., the upper bound of the injection rates and the lower bound of the production BHP's. As illustrated in Chapter 3 where we select N_w wells out of K_{max} potential paths, whether the subsequent control optimization can significantly increase the NPV's or not depends on the controls used to optimize the well locations and the bounds of the control settings. Fig. 4.7 shows the optimal well controls where most of the wells remain operating close to the upper bound of the injection rates and the lower bound of the production BHP's, but that "P4" and "P8" tend to produce less in the later stage of the production life. It is worth to mention that the optimal controls look a bit rough due to the fact that we set the temporal correlation length small, i.e., equivalent to two control steps.

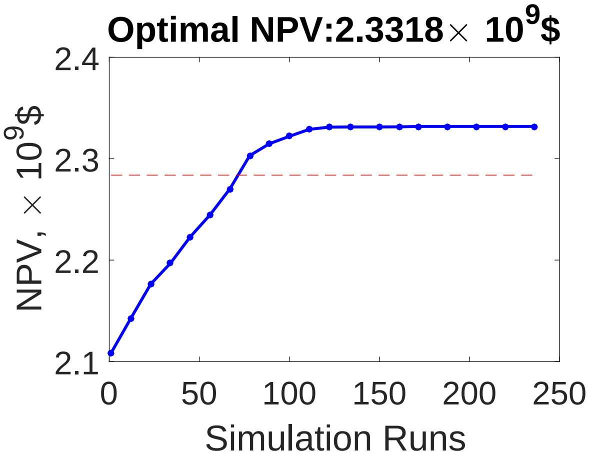
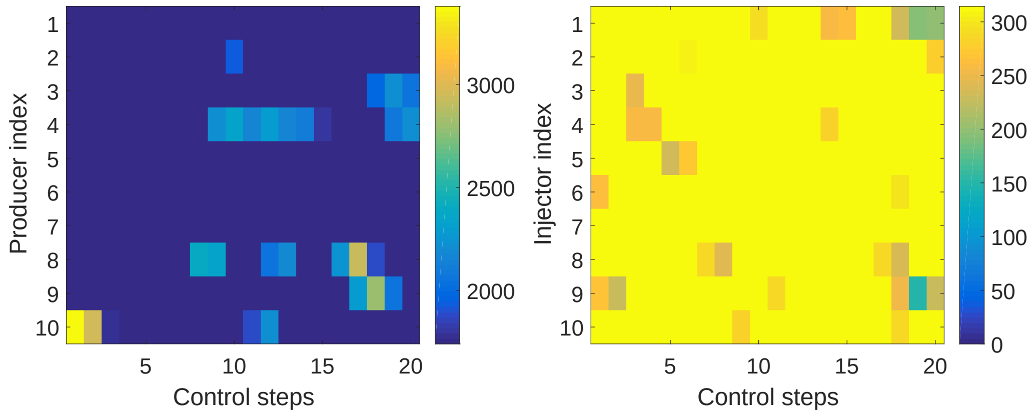


Figure 4.6: NPV versus number of simulation runs during the well control optimization using the optimal well locations of the reference scenario, PUNQ model. The red dashed line represents the optimal NPV's obtained after the well placement optimization.



(a) Production Control

(b) Injection Control

Figure 4.7: Optimal controls obtained for the reference scenario with optimal well locations, PUNQ model.

2. Joint Optimization of well status, types and locations

The definition of the initial population may impact the performance of the optimization process. It is likely that a population of poor individuals will require several generations until it produces a good solution. In this sense, employing the engineer's knowledge to set up the initial population is highly desirable. Hence, for all three algorithms to optimize the well status, types and locations, the reference scenario developed based on engineering judgment (given in Fig. 4.3(a)) is included in the initial population. The reference scenario gives an NPV of $\$1.88 \times 10^9$, which, as shown later, is higher than the rest of the randomly generated chromosomes in the initial population, indicating that reference scenario defines a good initial case.

In the iterative sequential algorithm, chromosomes within a population have identical well locations, but different well status and types. The initial population of the iterative sequential method uses the $K_{max} = 20$ well locations given in Fig. 4.3(a) and contains 50 candidate solutions of the well status and types where one chromosome is the same as given in Fig. 4.3(a) and the rest of the 49 chromosomes are generated randomly by sampling the well status s_j and well type T_j from $\{0, 1\}$ for each of the K_{max} wells. Considering that it is possible that we sample a producer in the aquifer and the fact that we force it to be an injector, we avoid placing more injectors than producers by increasing the probability of a well being a producer ($T_j = 0$) in the initial population. As a reservoir generally has more producers than injectors, we pre-set the the probability of a well being a producer ($T_j = 0$) is equal to twice the probability of a well being an injector ($T_j = 1$). In the mixed-encoded GA method, a chromosome contains information of the well status, types and locations of all K_{max} wells, and chromosomes within a population are different from each other. The initial population of mixed-encoded GA method again consist of one reference chromosome given in Fig. 4.3(a) and 49 random chromosomes. For each random chromosome, $(s_j, T_j),\ j = 1, 2, \cdots, K_{max}$ are sampled in the same way as in the iterative sequential algorithm while the location of well j is randomly sampled from the 35 locations given in Fig. 4.8. For the Genocop III method, both the initial search population and the initial reference population are generated using the same way as for the mixed-encoded GA method, but that the chromosome given in Fig. 4.3(a) is repeated 25 times in the initial reference population. Emmerick et al. [21] show that starting from a reference population in which, the reference scenario (engineer suggested case) represents 50 % of the individuals, leads to better solutions than starting the optimization with a completely random reference population. By repairing the infeasible chromosomes in the search population with the feasible chromosomes in the reference population, the Genocop III method tends to generate an optimal solution similar to the reference scenario, i.e., an “improved engineer defined case.” Since the distances between each pair of wells in Fig. 4.8 and Fig. 4.3(a) are all greater than $R_{\min} = 3$ gridblocks, all chromosomes of the initial population satisfy the minimum well spacing constraints.

The optimal NPV’s obtained with the three different algorithms using three random seeds are plotted in Fig. 4.9 and summarized in Table 4.1. As one can see, the iterative sequential algorithm converged in the first stage of optimization (i.e., a secondary well placement optimization is not used) while both the mixed-encoded GA and Genocop III method increased the NPV using the second-stage optimization to fine-tune the optimal well locations. During the first-stage optimization, the iterative sequential algorithm gives the highest NPV, the mixed-encoded GA gives the second highest NPV while the Genocop III method

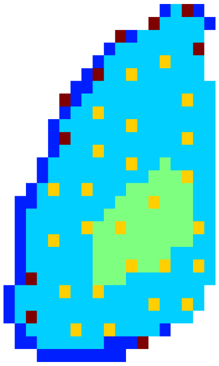


Figure 4.8: The set of potential locations (marked in yellow and red colors) for the initial population, plotted on the top layer of the PUNQ model.

gives the worst NPV. After the second-stage optimization, the Genocop III method still gives the lowest NPV on average, and the mixed-encoded GA gives a slightly higher NPV than the iterative sequential algorithm because the first-stage optimization of the mixed-encoded GA simultaneously optimized the well status, types and locations and hence finds a better solution than the iterative sequential optimization procedure. However, as we found in last chapter, simultaneous optimization does not always gives a higher estimated optimal NPV than the a sequential method. The Genocop III method converges the slowest among all three algorithms during the first-stage optimization since the linear combination process to repair infeasible chromosomes tends to produce similar chromosomes to the reference scenario (the engineer defined case), considering that the candidate solutions in the search population have a good chance to be combined with the reference scenario. However, this does not indicate that starting from a randomly generated reference population, Genocop III could converge faster since the randomly generated chromosomes generally have lower NPV's than the reference scenario and linear combination with these poor individuals may lead to an even slower converge. In any case, the Genocop III method produces the worst NPV. Fig. 4.10 shows the NPV's versus the number of simulations obtained with the best seed of each of the three algorithms during the first-stage optimization. From Fig. 4.10, the reference scenario given in Fig. 4.3(a) has the largest NPV among the initial populations generated for all three algorithms, indicating the reference scenario is indeed a good initial guess. In the iterative sequential algorithm, $*$'s represent the NPV's of current population re-evaluated with the updated locations after a StoSAG step, $*$'s represent the NPV's of current population re-evaluated with the updated locations after a GPS step. Even though only the locations drilled in the best chromosome are updated in the StoSAG and GPS steps, the NPV's for the whole population are shifted up with a similar variance to NPV's before the well location updates. The first StoSAG step increased the NPV by $\$0.17 \times 10^9$ while the second StoSAG step failed to improve the NPV for $N_{res} = 2$ iterations which invoked the GPS step after $N_{GA} = 4$ GA generations. The iterative sequential algorithm converged at around 1600 simulation runs. The best NPV's of each generation for the mixed-encoded GA and the Genocop III method increase slowly until they are terminated at 2,500 simulation runs. All three algorithms obtained higher optimal NPV's compared to the well placement optimization of the reference scenario, respectively by %11, %12 and %5. Table 4.2 shows the simulation cost for the three different algorithms using three random seeds. From Table 4.2, the iterative sequential method requires the least number of simulation runs while both the mixed-encoded GA and the Genocop III method require over 2,500 simulation runs. Since StoSAG and GPS are better at local search compared to GA, the simulation runs for the second-stage optimization is much less than the first-stage optimization.

Table 4.1: Comparison of the optimal NPV's obtained with three different algorithms using three seeds, PUNQ model.

	Iterative sequential		Mixed-encoded GA		Genocop III	
Seeds	Stage #1	Stage #2	Stage #1	Stage #2	Stage #1	Stage #2
Seed 1	2.55	-	2.47	2.53	2.29	2.36
Seed 2	2.46	-	2.48	2.56	2.12	2.48
Seed 3	2.54	-	2.48	2.60	2.35	2.36
Average	2.52	-	2.48	2.56	2.25	2.40

The number of injectors and producers in the optimal solutions obtained with the three different algorithms using three seeds are summarized in Table 4.3 and plotted in Fig. 4.11. The iterative sequential algorithm and the mixed-encoded GA obtained similar number of wells, i.e., 9 or 10 wells on average for three different runs. Due to the existence

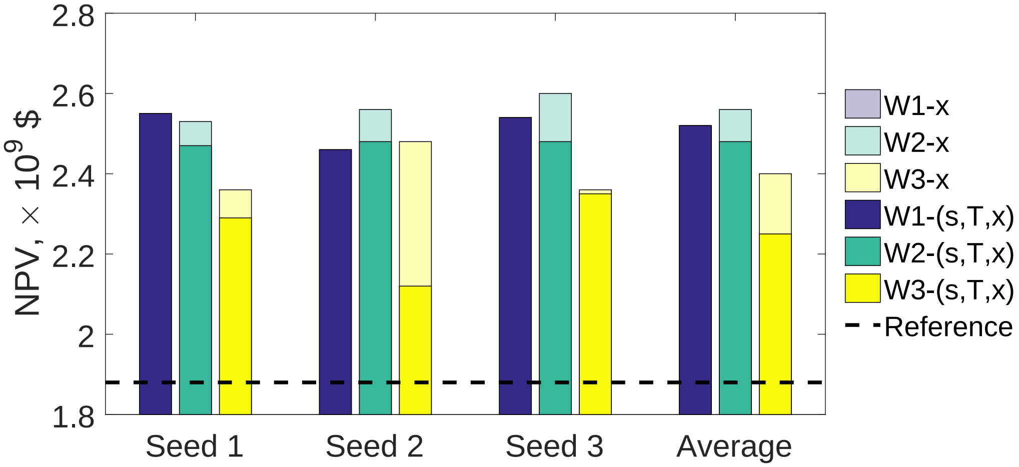
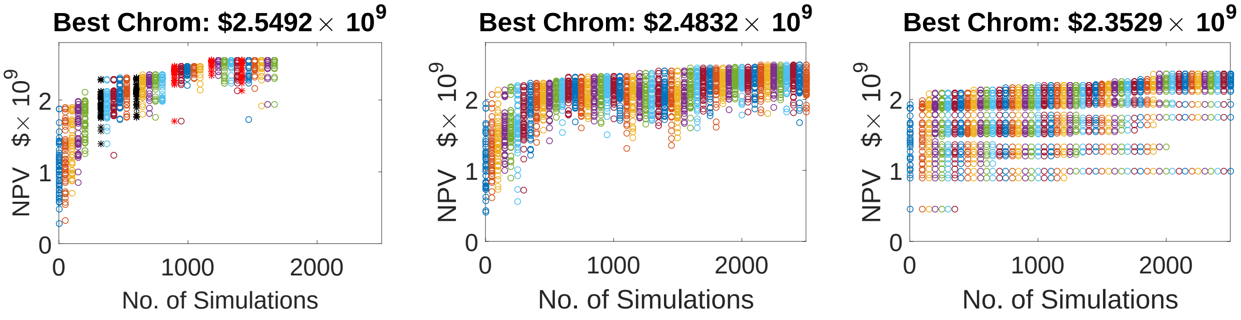


Figure 4.9: Comparison of the optimal well numbers obtained with three different algorithms using three seeds, PUNQ model. "W1-(s,T,x) and "W1-x" respectively represent the first-stage and the second-stage optimization of the iterative sequential algorithm, "W2-(s,T,x) and "W2-x" respectively represent the first-stage and the second-stage optimization of the mixed-encoded GA, "W3-(s,T,x) and "W3-x" respectively represent the first-stage and the second-stage optimization of the Genocop III method, and the black dashed line represents the NPV of the reference scenario.



(a) Iterative sequential - stage 1 (b) Mixed-encoded GA - stage 1 (c) Genocop III - stage 1

Figure 4.10: NPV's versus the number of simulation runs obtained with the best seed of each of the three algorithms, PUNQ model. In the iterative sequential algorithm, $*$'s represent the NPV's of current population re-evaluated with the updated locations after a StoSAG search step, $*$'s represent the NPV's of current population re-evaluated with the updated locations after a GPS search step.

of the strong aquifer which provides pressure support for producers, the optimal solutions tend to have no more than two injectors. As the Genocop III method tends to follow the guidance of the reference scenario which has 20 wells, it obtained 15 wells on average, based on three different runs, including 10 producers and 6 injectors. Fig. 4.12 shows the optimal well locations obtained with the best seed of the three runs for each algorithm. From Fig.

Table 4.2: Comparison of simulation cost obtained with three different algorithms using three seeds, PUNQ model.

	Iterative sequential		Mixed-encoded GA		Genocop III	
Seeds	Stage #1	Stage #2	Stage #1	Stage #2	Stage #1	Stage #2
Seed 1	1619	-	2500	128	2500	164
Seed 2	2004	-	2500	187	2500	559
Seed 3	2295	-	2500	175	2500	225
Average	1973	-	2500	163	2500	316

4.12, even though the total number of wells is different, the optimal number and locations of producers are somewhat similar to each other. Note that the inter-well distances are all greater than the pre-defined minimum well spacing, $R_{\text{min}} = 3$ gridblocks.

Table 4.3: Comparison of optimal well numbers obtained with three different algorithms using three seeds, PUNQ model.

	Iterative sequential			Mixed-encoded GA			Genocop III		
Seeds	N_w	N_{Pro}	N_{Inj}	N_w	N_{Pro}	N_{Inj}	N_w	N_{Pro}	N_{Inj}
Seed 1	9	8	1	11	9	2	13	9	4
Seed 2	9	8	1	10	10	0	18	10	8
Seed 3	10	9	1	10	10	0	15	10	5
Average	9	8	1	10	10	1	15	10	6

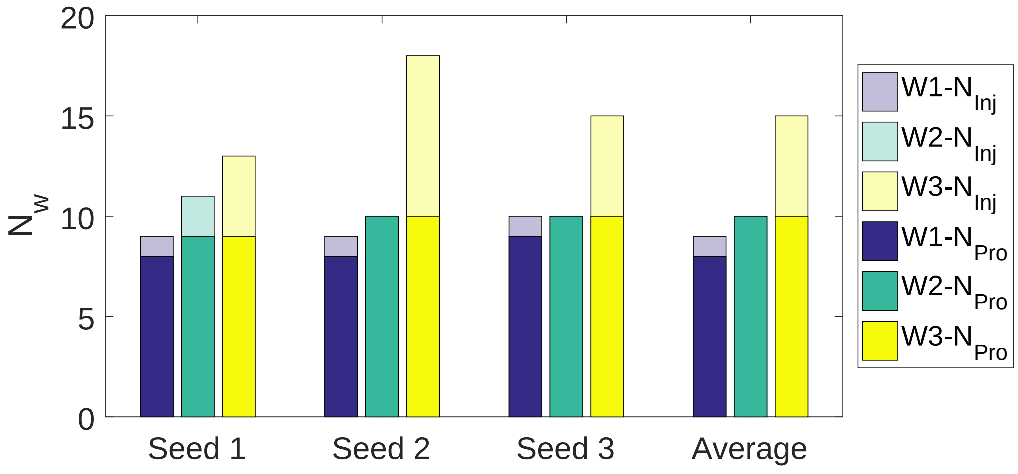
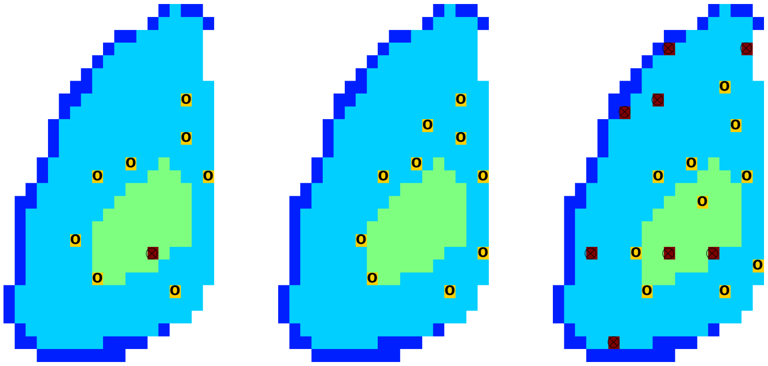


Figure 4.11: Comparison of optimal well numbers obtained with three different algorithms using three seeds, PUNQ model. "W1" represents the iterative sequential algorithm, "W2" represents the mixed-encoded GA, and "W3" represents the Genocop III method.



(a) Iterative sequential (b) Mixed-encoded GA (c) Genocop III

Figure 4.12: Optimal well locations obtained with the three algorithms using the best seed, plotted on the top layer of the PUNQ model.

3. Investigation of the effect of drilling cost

In the preceding optimization of well status, types and locations, the drilling cost of a well was set equal to $\$30 \times 10^6$. To investigate the effect of the drilling cost, we consider two more different drilling cost of a well, $\$100 \times 10^6$ and $\$8 \times 10^6$. The mix-encoded GA which performs the best during the optimization of $(\boldsymbol{s}, \boldsymbol{T}, \boldsymbol{x})$ is used in this comparison. Figs. 4.13 and 4.14 respectively show the optimal NPV's and the optimal number of wells obtained for three different drilling cost.

When drilling a well is relatively cheap (e.g., $C_{w,j} = \$8 \times 10^6$), the total drilling cost (i.e., $\sum_{j=1}^{K_{max}} s_j C_{w,j}$) represents a smaller penalty on the number of wells, and hence the optimal solution tends to drill more wells. The case of $C_{w,j} = \$8 \times 10^6$ obtains the largest number of wells (i.e., 9 producers and 5 injectors) along with the highest optimal NPV among all three cases. Since there is a strong aquifer in the PUNQ model to provide pressure support, with an increase in the drilling cost from $\$8 \times 10^6$ to $\$30 \times 10^6$, injectors are eliminated and only 10 producers are drilled. The fact that the two cases of $C_{w,j} = \$30 \times 10^6$ and $C_{w,j} = \$8 \times 10^6$ give similar optimal NPV's may indicate that 10 producers are more or less enough to deplete the PUNQ reservoir and the increase in well number beyond 10 producers only has a negligible affect on the overall production. With a further increase in the drilling cost, i.e., from $C_{w,j} = \$30 \times 10^6$ to $C_{w,j} = \$100 \times 10^6$, four more producers are eliminated, i.e., only 6 producers are obtained which significantly reduced the optimal NPV. The decrease in the optimal NPV is because of (i) the reduced production due to the reduced number of producers and (ii) the increase in the total drilling cost.

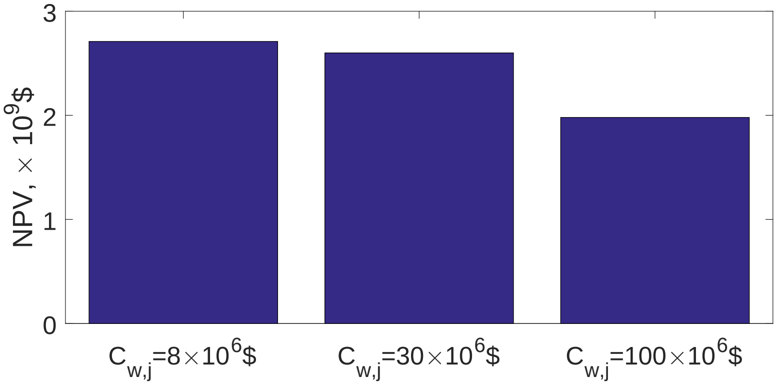


Figure 4.13: Optimal NPV's obtained with different drilling cost using the mixed-encoded GA algorithm, PUNQ model.

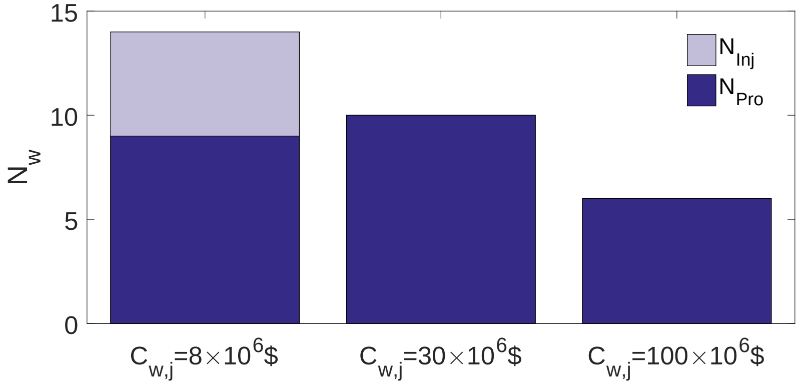


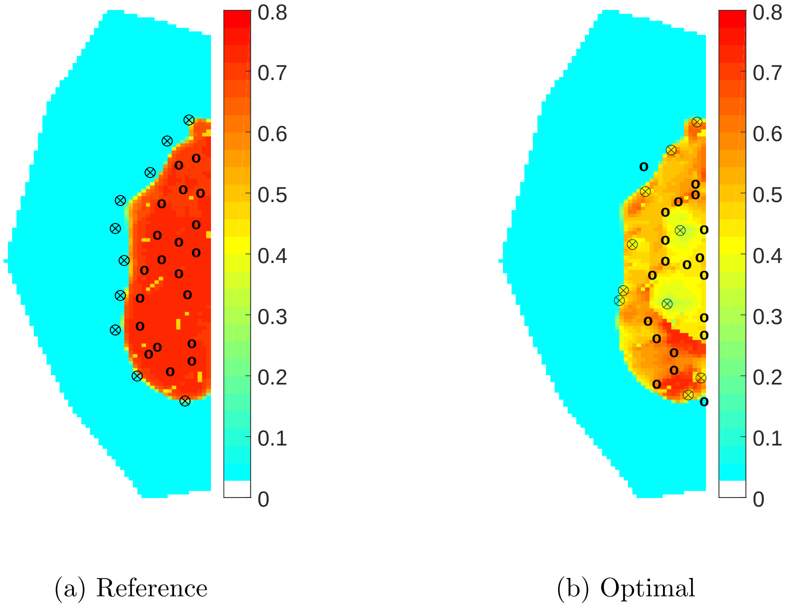
Figure 4.14: Optimal number of wells obtained with different drilling cost using the mixed-encoded GA algorithm, PUNQ model.

4.2.2 Example 2: Brugge model

A brief introduction of the Brugge model was provided in Chapter 3. In this example, we consider the case that all wells are vertical and a well can only be placed at the center of active gridblocks. Both injectors and producers operate under bottomhole pressure control where the BHP's for producers are specified as 725 psi and BHP's for injectors are specified as 2,662 psi. However, a producer is shut in when its water cut is greater than 0.94. In this model, we preset the maximum number of wells authorized to be drilled, K_{max}, equal to 30. The original 30 wells (including 20 producers and 10 injectors) are used as the reference scenario shown in Fig. 4.15(a). The reference scenario gives an NPV of $\$8.33 \times 10^9$, which, as shown later in Fig. 4.18, is higher than most of the randomly generated chromosomes in the initial population, indicating that the reference scenario defines a good initial case. The initial population for all three algorithms contains one chromosome of the reference scenario given in Fig. 4.15(a) and 49 random scenarios generated in a similar way as for the PUNQ model. For the mixed-encoded GA method and the Genocop III method, the well locations of the 49 random chromosomes are randomly sampled from the 64 locations given in Fig. 3.20(a). Since the distances between each pair of wells in Fig. 3.20(a) and Fig. 4.15(a) are all greater than $R_{\min} = 3$ gridblocks, all chromosomes of the initial population satisfy the minimum well spacing constraints.

1. Well placement optimization of the reference scenario

We first carried out a well placement optimization of the 30 wells given in the reference scenario (shown in Fig. 4.15(a)) to illustrate the efficacy of a GPS search after the StoSAG search and to benchmark the three proposed algorithms. Fig. 4.16 shows the NPV versus the number of simulation runs where o's represent the StoSAG iterations and $*$'s represent the GPS iterations. As one can see, the StoSAG algorithm increased the NPV from $\$8.33 \times 10^9$ to $\$11.18 \times 10^9$ in 283 simulations, and after two consecutive failures in improving the NPV, the optimization switched to the GPS algorithm at 313 simulation runs. The GPS algorithm increased the NPV by $\$10^9$, to $\$12.15 \times 10^9$, in 616 simulation runs. This observation is quite consistent with the result for the PUNQ model. Fig. 4.15(b) shows the optimal locations for the 30 wells after the well placement optimization. Compared with the initial locations in Fig. 4.15(a), two injectors are moved into the oil zone and two producers are moved into the aquifer. Since a producer will be shut-in if its water cut exceeds 0.94, the two producers placed in the aquifer are shut-in for the whole reservoir life to avoid extra water production. The fact that two producers are placed in the aquifer suggests that 20 producers are enough.



(a) Reference (b) Optimal

Figure 4.15: Locations of the 30 wells in the reference scenario (left) and their locations after well placement optimization (right), plotted on the top layer of the Brugge model. ⊗ represents an injector; o represents a producer.

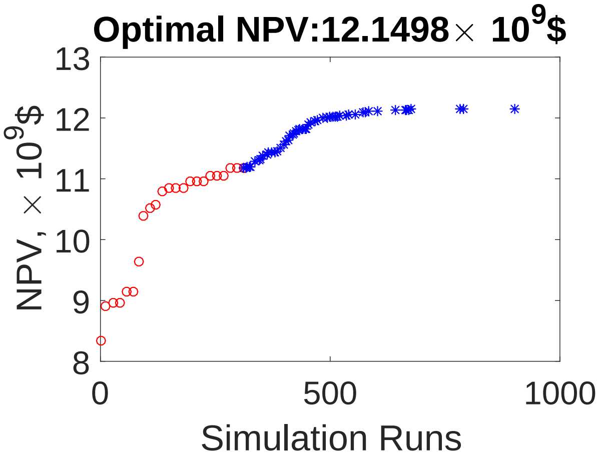


Figure 4.16: NPV versus the number of simulation runs in the well placement optimization of the 30 wells in the reference scenario of the Brugge model. o's represent the StoSAG iterations and $*$'s represent the GPS iterations.

2. Optimization of well status, types and locations

The optimal NPV's obtained with three different algorithms using three seeds are summarized in Table 4.4 and plotted in Fig. 4.17. For the iterative sequential method, only the first seed requires a second-stage optimization where the NPV is increased slightly, while for the other two seeds, the last GPS step of the first-stage optimization found local solutions of the well locations and subsequent GA generations before termination did not change the optimal well status and types. Both the mixed-encoded GA and the Genocop III method require the second-stage optimization where the NPV's are increased by more than $\$10^9$ on average. During the first stage of optimization, the iterative sequential algorithm gives the highest NPV, the mixed-encoded GA method gives the second highest NPV while the Genocop III method gives the worst NPV. After the second-stage optimization, the mixed-encoded GA gives a higher NPV than the iterative sequential algorithm while the Genocop III method still gives the worst NPV. This again is quite consistent with the results from the PUNQ model. The run obtained with seed 3 gives a relatively poor performance for all three algorithms. Considering only the runs obtained with seed 1 and seed 2, the optimal NPV's obtained with the three different algorithms are similar and comparable to the optimal NPV's obtained with well placement optimization of the reference scenario, indicating that optimizing well status and types only marginally improved the NPV. This is because the drilling cost of a well is relatively cheap in this example, i.e., $\$6 \times 10^6$, which imposes a small penalty on number of wells. Hence, the optimal well status are somewhat similar to the reference scenario.

Fig. 4.18 shows the NPV's versus the number of simulation runs obtained with the best seed of each of the three algorithms in the first-stage optimization. In the iterative sequential algorithm, the fourth StoSAG search step failed to improve the NPV for $N_{res} = 2$ which invoked the GPS search step after $N_{GA} = 4$ GA generations. Even though we terminated the iterative sequential algorithm after 2,500 simulation runs, its NPV stabilized at around 1,500 simulation runs. The best NPV's of each generation of the mixed-encoded GA and the Genocop III method increase slowly until they are terminated at 2,500 simulation runs. Table 4.5 shows the simulation cost for the three different algorithms using three random seeds. From Table 4.2, the iterative sequential method requires the least number of simulation runs while both the mixed-encoded GA and the Genocop III method require over 3,000 simulation runs on average of three runs. Since the Brugge model has many more active gridblocks and more wells in the optimal solution compared to the PUNQ model, the second-stage optimization requires more simulation runs for the Brugge model than for the PUNQ model.

Table 4.4: Comparison of optimal NPV's obtained with three different algorithms using three seeds, Brugge model.

	Iterative sequential		Mixed-encoded GA		Genocop III	
Seeds	Stage #1	Stage #2	Stage #1	Stage #2	Stage #1	Stage #2
Seed 1	12.27	12.32	11.00	12.23	11.28	12.32
Seed 2	12.28	–	11.39	12.60	10.63	11.53
Seed 3	11.94	–	11.35	12.46	11.22	12.12
Average	12.16	12.18	11.25	12.43	11.04	11.99

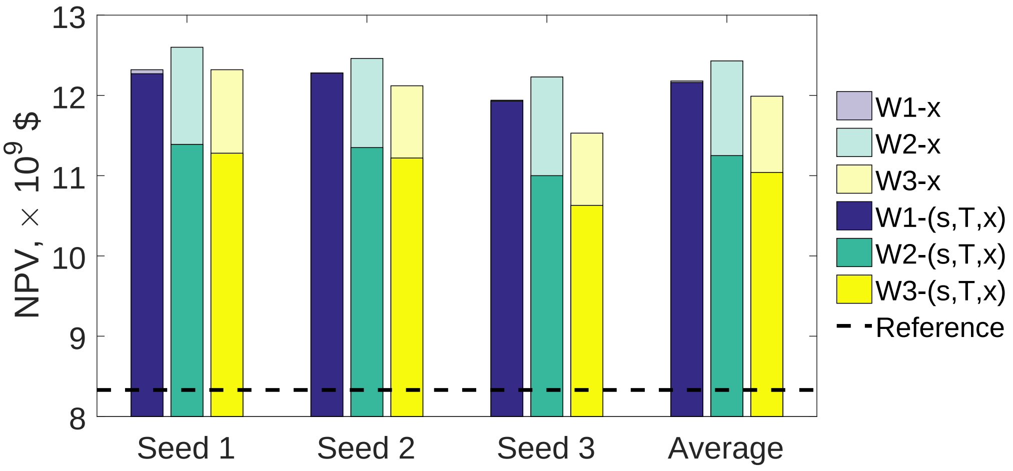
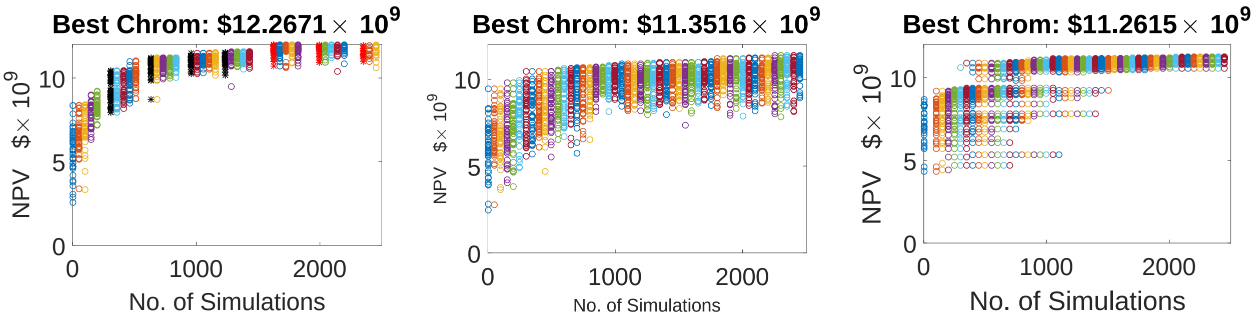


Figure 4.17: Comparison of optimal well numbers obtained with three different algorithms using three seeds, Brugge model. "W1-(s,T,x) and "W1-x" respectively represent the first-stage and the second-stage optimization of the iterative sequential algorithm, "W2-(s,T,x) and "W2-x" respectively represent the first-stage and the second-stage optimization of the mixed-encoded GA, "W3-(s,T,x) and "W3-x" respectively represent the first-stage and the second-stage optimization of the Genocop III method, and the black dashed line represent the NPV of the reference scenario.

The number of injectors and producers in the optimal solutions obtained with three different algorithms using three seeds are summarized in Table 4.6 and plotted in Fig. 4.19. All three algorithms obtained a similar number of wells on average (25 or 26), but the number of wells varies widely from seed to seed when the Genocop III is used. Fig. 4.20 shows the optimal wells obtained with the best of the three runs for each algorithm. From Fig. 4.12, even though the number of wells are different for each algorithm, in all cases, we obtain



(a) Iterative sequential - stage 1

(b) Mixed-encoded - stage 1

(c) Genocop III - stage 1

Figure 4.18: NPV's versus the number of simulations obtained with the best seed of each of the three algorithms, Brugge model. In the iterative sequential algorithm, $*$'s and $*$'s respectively represent the NPV's of current population re-evaluated with the updated locations after a StoSAG search step and a GPS search step.

Table 4.5: Comparison of the simulation cost for the three different algorithms using three seeds, Brugge model.

	Iterative sequential		Mixed-encoded GA		Genocop III	
Seeds	Stage #1	Stage #2	Stage #1	Stage #2	Stage #1	Stage #2
Seed 1	2500	433	2500	698	2500	720
Seed 2	2500	–	2500	630	2500	432
Seed 3	2500	–	2500	653	2500	500
Average	2500	144	2500	660	2500	552

Table 4.6: Comparison of optimal well numbers obtained with three different algorithms using three seeds, Brugge model.

	Iterative sequential			Mixed-encoded GA			Genocop III		
Seeds	N_w	N_{Pro}	N_{Inj}	N_w	N_{Pro}	N_{Inj}	N_w	N_{Pro}	N_{Inj}
Seed 1	25	17	8	24	16	8	28	15	13
Seed 2	26	16	10	26	18	8	29	15	14
Seed 3	23	14	9	26	17	9	20	13	7
Average	25	16	9	25	17	8	26	14	12

several injectors in the oil zone. As the Genocop III method tends to follow the guidance of the reference scenario, its optimal well placement is quite similar to that of the reference scenario. The inter-well distances are greater than or equal to the pre-defined minimum well spacing, $R_{\min} = 3$ gridblocks, for all results.

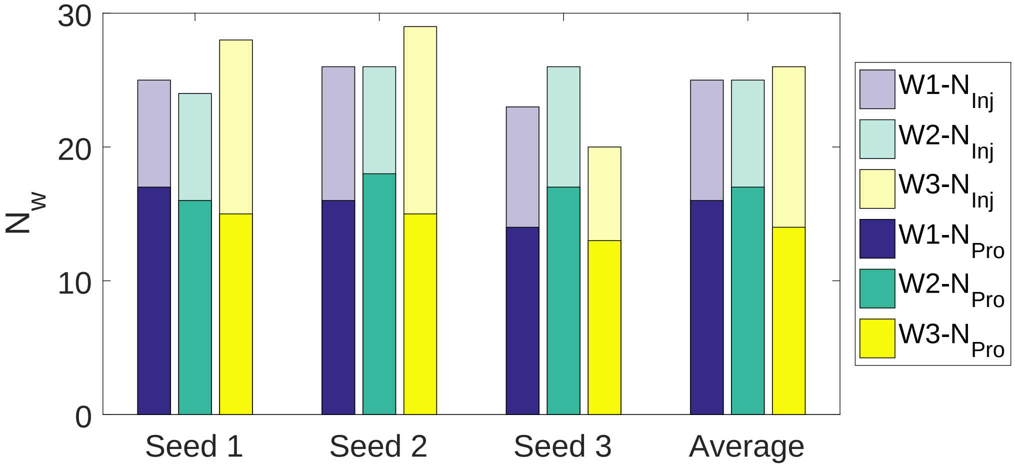
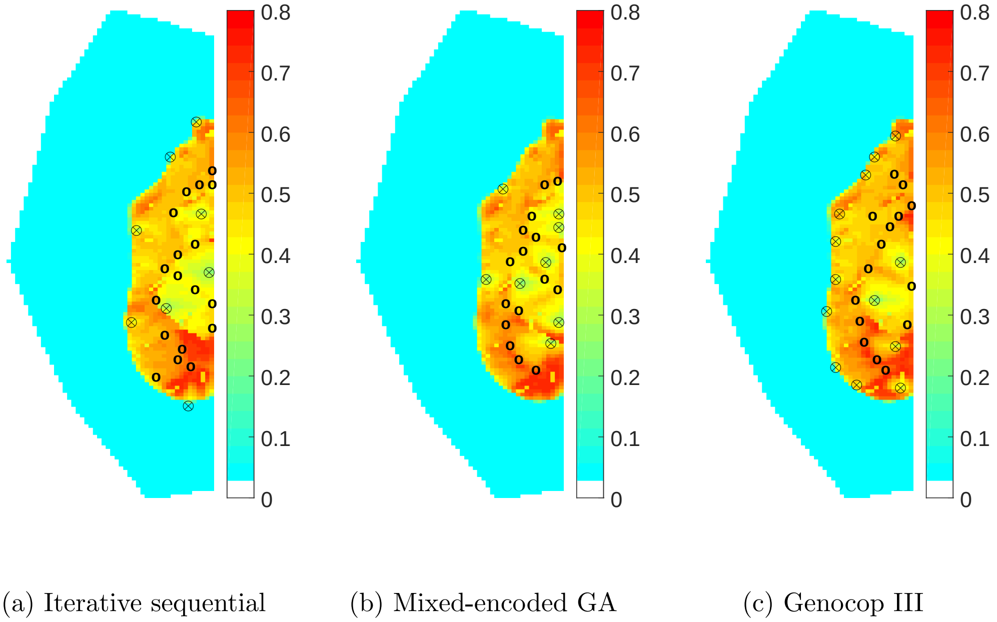


Figure 4.19: Comparison of the optimal well numbers obtained with three different algorithms using three seeds, Brugge model. “W1” represents the iterative sequential algorithm, “W2” represents the mixed-encoded GA, and “W3” represents the Genocop III method.



(a) Iterative sequential (b) Mixed-encoded GA (c) Genocop III

Figure 4.20: Optimal well locations obtained with the three algorithms using the best seed, plotted on the oil saturation field of the top layer after 20 years of production, Brugge model. ⊗ represents an injector; o represents a producer.

4.3 Field Development Optimization Considering Geological Uncertainty

There is significant geological uncertainty in the reservoir description due to the limited knowledge about the underground formation. The common approach to account for geological uncertainty is to use multiple plausible geostatistical realizations of the reservoir model. However, the computational cost when a large ensemble of realizations is used may be prohibitive. Hence, only a small ensemble is affordable in robust optimization, which may not be sufficient to capture the subsurface uncertainty. Thus, in this section, we propose a relatively efficient procedure for robust optimization of well status, types and locations when a large number of representative realizations are considered.

Denote the number of realizations to represent the geological uncertainty by N_e, and denote the kth realization (a vector of the model parameters) by $\boldsymbol{m}_k$, then the full-set ensemble of reservoir models is given by

$$\Omega_f = \{\boldsymbol{m}_k | k = 1, 2, \cdots, N_e\}.$$

The most common approach for handling geological uncertainty is to maximize the expected value of the objective function (NPV in this work) over the ensemble of N_e realizations [86, 22, 20, 49, 52], i.e.,

$$J_E^{\Omega_f}(\boldsymbol{s}, \boldsymbol{T}, \boldsymbol{x}) = \frac{1}{N_e} \sum_{k=1}^{N_e} J(\boldsymbol{m}_k, \boldsymbol{s}, \boldsymbol{T}, \boldsymbol{x}), \tag{4.4}$$

which is commonly referred to as robust optimization in the literature. The computational cost for robust optimization is much higher compared to deterministic optimization since each objective function evaluation requires N_e reservoir simulation runs. Hence, to reduce the computational cost, besides developing efficient optimization algorithms, we also focus on adaptively selecting a smaller subset of representative models and optimizing over the subset ensemble.

Denote the subset ensemble of the representative models by

$$\Omega_s = \{\boldsymbol{m}_{j_k} | \boldsymbol{m}_{j_k} \in \Omega_f, k = 1, 2, \cdots, N_{es}\},$$

where N_{es} is the number of realizations in the subset. We assume the realizations of the reservoir models in the full ensemble Ω_f could be represented by a subset of realizations Ω_s in the sense that optimizing the expected NPV over the subset gives a good approximation of that would be obtained by maximizing the expected NPV over N_e realization. Note this representative subset may change as the estimate of $(\boldsymbol{s}, \boldsymbol{T}, \boldsymbol{x})$ changes. Under the preceding assumption, the expected NPV over the full ensemble can be approximated by

$$J_E^{\Omega_s}(\boldsymbol{s}, \boldsymbol{T}, \boldsymbol{x}) = \frac{1}{N_{es}} \sum_{k=1}^{N_{es}} J(\boldsymbol{m}_{j_k}, \boldsymbol{s}, \boldsymbol{T}, \boldsymbol{x}). \tag{4.5}$$

Considering that the computational cost is the main concern of robust optimization, and that the iterative sequential algorithm is the most efficient of the three algorithms proposed for deterministic optimization of well status, types and locations (see Tables 4.2 and 4.5), the iterative sequential algorithm is adapted to robust optimization. Theoretically, at each iteration, whether an updated estimate of $(\boldsymbol{s}, \boldsymbol{T}, \boldsymbol{x})$ obtained using the subset ensemble should be be accepted or not, depends on whether the expected NPV of the whole ensemble models, $J_E^{\Omega_f}(\boldsymbol{s}, \boldsymbol{T}, \boldsymbol{x})$, is improved or not. However, considering that each evaluation of $J_E^{\Omega_f}(\boldsymbol{s}, \boldsymbol{T}, \boldsymbol{x})$ requires N_e simulation runs, we avoid evaluating it every iteration of the optimization algorithm. We emphasize that the base algorithm is an iterative sequential method which combines GA, StoSAG and GPS. If the population size of GA is set equal to 50, the number of GA iterations is set to 50, and the number of realizations in the subset Ω_s is set equal to 5, then it would require at least $50 * 50 * 5 = 12,500$ simulation runs to carry out the optimization of $(\boldsymbol{s}, \boldsymbol{T})$, without considering the simulation runs spent on adaptively selecting the subset realizations and on the StoSAG and GPS search steps to update $\boldsymbol{x}$. Hence, it is desirable to keep the the size of the subset ensemble small. Thus, in this chapter, the maximum number of realizations in Ω_s is set equal to 3. The representative models in the subset are selected as the models corresponding to the equally spaced percentiles of the cumulative distribution function (cdf) of the NPV's for N_e realizations. Suppose $N_e = 50$ and $N_{es} = 3$, then Ω_s includes the realizations corresponding to the P2, P50 and P100 (2th, 50th and 100th percentiles) of the cdf of NPV in Ω_f.

Below, we provide three different "procedures" in the context of Algorithm 6 to determine when to apply the estimates of $(\boldsymbol{s}, \boldsymbol{T}, \boldsymbol{x})$ obtained by maximizing $J_E^{\Omega_s}$ to all N_e realizations and how to update the subset ensemble. Procedures 1 and 2 are used as the benchmark for Procedure 3.

- Procedure 1. The cdf of NPV for the full ensemble, Ω_f, is only constructed once based on the initial settings of the well status, types and locations (i.e., the reference scenario generated using engineering judgment) and the realizations corresponding to P2, P50 and P100 are selected to form the subset ensemble. Throughout the optimization process, these three realizations are used to represent the geological uncertainty and each evaluation of the objective function (given in Eq. 4.5) requires 3 simulation runs.

- Procedure 2. The cdf of NPV for the full ensemble Ω_f is initially constructed based on the initial settings of $(\boldsymbol{s}, \boldsymbol{T}, \boldsymbol{x})$ and reservoir models corresponding to P2, P50 and P100 of the initial cdf of NPV are selected to form the initial subset. For each search step (the GA step, the StoSAG or the GPS step), we use the best estimate of the optimal solution (the estimate of $(\boldsymbol{s}, \boldsymbol{T}, \boldsymbol{x})$ that gives the largest expected NPV for the N_e realizations, denoted as $\boldsymbol{w}^{opt}$) as the initial guess. Denote the estimate of $(\boldsymbol{s}, \boldsymbol{T}, \boldsymbol{x})$ after carrying out the kth search step using the selected representative subset, Ω_s^{k-1}, as $\boldsymbol{v}^{k,opt}$. After each search step, the NPV's of the N_e realizations in the full ensemble are evaluated using $\boldsymbol{v}^{k,opt}$ and the expected NPV of the N_e realizations, $J_E^{\Omega_f}(\boldsymbol{v}^{k,opt})$, is calculated and a trial cdf of NPV is built. If $J_E^{\Omega_f}(\boldsymbol{v}^{k,opt})$ is no greater than the one calculated using the best estimate of the optimal solution, $J_E^{\Omega_f}(\boldsymbol{w}^{opt})$, we do not accept the updated solution, i.e., we replace the updated solution by the solution at the beginning of the step. Otherwise, we (i) accept $\boldsymbol{v}^{k,opt}$ as the best estimate of the optimal solution and accept the trial cdf of NPV, and (ii) select a new representative subset of realizations (reservoir models corresponding to P2, P50 and P100 of the newly built cdf of NPV), Ω_s^k, to represent the geological uncertainty for the next GA, StoSAG or GPS search step, whichever comes first.

- Procedure 3. In Procedure 2, to decide if an estimate of $(\boldsymbol{s}, \boldsymbol{T}, \boldsymbol{x})$ obtained with the subset is accepted as the best estimate of the optimal solution, the NPV's of the N_e realizations are evaluated after each search step (the GA step, StoSAG step or GPS step). However, it is possible that the preceding search step leads to a big change to the design variables $(\boldsymbol{s}, \boldsymbol{T}, \boldsymbol{x})$ with which the subset realizations are no longer representative. Hence, to decide if the subset needs to be re-selected, we also evaluate the N_e realizations within each search step provided that a specified criterion is satisfied. The cdf of the full ensemble Ω_f is initially constructed based on the reference scenario (the initial settings of the $(\boldsymbol{s}, \boldsymbol{T}, \boldsymbol{x})$). A set of reference realizations (Ω_{co}^0) are chosen as the reservoir models corresponding to P2, P25, P50, P27 and P100 of the initial cdf of NPV and one representative realization, the P50 of the initial cdf of NPV, is selected to form the representative subset. Denote the estimate of $(\boldsymbol{s}, \boldsymbol{T}, \boldsymbol{x})$ obtained with the subset realizations at the ℓth iteration as $\boldsymbol{v}^\ell$ and denote the best estimate of the optimal solution obtained till ℓth iteration as $\boldsymbol{w}^{opt}$. If $\boldsymbol{v}^\ell$ improves the expected NPV of the subset ensemble compared to last iteration, then the NPV's of the realizations in Ω_{co}^ℓ are evaluated using $\boldsymbol{v}^\ell$. If the ordering of the NPV's of the realizations in Ω_{co}^ℓ remains the same as the ordering at the best estimate of the optimal solution, the subset realizations are still considered to be representative and we move on to next iteration; otherwise the NPV's of the N_e realizations are evaluated using $\boldsymbol{v}^\ell$ and the expected NPV, $J_E^{\Omega_f}(\boldsymbol{v}^\ell)$, is calculated. If $J_E^{\Omega_f}(\boldsymbol{v}^\ell)$ is not greater than the one obtained with the best estimate of the optimal solution, $J_E^{\Omega_f}(\boldsymbol{w}^{opt})$, we do not accept the updated solution, i.e., we replace the updated solution by $\boldsymbol{w}^{opt}$, and enlarge the size of the subset to 3 if there is only one realization in the subset, i.e., the reservoir models corresponding to the P2, P50 and P100 of the best estimate of the cdf are selected. Otherwise, we (i) accept $\boldsymbol{v}^\ell$ as the next estimate of the optimal solution (set $\boldsymbol{w}^{opt} = \boldsymbol{v}^\ell$) and build the cdf of NPV based on N_e realizations using $\boldsymbol{v}^\ell$, and (ii) select a new representative subset of realizations (models corresponding to P2, P50 and P100 of the newly built cdf of NPV), Ω_s^ℓ, and a new reference set, Ω_{co}^ℓ, for next iteration.

For cases where the representative models are relatively independent of the design variables, $(\boldsymbol{s}, \boldsymbol{T}, \boldsymbol{x})$, using only one realization (the adaptively selected P50 realization) can greatly save the simulation cost; for cases where the representative models heavily rely on the design variables, $(\boldsymbol{s}, \boldsymbol{T}, \boldsymbol{x})$, the adaptively selected three realizations (P2, P50 and P100 realizations of the newly built cdf) can roughly represent the full ensemble. To estimate the StoSAG search direction, if the subset ensemble size is increased, the number of perturbations per realization, N_{pert}, is reduced in order to keep the computational cost low, i.e., we set $N_{pert} = 15$ for $N_{es} = 1$ and $N_{pert} = 5$ for $N_{es} = 3$. Algorithm 6 coupled with Procedure 3 is summarized in Algorithm 10. In the context of Algorithm 10, Procedure 3 is given in Algorithm 11.

In Algorithm 10, AL_{idx} represents the indices of the three search step, i.e., $AL_{idx} = 1$ represents the GA search step, $AL_{idx} = 2$ represents the StoSAG search step, $AL_{idx} = 3$ represents the GPS search step; $FL_{Sto} = true$ and $FL_{GPS} = true$ respectively represent failures in StoSAG search step and the GPS search step; $eval_{\Omega_f} = true$ represents evaluating the NPV's of the N_e realizations in Ω_f; $\boldsymbol{s}^0$, $\boldsymbol{T}^0$ and $\boldsymbol{x}^0$ respectively represent the initial settings of well status, types and locations (i.e., the reference scenario generated using engineering judgment); the superscript *opt* denotes the best estimate of the optimal solution; $\boldsymbol{x}^{opt}$, $\boldsymbol{c}^{opt}$ and $\boldsymbol{POP}^{opt}$ respectively represent the best estimate of well locations, status and types, and the population which contains $\boldsymbol{c}^{opt}$; ℓ denotes the number of iterations within each search step; Ω_s^ℓ and Ω_{co}^ℓ respectively represent the subset ensemble to account for the uncertainty and the reference ensemble to check the ordering of NPV's at ℓth iteration of a search step; $\boldsymbol{POP}_s^\ell$ represent the population at ℓth iteration; $\boldsymbol{c}_i^\ell$ and $\boldsymbol{c}_b^\ell$ respectively represent the ith chromosome and the best chromosome in $\boldsymbol{POP}_s^\ell$; $\boldsymbol{x}^{\ell,trial}$ and $\boldsymbol{x}^\ell$ respectively represent the well locations after a trial update and the actual value of well locations at ℓth iteration.

Robust optimization of well status, types and locations given the well controls, using the iterative sequential method coupled with Procedure 1, Procedure 2 and Procedure 3 are referred to Robust-1, Robust-2 and Robust-3, respectively. In this section, the convergence

Algorithm 10 Pseudo-code for adaptive robust optimization using Procedure 3

```
Preset $N_p$, $N_g$, $p_c$, $p_m$, $N_{GA}$, $N_{Sto}$ and $\Delta_0 = 1$. Set $FL_{Sto} = false$ and $FL_{GPS} = false$. Generate the initial settings of well status $\boldsymbol{s}^0$, types $\boldsymbol{T}^0$, and locations $\boldsymbol{x}^0$. Set $\boldsymbol{c}_b^0 = [s_1^0, T_1^0, s_2^0, T_2^0, \cdots, s_{K_{max}}^0, T_{K_{max}}^0]$, $\boldsymbol{c}^{opt} = \boldsymbol{c}^0$ and $\boldsymbol{x}^{opt} = \boldsymbol{x}^0$. Evaluate $J_E^{\Omega_f}(\boldsymbol{c}^{opt}, \boldsymbol{x}^{opt})$ and form $cdf^{opt}$. Set the size of the initial subset, $N_{es} = 1$, select the P50 realization of $cdf^{opt}$ to form $\Omega_s^0$ and select the "P2, P25, P50, P75 and P100" realizations of $cdf^{opt}$ to form $\Omega_{co}^0$. Set the number of perturbations used to estimate the StoSAG gradient at each iteration, $N_{pert} = 15$.
1. Generate the initial population, $\boldsymbol{POP}_s^0$, with $N_p$ random chromosomes and replace one random chromosome by $\boldsymbol{c}^0$. Evaluate $\boldsymbol{POP}_s^0$ and set $\boldsymbol{POP}^{opt} = \boldsymbol{POP}_s^0$.
2. Set $\ell = 0$, $\boldsymbol{POP}_s^\ell = \boldsymbol{POP}^{opt}$, $\boldsymbol{c}_b^\ell = \boldsymbol{c}^{opt}$ and $\boldsymbol{x}^\ell = \boldsymbol{x}^{opt}$. If $AL_{idx} = 1$, evaluate each chromosome in $\boldsymbol{POP}_s^\ell$ using $\boldsymbol{x}^{opt}$, where the fitness of a chromosome is the expected NPV of the realizations in $\Omega_s^\ell$, and go to step 3; if $AL_{idx} = 2$, go to step 4; if $AL_{idx} = 3$, go to step 5.
3. Continue for $N_{GA}$ generations of GA to maximize $J_E^{\Omega_s^\ell}$ by adjusting the well status and types. At each iteration, (i) set $\Omega_s^{\ell+1} = \Omega_s^\ell$, $\Omega_{co}^{\ell+1} = \Omega_{co}^\ell$ and $\ell = \ell + 1$; (ii) generate $\boldsymbol{POP}_s^\ell$ using GA operators based on $\boldsymbol{POP}_s^{\ell-1}$; (iii) evaluate $J_E^{\Omega_s^{\ell-1}}(\boldsymbol{c}_i^\ell, \boldsymbol{x}^{opt})$ for $\boldsymbol{c}_i^\ell$ in $\boldsymbol{POP}_s^\ell$ and find the chromosome with the largest $J_E^{\Omega_s^{\ell-1}}(\boldsymbol{c}_i^\ell, \boldsymbol{x}^{opt})$, $\boldsymbol{c}_b^\ell$; (iv) if $J_E^{\Omega_s^{\ell-1}}(\boldsymbol{c}_b^\ell, \boldsymbol{x}^{opt}) > J_E^{\Omega_s^{\ell-1}}(\boldsymbol{c}_b^{\ell-1}, \boldsymbol{x}^{opt})$, apply Procedure 3 given in Algorithm 11.
4. Maximize $J_E^{\Omega_s^\ell}$ by adjusting $\boldsymbol{x}$, using StoSAG for $N_{Sto}$ iterations. The number of perturbations used to estimate the StoSAG gradient per realization is set equal to $\frac{N_{pert}}{N_{es}}$. At each iteration, (i) set $\Omega_s^{\ell+1} = \Omega_s^\ell$, $\Omega_{co}^{\ell+1} = \Omega_{co}^\ell$ and $\ell = \ell+1$; (ii) search along the StoSAG gradient to obtain a trial update $\boldsymbol{x}^{\ell,trial}$; (iii) if $J_E^{\Omega_s^{\ell-1}}(\boldsymbol{c}^{opt}, \boldsymbol{x}^{\ell,trial}) > J_E^{\Omega_s^{\ell-1}}(\boldsymbol{c}_b^{opt}, \boldsymbol{x}^{\ell-1})$, set $\boldsymbol{x}^\ell = \boldsymbol{x}^{\ell,trial}$ and apply Procedure 3 given in Algorithm 11, otherwise set $\boldsymbol{x}^\ell = \boldsymbol{x}^{\ell-1}$ and go to next iteration. If StoSAG failed to improve $J_E^{\Omega_s^{\ell-1}}$ for $N_{res}$ iterations, set $FL_{Sto} = true$; otherwise, set $FL_{Sto} = false$. Go to Step 7.
5. Maximize $J_E^{\Omega_s^\ell}$ by adjusting $\boldsymbol{x}$ along $N_x$ coordinate directions. At each iteration, (i) set $\Omega_s^{\ell+1} = \Omega_s^\ell$, $\Omega_{co}^{\ell+1} = \Omega_{co}^\ell$ and $\ell = \ell+1$; (ii) search along the coordinate direction to obtain a trial update $\boldsymbol{x}^{\ell,trial}$; (iii) if $J_E^{\Omega_s^{\ell-1}}(\boldsymbol{c}^{opt}, \boldsymbol{x}^{\ell,trial}) > J_E^{\Omega_s^{\ell-1}}(\boldsymbol{c}_b^{opt}, \boldsymbol{x}^{\ell-1})$, set $\boldsymbol{x}^\ell = \boldsymbol{x}^{\ell,trial}$ and apply Procedure 3 given in Algorithm 11, otherwise set $\boldsymbol{x}^\ell = \boldsymbol{x}^{\ell-1}$ and go to next iteration. If GPS failed to improve $J_E^{\Omega_s}$ for $N_x$ coordinate directions, set $FL_{GPS} = true$; otherwise, set $FL_{GPS} = false$.
6. If $AL_{idx} \neq 1$, set $AL_{idx} = 1$; otherwise, set $AL_{idx} = 2$ if $FL_{Sto} = false$, and set $AL_{idx} = 3$ if $FL_{Sto} = true$ and $FL_{GPS} = false$. Go to step 2.
```

criterion for Robust-1 is set as failure to improve the expected NPV of the three fixed realizations in a GA step followed by a GPS step or a GPS step followed by a GA step. The convergence criterion for Robust-2 and Robust-3 is set as consecutive failures in updating

Algorithm 11 Procedure 3 for adaptive robust optimization of well status, types and locations

```
• Set $eval_{\Omega_f} = false$.
• If $AL_{idx} = 1$,
    – set $\boldsymbol{v}_b^\ell = (\boldsymbol{c}_b^\ell, \boldsymbol{x}^{opt})$;
    – if $\ell = N_{GA}$, set $eval_{\Omega_f} = true$.
• If $AL_{idx} = 2$,
    – set $\boldsymbol{v}_b^\ell = (\boldsymbol{c}^{opt}, \boldsymbol{x}^\ell)$;
    – if $\ell = N_{Sto}$ or $FL_{sto} = true$, set $eval_{\Omega_f} = true$.
• If $AL_{idx} = 3$,
    – set $\boldsymbol{v}_b^\ell = (\boldsymbol{c}^{opt}, \boldsymbol{x}^\ell)$;
    – if $\ell = N_x$, set $eval_{\Omega_f} = true$.
• If $eval_{\Omega_f} = false$,
    – evaluate $J(\boldsymbol{m}_k, \boldsymbol{v}_b^\ell), \forall\, \boldsymbol{m}_k \in \Omega_{co}^{\ell-1}$;
    – if the ordering of the NPV's in $\Omega_{co}^{\ell-1}$ change, set $eval_{\Omega_f} = true$.
• If $eval_{\Omega_f} = true$,
    – evaluate $J_E^{\Omega_f}(\boldsymbol{v}_b^\ell)$ and build a trial cdf, $cdf^{trial}$;
    – if $J_E^{\Omega_f}(\boldsymbol{v}_b^\ell) > J_E^{\Omega_f}(\boldsymbol{c}^{opt}, \boldsymbol{x}^{opt})$,
        ∗ if $AL_{idx} = 1$, set $\boldsymbol{c}^{opt} = \boldsymbol{c}_b^\ell$ and $\boldsymbol{POP}^{opt} = \boldsymbol{POP}_s^\ell$;
        ∗ if $AL_{idx} \neq 1$, set $\boldsymbol{x}^{opt} = \boldsymbol{x}^\ell$;
        ∗ set $cdf^{opt} = cdf^{trial}$ and re-select realizations from $cdf^{opt}$ to form $\Omega_s^\ell$ and $\Omega_{co}^\ell$;
        ∗ evaluate $J_E^{\Omega_s^\ell}(\boldsymbol{v}_b^\ell)$.
    – otherwise,
        ∗ set $\boldsymbol{x}^\ell = \boldsymbol{x}^{opt}$, $\boldsymbol{c}_b^\ell = \boldsymbol{c}^{opt}$ and $\boldsymbol{POP}_s^\ell = \boldsymbol{POP}^{opt}$;
        ∗ if $N_{es} = 1$, set $N_{es} = 3$, and re-select the "P2, P50 and P100" realizations of $cdf^{opt}$ to form $\Omega_s^\ell$.
• Otherwise, move to next iteration.
```

the cdf of the N_e realizations in a GA step followed by a GPS step or a GPS step followed by a GA step. The three methods (i.e., Robust-1, Robust-2 and Robust-3) are terminated either at convergence or when the number of simulation runs exceeds a pre-defined value.

4.4 Numerical Examples of Robust Optimization

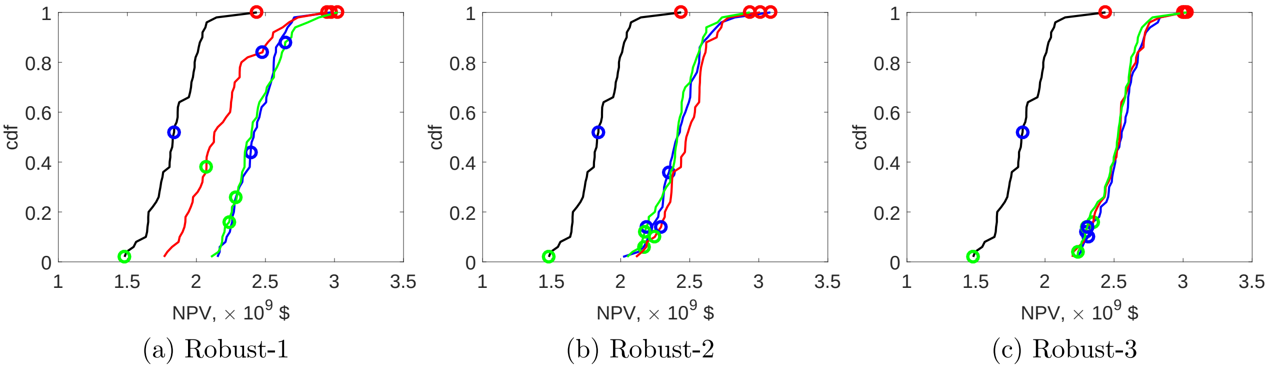
In this section, the three robust optimization procedures (i.e., Robust-1, Robust-2 and Robust-3) are applied to both the PUNQ model and the Brugge model. For both examples, 50 realizations of the reservoir model are generated to represent the geological uncertainty. The maximum number of simulation runs is set as 10,000 for the PUNQ model and 8,000 for the Brugge model.

4.4.1 Example 1: PUNQ model

The PUNQ model is considered where 50 realizations of the horizontal and vertical permeability fields are used to represent the geological uncertainty. The same reference well placement scenario, well control settings and economic parameters are used as in deterministic optimization presented in Section 4.2.1. The only difference is that we consider the minimum well spacing as one gridblock, i.e., two wells cannot be placed in the same gridblock. The NPV's for all 50 realizations are calculated using the reference scenario given in Fig. 4.3(a) where the cdf of NPV is shown as the black curve in Fig. 4.21. The P2, P50 and P100 realizations correspond to realization #48, #8 and #27 respectively, whose log-permeability fields are shown in Fig. 4.22. From Fig. 4.22, the permeability fields of these three realizations are quite different, indicating the large geological uncertainty in the reservoir models. We first carry out the adaptive robust optimization using iterative sequential steps of GA, StoSAG and GPS, and then we consider the case where the StoSAG search step is not used.

1. Robust optimization using the iterative sequential algorithm with GA, StoSAG and GPS

Since GA and StoSAG are stochastic algorithms, three initial seeds are tried for each of the three robust optimization procedures. Fig. 4.21 shows the cdf's of NPV obtained with the reference scenario and the optimal well placement using three seeds. The cdf's of NPV are generated based on evaluating the NPV of all $N_e = 50$ reservoir models given an estimate of $(\boldsymbol{s}, \boldsymbol{T}, \boldsymbol{x})$. The green, blue and red circles respectively represent the P2, P50 and P100 realizations of the initial cdf of NPV (the black curve) obtained with the reference scenario. From Fig. 4.21, the subset realizations selected from the initial cdf no longer correspond to the 2th, 50th and 100th percentiles of the cdf's of NPV obtained with the optimal solutions, indicating that the representative realizations heavily depend on the optimization variables for the PUNQ model and that it is necessary to adaptively re-select the representative realizations.



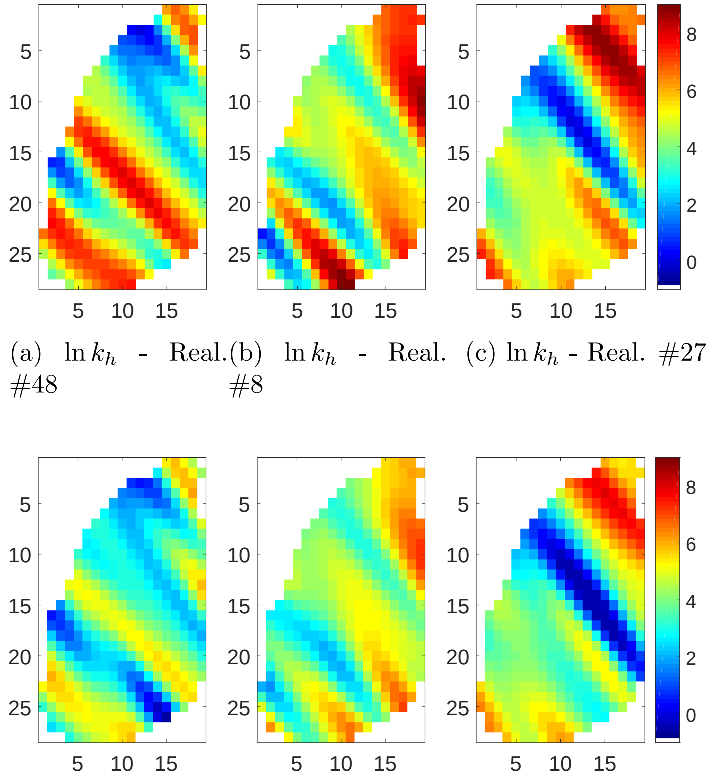
(a) Robust-1

(b) Robust-2

(c) Robust-3

Figure 4.21: Cdf's of NPV obtained with the reference scenario (black curve) and the optimal well placements for the three robust procedures using three seeds (blue curve for seed 1, red curve for seed 2 and green curve for seed 3), robust optimization of the PUNQ model. The green, blue and red circles respectively represent the P2, P50 and P100 of the cdf of NPV obtained with the reference scenario.

From Fig. 4.21(a), using Robust-1 (where the subset is fixed during the optimization process), the optimal cdf's obtained with the three seeds are quite different in that seed 2 obtained lower optimal NPV's for almost all 50 realizations compared to those of the other two seeds. From Figs. 4.21(b) and (c), the optimal results obtained with Robust-2 (where the subset is re-selected at each switch of the search steps) and Robust-3 (where the subset is re-selected following Algorithm 11), are almost independent of the seed. For example, the difference between the best and worst expected NPV of the full ensemble for the optimal solutions obtained with Robust-2 and Robust-3 are respectively $\$50 \times 10^6$ (2% of the average optimal NPV) and $\$30 \times 10^6$ (1.2% of the average optimal NPV). Besides being consistent, the optimal expected NPV obtained with Robust-2 and Robust-3 are respectively 3.6% and 7.7% higher than the one obtained with Robust-1, on average of three seeds.



(a) $\ln k_h$ - Real. #48 (b) $\ln k_h$ - Real. #8 (c) $\ln k_h$ - Real. #27

(d) $\ln k_v$ - Real. #48 (e) $\ln k_v$ - Real. #8 (f) $\ln k_v$ - Real. #27

Figure 4.22: Realizations of the log-horizontal permeability fields (upper row) and the log-vertical permeability fields (bottom row), corresponding to P2, P50 and P100 of the cdf of NPV obtained with the reference well placement scenario, PUNQ model. k_h and k_v represent the horizontal and vertical permeability respectively.

Fig. 4.23 shows the expected NPV's of the subset versus number of simulations using three fixed subset realizations (realizations #48, #8 and #27) during three runs of Robust-1. We apply the optimal solutions obtained with the subset ensemble to all N_e reservoir models and presented the expected NPV's of the full ensemble using dashed lines in color in Fig. 4.23. From Fig. 4.23, the optimal solution obtained with seed 2 gives an expected NPV of \$$2.5 \times 10^9$ for the subset, but only \$$2.18 \times 10^9$ for the full ensemble which is 13% less, due to the loss of representativeness of the subset realizations.

Fig. 4.24 shows the expected NPV's for the full ensemble with respect to number of simulation runs using Robust-2 where each point represents a switch of the search steps. From Fig. 4.24, both GA and GPS search step increased the expected NPV of the full

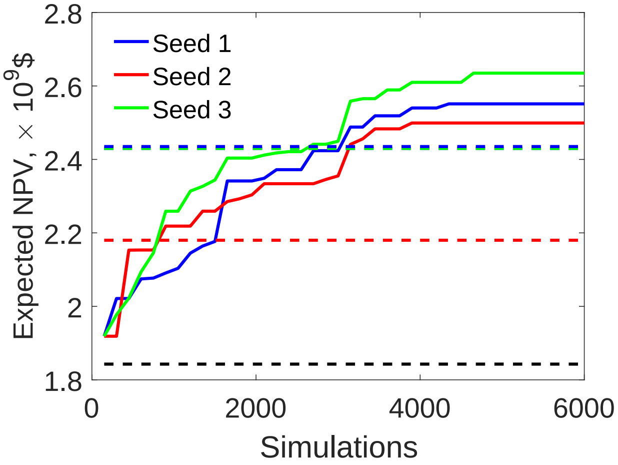
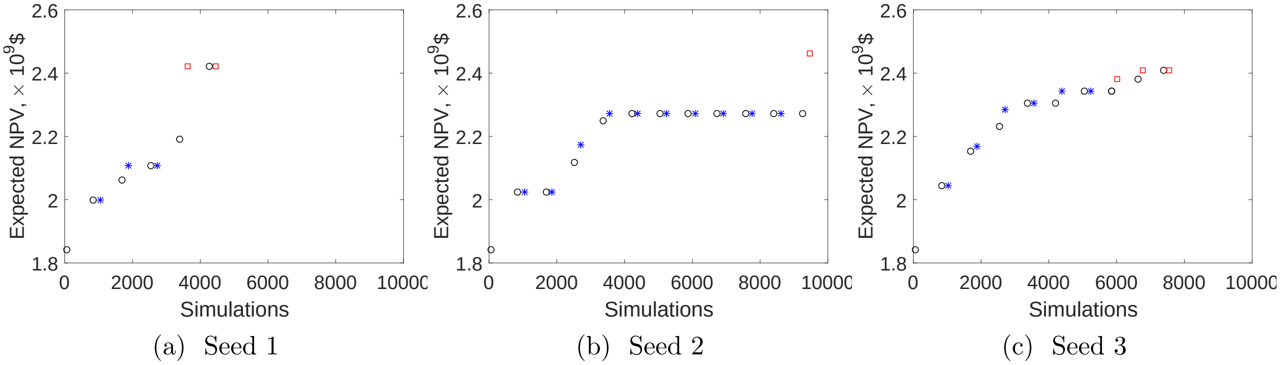


Figure 4.23: Expected NPV of the susbet ensemble versus number of simulation runs obtained with Robust-1 using three seeds (blue curve for seed 1, red curve for seed 2 and green curve for seed 3), robust optimization of the PUNQ model. The solid lines represent the expected NPV's of the subset ensemble; the dashed lines in color represent the optimal expected NPV's of the full ensemble; the black dashed line is the expected NPV of the full ensemble obtained with the reference scenario.

ensemble significantly, however, the StoSAG search step only improved it slightly. For seed 2, the optimization got stuck from around 4,000 simulation runs to 9,000 simulation runs, but then provided a sharp increase in the expected NPV of the full ensemble. Fig. 4.25 shows both the expected NPV's over the subset and the full ensemble during the StoSAG and GPS search step using seed 2. From Fig. 4.25, the StoSAG step could efficiently improve the expected NPV of the subset (the blue dots), but not the average NPV of the full ensemble (the blue stars). This is because the StoSAG search direction is only a stochastic uphill direction, and $N_{Sto} = 5$ iterations along the StoSAG search direction lead to a big change in the well locations with which the selected subset realizations are no longer representative, even though the improvement in the expected NPV's of the subset seems to be less than that in the GPS step. Hence, optimizing over the subset realizations using StoSAG only slightly improves the expected NPV of the full ensemble.

Fig. 4.26 shows the expected NPV's for the full ensemble with respect to number of simulation runs using Robust-3 where it is less frequent for the expected NPV of the full ensemble to get stuck. This is because that Robust-3 procedure not only constructs the cdf's



(a) Seed 1 (b) Seed 2 (c) Seed 3

Figure 4.24: Expected NPV of the full ensemble versus number of simulations obtained for Robust-2 using three seeds, robust optimization of PUNQ model. o represents an update by GA, $*$ represents an update by StoSAG, and □ represents an update by GPS.

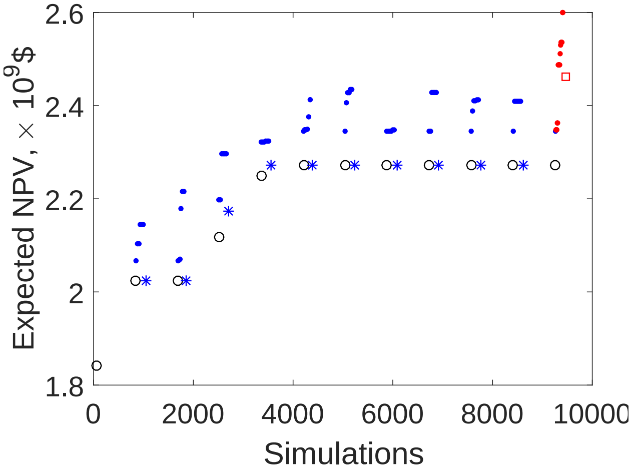
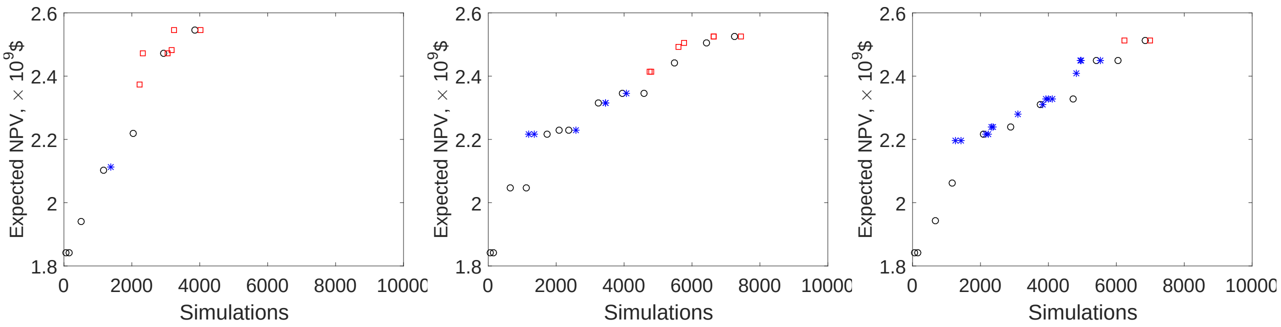


Figure 4.25: Expected NPV versus number of simulations obtained with Robust-2 using seed 2, robust optimization of PUNQ model. o, $*$, and □ respectively represent the expected NPV of the full ensemble obtained using GA, StoSAG and GPS; . and . respectively represent the expected NPV of the subset ensemble obtained using StoSAG and GPS.

of NPV at each switch of the search steps, but also when the ordering of NPV's of a set of reference realizations changes. This tends to prevent large updates to the well status, types and locations between two neighboring cdf constructions, which may cause the representative subset to become non-representative. It is worth to mention that Robust-3 procedure starts with a subset containing only one realization, however, the size of the subset is increased to 3 after the first generation of GA, since as we have observed, the representativeness of realizations for the PUNQ model heavily depends on the optimization variables.

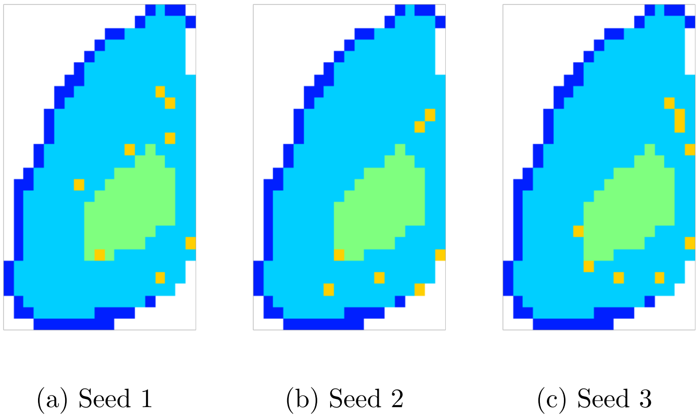


(a) Seed 1 (b) Seed 2 (c) Seed 3

Figure 4.26: Expected NPV of the full ensemble versus number of simulations obtained for Robust-3 using three seeds, robust optimization of PUNQ model. o represents an update by GA, $*$ represents an update by StoSAG, and □ represents an update by GPS.

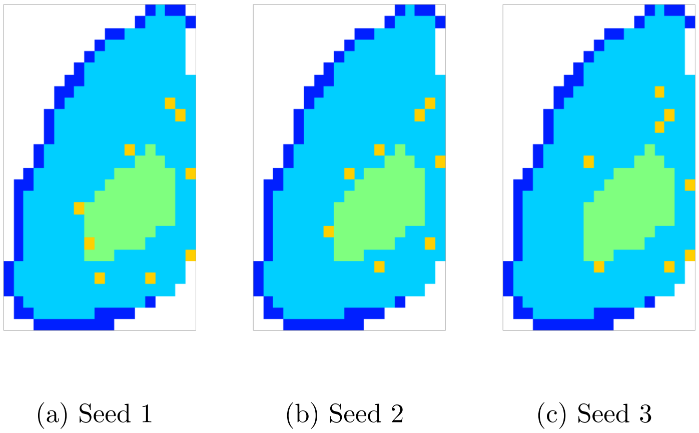
Figs. 4.27, 4.28 and 4.29 respectively show the optimal wells drilled, their types and locations obtained with Robust-1, Robust-2 and Robust-3 where each optimization procedure is repeated three times using different seeds. Due to the existence of a strong aquifer which provides pressure support for producers, the optimal solutions tend to avoid drilling any injectors except that one injector is drilled in the optimal solution of Robust-3 using seed 1. Meanwhile, the number of producers obtained using three seeds ranges from 7 to 10, which are quite consistent with each other.

The expected NPV's of the full ensemble and the subset ensemble obtained with optimal solutions, the optimal number of producers along with the simulation costs to obtain convergence are summarized in Table 4.7. Note that the maximum number of simulation runs is set to be 10,000 and seed 2 of Robust-2 didn't reach convergence before the algorithm termination. From Table 4.7, based on the average performance of three stochastic runs, (i) Robust-1 costs the least number of simulation runs, but provides the lowest expected NPV's of the full ensemble; (ii) Robust-2 gives higher expected NPV's of the full ensemble compared to Robust-1, but costs the highest number of simulation runs; (iii) Robust-3 requires 1,200 simulation runs less than Robust-2, but 500 simulation runs more than Robust-1, however, it provides the highest expected NPV's of the full ensemble; (iv) the expected NPV's of the subset ensemble are generally higher than those of the full ensemble at the optimal solutions since the former is the objective that we actually maximize, and the difference between the former and the latter is much larger for Robust-1 compared to Robust-2 and Robust-3.



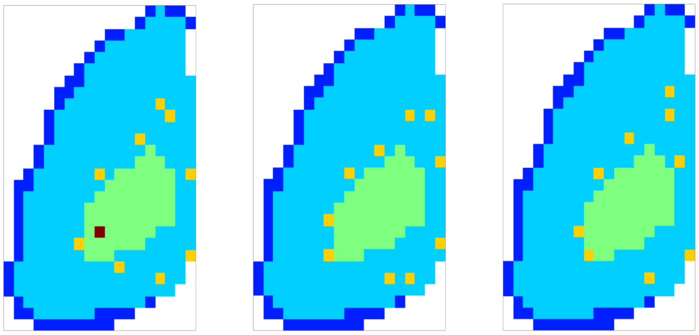
(a) Seed 1 (b) Seed 2 (c) Seed 3

Figure 4.27: Optimal types and locations of drilled wells obtained with Robust-1 using three seeds, robust optimization of the PUNQ model. Injector locations are marked in brown color and labeled with ⊗; producer locations are marked in yellow color and labeled with o; the gas cap is marked in green color; the aquifer is marked in dark blue and the oil zone is marked in light blue color.



(a) Seed 1 (b) Seed 2 (c) Seed 3

Figure 4.28: Optimal types and locations of drilled wells obtained for Robust-2 using three seeds, robust optimization of PUNQ model. Injector locations are marked in brown color and labeled with ⊗; producer locations are marked in yellow color and labeled with o; the gas cap is marked in green color; the aquifer is marked in dark blue and the oil zone is marked in light blue color.



(a) Seed 1 (b) Seed 2 (c) Seed 3

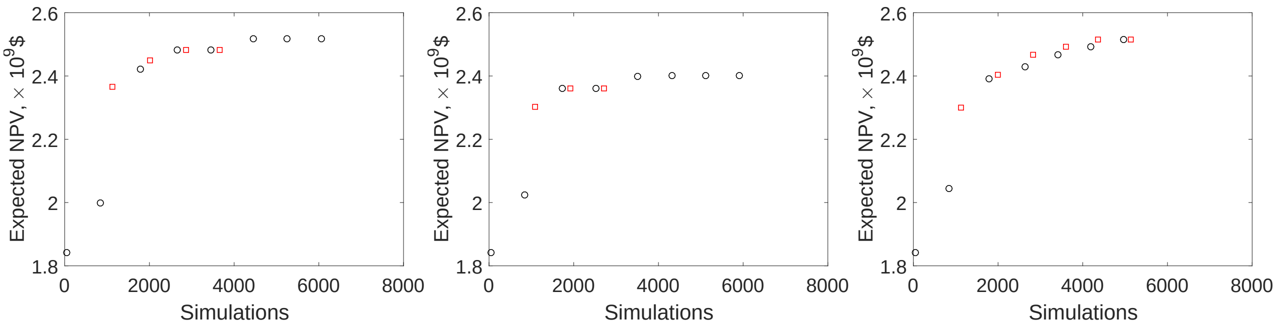
Figure 4.29: Optimal types and locations of drilled wells obtained for Robust-3 using three seeds, robust optimization of PUNQ model. Injector locations are marked in brown color and labeled with ⊗; producer locations are marked in yellow color and labeled with o; the gas cap is marked in green color; the aquifer is marked in dark blue and the oil zone is marked in light blue color.

Table 4.7: Summary of the optimal solutions and computational cost obtained with the three robust optimization procedures using three seeds, robust optimization of PUNQ model.

Seeds	Robust-1				Robust-2				Robust-3			
	Sim.	$\$ \times 10^9$		N_{Pro}	Sim.	$\$ \times 10^9$		N_{Pro}	Sim.	$\$ \times 10^9$		N_{Pro}
	#	$J_E^{\Omega_f}$	$J_E^{\Omega_s}$		#	$J_E^{\Omega_f}$	$J_E^{\Omega_s}$		#	$J_E^{\Omega_f}$	$J_E^{\Omega_s}$	
Seed 1	5950	2.43	2.56	8	4465	2.42	2.51	9	4031	2.55	2.60	9
Seed 2	5500	2.18	2.50	7	10000	2.46	2.54	8	7443	2.53	2.58	10
Seed 3	5500	2.43	2.63	9	7547	2.41	2.46	8	6984	2.52	2.60	10
Average	5650	2.35	2.56	8	7337	2.43	2.50	8	6153	2.53	2.59	10

2. Robust optimization using the iterative sequential algorithm with GA and GPS

Due to the observation that the StoSAG optimization for the subset realizations after $N_{Sto} = 5$ iterations only slightly improves the expected NPV of the full ensemble, we investigate Robust-1, Robust-2 and Robust-3 optimization procedures using only the GA search step and the GPS search step. Figs. 4.30 and 4.31 show the expected NPV of the full ensemble versus number of simulations obtained with Robust-2 and Robust-3 respectively. The convergence speed of Figs. 4.30 and 4.31 has been greatly improved compared to Figs. 4.24 and 4.26, i.e., saving over 1,300 simulation runs per seed. The expected NPV's of the full ensemble and the subset ensemble obtained with the optimal solutions, the optimal number of producers along with the simulation cost to reach convergence are summarized in Table 4.8. From Table 4.8, based on the average performance of three stochastic runs, (i) Robust-3 remains giving highest expected NPV's of the full ensemble while Robust-1 still gives the worst expected NPV for $N_e = 50$ realizations; (ii) the expected NPV's of the subset ensemble are higher than those of the full ensemble at the optimal solutions and the difference in between is much larger for Robust-1 compared to Robust-2 and Robust-3; (iii) the optimal number of producers are consistent using different optimization procedures and different seeds, i.e., 7 to 10 producers are drilled. Note almost no injectors are drilled, though we do not show it.

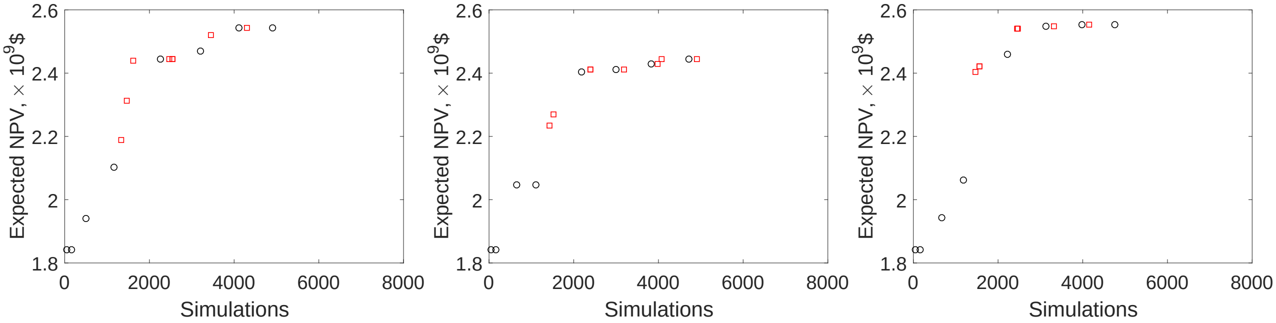


(a) Seed 1 (b) Seed 2 (c) Seed 3

Figure 4.30: Expected NPV of the full ensemble versus number of simulations obtained for Robust-2 using the iterative sequential method with GA and GPS steps, robust optimization of the PUNQ model. o's and □'s respectively represent updates by GA and GPS.

Table 4.8: Summary of the optimal solutions and computational cost obtained with the three robust optimization procedures with GA and GPS, robust optimization of PUNQ model.

Seeds	Robust-1				Robust-2				Robust-3			
	Sim.	$ × 10^9		N_{Pro}	Sim.	$ × 10^9		N_{Pro}	Sim.	$ × 10^9		N_{Pro}
	#	$J_E^{\Omega_f}$	$J_E^{\Omega_s}$		#	$J_E^{\Omega_f}$	$J_E^{\Omega_s}$		#	$J_E^{\Omega_f}$	$J_E^{\Omega_s}$	
Seed 1	6100	2.39	2.59	9	6055	2.52	2.60	9	4905	2.54	2.59	10
Seed 2	3100	2.36	2.57	7	5915	2.40	2.46	9	4912	2.44	2.50	8
Seed 3	6400	2.44	2.59	8	5135	2.52	2.60	9	4749	2.55	2.60	9
Average	5200	2.40	2.58	8	5700	2.48	2.55	9	4855	2.51	2.56	9



(a) Seed 1 (b) Seed 2 (c) Seed 3

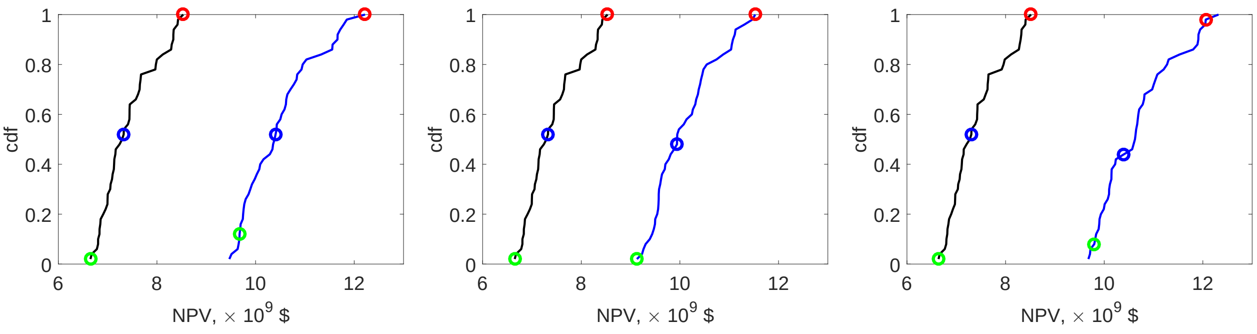
Figure 4.31: Expected NPV of the full-ensemble versus number of simulations obtained for Robust-3 using the iterative sequential method with GA and GPS steps, robust optimization of PUNQ model. o's and □'s respectively represent updates by GA and GPS.

4.4.2 Example 2: Brugge model

In the original Brugge model, 104 geological realizations of the horizontal permeability, vertical permeability, porosity, connate water saturation and the net-gross ratio are generated to describe the reservoir uncertainty. However, due to our limited computational resources, we only use 50 realizations where 10 realizations are generated using each of the 5 geo-statistical methods (see [71] for more details). The same reference well placement scenario, well control settings, economic parameters and minimum well spacing are used as in deterministic optimization presented in Section 4.2.2. The NPV's for all 50 realizations are calculated using the reference scenario given in Fig. 4.15(a) where the cdf of NPV is shown as the black curve in Fig. 4.32. The P2, P50 and P100 realizations correspond to realization #32, #14 and #9 respectively, whose petrophysical parameters for layer 1 are shown in Fig. 4.33. From Fig. 4.33, the petrophysical parameters of these three realizations are quite different, indicating the large geological uncertainty in the reservoir models.

1. Robust optimization using the iterative sequential method with GA, StoSAG and GPS

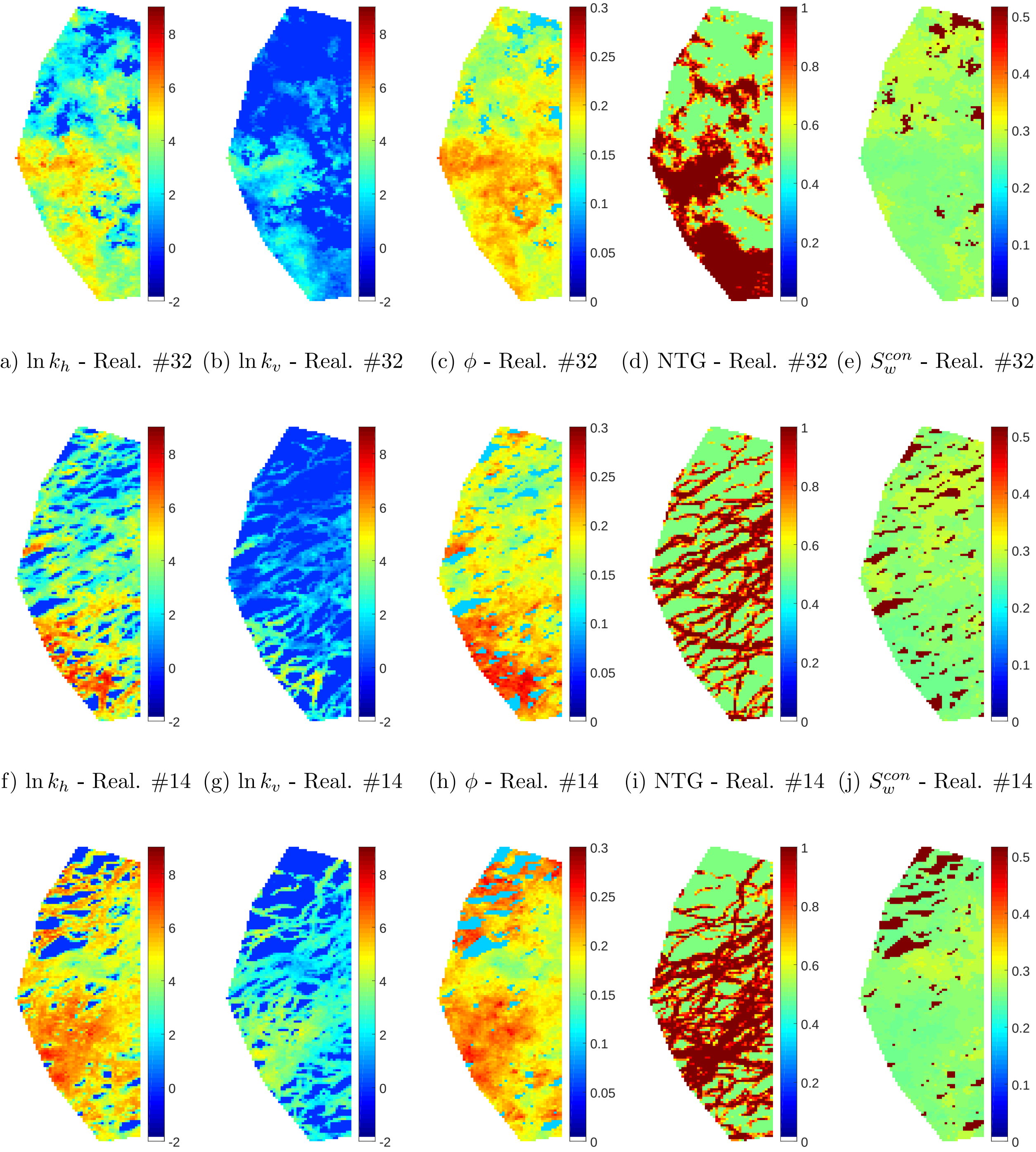
Since each simulation run for the Brugge model costs approximately 2 minutes and it has been observed for the PUNQ model that the inclusion of StoSAG search step in the iterative sequential method may lead to a slow convergence speed, only one run is tried for each of the three robust optimization procedures using GA, StoSAG and GPS search steps. Fig. 4.32 shows the cdf's of NPV obtained with the reference scenario (given in Fig. 4.15(a)) and the optimal well placement for Robust-1, Robust-2 and Robust-3 respectively. The cdf's of NPV are generated based on evaluating the NPV of all $N_e = 50$ reservoir models given an estimate of $(\boldsymbol{s}, \boldsymbol{T}, \boldsymbol{x})$. The green, blue and red circles respectively represent the P2, P50 and P100 realizations of the cdf of NPV obtained with the reference scenario (the black curve). From Fig. 4.32, the subset models selected from the initial cdf still roughly correspond to the 2th, 50th and 100th percentiles of the cdf's of NPV obtained with the optimal solutions, indicating that the representative realizations are relatively independent of the optimization variables (well status, types and locations) for the Brugge model, and that it is unnecessary to frequently re-select the representative realizations.



(a) Robust-1 (b) Robust-2 (c) Robust-3

Figure 4.32: Cdf's of NPV's obtained with the reference scenario (black curve) and the optimal well placement for three robust optimization procedures, robust optimization of Brugge model. The green, blue and red circles, respectively, represent the P2, P50 and P100 of the cdf of NPV obtained with the reference scenario.

With Robust-1 procedure where the subset ensemble (containing 3 realizations) are fixed during the optimization process, the optimal solution gives an expected NPV of \$10.77 × 10^9 for the subset and \$10.47 × 10^9 for the full ensemble which is 3% less; see Fig. 4.34. With Robust-2 procedure where the subset ensemble (containing 3 realizations) is re-selected at each switch of the algorithms, i.e., from GA to StoSAG/GPS or vice versa, the optimal solution gives an expected NPV of \$10.10 × 10^9 for the full ensemble, which is far less than the value obtained with Robust-1; see Fig. 4.35. This is because the StoSAG step converges



(a) $\ln k_h$ - Real. #32 (b) $\ln k_v$ - Real. #32 (c) ϕ - Real. #32 (d) NTG - Real. #32 (e) S_w^{con} - Real. #32

(f) $\ln k_h$ - Real. #14 (g) $\ln k_v$ - Real. #14 (h) ϕ - Real. #14 (i) NTG - Real. #14 (j) S_w^{con} - Real. #14

(k) $\ln k_h$ - Real. #9 (l) $\ln k_v$ - Real. #9 (m) ϕ - Real. #9 (n) NTG - Real. #9 (o) S_w^{con} - Real. #9

Figure 4.33: Petrophysical parameters for the top layer of the realizations corresponding to P2, P50 and P100 of the cdf of NPV obtained with the reference well placement scenario, Brugge model. k_h and k_v represents the horizontal and vertical permeability respectively, ϕ is the porosity, NTG is the net to gross ratio and S_w^{con} is the connate water saturation.

slowly when it comes near a local solution, but the slight improvements of the expected NPV's of the subset ensemble and the full ensemble are sufficient to avoid triggering the convergence criteria for the StoSAG search step until about 8,000 simulations and hence delayed the GPS fine-tune step for Robust-2. Given the maximum number of simulation runs as 8,000 for robust optimization, the inclusion of StoSAG search steps seems disadvantageous. In fact the resulting slow convergence could happen to all three robust optimization procedures and Robust-2 happens to be an unlucky run. With Robust-3 procedure where the subset ensemble (starting with ensemble size 1) is re-selected following Algorithm 11, the optimal solution gives the highest expected NPV of the full ensemble, $\$10.67\times10^9$, and the fastest convergence speed; see Fig. 4.36. For Robust-3, the subset ensemble only contains one realization in the early iterations and hence can save simulation costs and search more thoroughly within the specified number of simulation runs.

The optimal expected NPV's for the full ensemble and subset ensemble, the optimal number of injectors and producers along with the simulation costs to achieve convergence are summarized in Table 4.9. Note we terminate the optimization process when the number of simulation runs exceeds 8,000 even if it hasn't converged yet. From Table 4.9, (i) Robust-3 requires less simulation runs compared to Robust-1 and Robust-2, and provides the highest expected NPV's for the full ensemble; (ii) the expected NPV's of the subset ensemble are generally higher than those of the full ensemble at the optimal solutions since the former is the objective that we actually maximize, and the difference between the former and the latter is quite similar for all three optimization procedures which is consistent with Fig. 4.32; and (iii) the optimal number of wells are similar using different optimization procedures, i.e., 23 to 25 wells are drilled. Figs. 4.37, 4.38 and 4.39 respectively show the optimal wells drilled, their types and locations obtained with the three optimization procedures, along with the oil saturation of the top layer after 20 years of production obtained with realization #14.

2. Robust optimization using the iterative sequential method with GA and GPS

Due to the observation that the inclusion of StoSAG optimization may lead to a slow convergence speed, we investigate the Robust-1, Robust-2 and Robust-3 optimization

Table 4.9: Summary of the optimal solutions and computational cost obtained with three robust optimization procedures, robust optimization of the Brugge model.

Seeds	Robust-1					Robust-2					Robust-3				
	Sim.	$\$ \times 10^9$		N_{Inj}	N_{Pro}	Sim.	$\$ \times 10^9$		N_{Inj}	N_{Pro}	Sim.	$\$ \times 10^9$		N_{Inj}	N_{Pro}
	#	$J_E^{\Omega_f}$	$J_E^{\Omega_s}$			#	$J_E^{\Omega_f}$	$J_E^{\Omega_s}$			#	$J_E^{\Omega_f}$	$J_E^{\Omega_s}$		
Seed 1	7798	10.47	10.77	8	15	8000	10.10	10.20	14	11	7159	10.67	10.88	15	8

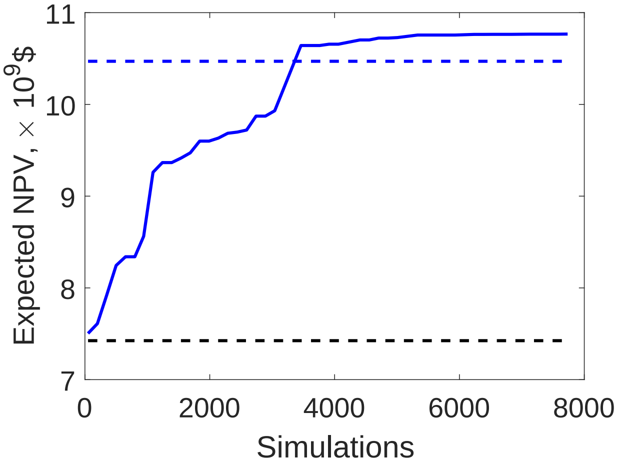


Figure 4.34: Expected NPV of the subset ensemble obtained with the best candidate solution of a population versus number of simulation runs obtained with Robust-1, robust optimization of Brugge model. The solid line in blue represents the expected NPV's of the subset ensemble, the blue dashed line represents optimal expected NPV of the full ensemble, and the black dashed line represents the expected NPV of the full ensemble obtained with the reference scenario.

procedures using the iterative sequential method with GA and GPS search steps only. Three runs are tried for each of the robust optimization procedures. Fig. 4.40 shows the cdf of NPV for the 50 realizations obtained with the reference scenario given in Fig. 4.15(a) and the optimal solutions for the three optimization procedures. Similar to Fig. 4.32, the subset models selected from the initial cdf still roughly correspond to the 2th, 50th and 100th percentiles of the cdf's of NPV obtained with the optimal solutions, indicating that the representative realizations are relatively independent of the optimization variables (well status, types and locations) for the Brugge model. This indicates that the optimal expected NPV's of the full ensemble obtained with Robust-1, Robust-2 and Robust-3, should be more or less the same, since it is not necessary to re-select the representative subset ensemble frequently.

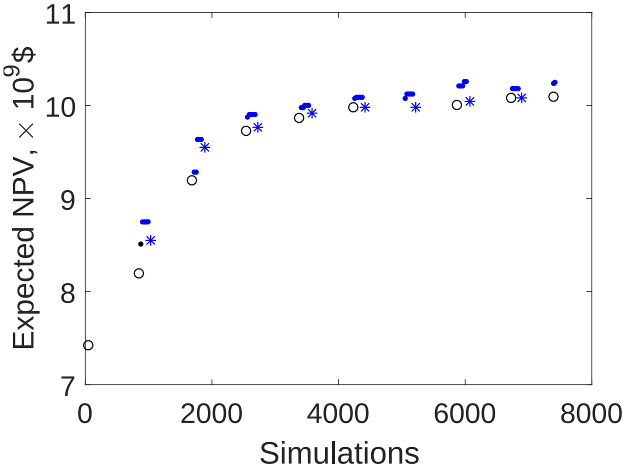


Figure 4.35: Expected NPV versus number of simulations obtained with Robust-2, robust optimization of the Brugge model. o, ∗ respectively represent the expected NPV of the full ensemble obtained using GA and StoSAG; . represents the expected NPV of the subset ensemble obtained using StoSAG.

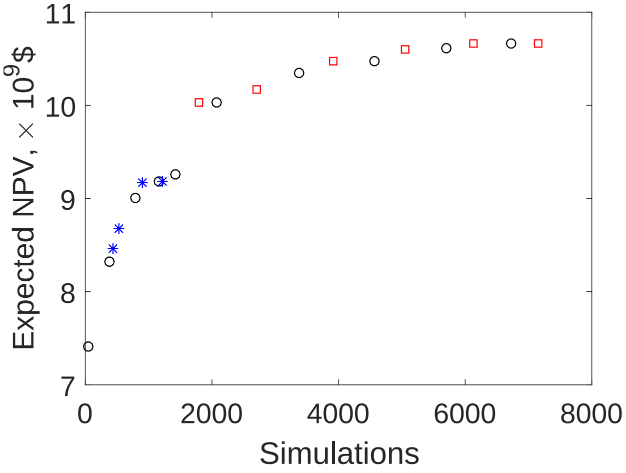
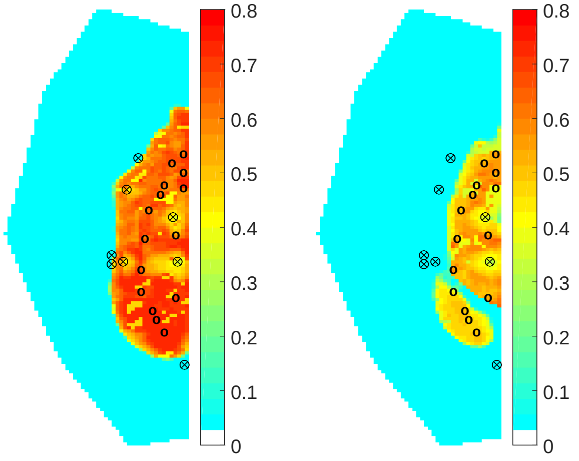


Figure 4.36: Expected NPV of the full ensemble versus number of simulations obtained for Robust-3, robust optimization of the Brugge model. o represents an update by GA, ∗ represents an update by StoSAG, and □ represents an update by GPS.

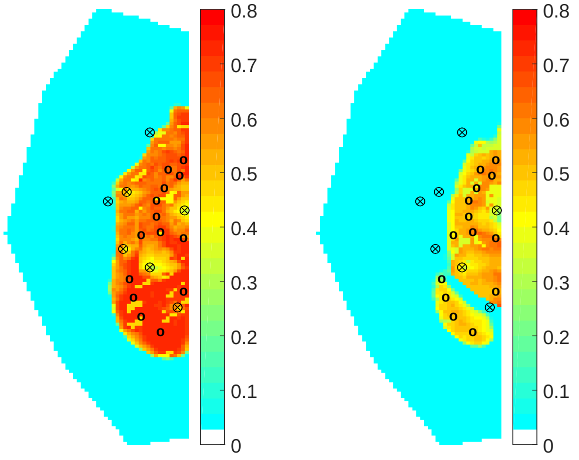
The expected NPV's of the full ensemble and the subset ensemble obtained with optimal solutions, the optimal number of injectors and producers along with the simulation costs to obtain convergence are summarized in Table 4.10. Note that the maximum number of simulation runs is set to be 8,000 and seed 1 of Robust-1 didn't reach convergence before the algorithm termination. From Table 4.10, based on the average performance of three stochastic runs, (i) Robust-1 costs the least number of simulation runs because the runs



(a) Layer 1

(b) Layer 9

Figure 4.37: Optimal types and locations of drilled wells obtained for Robust-1, plotted on the oil saturation field at the top layer of realization #14 after 20 years of production, robust optimization of the Brugge model. ⊗ represents an injector and o represents a producer.

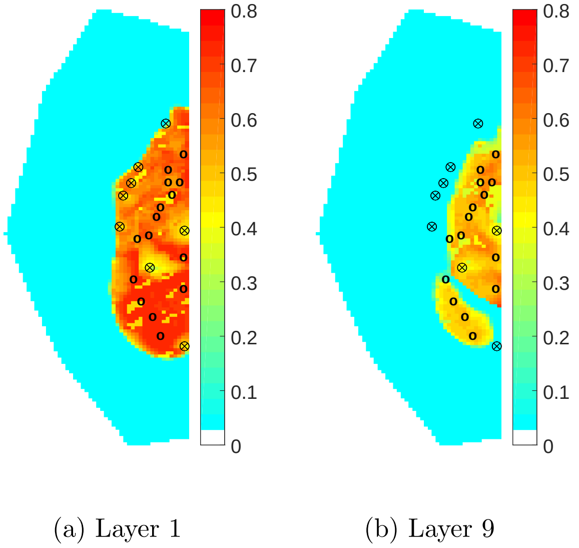


(a) Layer 1

(b) Layer 9

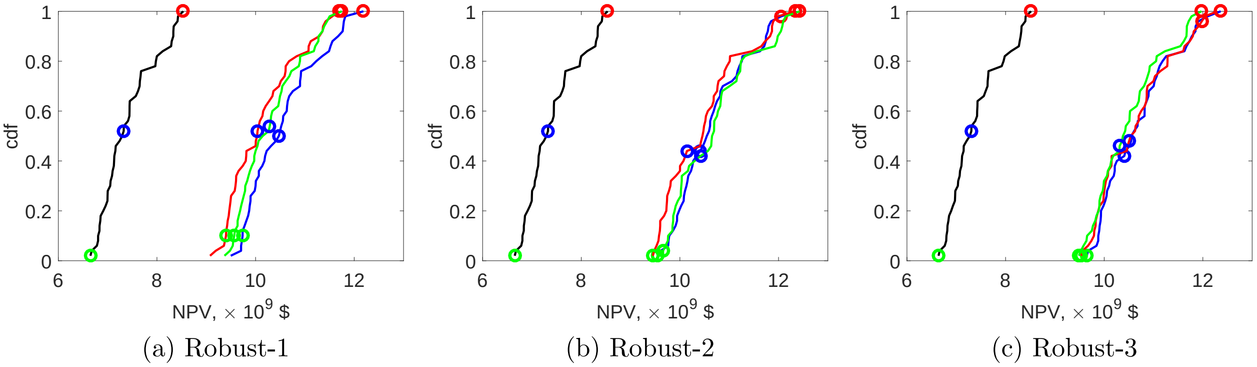
Figure 4.38: Optimal types and locations of drilled wells obtained for Robust-2, plotted on the oil saturation field at the top layer of realization #14 after 20 years of production, robust optimization of the Brugge model. ⊗ represents an injector and o represents a producer.

with seed 2 and seed 3 converged at local solutions which are far from the global solution; (ii) as expected, Robust-2 and Robust-3 give almost the same optimal expected NPV's of the full ensemble which is higher than the value of Robust-1, but Robust-3 requires 700



(a) Layer 1 (b) Layer 9

Figure 4.39: Optimal types and locations of drilled wells obtained for Robust-3, plotted on the oil saturation field at the top layer of realization #14 after 20 years of production, robust optimization of the Brugge model. ⊗ represents an injector and o represents a producer.



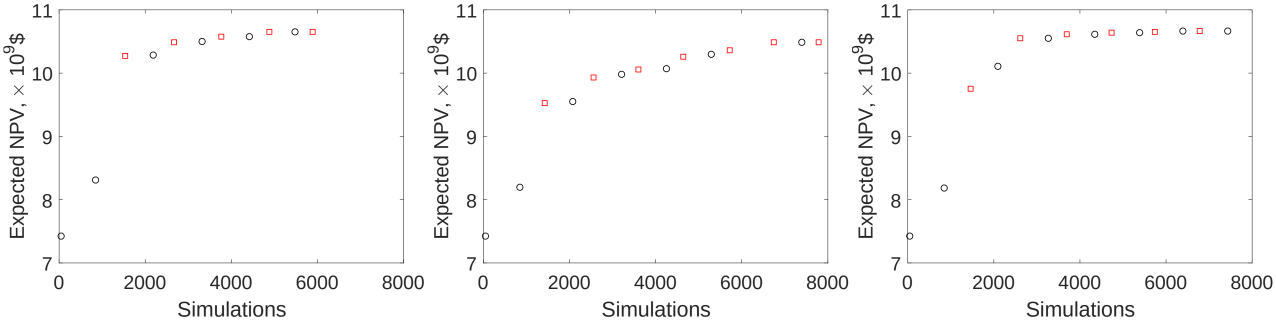
(a) Robust-1 (b) Robust-2 (c) Robust-3

Figure 4.40: Cdf's of NPV's obtained with the reference scenario (black curve) and optimal well placement for the three robust optimization procedures with GA and GPS search steps, robust optimization of Brugge model. The green, blue and red circles respectively represent the P2, P50 and P100 of the cdf of NPV obtained with the reference scenario.

simulation runs less than Robust-2; (iii) the expected NPV's of the subset ensemble are generally higher than those of the full ensemble at the optimal solutions since the former is the objective that we actually maximize, and the difference between the former and the latter is almost the same for Robust-1, Robust-2 and Robust-3; (iv) the number of wells obtained with the optimal solutions whose expected NPV's are larger than $\$10.50 \times 10^9$, are quite similar, i.e., 8 to 9 injectors and 13 to 19 producers. Figs. 4.41 and 4.42 respectively show the expected NPV of the full ensemble versus number of simulations obtained with Robust-2 and Robust-3. Compared to Figs. 4.35 and 4.36, the convergence speed of Robust-2 and Robust-3 without the StoSAG search step have been greatly improved, i.e., saving about 1,000 simulations per run.

Table 4.10: Summary of the optimal solutions and computational cost obtained with the three robust optimization procedures using the iterative sequential method with GA and GPS search steps, robust optimization of Brugge model.

Seeds	Robust-1					Robust-2					Robust-3				
	Sim.	$ × 10^9		N_{Inj}	N_{Pro}	Sim.	$ × 10^9		N_{Inj}	N_{Pro}	Sim.	$ × 10^9		N_{Inj}	N_{Pro}
	#	$J_E^{\Omega_f}$	$J_E^{\Omega_s}$			#	$J_E^{\Omega_f}$	$J_E^{\Omega_s}$			#	$J_E^{\Omega_f}$	$J_E^{\Omega_s}$		
Seed 1	8000	10.55	10.80	9	13	5894	10.66	10.83	9	19	5054	10.68	10.87	7	14
Seed 2	5345	10.14	10.36	9	11	7795	10.49	10.76	11	14	6247	10.63	10.82	9	17
Seed 3	4796	10.31	10.52	7	13	7429	10.66	10.89	7	18	7227	10.50	10.63	10	12
Average	6047	10.33	10.56	8	12	6908	10.60	10.79	9	17	6176	10.60	10.77	9	14



(a) Seed 1

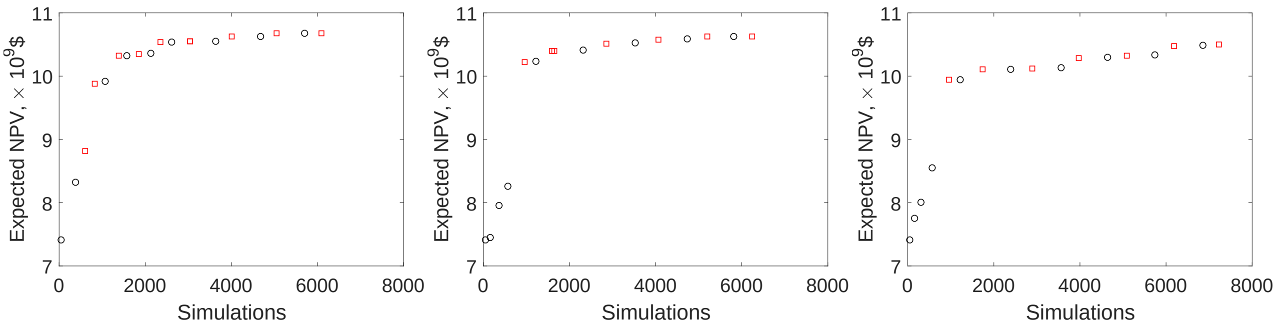
(b) Seed 2

(c) Seed 3

Figure 4.41: The expected NPV of the full ensemble versus number of simulation runs obtained for Robust-2 using the iterative sequential method with GA and GPS steps, robust optimization of Brugge model. o's and □'s respectively represent updates by GA and GPS.

4.5 Summary and Discussion

In this chapter, we considered maximizing the NPV by optimizing the well status, types, locations given a maximum number of wells authorized to drill. The optimization problem considers both deterministic reservoir models and stochastic reservoir models whose uncertainty is represented by a large number of realizations of the reservoir properties. For deterministic optimization, three algorithms are designed to combine the global search ability of GA with the local search ability of StoSAG and GPS methods, including an iterative



(a) Seed 1 (b) Seed 2 (c) Seed 3

Figure 4.42: The expected NPV of the full ensemble versus number of simulation runs obtained for Robust-3 using the iterative sequential method with GA and GPS steps, robust optimization of Brugge model. o's and □'s respectively represent updates by GA and GPS.

sequential procedure and two simultaneous procedures followed by StoSAG and GPS search steps, one using mixed-encoded and one using the Genocop III method. Numerical results on the PUNQ model and the Brugge model show that both the iterative sequential method and the mixed-encoded GA outperform the Genocop III method and that the iterative sequential method has the fastest convergence speed. For the optimization under geological uncertainty (also referred to as robust optimization), we developed an efficient subset realization selection procedure and coupled it with the iterative sequential method. Numerical results on the PUNQ model and the Brugge model show that with the adaptive procedure, we can achieve robust optimization within a few thousand simulation runs, given a large ensemble to characterize the geological uncertainty. Besides, we find it desirable to exclude StoSAG from the iterative sequential method for robust optimization.

CHAPTER 5

WELL PLACEMENT OPTIMIZATION WITH COMPLICATED WELL TRAJECTORIES

In this chapter, we consider well placement optimization of complicated well trajectories, e.g. slanted wells and multi-segmented wells, given the well number, their types and controls. Corresponding parameterizations for the well trajectories are firstly proposed and applied to deterministic optimization using a hybrid of StoSAG and GPS. Afterwards, bi-objective well placement optimization under geological uncertainty is investigated where the two objectives are to maximize the expected value of the life-cycle net-present-value (NPV) and to maximize the minimum NPV of the set of realizations representing the geological uncertainty, using an efficient implementation of the lexicographic method.

5.1 Well Trajectory Parameterization

5.1.1 Slanted well trajectory parameterization

Here, a slanted well trajectory refers to the case where the center-line of a well is a straight line in the 3D space. Emerick et al. [21] proposed to use the (i, j, k) gridblock indices of the heel point and toe point of a well to characterize the trajectory of a slanted well . We propose to use the (x, y, z) coordinates of the two end points of a well inside the reservoir, which is a slightly more general parameterization. We consider a procedure to obtain the path of well iw given the vector

$$\boldsymbol{X}_{iw} = [x_0^{iw}, y_0^{iw}, z_0^{iw}, x_1^{iw}, y_1^{iw}, z_1^{iw}], \tag{5.1}$$

where, $(x_0^{iw}, y_0^{iw}, z_0^{iw})$ and $(x_1^{iw}, y_1^{iw}, z_1^{iw})$ are the coordinates of the two end points of well

iw; see Fig. 5.1(a). The entry point $(x_0^{iw}, y_0^{iw}, z_0^{iw})$ and the exit point $(x_1^{iw}, y_1^{iw}, z_1^{iw})$ are required to fall in the active gridblocks, Ω_{active}. Two types of the most commonly considered constraints are the minimum well spacing constraints where the distance between any pair of well trajectories, $d(\boldsymbol{X}_{iw}, \boldsymbol{X}_{jw})$ should be greater than a pre-specified value $R_{\min}$ and the maximum/minimum well length constraints, these two constraints are given by

$$d(\boldsymbol{X}_{iw}, \boldsymbol{X}_{jw}) > R_{\min} \qquad iw, jw = 1, 2, \cdots, N_w, iw \neq jw, \tag{5.2a}$$

$$l_{iw}^{\mathrm{low}} \leq l_{iw} \leq l_{iw}^{\mathrm{up}} \qquad iw = 1, 2, \cdots, N_w, \tag{5.2b}$$

where, $d(\boldsymbol{X}_{iw}, \boldsymbol{X}_{jw})$ is the Euclidean distance between the iwth and jwth wells (see Appendix A for the detailed calculation), $R_{\min}$ is the minimum allowable inter-well distance, l_{iw} is the length of the iwth well and is calculated as $l_{iw} = \sqrt{(x_1^{iw} - x_0^{iw})^2 + (y_1^{iw} - y_0^{iw})^2 + (z_1^{iw} - z_0^{iw})^2}$, l_{iw}^{low} and l_{iw}^{up} correspond to the minimum and maximum allowable well lengths for the iwth well respectively, and N_w is the number of wells. There are $\frac{N_w(N_w-1)}{2}$ well spacing constraints and $2N_w$ well length constraints in total. For the parameterization given in Eq. 5.1, both constraints in Eq. 5.2 are nonlinear constraints on the design variables, i.e., the evaluations of $d(\boldsymbol{X}_{iw}, \boldsymbol{X}_{jw})$ and l_{iw} do not require simulation runs.

In order to evaluate the objective function, we simulate the reservoir dynamic performance using the commercial simulator, CMG, where deviated perforations can be specified by the (x, y, z) coordinates of their entry and exit points running through each perforated gridblock. With this information, CMG is able to calculate an accurate deviated well productivity/injectivity index using a Peaceman type well model ([68, 69, 70]). Thus, a routine which identifies the grid cells crossed by each well and calculates the Cartesian coordinates of the entry point and exit point of each perforation of a well in its perforated cell is implemented, see Appendix C. In this dissertation, this routine is only developed for reservoirs with (transformed) Cartesian grids.

Denote the full set of the variable vector for vertical well placement optimization as $\boldsymbol{x} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_{N_w}]$, where the dimension of $\boldsymbol{x}$ is $N_x = 6N_w$. The net present value

(NPV) of production is defined as

$$J(\boldsymbol{m},\boldsymbol{x}) = \sum_{n=1}^{N_t}\left\{\frac{\Delta t^n}{(1+b)^{\frac{t^n}{365}}}\left[\sum_{j=1}^{N_P}\left(r_o^n \cdot \overline{q}_{o,j}^n - c_w^n \cdot \overline{q}_{w,j}^n - c_g^n \cdot \overline{q}_{g,j}^n\right) - \sum_{k=1}^{N_I}\left(c_{wi}^n \cdot \overline{q}_{wi,k}^n + c_{gi}^n \cdot \overline{q}_{gi,k}^n\right)\right]\right\} - \sum_{iw=1}^{N_w} C_w(\boldsymbol{X}_{iw}),$$

where n, N_t, Δt^n, N_P, N_I, r_o^n, c_w^n, c_g^n, c_{wi}^n, c_{gi}^n, $\overline{q}_{o,j}^n$, $\overline{q}_{w,j}^n$, $\overline{q}_{g,j}^n$, $\overline{q}_{wi,k}^n$, $\overline{q}_{gi,k}^n$ denote the same terms as in Section 2.1, and $C_w(\boldsymbol{X}_{iw})$ denotes the drilling cost for the iwth well which depends on the trajectory variables defining this well, $\boldsymbol{X}_{iw}$. Given a vector of the reservoir model parameter, $\boldsymbol{m}$, the trajectory optimization problem of vertical wells considered in this chapter is stated as follows,

$$\max_{\boldsymbol{x} \in R^{N_x}} \quad J(\boldsymbol{m},\boldsymbol{x}) \tag{5.3a}$$

$$s.t. \quad (x_0^{iw}, y_0^{iw}, z_0^{iw}) \in \Omega_{active}, \qquad iw = 1, 2, \cdots, N_w, \tag{5.3b}$$

$$(x_1^{iw}, y_1^{iw}, z_1^{iw}) \in \Omega_{active}, \qquad iw = 1, 2, \cdots, N_w, \tag{5.3c}$$

$$d(\boldsymbol{X}_{iw}, \boldsymbol{X}_{jw}) > R_{\min} \qquad iw, jw = 1, 2, \cdots, N_w, iw \neq jw, \tag{5.3d}$$

$$l_{iw}^{\text{low}} \leq l_{iw} \qquad iw = 1, 2, \cdots, N_w, \tag{5.3e}$$

$$l_{iw} \leq l_{iw}^{\text{up}} \qquad iw = 1, 2, \cdots, N_w. \tag{5.3f}$$

5.1.2 Multi-segmented well trajectory parameterization

For a multi-segmented well, using the (x, y, z) coordinates of the end points of each segment as design variables may destroy the smoothness of the well trajectory after the variable updates in the optimization process. Hence, we propose a different parameterization (see Fig. 5.1(b) and Eq. 5.4 below) to transform the drillability (smoothness) constraint to the bound constraints on optimization variables which then can be effectively enforced by truncation. Specifically, the vector of parameters for well iw is given by,

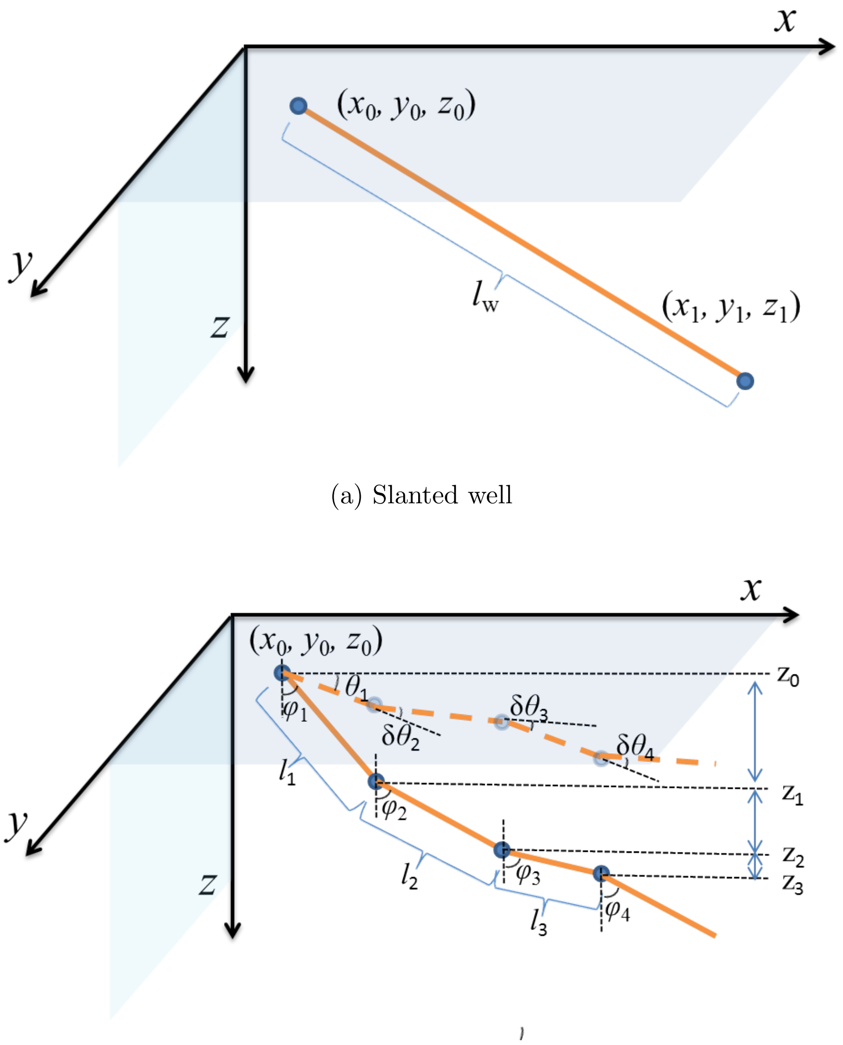
$$\boldsymbol{X}_{iw} = [x_0^{iw}, y_0^{iw}, z_0^{iw}, z_1^{iw}, l_1^{iw}, \theta_1^{iw}, z_2^{iw}, l_2^{iw}, \delta\theta_2^{iw}, \cdots, z_{N_{seg}}^{iw}, l_{N_{seg}}^{iw}, \delta\theta_{N_{seg}}^{iw}], \tag{5.4}$$

where, $(x_0^{iw}, y_0^{iw}, z_0^{iw})$ are the coordinates of the entry point of the iwth well into the reservoir

($(x_0^{iw}, y_0^{iw}, z_0^{iw})$ is required to fall into an active gridblock of the reservoir), see Fig. 5.1(b); θ_1^{iw} is the azimuth angle of the first segment (the angle between the x-axis and the projection of the first segment onto the x-y plane); $\delta\theta_i^{iw},\ i = 2, 3, \cdots, N_{seg}$ represents the change in azimuth angle of the ith segment compared to the $(i-1)$th segment so the azimuth angle of each segment can be calculated recursively using $\theta_i^{iw} = \theta_{i-1}^{iw} + \delta\theta_i^{iw}, i = 2, 3, \cdots, N_{seg}$; $z_i^{iw}, i = 1, \cdots, N_{seg}$ represents the z coordinate of the end point of the ith segment of the iwth well; $l_i^{iw}, i = 1, \cdots, N_{seg}$ represents the length of the ith segment. The (x, y) coordinates of the end point of the ith segment of the iwth well can be calculated as $x_i^{iw} = x_{i-1}^{iw} + \sqrt{(l_i^{iw})^2 - \left(z_i^{iw} - z_{i-1}^{iw}\right)^2} * \cos(\theta_i^{iw})$ and $y_i^{iw} = y_{i-1}^{iw} + \sqrt{(l_i^{iw})^2 - \left(z_i^{iw} - z_{i-1}^{iw}\right)^2} * \sin(\theta_i^{iw})$. Similar to the azimuth angle concept, we define φ_1^{iw} as the inclination angle of the first segment (the angle between z-axis and the first segment of the well trajectory), and define $\delta\varphi_i^{iw},\ i = 2, 3, \cdots, N_{seg}$ as the change in inclination angle of the ith segment compared to the $(i-1)$th segment. Note that φ_1^{iw} and $\delta\varphi_i^{iw}$ are not in the well trajectory parameter $\boldsymbol{X}_{iw}$.

For a multi-segmented well, there may be two segments of perforations in the gridblock containing the end point of a well segment (except the last well segment). Since in CMG as well as other commercial simulators, one can only define a perforation with one line segment, thus a slanted pseudo-perforation is required. We propose to adjust the completion factor of the pseudo-perforation based on CMG [55] so that equal injectivity/productivity can be obtained. The calculation of the coordinates of the entry and exit point and the completion factor of this pseudo-perforation is presented in Appendix C.2.

In fact, the multi-segmented well trajectory depicted in Fig. 5.1(b) is not drillable using current directional drilling techniques, and a good-quality wellbore should be smooth in practice. Hence, when applying the optimal multi-segmented well trajectory, we smooth the ith and the $(i+1)$th well segments as a circular arc located in the plane of these two segments. Fig. 5.2 illustrates the smoothing of two consecutive well segments for a multi-segmented well where points A and C denote the two end points of the ith well segment, points C and E denote the two end points of the $(i+1)$th segment and points B and D denote the points on segments AC and CE such that $|BC| = |CD| = \min(\frac{|AC|}{2}, \frac{|CE|}{2})$ and O



(a) Slanted well

(b) Multi-segmented well

Figure 5.1: Illustration of well trajectory parameterization for a slanted and a multi-segmented well.

is the center of the arc $\widehat{BD}$. The calculation of the segment end points and the smoothing of the trajectory of well iw are given in Appendix B.

To be drillable, the dogleg severity (DLS) of the well trajectory in Fig. 5.2 has to stay below a maximum value. DLS ([58]) is the dogleg angle (defined as the overall change in inclination and azimuth of a borehole from the ith segment to the $(i+1)$th segment) per 100 feet of the measured depth. Following the notation in Fig. 5.2, DLS is given by

$$DLS_i^{iw} = \frac{100\beta_i^{iw}}{\widehat{BD}} = \frac{100}{OB}\frac{180}{\pi}, \tag{5.5}$$

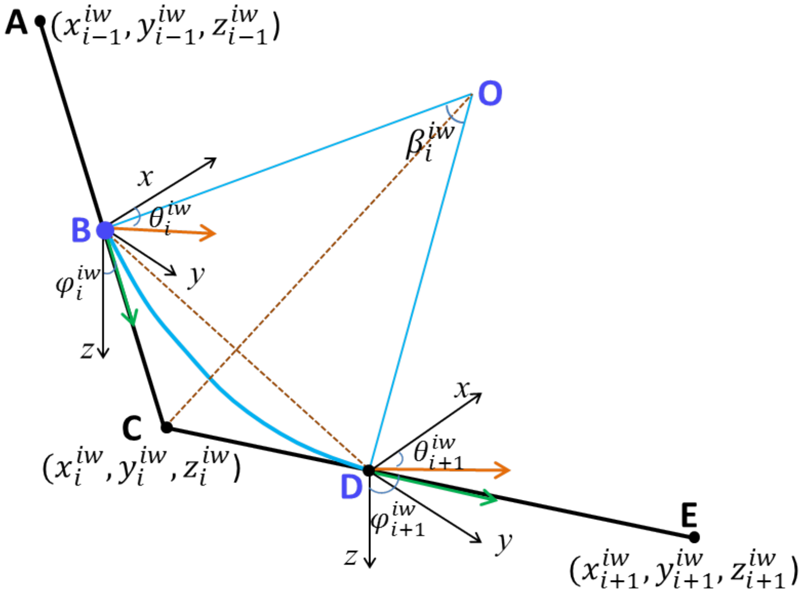

A
$(x_{i-1}^{iw}, y_{i-1}^{iw}, z_{i-1}^{iw})$
O
β_i^{iw}
x
B
θ_i^{iw}
y
φ_i^{iw}
z
C
$(x_i^{iw}, y_i^{iw}, z_i^{iw})$
x
θ_{i+1}^{iw}
D
φ_{i+1}^{iw}
y
z
E
$(x_{i+1}^{iw}, y_{i+1}^{iw}, z_{i+1}^{iw})$

Figure 5.2: Illustration of the smoothed trajectory of two consecutive well segments for a multi-segmented well.

where,

$$OB = \frac{BC}{\tan\frac{\beta_i^{iw}}{2}}, \tag{5.6}$$

and

$$\beta_i^{iw} = 2\text{arcsin}\sqrt{\left(\text{sin}\frac{\delta\varphi_i^{iw}}{2}\right)^2 + \text{sin}\varphi_i^{iw}\text{sin}(\varphi_i^{iw} + \delta\varphi_i^{iw})\left(\text{sin}\frac{\delta\theta_i^{iw}}{2}\right)^2}. \tag{5.7}$$

Substituting Eq. 5.6 into Eq. 5.5, we obtain

$$DLS_i^{iw} = \frac{100}{BC}\frac{180}{\pi}\tan(\frac{\beta_i^{iw}}{2}). \tag{5.8}$$

From Eq. 5.8, the dogleg severity is low if β_i^{iw} is low or BC (i.e., $\min(\frac{l_i^{iw}}{2}, \frac{l_{i+1}^{iw}}{2})$) is large. Since the thickness of a reservoir is usually much smaller than its width or length and the perforated length of a directional well is usually quite large, the change in inclination $\delta\varphi_i^{iw}$ is usually kept at a small value if a well trajectory is bounded within the a reservoir (or a specific target zone). To avoid handling the nonlinear constraints that the dogleg severity has to be below a maximum value, we put bound constraints on the changes in the azimuth angle $\delta\theta_i^{iw}$ and the segment length l_i^{iw}. The lower and upper bounds of $\delta\theta_i^{iw}$ and l_i^{iw} are denoted as $\delta\theta_i^{\mathrm{low},iw}$ and $\delta\theta_i^{\mathrm{up},iw}$, and, $l^{\mathrm{low},iw}$ and $l^{\mathrm{up},iw}$, respectively. Thus the total set of bound constraints on optimization variables are given by

$$\begin{aligned}
(x_0^{iw}, y_0^{iw}, z_0^{iw}) &\in \Omega_{active}, \quad iw = 1, 2, \cdots, N_w, \\
z_i^{\mathrm{low},iw} < z_i^{iw} &< z_i^{\mathrm{up},iw}, \quad iw = 1, 2, \cdots, N_w, \qquad i = 1, 2, \cdots, N_{seg}, \\
l_i^{\mathrm{low},iw} < l_i^{iw} &< l_i^{\mathrm{up},iw}, \quad iw = 1, 2, \cdots, N_w, \qquad i = 1, 2, \cdots, N_{seg}, \\
\theta_1^{\mathrm{low},iw} < \theta_1^{iw} &< \theta_1^{\mathrm{up},iw}, \quad iw = 1, 2, \cdots, N_w, \\
\delta\theta_i^{\mathrm{low},iw} < \delta\theta_i^{iw} &< \delta\theta_i^{\mathrm{up},iw}, \quad iw = 2, \cdots, N_w, \qquad i = 2, 3, \cdots, N_{seg}.
\end{aligned} \tag{5.9}$$

Note that even though the bound constraints in Eq. 5.9 can be efficiently enforced by truncation, they only roughly enforce the drillability constraint. If the nonlinear constraints given in Eq. 5.2 are considered in the multi-segmented well optimization, then $d(\boldsymbol{X}_{iw}, \boldsymbol{X}_{jw})$ is defined as the minimum distance between any pair of well segments of the iwth and jwth wells ($iw \neq jw$).

Denote the full set of the variable vector for multi-segmented well placement as $\boldsymbol{x} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_{N_w}]$, where the dimension of $\boldsymbol{x}$ is $N_x = \sum_{iw=1}^{N_w} 3(N_{seg}^{iw} + 1)$. Given a vector of reservoir model parameter, $\boldsymbol{m}$, the well trajectory optimization of multi-segmented wells considered in this chapter is stated as follows,

$$\max_{\boldsymbol{x} \in R^{N_x}} \quad J(\boldsymbol{m}, \boldsymbol{x}) \tag{5.10a}$$

$$s.t. \quad (x_0^{iw}, y_0^{iw}, z_0^{iw}) \in \Omega_{active}, \qquad iw = 1, 2, \cdots, N_w, \tag{5.10b}$$

$$z_i^{\mathrm{low},iw} < z_i^{iw} < z_i^{\mathrm{up},iw}, \qquad iw = 1, 2, \cdots, N_w, \quad i = 1, 2, \cdots, N_{seg}, \tag{5.10c}$$

$$l_i^{\mathrm{low},iw} < l_i^{iw} < l_i^{\mathrm{up},iw}, \qquad iw = 1, 2, \cdots, N_w, \quad i = 1, 2, \cdots, N_{seg}, \tag{5.10d}$$

$$\theta_1^{\mathrm{low},iw} < \theta_1^{iw} < \theta_1^{\mathrm{up},iw}, \qquad iw = 1, 2, \cdots, N_w, \tag{5.10e}$$

$$\delta\theta_i^{\mathrm{low},iw} < \delta\theta_i^{iw} < \delta\theta_i^{\mathrm{up},iw}, \qquad iw = 2, \cdots, N_w, \quad i = 2, 3, \cdots, N_{seg} \tag{5.10f}$$

$$d(\boldsymbol{X}_{iw}, \boldsymbol{X}_{jw}) > R_{\min} \qquad iw, jw = 1, 2, \cdots, N_w, \quad iw \neq jw. \tag{5.10g}$$

5.2 Constrained Optimization using Augmented Lagrangian Method

The problems described in Eqs. 5.3 and 5.10 are maximization problems under both bound constraints and nonlinear constraints on the design variables. For the vertical well placement optimization, Eqs. 5.3b to 5.3c are bound constraints and Eqs. 5.3d to 5.3f are nonlinear constraints. For the multi-segmented well trajectory optimization, Eqs. 5.10b to 5.10f are bound constraints and Eq. 5.10g is a nonlinear constraint. Denote the upper and lower bounds of $\boldsymbol{x}$ respectively as $\boldsymbol{x}^{\mathrm{up}}$ and $\boldsymbol{x}^{\mathrm{low}}$, then the bound constraints can be re-written in a general form that $x_i^{\mathrm{low}} \leq x_i \leq x_i^{\mathrm{up}}, i = 1, 2, \cdots, N_x$ where x_i^{low}, x_i and x_i^{up}, respectively denote the ith component of the vectors $\boldsymbol{x}^{\mathrm{low}}$, $\boldsymbol{x}$ and $\boldsymbol{x}^{\mathrm{up}}$. In this chapter, the bound constraints are enforced by truncating the design variables beyond the bounds to their nearest bounds while the nonlinear constraints are handled using the augmented Lagrangian method (ALM).

As the nonlinear constraints in Eqs. 5.3d, 5.3e, 5.3f and 5.10g may be of different scales, and C. Chen [10] observed that the convergence rate of ALM can be slowed appreciably by poor scaling of the constraints based on numerical examples, we scale these nonlinear constraints so that the scaled constraints are more or less on the same magnitude. The scaled nonlinear constraints in Eqs. 5.3f and 5.10g are given below,

$$c\left(\boldsymbol{x}\right) = \frac{R_{\min} - d(\boldsymbol{X}_{iw}, \boldsymbol{X}_{jw})}{R_{\min}} \leq 0, \quad iw, jw = 1, 2, \cdots, N_w, iw \neq jw. \tag{5.11}$$

The scaled constraints of Eqs. 5.3d and 5.3e are given by

$$\begin{aligned} c\left(\boldsymbol{x}\right) &= \frac{l_{iw}^{\mathrm{low}} - l_i^{iw}}{l_{iw}^{\mathrm{low}}} \leq 0, \quad iw = 1, 2, \cdots, N_w, \\ c\left(\boldsymbol{x}\right) &= \frac{l_i^{iw} - l_{iw}^{\mathrm{up}}}{l_{iw}^{\mathrm{up}}} \leq 0, \quad iw = 1, 2, \cdots, N_w. \end{aligned} \tag{5.12}$$

The components of the design variable vector $\boldsymbol{x}$ are also of different scales, e.g., for a slanted well, z_0^{iw} and z_1^{iw} may be of different scales with x_0^{iw}, x_1^{iw}, y_0^{iw} and y_1^{iw}; for a multi-segmented well, $\delta\theta_i^{iw}$ are of completely different scales with x_0^{iw}, y_0^{iw} and z_0^{iw}. Similar changes to each component of $\boldsymbol{x}$ may produce different magnitude of changes in the NPV function $J(\boldsymbol{m}, \boldsymbol{x})$. Hence, we introduce a new variable vector, $\boldsymbol{v}$, which has similar magnitude in all its components and then solve the optimization problem in terms of $\boldsymbol{v}$. The ith component of $\boldsymbol{v}$ is given by

$$v_i = \frac{x_i - x_i^{\text{low}}}{x_i^{\text{up}} - x_i^{\text{low}}}, \quad \text{for } i = 1, 2, \cdots, N_x, \tag{5.13}$$

where $0 \leq v_i \leq 1$. Given v_i, the ith component of $\boldsymbol{x}$ can be obtained by

$$x_i = x_i^{\text{low}} + v_i(x_i^{\text{up}} - x_i^{\text{low}}), \quad \text{for } i = 1, 2, \cdots, N_x. \tag{5.14}$$

Denote the total number of nonlinear constraints as N_{nc} and the nonlinear constraints as $c_i(\boldsymbol{v}) \leq 0, i = 1, 2, \cdots, N_{nc}$, the problems defined in Eqs. 5.3 and 5.10 can be written in a general form of

$$\begin{aligned} \max_{\boldsymbol{v} \in R^{N_x}} \quad & J(\boldsymbol{m}, \boldsymbol{v}) \\ s.t. \quad & v_i \in [0, 1], \quad i = 1, 2, \cdots, N_x, \\ & c_i(\boldsymbol{v}) \leq 0, \quad i = 1, 2, \cdots, N_{\text{nc}}. \end{aligned} \tag{5.15}$$

With ALM, the solution of a constrained optimization problem is obtained by solving a series of bound constrained optimization sub-problems with a penalized objective function defined by

$$L_A(\boldsymbol{m}, \boldsymbol{v}, \lambda_i^{k_L}, \mu^{k_L}) = J(\boldsymbol{m}, \boldsymbol{v}) - \sum_{i=1}^{N_{\text{nc}}} \left[\lambda_i^{k_L}(c_i(\boldsymbol{v}) + s_i) + \frac{1}{2\mu^{k_L}}(c_i(\boldsymbol{v}) + s_i)^2 \right], \tag{5.16}$$

where $\lambda_i^{k_L}$ is the Lagrangian multiplier within the k_Lth inner loop for the nonlinear constraint $c_i(\boldsymbol{v})$, μ^{k_L} is the penalty parameter within the k_Lth inner loop iterates where all c_i's share the same μ^{k_L}, and s_i is a positive slack variable used to convert the ith inequality constraint $(c_i(\boldsymbol{v}) \leq 0)$ into an equality constraint. The Lagrangian function of Eq. 5.16 is a concave quadratic function of the slack variable s_i. Denote the term in the bracket of Eq. 5.16 as

ψ_{c_i}, then

$$L_A(\boldsymbol{m}, \boldsymbol{v}, \lambda_i^{k_L}, \mu^{k_L}) = J(\boldsymbol{m}, \boldsymbol{v}) - \sum_{i=1}^{N_{\mathrm{nc}}} \psi_{c_i}. \tag{5.17}$$

Following Nocedal and Wright [62], we eliminate the slack variables for the inequality constraints, s_i, by solving s_i from $\frac{\partial \psi_{c_i}}{\partial s_i} = 0$ and substitute s_i back to ψ_{c_i} so that

$$\psi_{c_i}(\boldsymbol{v}, \lambda_i^{k_L}, \mu^{k_L}) = \begin{cases} -\frac{\mu^{k_L}}{2}\left(\lambda_i^{k_L}\right)^2 & c_i < -\mu^{k_L}\lambda_i^{k_L} \\ \lambda_i^{k_L} c_i + \frac{c_i^2}{2\mu^{k_L}} & c_i \geq -\mu^{k_L}\lambda_i^{k_L} \end{cases}. \tag{5.18}$$

Eq. 5.18 indicates that in ALM, if $c_i(\boldsymbol{v})$ is less than the switch-point, $-\mu^{k_L}\lambda_i^{k_L}$, ψ_{c_i} is a constant that has no effect on the optimization; otherwise, ψ_{c_i} is a quadratic function of $c_i(\boldsymbol{v})$.

The optimization process using ALM consists of two coupled loops. In the inner loop, an iterative optimization procedure is used to maximize $L_A(\boldsymbol{m}, \boldsymbol{v}, \lambda_i^{k_L}, \mu^{k_L})$ given in Eq. 5.17 with the value of the Lagrangian multiplier $\lambda_i^{k_L}$ and penalty parameter μ^{k_L} fixed. In the outer loop of the optimization process, the value of $\lambda_i^{k_L}$ and μ^{k_L} are updated in order to minimize the violation of constraints. When the constraint violation is small, $\lambda_i^{k_L}$'s are updated and μ^{k_L} is unchanged; otherwise, $\lambda_i^{k_L}$'s are kept the same and μ^{k_L} is updated (decreased), see Chen and Reynolds [11]. In this chapter, following [11], the initial Lagrangian multipliers are estimated as

$$\lambda_i^0 = \max\left\{0, \frac{c_i(\boldsymbol{v}_0)}{\mu^0}\right\}, i = 1, 2, \cdots, N_{nc}, \tag{5.19}$$

where $\boldsymbol{v}_0$ is the initial guess of the scaled design variables. It is very important to choose a reasonable value for the initial penalty parameter μ^0. From Eq. 5.16, if μ^0 takes a large value which imposes negligible penalty to the NPV, then the first few inner loops, perform like unconstrained optimization which may lead to a great constraint violation. When this occurs, many extra simulations are needed to eliminate the constraint violation in the following inner loops so that the updated μ^{k_L} is small enough. However, if μ^0 takes a very small value which corresponds to a large penalty term, we may not be able to increase the NPV once the

the current estimate is infeasible or close to the boundaries of the feasible region, since the objective function will be dominated by the penalty term. In this work, we chose μ^0 by ensuring that the average penalty term is on the order of a reasonable portion of the initial NPV value.

$$\gamma J(\boldsymbol{v}_0) = \frac{N_{nc}}{2\mu^0}\hat{c}, \tag{5.20}$$

where, γ is the percentage of the average penalty term over the initial NPV value (we use $\gamma = 10\%$), $\hat{c}$ is the average value of the constraints c_i's (we use $\hat{c} = 1$). From Eq. 5.20, we can obtain

$$\mu^0 = \frac{N_{nc}}{2\gamma J(\boldsymbol{v}_0)}\hat{c}. \tag{5.21}$$

Please see Algorithm 12 for a detailed implementation of the ALM method.

Due to the gradient estimation error in StoSAG and the fact that the objective function surface of the well placement optimization is rough, ALM is applied twice. In the first application of ALM, the inner loop optimization represented by Eq. 5.17 is solved using StoSAG. Then starting from the optimal estimates of $\boldsymbol{v}$, λ_i's and μ obtained, ALM is applied again where the inner loop optimization is solved using GPS. Both applications of ALM are terminated either at convergence or when the maximum number of simulation runs exceeds a specified number. Algorithms 2 and 3 in Chapter 2, respectively, present the procedures of StoSAG and GPS algorithms. The convergence criteria for StoSAG is set as five consecutive failures in improving $L_A(\boldsymbol{m}, \boldsymbol{v}, \lambda^{k_L}, \mu^{k_L})$ along the StoSAG search direction using backtracking where the maximum number of step size cuts is set equal to 5. However, the perturbation size and the initial step size are chosen according to the optimization problem and are specified for each numerical example presented later. The convergence criteria for GPS is that the relative change of $L_A(\boldsymbol{m}, \boldsymbol{v}, \lambda^{k_L}, \mu^{k_L})$ is less or equal to 10^{-4} after searching along N_x coordinates directions or a maximum number of simulations is reached. In both StoSAG and GPS, the bound constraints are solved using truncation, i.e., truncating the design variables beyond the bounds to their nearest bounds.

Algorithm 12 Augmented Lagrangian method to solve Eq. 5.15

- Initialization: $k_L = 0$. Choose $\boldsymbol{x}_0, \boldsymbol{v}_0, \lambda_i^0$ and μ^0. Set $\beta = 0.2, \alpha = 0.1, \tau = 0.25, \eta^* = 0.01, \eta^{k_L} = 0.1$.
- Step 1: Find the approximate optimum of the augmented Lagrangian function defined in Eq. 5.17 subject to the bound constraints using StoSAG or GPS, starting from $\boldsymbol{v}_0$. The optimal solution is denoted by $\boldsymbol{v}^*_{k_L}$ in the transformed domain or $\boldsymbol{x}^*_{k_L}$ in the original domain.
- Step 2: Check constraint violation, σ_{cv}. If $\sigma_{cv} < \eta^*$, set $\boldsymbol{s}^* = \boldsymbol{x}^*_{k_L}$ and terminate the optimization algorithm; otherwise go to Step 3.

$$\sigma_{cv} = \sqrt{\sum_{i=1}^{N_{nc}} \left\{ \max\bigl(0, c_i\left(\boldsymbol{v}^*_{k_L}\right)\bigr) \right\}^2}$$

- Step 3: If $\sigma_{cv} \leq \eta^{k_L}$, update the Lagrangian multipliers and tighten the constraint violation tolerance by

  $\lambda_i = \max\{0, \lambda_i + \frac{c_i(\boldsymbol{v}^*_{k_L})}{\mu^{k_L}}\}$ and $\eta^{k_L+1} = \max\{\eta^{k_L} * \min\left\{(\mu^{k_L})^{\beta}, 0.5\right\}, \eta^*\}$,

  and set

  $\mu^{k_L+1} = \mu^{k_L}$, $\boldsymbol{v}_0 = \boldsymbol{v}^*_{k_L}$ and $k_L = k_L + 1$.
- Step 4: If $\sigma_{cv} > \eta^{k_L}$, update the penalty parameter and tighten the constraint violation tolerance by

  $\lambda_i^{k_L+1} = \lambda_i^{k_L}$ and $\eta^{k_L+1} = \max\{\eta^{k_L} * \min\left\{(\mu^{k_L})^{\alpha}, 0.5\right\}, \eta^*\}$,

  and set

  $\mu^{k_L+1} = \tau\mu^{k_L}$, $\boldsymbol{v}_0 = \boldsymbol{v}^*_{k_L}$, $k_L = k_L + 1$.

Where, η^{k_L} and η^* are the constraint violation tolerance at the k_Lth outer loop and that at convergence, respectively.

5.3 Bi-objective Well Placement Optimization

Given an ensemble of realizations which reasonably covers the uncertainty of the model parameters, it is desirable to maximize the average NPV over all realizations and, in order to minimize downside risk, to maximize the minimum NPV. The lexicographical method optimizes the two objectives sequentially according to the order of importance where the most important objective is firstly optimized and then the less significant objective is optimized with the value of previous objective treated as an additional constraint.

In this section, the primary objective is to maximize the expected value of NPV over the set of N_e realizations of the reservoir model parameter chosen to represent the geological uncertainty, $\{\boldsymbol{m}_k\}_{k=1}^{N_e}$, i.e., the optimization problem pertaining to the primary objective is given by

$$\begin{aligned}
\max_{\boldsymbol{v}\in\mathbb{R}^{\mathrm{N}_x}} \quad & J_E(\boldsymbol{v}) = \frac{1}{N_e}\sum_{k=1}^{N_e} J(\boldsymbol{v},\boldsymbol{m}_k) && \text{(5.22a)}\\
\text{s.t.} \quad & 0 \leq \ v_i \ \leq \ 1, \qquad i \ = 1,2,\ldots,N_x && \text{(5.22b)}\\
& c_i(\boldsymbol{v}) \leq 0, \qquad i \ = 1,2,\cdots,N_{nc} && \text{(5.22c)}
\end{aligned}$$

where N_e is the number of realizations representing the geological uncertainty, $\boldsymbol{v}$ is the scaled well trajectory variables, $c_i(\boldsymbol{v})$'s represent the nonlinear constraints, and N_{nc} is the total number of nonlinear constraints. The primary optimization problem defined in Eq. 5.22 is simply the robust optimization under nonlinear constraints which can be solved using StoSAG followed by GPS coupled with ALM. When the number of reservoir models, N_e, is greater or equal to 10, following [24], we use only one perturbation per reservoir model to estimate a StoSAG gradient which requires N_e simulation runs. In GPS, each coordinate search requires at least $2 * N_e$ simulation runs. By solving Eq. 5.22, we obtain the estimate of the optimal (scaled) vector of well trajectories, denoted by $\boldsymbol{v}_E^*$, and the corresponding expected NPV, denoted by J_E^*. To minimize the uncertainty in the NPV's resulted from the geological uncertainties, previous researchers have tried to minimize the standard deviation of the optimal NPV's, or to maximize the minimum NPV. To avoid the situation that minimization of the standard deviation is achieved by decreasing the highest NPV, as observed for fluvial models by [49], we maximize the minimum NPV in the secondary optimization, subject to the constraint that the average NPV should not be smaller than J_E^* or some percentage of J_E^*, i.e., ηJ_E^* where $0.95 \leq \eta \leq 1.0$. Note that the constrained optimization problem starts from $\boldsymbol{v}_E^*$, which is a feasible point.

The second optimization problem is given by

$$\max_{\boldsymbol{v}\in\mathbb{R}^{\mathrm{N}_x}} \quad J_R(\boldsymbol{v}) = \min\{J(\boldsymbol{v},\boldsymbol{m}_k)\}_{k=1}^{N_e} \tag{5.23a}$$

$$\text{s.t.} \quad J_E(\boldsymbol{v}) \geq \eta J_E(\boldsymbol{v}_E^*), \tag{5.23b}$$

$$0 \leq \ v_i \ \leq \ 1, \qquad i \ = 1,2,\ldots,N_x, \tag{5.23c}$$

$$c_i(\boldsymbol{v}) \leq 0, \qquad i \ = 1,2,\cdots,N_{nc} \tag{5.23d}$$

and solved using ALM. To avoid poor scaling of the constraints, Eq. 5.23b is reformulated as

$$c_J = \frac{\eta J_E(\boldsymbol{v}_E^*) - J_E(\boldsymbol{v})}{J_E(\boldsymbol{v}_E^*)} \leq 0. \tag{5.24}$$

The corresponding augmented Lagrangian function is defined similar to Eq. 5.17, but with an extra term. Specifically,

$$L_A(\boldsymbol{v},\lambda_{c_i}^{k_L},\lambda_{c_J}^{k_L},\mu^{k_L}) = J_R(\boldsymbol{v}) - \sum_{i=1}^{N_{nc}} \psi_{c_i}(\boldsymbol{v},\lambda_i^{k_L},\mu^{k_L}) - \psi_{c_J}(\boldsymbol{v},\lambda_{c_J}^{k_L},\mu^{k_L}), \tag{5.25}$$

where

$$\psi_{c_J}(\boldsymbol{v},\lambda^{k_L},\mu^{k_L}) = \begin{cases} -\frac{\mu^{k_L}}{2}\left(\lambda_{c_J}^{k_L}\right)^2 & c_J < -\mu^{k_L}\lambda_{c_J}^{k_L} \\ \lambda_{c_J}^{k_L} c_J + \frac{c_J^2}{2\mu^{k_L}} & c_J \geq -\mu^{k_L}\lambda_{c_J}^{k_L} \end{cases}. \tag{5.26}$$

Here, $\lambda_{c_J}^{k_L}$ is the Lagrangian multiplier associated with the nonlinear constraint $c_J(\boldsymbol{v}) \leq 0$ within the k_Lth inner loop iterates. Following a a similar approach as presented in Eq. 5.19, the initial value of λ_{c_J} is set to 0 since

$$\lambda_{c_J}^0 = \max\left\{0, \frac{c_J(\boldsymbol{v}_E^*)}{\mu^0}\right\} = 0. \tag{5.27}$$

Maximization of the minimum NPV over all realizations, $J_R(\boldsymbol{v})$, is a max-min problem. $J_R(\boldsymbol{v})$ is continuous with respect to the optimization variables $\boldsymbol{v}$, but not differentiable everywhere. When the NPV of the risk realization becomes higher than or equal to another realization, the function $J_R(\boldsymbol{v})$ becomes non-differentiable and the gradient becomes undefined. Liu and Reynolds [50] used the adjoint gradient to solve a bi-objective well control optimization problem. They had to transform the maximization of the expected shortfall, $J_R(\boldsymbol{v})$, to a constrained optimization problem which maximizes a newly introduced parameter α subject to the constraints that NPV for each realization is greater than α. However, with the stochastic approximate gradient coupled with steepest ascent algorithm, the non-differentiability is not an issue.

Denote $\hat{J}_R(\boldsymbol{v}, \lambda_i^{k_L}, \mu^{k_L}) = J_R(\boldsymbol{v}) - \sum_{i=1}^{N_{nc}} \psi_{c_i}(\boldsymbol{v}, \lambda_i^{k_L}, \mu^{k_L})$, Eq. 5.25 can be rewritten as

$$L_A(\boldsymbol{v}, \lambda_{c_i}^{k_L}, \lambda_{c_J}^{k_L}, \mu^{k_L}) = \hat{J}_R(\boldsymbol{v}, \lambda_i^{k_L}, \mu^{k_L}) - \psi_{c_J}(\boldsymbol{v}, \lambda_{c_J}^{k_L}, \mu^{k_L}). \tag{5.28}$$

Denote the realization with the lowest NPV value among the N_e realizations at ℓth iteration as $\boldsymbol{m}_r$, then $\boldsymbol{m}_r$ is given by

$$\boldsymbol{m}_r = \min_k \quad \{J(\boldsymbol{v}^\ell, \boldsymbol{m}_k)\}_{k=1}^{N_e}. \tag{5.29}$$

$\boldsymbol{m}_r$ is not fixed and may change whenever the (scaled) design variable $\boldsymbol{v}$ is updated.

When maximizing Eq. 5.28, a basic simplex search direction $\boldsymbol{d}_{\text{sim}}$ can be obtained in a similar way as presented in Eq. 2.8, i.e.,

$$\boldsymbol{d}_{\text{sim}} \equiv \boldsymbol{d}_{L_A} = (\Delta \boldsymbol{V} \Delta \boldsymbol{V}^T)^{+} \Delta \boldsymbol{V} \boldsymbol{\Delta L}_{\text{A}} \left(\boldsymbol{v}, \lambda_i^{k_L}, \lambda_{c_J}^{k_L}, \mu^{k_L}\right), \tag{5.30}$$

where $\Delta \boldsymbol{V} = [\delta \boldsymbol{v}_1, \delta \boldsymbol{v}_1, \ldots, \delta \boldsymbol{v}_{N_{pert}}]$, and $\boldsymbol{\Delta L}_{\text{A}} = [\delta L_{\text{A},1}, \delta L_{\text{A},2}, \ldots, \delta L_{\text{A},N_{pert}}]^T$. To obtain a reliable stochastic gradient of L_A, five to fifteen perturbations have to be used (i.e., $5 \leq N_{pert} \leq 15$). Since the evaluation of Eq. 5.28 requires N_e simulation runs, Eq. 5.30 requires $N_e * N_{pert}$ simulation runs, which is quite expensive.

Note that at beginning of each iteration of an optimization algorithm, $\boldsymbol{m}_r$ (Eq. 5.29) is computed so $\hat{J}_R(\boldsymbol{v}, \lambda_i^{k_L}, \mu^{k_L})$ only involves a single model $\boldsymbol{m}_r$ while ψ_{c_J} (Eq. 5.26) involves the entire ensemble of realizations used to represent the geological uncertainty. When computing the stochastic gradient of $c_J(\boldsymbol{v}, \lambda_{c_J}^{k_L}, \mu^{k_L})$ in order to obtain the stochastic gradient of ψ_{c_J}, it is appropriate to use one perturbation per realization as discussed previously in Section 2.2.2. However, $\hat{J}_R(\boldsymbol{v}, \lambda_i^{k_L}, \mu^{k_L})$ depends only on $\boldsymbol{m_r}$ so we will need to use five to fifteen perturbations to compute the stochastic gradient of $\hat{J}_R$ where the simulation will need to be run for each perturbation but based on the single model $\boldsymbol{m}_r$. In essence, the stochastic gradient of the augmented Lagrangian function can be approximated using the sum of two stochastic gradients. The details of this procedure are formulated below.

The gradient of Eq. 5.28 (or equivalently Eq. 5.25) is given by

$$\nabla_{\boldsymbol{v}} L_A(\boldsymbol{v}, \lambda_{c_i}^{k_L}, \lambda_{c_J}^{k_L}, \mu^{k_L}) = \nabla_{\boldsymbol{v}} \hat{J}_R(\boldsymbol{v}, \lambda_i^{k_L}, \mu^{k_L}) - \nabla_{\boldsymbol{v}} \psi_{c_J}(\boldsymbol{v}, \lambda_{c_J}^{k_L}, \mu^{k_L}), \tag{5.31}$$

where

$$\nabla_{\boldsymbol{v}} \psi_{c_J}(\boldsymbol{v}, \lambda_{c_J}^{k_L}, \mu^{k_L}) = \begin{cases} 0 & c_J \leq -\mu_{c_J}^{k_L} \lambda^{k_L} \\ \left(-\lambda_{c_J}^{k_L} + \frac{c_J}{\mu^{k_L}}\right) \nabla_{\boldsymbol{v}} c_J & c_J > -\mu^{k_L} \lambda_{c_J}^{k_L} \end{cases}, \tag{5.32}$$

and from Eq. 5.24,

$$\nabla_{\boldsymbol{v}} c_J(\boldsymbol{v}) = -\frac{\nabla_{\boldsymbol{v}} J_E(\boldsymbol{v})}{J_E(\boldsymbol{v}_E^*)}. \tag{5.33}$$

We propose to separate the StoSAG direction $\boldsymbol{d}_{L_A}$ into two parts where we estimate StoSAG gradients for the two terms $\hat{J}_{\text{R}}$ and ψ_{c_J} separately, denoted as $\boldsymbol{d}_{\hat{J}_R}$ and $\boldsymbol{d}_{\psi_{c_J}}$ respectively. $\boldsymbol{d}_{L_A}$ obtained in this way is denoted by

$$\boldsymbol{d}_{\text{sep}} = \boldsymbol{d}_{\hat{J}_R} - \boldsymbol{d}_{\psi_{c_J}}. \tag{5.34}$$

$\boldsymbol{d}_{\hat{J}_R}$ is the stochastic gradient of the summation of J_R and $-\sum_{i=1}^{N_{nc}} \psi_{c_i}$, and is approximated using N_{pert} perturbations (we use $N_{pert} = 10$ in this chapter):

$$\boldsymbol{d}_{\hat{J}_R} = (\Delta \boldsymbol{V} \Delta \boldsymbol{V}^T)^{+} \Delta \boldsymbol{V} \boldsymbol{\Delta \hat{J}} \left(\boldsymbol{v}, \boldsymbol{m}_r, \lambda^{k_L}, \mu^{k_L}\right). \tag{5.35}$$

$\boldsymbol{d}_{\psi_{c_J}}$ is obtained as

$$\boldsymbol{d}_{\psi_{c_J}} = \begin{cases} 0 & c_J \le -\mu^{k_L}\lambda^{k_L} \\ \left(-\lambda^{k_L} + \frac{c_J}{\mu^{k_L}}\right)\boldsymbol{d}_{c_J} & c_J > -\mu^{k_L}\lambda^{k_L}, \end{cases} \tag{5.36}$$

where $\boldsymbol{d}_{c_J}$ is approximated using one perturbation per realization given by

$$\boldsymbol{d}_{c_J} = \frac{1}{N_e * J_E^*}\sum_{k=1}^{N_e}(\delta\boldsymbol{v}_k\delta\boldsymbol{v}_k^+)\delta\boldsymbol{v}_k\delta J\left(\boldsymbol{v}, \boldsymbol{m}_k, \lambda^{k_L}, \mu^{k_L}\right). \tag{5.37}$$

The number of simulation runs required to approximate $\boldsymbol{d}_{\text{sep}}$ is N_{pert} if $c_J \le -\mu^{k_L}\lambda_{c_J}^{k_L}$ and $N_e + N_{pert}$ if $c_J > -\mu^{k_L}\lambda_{c_J}^{k_L}$. If $N_e = 10$ and $N_{pert} = 10$, estimating $\boldsymbol{d}_{\text{sep}}$ requires only 10 to 20 simulation runs while estimating $\boldsymbol{d}_{\text{sim}}$ requires 100 simulation runs.

5.4 Numerical Example: Channelized Reservoir

In this section, we present numerical results for the channelized reservoir model whose description is given in Section 3.2.1. Recall that the channelized reservoir is box-shaped with 50 × 50 × 14 gridblocks where each gridblock is 200 ft × 200 ft × 10 ft. We firstly consider the well trajectory optimization of slanted wells and multi-segmented wells for deterministic optimization. Then we consider the bi-objective well placement optimization where we maximize the expected NPV over a set of realizations characterizing the geological uncertainty and maximize the minimum NPV.

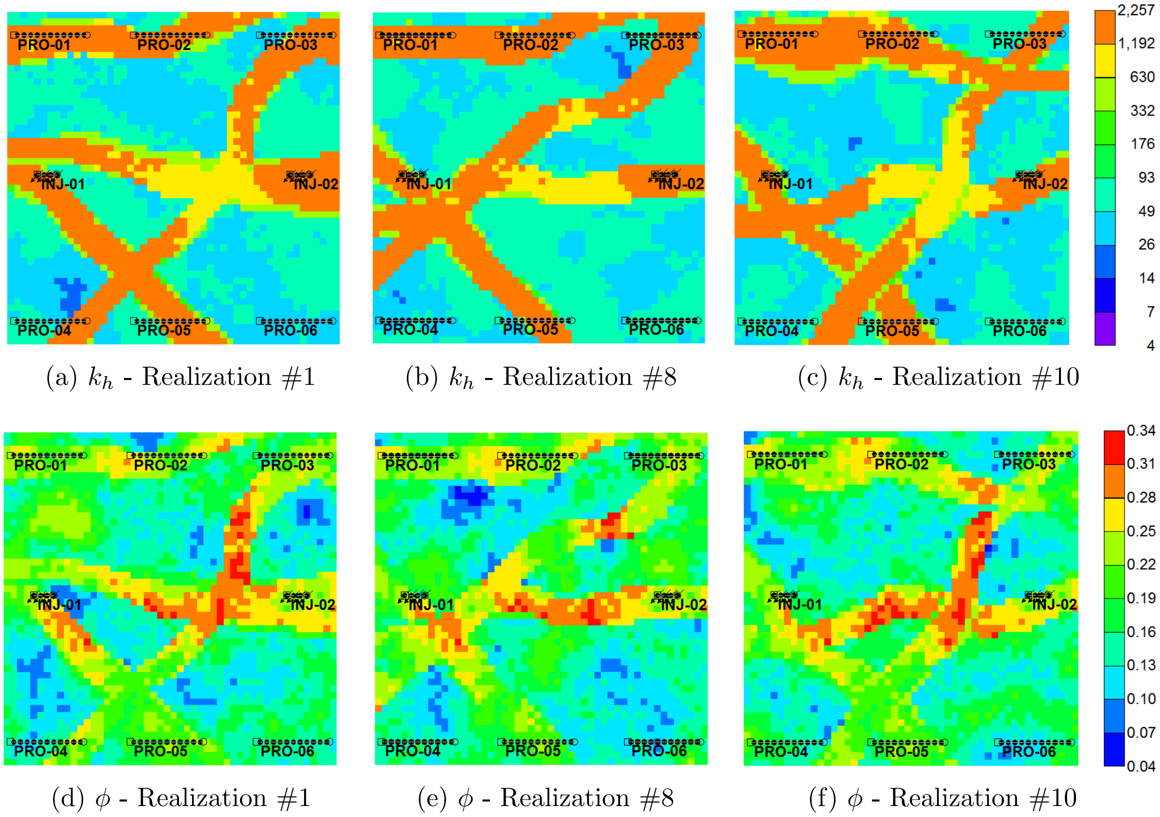
Ten realizations are used to represent the geological uncertainty where the facies depositions are generated using object-based modeling and the petrophysical parameters for each facies (i.e., permeability and porosity) are generated using sequential Gaussian Simulation; see related parameters in Table 3.1. The horizontal permeability along the x and y directions are set equal while the vertical permeability is set equal to $\frac{1}{10}$th of the horizontal permeability. The reservoir life is set to be 3,000 days throughout where all wells operate under BHP control. The BHP's for injectors are 6,000 psi and the BHP's for producers are 2,000 psi. We consider placing 8 wells including 6 producers initialized as horizontal wells of $2,000$ ft in length which are completed at different layers and 2 injectors initialized as slightly slanted wells of 616 ft in length which fully penetrate all 14 layers. Fig. 5.3 shows the initial well trajectories plotted on the horizontal permeability field (upper row) and the porosity field (bottom row) of layer 7 for three different realizations. For easier view, the projected initial paths are plotted on layer 7 in Figs. 5.6(a) and 5.11(a) without any value of petrophysical parameters. In Figs. 5.3, the squares, circles and black dots represent the projections of the entry points, exit points and perforations of a well respectively; the black solid line represent the projections of the perforations in the current layer. PRO-01 and PRO-06 are initially completed in layer 1, PRO-03 and PRO-04 are initially completed in layer 5, and PRO-02 and PRO-05 are initially completed in layer 11. The economic parameters to calculate the NPV are set as: r_o^n = \$50.0/STB, c_w^n = \$5.0/STB, c_{wi}^n = \$5.0/STB, b=10.0%. The drilling cost for well iw is defined as

$$C_w^{iw} = \sum_{i=1}^{N_{seg}} \left\{ 1,524\sqrt{(z_i^{iw} - z_{i-1}^{iw})^2} + 3,048\sqrt{(x_i^{iw} - x_{i-1}^{iw})^2 + (y_i^{iw} - y_{i-1}^{iw})^2} \right\}. \tag{5.38}$$

A minimum well spacing of 500 ft is also enforced, see Eq. 5.11. Meanwhile, we also enforce the maximum length of a well be no greater than 4,000 ft for producers and 2,000 ft for injectors, see Eq. 5.12.

We consider the optimization of two scenarios, where in one case, all eight wells are considered as slanted wells and in another case, the six producers are parameterized as multi-segmented wells and the two injectors remain as slanted wells. Each multi-segmented well is characterized by 8 segments where the entry point of a well (x_0, y_0, z_0) is bounded into the reservoir, the lower and upper bounds on the azimuth angle of the entry segment θ_1^{iw} are set equal to 0° and 360°, the lower and upper bounds on the change in the azimuth angle between two consecutive well segments, $\delta\theta_i^{iw}$, are set equal to $-20°$ and $20°$, and the lower and upper bounds on the segment length, l_i^{iw}, are set equal to 200 ft and 500 ft.

All the optimization variables are normalized to [0,1]. In StoSAG, the perturbations of the ith design variable are sampled from the normal distribution $N(0, \sigma_i^2), i = 1, 2, \cdots, N_x$, where σ_i's are chosen so that a perturbation causes a reasonable change in the objective



(a) k_h - Realization #1 (b) k_h - Realization #8 (c) k_h - Realization #10

(d) ϕ - Realization #1 (e) ϕ - Realization #8 (f) ϕ - Realization #10

Figure 5.3: Projections of the initial well trajectories on the horizontal permeability field (upper row) and the porosity field (bottom row) of layer 7 for three different realizations, channelized model.

function that is being maximized. Here, σ_i's are chosen so that the two end points of each well move by approximately one gridblock when the perturbation of each parameter have a magnitude of σ_i. For example, if the ith design variable is a coordinate variable and there are N gridblocks along the coordinate direction, we approximately take $\sigma_i = \frac{1}{N}$. Hence, σ_i is set to 0.02 in the x and y directions and 0.07 in the z direction. Denote the initial length of each multi-segmented well as $l^0_{iw}, iw = 1, 2, \cdots, N_w$ and denote the minimum value of the length and width of a gridblock as ΔX. The perturbations in θ^{iw}_1, $\delta\theta^{iw}_i$ and l^{iw}_i in the real domain should move the end point of a multi-segmented well roughly by one gridblock. σ_i's for θ^{iw}_1's are chosen so that $l^0_{iw}\sin(\sigma_i(\theta^{\text{up},iw}_1 - \theta^{\text{low},iw}_1)) \approx \Delta X$, σ_i's for $\delta\theta^{iw}_i$'s are chosen so that $l^0_{iw}\sin(\sigma_i(\delta\theta^{\text{up},iw}_i - \delta\theta^{\text{low},iw}_i)N_{seg}) \approx \Delta X$, and σ_i's for l^{iw}_i's are chosen so that $\sigma_i(l^{\text{up},iw}_i - l^{\text{low},iw}_i)N_{seg} \approx \Delta X$. So for variables θ^{iw}_1, $\delta\theta^{iw}_i$ and l^{iw}_i, the standard deviations σ_i's are set as 0.01, 0.02 and 0.05 respectively, which correspond to 3.6°, 0.8° and 32 ft.

The maximum step size is set as 0.1, and the maximum number of step size cuts and the maximum number of allowable resamples are both set equal to 5. After the application of ALM using StoSAG, ALM using GPS is used to fine tune the optimal well locations. In ALM using GPS, a fixed mesh is used where Δ_i's are set equal to σ_i's. GPS is considered to converge if NPV cannot be improved after searching all coordinate directions of the design variable. Note that there is still a possibility that higher NPV can be obtained if we continue the optimization process with a smaller mesh size Δ_i. However, we avoid doing that to save computational resources. In deterministic optimization, the maximum number of simulation runs is set as 3,000. In bi-objective well placement optimization, the maximum number of simulation runs is set as 4,500 for each stage of the optimization runs (the robust optimization and the secondary optimization).

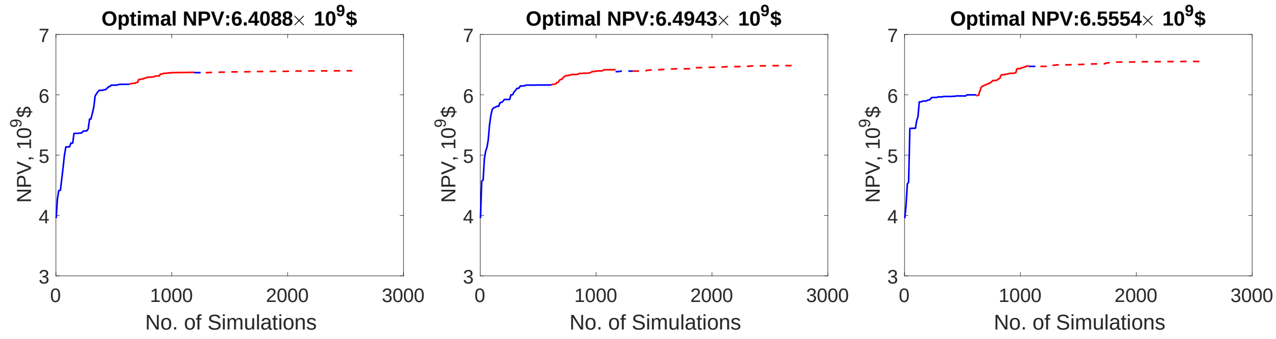
5.4.1 Deterministic optimization using realization #1

In this section, deterministic optimization using realization #1 of the channelized reservoir example (see Fig. 5.3(a) and (d)) is conducted starting from the initial well placement given in Fig. 5.3. We consider the optimization of both slanted wells and multi-segmented wells. For slanted well parameterization, (i) bound constraints on the design variables ((x, y, z) coordinates of the two end points of wells) enforce the condition that all wells fall in the reservoir and (ii) the nonlinear constraints include the well spacing constraints and well length constraints. However, for multi-segmented well parameterization, (i) the bound constraints on design variables do not guarantee that the whole well lays inside the reservoir and well trajectories that fall outside the reservoir are not considered as perforations, and (ii) since the maximum well length constraints can be enforced by bound constraints on l_i^{iw}, minimum well spacing constraint is the only type of nonlinear constraints in this example. Different than the initial guess for slanted wells, the circle symbols of PRO-03 and PRO-06 in Fig. 5.3 are selected as the entry points of the multi-segmented wells, in order to avoid the truncation of the well trajectories when the segment lengths are increased so that PRO-03 and PRO-06 go beyond the eastern boundary of the reservoir.

1. Slanted trajectory optimization followed by multi-segmented trajectory optimization

The 8 wells are initially parameterized as slanted wells and StoSAG followed by GPS coupled with ALM is used to optimize the well trajectories where the total number of design variables (coordinates of well end points) is 48. Then the optimal solutions of the producers found in the first optimization are parameterized as multi-segmented wells with 8 segments of equal lengths while the injectors are parameterized as one-segmented wells, i.e., slanted wells, then using the reparameterized optimal solution for the slanted well case as the initial guess, StoSAG followed by GPS is applied. The total number of design variables in this case is 174. Three optimization runs with different seeds are tried. Fig. 5.4 shows the NPV versus number of simulation runs obtained where the blue and red solid lines represent the slanted well trajectory optimization using StoSAG and GPS respectively, and the blue and red dashed lines represent the subsequent multi-segmented well trajectory optimization using StoSAG and GPS respectively. When optimizing the slanted well trajectories, StoSAG improved the NPV significantly, by approximately 55% on average, within 600 simulation runs. However, the difference between the optimal NPV's obtained with three seeds can be up to $\$0.17\times10^9$, which is equivalent to 4.4% of the initial NPV, indicating that StoSAG can suffer from pre-mature convergence due to the roughness of the objective function surface and the gradient estimation error. A subsequent optimization using GPS is able to improve the NPV by approximately another 8% and the difference between the optimal NPV's for the three seeds is reduced to $\$0.15 \times 10^9$, or equivalently 3.7% of the initial NPV. The follow-up multi-segmented trajectory optimization only improves the NPV slightly (from $\$6.41 \times 10^9$ to $\$6.49 \times 10^9$ on average of three seeds, which is equivalent to 2% of the initial NPV), indicating that it may be unnecessary to use over complicated parametrization for wells in a box-shaped reservoir.

Fig. 5.5 shows the cumulative oil production, cumulative water production and injection for the initial and optimal well placement obtained using seed 3 (the one with the highest NPV among three seeds) after the optimization using slanted well parameteriza-

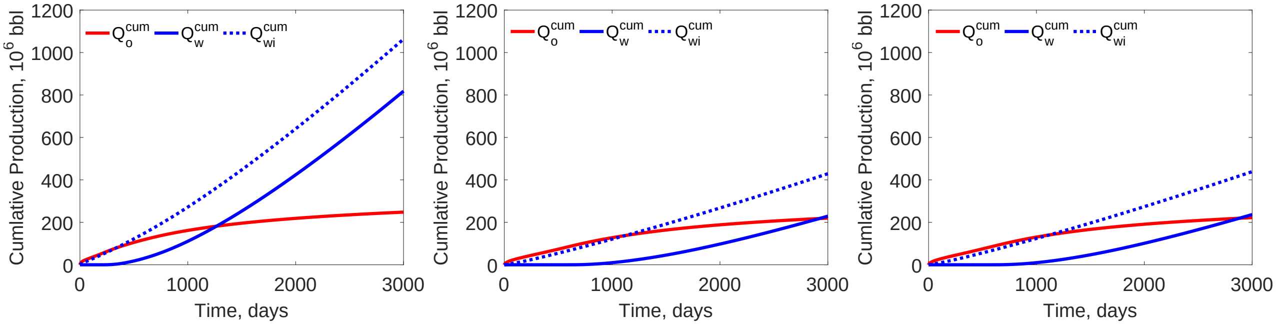


(a) Seed 1 (b) Seed 2 (c) Seed 3

Figure 5.4: NPV versus number of simulation runs obtained for slanted well trajectory optimization followed by multi-segmented trajectory optimization, deterministic optimization using realization #1, channelized model. The solid lines represent the slanted well trajectory optimization (StoSAG in blue and GPS in red) and the dashed lines represent the subsequent multi-segmented well trajectory optimization (StoSAG in blue and GPS in red).

tion followed by multi-segmented well parameterization. In Fig. 5.5, Q_o^{cum}, Q_w^{cum} and Q_{wi}^{cum} respectively represent the cumulative oil production, cumulative water production and injection. From Fig. 5.5, after the optimization using slanted well parameterization, the amounts of injected and produced water are both greatly reduced while the amount of produced oil remains similar. Consistent with Fig. 5.4 that the multi-segmented trajectory optimization only improves the NPV slightly, there is only slight change in Q_o^{cum}, Q_w^{cum} and Q_{wi}^{cum} after the follow-up optimization.

In Fig. 5.6, the well trajectories of the initial and optimal well placement (obtained with seed 3) are projected onto layer 7 without any value of petrophysical parameters. These paths are also plotted on the oil saturation field of layer 7 after 3,000 days of production in Fig. 5.7. Since in Figs. 5.6 and 5.7, the projections of the optimal trajectories of the multi-segmented wells are close to straight, we only present the coordinates of the starting and end points for all eight wells in Table 5.1. From Fig. 5.7 and Table 5.1, in the optimal solution of the slanted wells, (i) after using StoSAG, both injectors are moved in the north direction with shorter trajectories where the lengths of INJ-01 and INJ-02 are reduced to 315 ft and 324 ft respectively from 616 ft, and all producers are moved closer to the reservoir bounds; (ii) the subsequent application of ALM with GPS tends to increase the lengths of some wells, i.e. PRO-01, PRO-02 and PRO-04, and also causes PRO-04 to be moved closer to the southwest corner. Considering that the subsequent multi-segmented trajectory optimization only slightly increases the optimal NPV, the optimal well trajectories look quite similar to the ones obtained after the slanted well optimization. Fig. 5.8 shows the projections of the initial and optimal injector perforations obtained with slanted followed by multi-segmented trajectory optimization using seed 3, onto the $x - z$ plane of the horizontal permeability field of realization #1. From Fig. 5.8, after optimization, the two injectors not only have shorter trajectories, but also perforate gridblocks with relative low permeabilities compared to the initial well placement given in Fig. 5.8(a), leading to a significant decrease in the amount of cumulative water injection and production (see Fig. 5.5).

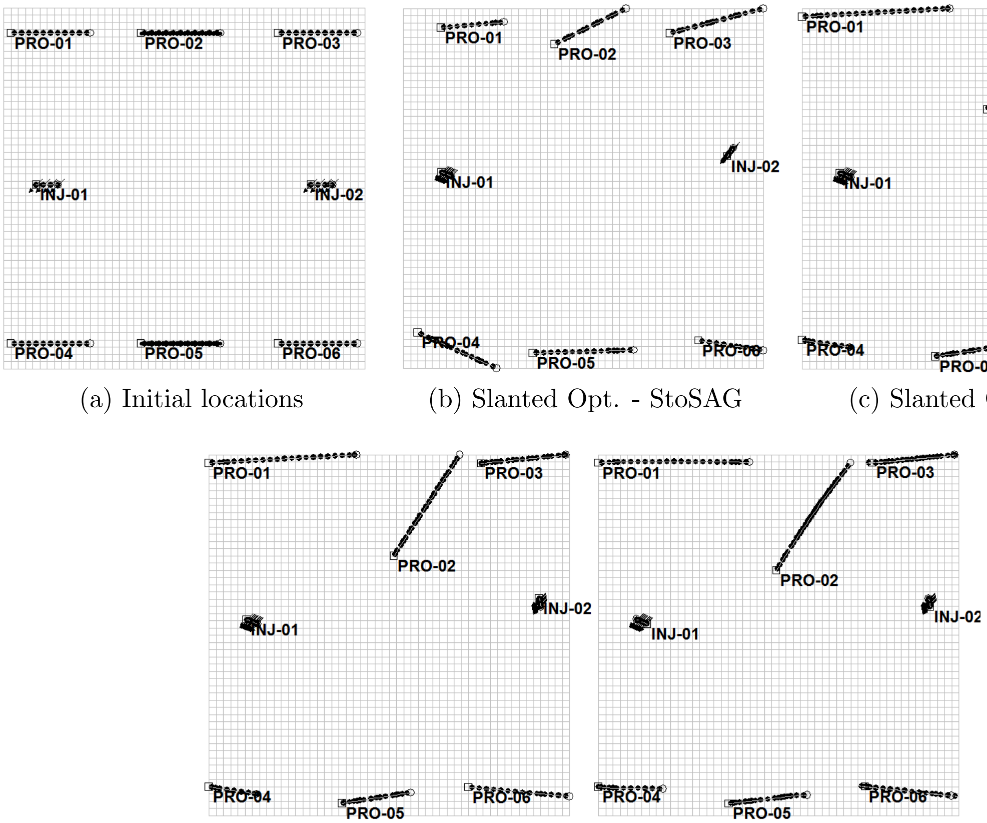


(a) Initial locations (b) Slanted Opt. (c) Slanted - multi-segmented Opt.

Figure 5.5: Cumulative oil production, water production and water injection obtained with the initial and optimal well placements after slanted well trajectory optimization followed by multi-segmented trajectory optimization using seed 3, deterministic optimization using realization #1, channelized model.

2. Multi-segmented trajectory optimization

To compare with the optimization process of slanted wells followed by multi-segmented wells, we also carried out optimization of multi-segmented wells by itself starting from the initial well trajectories given in Fig. 5.7(a). Fig. 5.9 shows the NPV versus number of simulation runs obtained using StoSAG (solid blue line) and GPS (solid red line) respectively. From Fig. 5.9, StoSAG improves the NPV significantly, by approximately 52% on average, within 600 simulation runs. However, the difference between the optimal NPV obtained with three seeds can be up to \$0.34 × 10^9, or equivalently 9% of the initial NPV,



(a) Initial locations (b) Slanted Opt. - StoSAG (c) Slanted Opt. - GPS

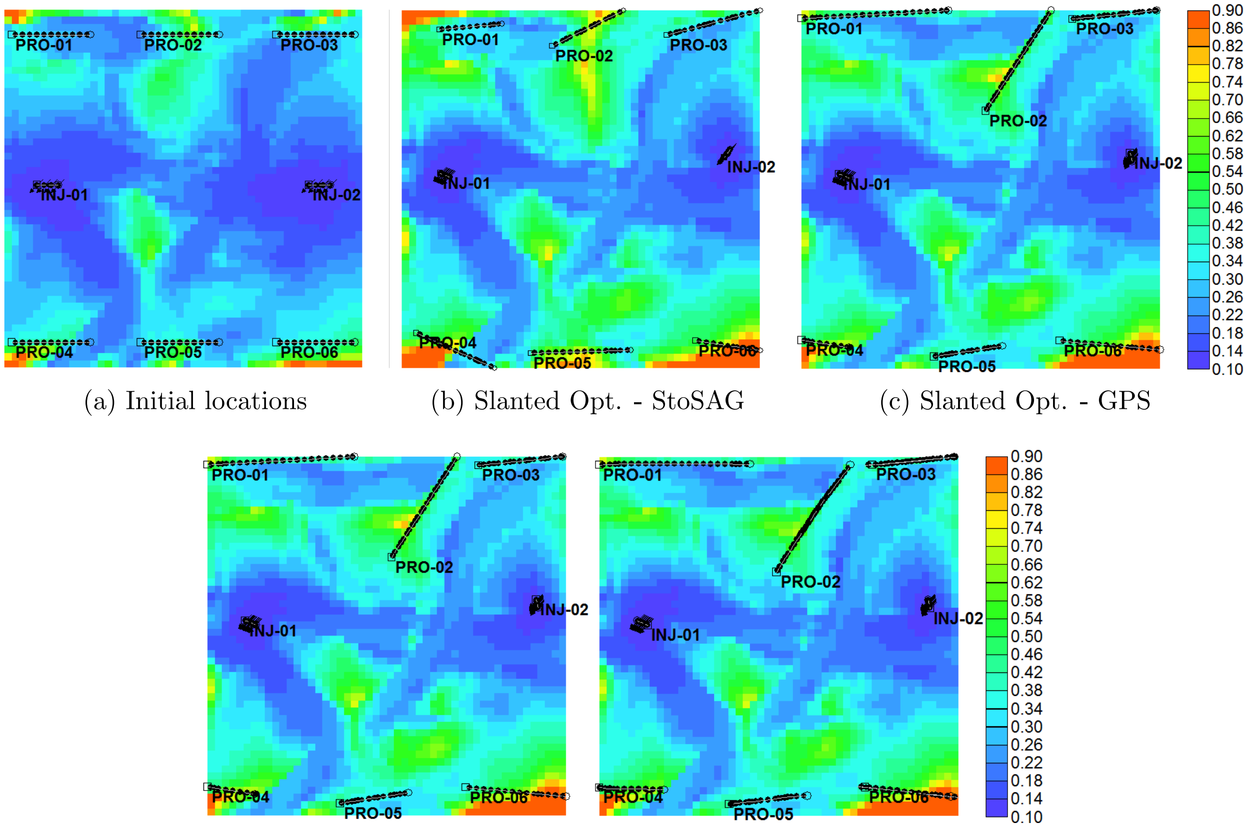
(d) Multi-segmented Opt. - StoSAG (e) Multi-segmented Opt. - GPS

Figure 5.6: Projections of the initial and optimal well placements after slanted well trajectory optimization followed by multi-segmented trajectory optimization on layer 7 using seed 3, deterministic optimization using realization #1, channelized model.

Table 5.1: Summary of the trajectories of all 8 wells obtained with the initial and optimal well placements after slanted well trajectory optimization followed by multi-segmented trajectory optimization using seed 3, deterministic optimization using realization #1, channelized model.

Wells	Initial setting		Slanted Opt.		Slanted-multi-segmented Opt.	
	$(x_0^{iw}, y_0^{iw}, z_0^{iw})$	$(x_{N_{seg}}^{iw}, x_{N_{seg}}^{iw}, x_{N_{seg}}^{iw})$	$(x_0^{iw}, y_0^{iw}, z_0^{iw})$	$(x_{N_{seg}}^{iw}, x_{N_{seg}}^{iw}, x_{N_{seg}}^{iw})$	$(x_0^{iw}, y_0^{iw}, z_0^{iw})$	$(x_{N_{seg}}^{iw}, x_{N_{seg}}^{iw}, x_{N_{seg}}^{iw})$
PRO-01	$(300, 700, 4806)$	$(2300, 700, 4806)$	$(100, 218, 4800)$	$(4092, 0, 4800)$	$(100, 218, 4800)$	$(4099, 202, 4800)$
PRO-02	$(3900, 700, 4905)$	$(5900, 700, 4905)$	$(5194, 2707, 4880)$	$(6958, 0, 4888)$	$(4994, 3107, 4880)$	$(6976, 252, 4888)$
PRO-03	$(7700, 700, 4845)$	$(9700, 700, 4845)$	$(7549, 236, 4800)$	$(9900, 0, 4850)$	$(7550, 220, 4800)$	$(9887, 1, 4850)$
PRO-04	$(300, 9300, 4845)$	$(2300, 9300, 4845)$	$(0, 9202, 4831)$	$(1322, 9400, 4826)$	$(0, 9202, 4831)$	$(1694, 9259, 4805)$
PRO-05	$(3900, 9300, 4905)$	$(5900, 9300, 4905)$	$(3701, 9654, 4860)$	$(5553, 9366, 4823)$	$(3701, 9654, 4863)$	$(5659, 9435, 4800)$
PRO-06	$(7700, 9300, 4806)$	$(9700, 9300, 4806)$	$(7335, 9223, 4800)$	$(10000, 9471, 4800)$	$(7320, 9184, 4800)$	$(9800, 9450, 4800)$
INJ-01	$(900, 4900, 4940)$	$(1500, 4900, 4800)$	$(1047, 4564, 4940)$	$(1349, 4685, 4804)$	$(1052, 4556, 4940)$	$(1349, 4685, 4801)$
INJ-02	$(8500, 4900, 4800)$	$(9100, 4900, 4940)$	$(9164, 3991, 4936)$	$(9211, 4213, 4800)$	$(9164, 3991, 4936)$	$(9211, 4213, 4800)$

again indicating that StoSAG by itself cannot find the optimal solution of the well placement optimization problem. A subsequent optimization using GPS is able to improve the



(a) Initial locations

(b) Slanted Opt. - StoSAG

(c) Slanted Opt. - GPS

(d) Multi-segmented Opt. - StoSAG

(e) Multi-segmented Opt. - GPS

Figure 5.7: Oil saturation after 3,000 days of production plotted on layer 7, obtained with the initial and optimal well placements after slanted well trajectory optimization followed by multi-segmented trajectory optimization using seed 3, deterministic optimization using realization #1, channelized model.

NPV by another 10% and the difference between the optimal NPV of any pair of results for two distinct seeds is narrowed to \$0.09 × 10^9, or equivalently 2.4% of the initial NPV. More importantly, the average optimal NPV obtained with 3 runs for the multi-segmented well optimization is 3.3% less than the optimization process of slanted trajectories followed by multi-segmented trajectories within 3,000 simulation runs, which again indicates that it is unnecessary to use over-complicated well trajectory parameterization. In fact, the number of optimization variables for a multi-segmented well trajectory in this example is 3.5 times more than the number needed for slanted well trajectory parameterization so the multi-segmented case requires more simulation runs to converge. This explains the observation that a smaller NPV is obtained for a given number of total reservoir simulation runs when

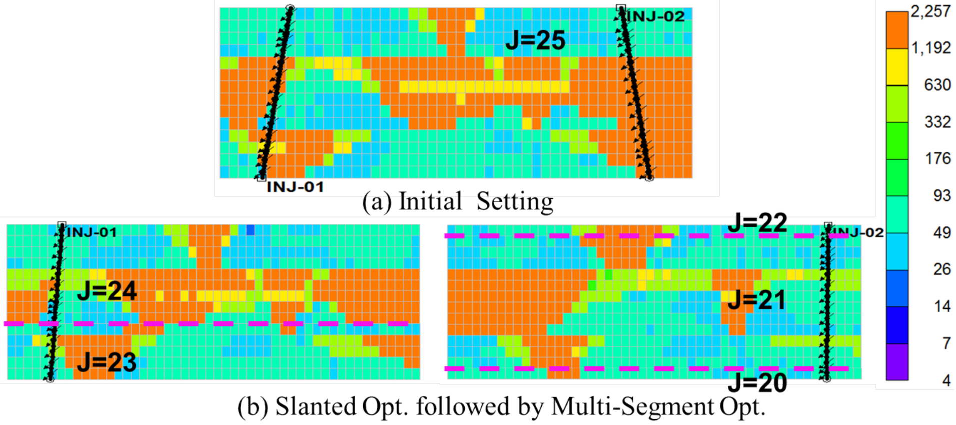
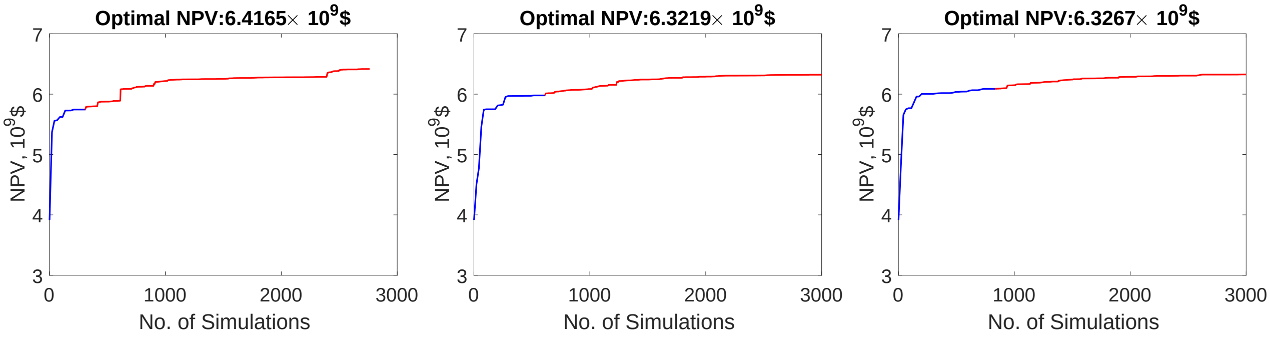


Figure 5.8: Projections of the initial and optimal injector perforations obtained with slanted followed by multi-segmented trajectory optimization using seed 3, onto the $x-z$ plane of the horizontal permeability field, deterministic optimization using realization #1, channelized model.

using the multi-segmented parameterization throughout the optimization process.



(a) Seed 1 (b) Seed 2 (c) Seed 3

Figure 5.9: NPV versus number of simulations obtained for multi-segmented trajectory optimization, deterministic optimization using realization #1, channelized reservoir. The blue lines represent the StoSAG algorithm and the red lines represent the GPS algorithm.

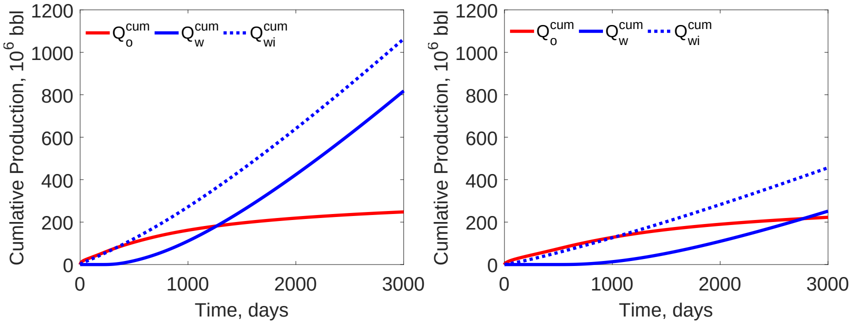
Fig. 5.10 shows the cumulative oil production, cumulative water production and injection for the initial and optimal well placement obtained using seed 1 (the one with the highest NPV among three seeds) after the optimization using multi-segmented well parameterization throughout the optimization process. From Fig. 5.10, similar to the previous case, after the optimization, the amounts of injected and produced water are both greatly reduced while the amount of produced oil remains similar. In Fig. 5.11, the well trajectories of the optimal well placement (obtained with seed 1) after StoSAG and GPS are projected onto layer 7 without any value of petrophysical parameters. These paths are also plotted on the oil saturation field of layer 7 after 3,000 days of production in Fig. 5.12. Since in Figs. 5.12, the projections of the optimal trajectories of the multi-segmented wells are close to straight, we only present the coordinates of the starting and end points, the total lengths, azimuth angles of the entry segments, and the intervals of the perforated layers for all eight wells in Table 5.2.

From Fig. 5.12, (i) similar to the optimal solution of slanted wells, both injectors are moved in the north direction and all producers are moved closer to the reservoir bounds, after using StoSAG; (ii) the subsequent application of ALM with GPS significantly changes some well trajectories, i.e., the lengths of both PRO-01 and PRO-02 are increased, PRO-02 is shifted downward by almost 2,000 ft, and PRO-04 whose trajectory lays outside of the reservoir after StoSAG is now shifted back into the reservoir again. Note that with the multi-segmented well parameterization, the bound constraints on the design variables do not guarantee that the whole well lays inside the reservoir and trajectories outside of the reservoir boundaries are not considered as perforations, which is the case of PRO-04 in Figs. 5.11(b) and 5.12(b). Fig. 5.13 shows the projections of the initial and optimal injector perforations obtained with the multi-segmented trajectory optimization using seed 1, onto the $x-z$ plane of the horizontal permeability field of realization #1. From Fig. 5.13, after optimization, the two injectors not only have shorter trajectories, but also perforate gridblocks with relative low permeabilities compared to the initial well placement given in Fig. 5.13(a), which is similar to Fig. 5.8.

For all the optimal well placement scenarios shown in Figs. 5.7 and 5.12, both the minimum well spacing constraints and the maximum well length constraints are satisfied. In fact, these two types of constraints are rarely violated since maximizing the NPV tends to penalize long well trajectories and to avoid wells intersecting with each other.

5.4.2 Bi-objective optimization using 10 realizations

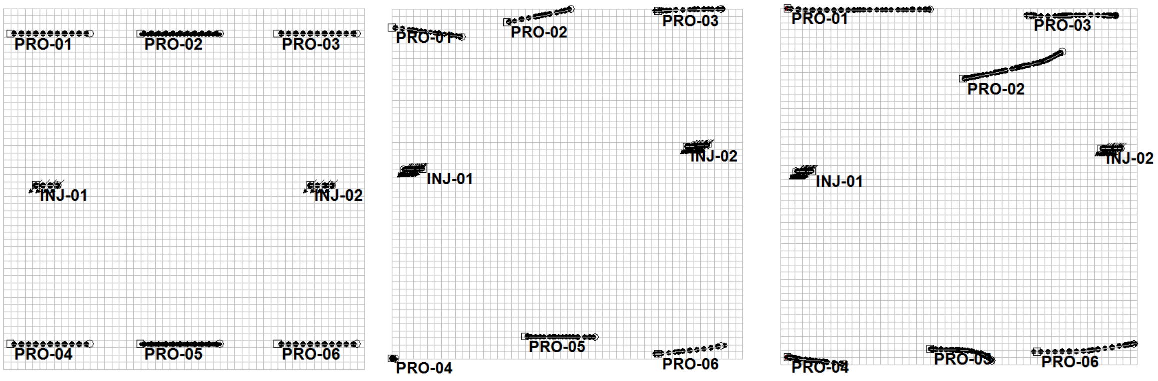
In this section, we consider the bi-objective well placement optimization where we



(a) Initial

(b) Multi-segmented Opt.

Figure 5.10: Cumulative oil production, water production and water injection obtained for the initial and optimal well placements after multi-segmented well trajectory optimization using seed 1, deterministic optimization using realization #1, channelized model.



(a) Initial locations

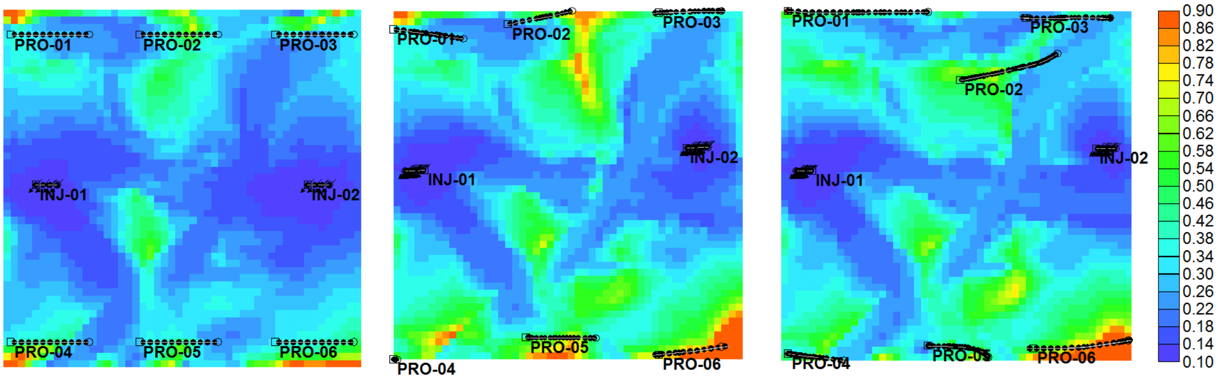
(b) Multi-segmented Opt. - StoSAG

(c) Multi-segmented Opt. - GPS

Figure 5.11: Projections of the initial and optimal well placement after multi-segmented trajectory optimization onto layer 7 using seed 1, deterministic optimization using realization #1, channelized model.

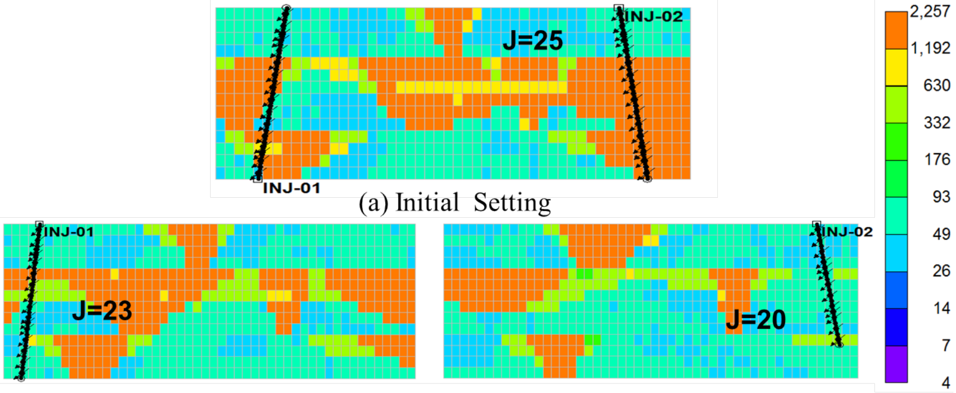
Table 5.2: Summary of the trajectories of all 8 wells obtained for multi-segmented trajectory optimization using seed 1, deterministic optimization using realization #1, channelized model.

Wells	Initial Setting					Slanted-multi-segmented Opt.				
	$(x_0^{iw}, y_0^{iw}, z_0^{iw})$	$(x_{N_{seg}}^{iw}, x_{N_{seg}}^{iw}, x_{N_{seg}}^{iw})$	θ_1^{iw}	l_{iw}	K1-K2	$(x_0^{iw}, y_0^{iw}, z_0^{iw})$	$(x_{N_{seg}}^{iw}, x_{N_{seg}}^{iw}, x_{N_{seg}}^{iw})$	θ_1^{iw}	l_{iw}	K1-K2
PRO-01	(300, 700, 4806)	(2300, 700, 4806)	0	2000	1	(200, 0, 4800)	(4198, 32, 4810)	4.7	4000	1-2
PRO-02	(3900, 700, 4905)	(5900, 700, 4905)	0	2000	11	(5159, 1968, 4926)	(7898, 1219, 4940)	-10.5	2868	5-14
PRO-03	(9700, 700, 4845)	(7700, 700, 4845)	180	2000	5	(9400, 200, 4860)	(6858, 191, 4807)	180.0	2463	1-7
PRO-04	(300, 9300, 4845)	(2300, 9300, 4845)	0	2000	5	(200, 9789, 4909)	(1650, 9971, 4837)	7.5	1467	4-11
PRO-05	(3900, 9300, 4905)	(5900, 9300, 4905)	0	2000	11	(4202, 9564, 4921)	(5929, 9890, 4940)	3.9	1813	6-14
PRO-06	(9700, 9300, 4806)	(7700, 9300, 4806)	180	2000	1	(9919, 9416, 4820)	(7105, 9640, 4800)	171.2	2917	1-3
INJ-01	(1500, 4900, 4800)	(900, 4900, 4940)	180	2000	1-14	(883, 4567, 4800)	(437, 4568, 4940)	179.8	467	1-14
INJ-02	(8500, 4900, 4800)	(9100, 4900, 4940)	0	2000	1-14	(9005, 3930, 4800)	(9528, 3933, 4905)	0.3	534	1-11



(a) Initial locations (b) Multi-segmented Opt. - StoSAG (c) Multi-segmented Opt. - GPS

Figure 5.12: Oil saturation after 3,000 days of production plotted on layer 7, obtained for the initial and optimal well placement after multi-segmented trajectory optimization using seed 1, deterministic optimization using Realization #1, channelized model.



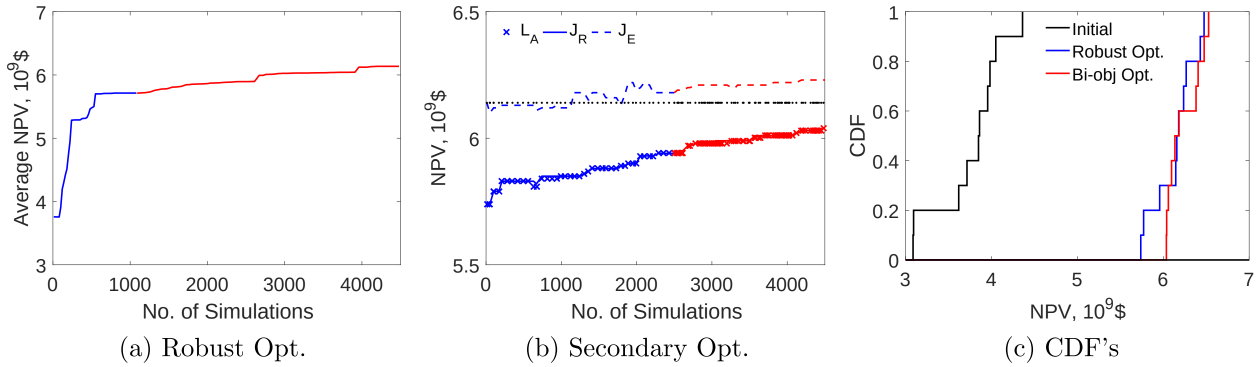
(a) Initial Setting

(b) Multi-Segmented Opt.

Figure 5.13: Projections of the initial and optimal injector perforations obtained with multi-segmented well trajectory optimization using seed 1, onto the $x - z$ plane of the horizontal permeability field, deterministic optimization using Realization #1, channelized model.

maximize the expected NPV over ten realizations characterizing the geological uncertainty and maximize the minimum NPV, see Eqs. 5.22 and 5.23, using Lexicographical method. We parameterize all 8 wells as slanted wells so the total number of design variables is 48 starting from the initial guess given in Fig. 5.7(a) and considering minimum well spacing constraints and maximum well length constraints, see Eq. 5.2. Fig. 5.14 shows a summary of the bi-objective well trajectory optimization using StoSAG followed by GPS coupled with ALM. According to the lexicographic method, the primary objective (the average NPV over 10 realizations, J_E) is optimized first where J_E is improved from \$3.76 × 10^9 to \$6.14 × 10^9, which represents a 63% increase in J_E. The primary optimization is terminated after 4,500 simulation runs where GPS has not converged yet. During the primary optimization, the nonlinear constraints are rarely violated. We denote the optimal solution of the primary optimization as $\boldsymbol{v}_E^*$. Starting from $\boldsymbol{v}_E^*$, a secondary optimization is carried out where the minimum NPV is maximized subject to an extra constraint that the average NPV is no less than $0.99J_E(\boldsymbol{v}_E^*)$, or equivalently $\$6.08 \times 10^9$. In the secondary optimization, $N_{pert} = 10$ perturbations are used to estimate the StoSAG search direction of J_R, $d_{\hat{J}_R}$, using Eq. 5.35 while one perturbation per realization is used to estimate the StoSAG search direction of ψ_{c_J}, $d_{\psi_{c_J}}$, using Eq. 5.36 when $c_J > -\mu^{k_L}\lambda^{k_L}$. Fig. 5.14(b) shows the changes in the augmented Lagrangian Function (L_A), the minimum NPV (J_R) and the average NPV (J_E) with respect to the number of simulation runs. It is important to note that L_A increases with respect to number of simulations monotonically only in the inner loop of the ALM. The updates to the Lagrangian multipliers and the penalty parameter in the outer loop will change the initial value of L_A in the next inner loop. Both J_R and J_E do not increase monotonically with respect to the number of simulation runs. From Fig. 5.14(b), the minimum NPV is increased from $\$5.74 \times 10^9$ to $\$6.04 \times 10^9$, or by approximately 5%; although the result is unexpected, the average NPV is also increased from $\$6.14 \times 10^6$ to $\$6.24 \times 10^9$, or by approximately 1.5%. Fig. 5.15 shows the indices of the model with the lowest NPV value at each iteration and the number of violated constraints during the secondary optimization process. As is observed, the secondary optimization does not appear to suffer from the non-differentiability issue even though the "risk model" (model $\boldsymbol{m}_r$) can change from iteration to iteration. At most iterations, the results of Fig. 5.15 indicate that realization #8 is the risk model, but for a few iterations, either realization #6 or #7 is the realization with the lowest NPV. Fig. 5.14(c) shows the cumulative distribution function (cdf) of NPV's of the 10 realizations obtained with the well placement of the initial guess and the optimal solution of the bi-objective optimization. From Fig. 5.14(c), robust optimization shifted the cdf of NPV to the right and the secondary optimization makes the cdf of NPV more compact. The realizations with the lowest, second lowest and the third lowest NPV respectively are realizations #8, #7 and

#6, i.e., the three realizations that have served as the risk realization at some iterations of the secondary optimization, see Fig. 5.15(b).



(a) Robust Opt.

(b) Secondary Opt.

(c) CDF's

Figure 5.14: Bi-objective well placement optimization using 10 realizations, channelized model. (a) Average NPV over 10 realizations versus number of simulation runs in primary optimization using StoSAG (blue) and GPS (red); (b) the augmented Lagrangian function, L_A, minimum NPV, J_R, and average NPV, J_E, versus number of simulation runs in secondary optimization using StoSAG (blue) and GPS (red) where L_A is represented by marker x, J_R is represented by the solid line, J_E is represented by the dashed line and $J_E(\boldsymbol{v}_E^*)$ is represented by the dotted line; (c) Cdf's of NPV's for the initial well placement and the optimal well placement after primary and secondary optimization.

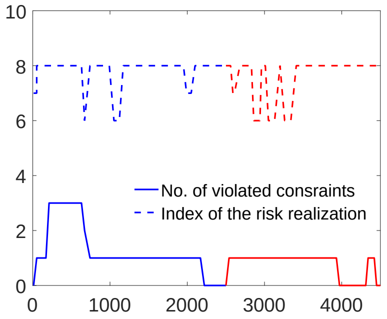
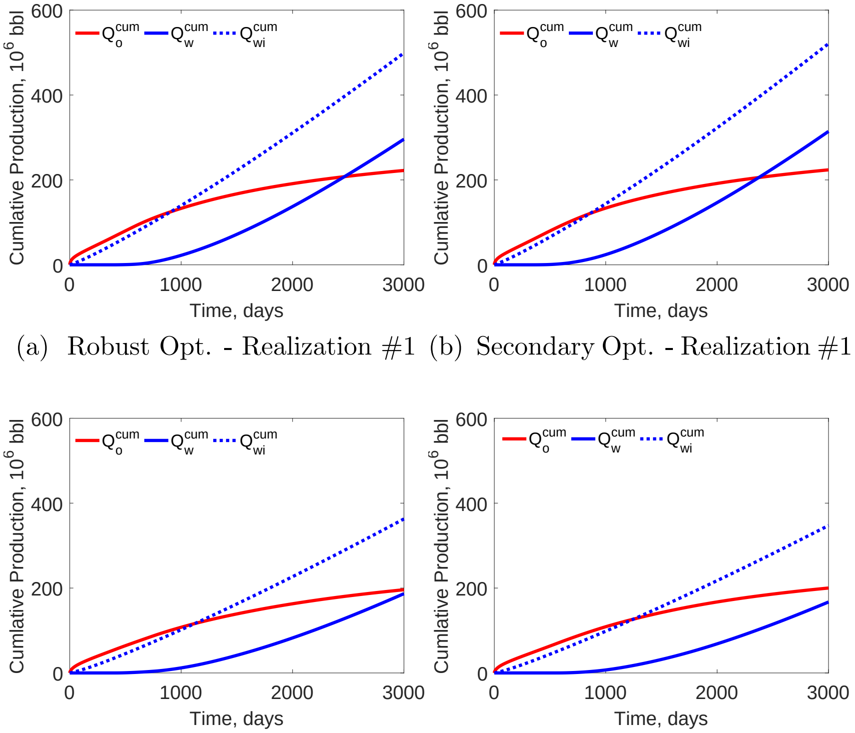


Figure 5.15: Number of violated constraints and indicies of the risk realization during the secondary optimization of the bi-objective optimization of slanted well trajectories using 10 realizations, channelized model.

Fig. 5.16 shows the cumulative oil production, cumulative water production and cumulative water injection after the secondary optimization for realization #1 whose NPV is reduced by 1.3% and for realization #8 whose NPV is increased by 4.6%. From Fig. 5.16, the secondary optimization slightly increases the volume of injected water and produced water for realization #1 and slightly decreases the volume of injected water and produced water for realization #8. For both realizations #1 and #8, the secondary optimization causes negligible change in the cumulative oil production.



(a) Robust Opt. - Realization #1 (b) Secondary Opt. - Realization #1

(c) Robust Opt. - Realization #8 (d) Secondary Opt. - Realization #8

Figure 5.16: Cumulative oil production, water production and water injection obtained after the primary and secondary optimization of the bi-objective optimization, channelized model.

The minimum NPV of the cdf obtained after the secondary optimization in Fig. 5.14 corresponds to realization #8 which was the risk realization during the last several iterations of the bi-objective optimization procedure. This means during the last iteration, the algorithm was maximizing the NPV of realization #8 but subject to the constraint that the expected NPV obtained during the first optimization step of bi-objective lexicographic optimization does not decrease by more than one percent. If the constraint is removed and we simply apply deterministic optimization to maximize the NPV of realization #8, we obtain an NPV value equal to $\$6.45 \times 10^9$, which is 6.8% higher than the minimum NPV obtained by bi-objective optimization.

It is worthwhile to mention that if robust optimization found a solution which is close to a global optimum, it is unlikely that the secondary optimization would improve

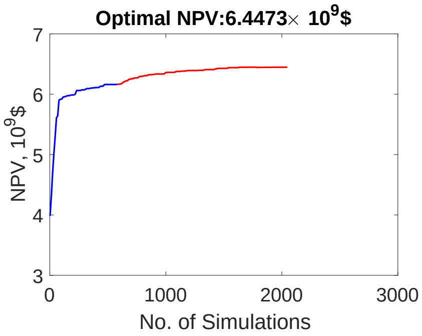
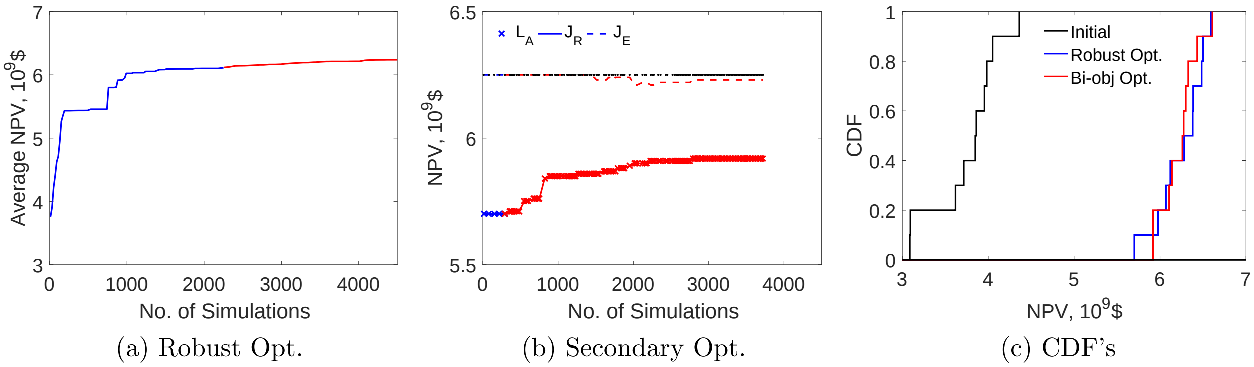


Figure 5.17: NPV versus number of simulation runs for the deterministic optimization using realization #8, channelized model. The blue line represents the StoSAG algorithm and the red line represents the GPS algorithm.

the average NPV since these two objectives now conflict with each other. To illustrate the more normal behavior, we tried a different run of the bi-objective optimization where we obtained an average NPV of \$$6.25 \times 10^9$ after the primary optimization which is decreased to \$$6.23\times10^9$ after the secondary optimization, even through the minimum NPV is improved from \$$5.70 \times 10^9$ to \$$5.92 \times 10^9$; see Fig. 5.18.



(a) Robust Opt.

(b) Secondary Opt.

(c) CDF's

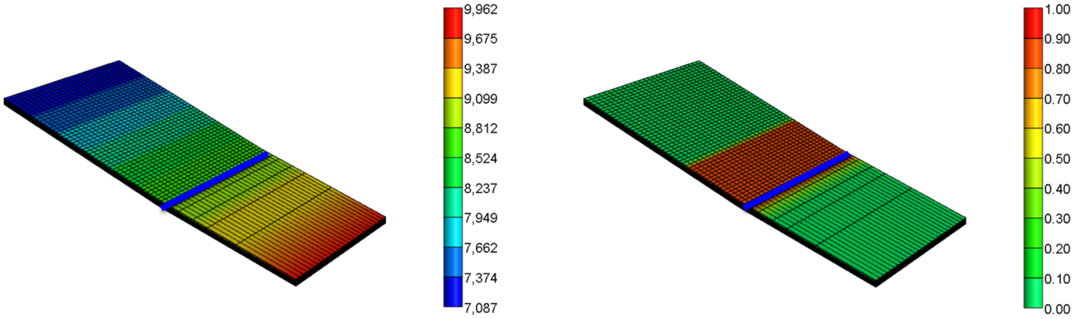
Figure 5.18: Bi-objective well placement optimization using 10 realizations using a different seed, channelized model. (a) Average NPV over 10 realizations versus number of simulation runs in primary optimization using StoSAG (blue) and GPS (red); (b) the augmented Lagrangian function, L_A, minimum NPV, J_R, and average NPV, J_E, versus number of simulation runs in secondary optimization using StoSAG (blue) and GPS (red) where L_A is represented by marker x, J_R is represented by the solid line, J_E is represented by the dashed line and $J_E(\boldsymbol{v}_E^*)$ is represented by the dotted line; (c) Cdf's of NPV for the initial well placement and the optimal well placement after primary and secondary optimization.

5.5 Numerical Example: Oseberg Reservoir

We consider a problem which is based on a modification of the original reservoir simulation model of Oseberg reservoir in the Norwegian sector of the North Sea. The modification simply uses a more uniform grid. The reservoir consists of three distinct geological zones, Etive, Rannoch and Oseberg. Etive is the top zone and Oseberg is the bottom zone. These two zones are separated by Rannoch which is a relatively tight layer. There is vertical communication between the three zones. We use a $54 \times 27 \times 10$ simulation grid with non-uniformed grid sizes in the oil and gas zones along x and z directions. The gridblock size in the x direction is equal to 328 ft in the top and middle part of the reservoir, and the gridblock size increases gradually from 328 ft to 2,624 ft as we approach the bottom of the reservoir boundary in the x direction. The gridblock size in the y direction is equal to 656 ft. Only one vertical gridblock is used in the Etive and Rannoch layers, and layers 3 through 10 correspond to the Oseberg zone. Gridblock sizes in the z direction are non-uniform with values equal to 23.0 ft in Etive, 16.5 ft in Rannoch and 11.5 ft in Oseberg.

The initial reservoir pressure is 4,071 psi at the depth of 8,192 ft subsea, and the original reservoir fluid bubble point pressure is 3,771 psi. The top structure of the Oseberg model and the initial oil saturation field are shown in Fig. 5.19. Note that the reservoir is tilted with a significant dip. The reservoir has a gas cap at the top and an aquifer at the bottom. The initial gas-oil contact is at 8,192 ft subsea and the water-oil contact is at 8,918 ft subsea. The oil zone is separated from the aquifer by an impermeable tar mat which is highlighted with a blue line in Fig. 5.19. In the oil zone, the initial oil saturation is 0.885 and the initial water saturation is equal to the irreducible water saturation which is equal to 0.115. The initial gas saturation in the gas cap is equal to 0.885. The water saturation of the gas cap is at the irreducible water saturation and there is no oil in the gas cap initially.

In this case study, all fluid related properties and statistical parameters to generate the model realizations are obtained from Zhang [88]. When generating the petrophycical parameters, a 2D non-isotropic exponential covariance function is used where the major correlation direction is along the x-axis direction and the minor correlation direction is along the y-axis direction. The correlation length is set to 6,555 ft in the major correlation direction, 1,968 ft in the minor correlation direction and 11.64 ft in the vertical direction. The horizontal log-permeability, vertical log-permeability and the porosity field are all generated following normal distribution where the mean and deviation of the petro-physical parameters for each geological zone are given in Table 5.3. Note that horizontal and vertical log-permeability in Oseberg decrease linearly from top to bottom, i.e., from layer 3 to layer 10. The correlation between horizontal log-permeability and vertical log-permeability is set equal to 0.8, and the correlation between horizontal log-permeability and porosity is set equal to 0.3, following Zhang [88]. The minimum porosity for any gridblock in the model is set to 0.03. In the tar mat, the horizontal permeability is set to 0.03 mD, and vertical permeability is set to 0.05 mD. Ten realizations of the horizontal permeability, vertical permeability and the porosity fields are generated to represent the geological uncertainty where realizations #1 and #7 are shown in Figs. 5.20 and 5.21 respectively.



(a) Reservoir depth

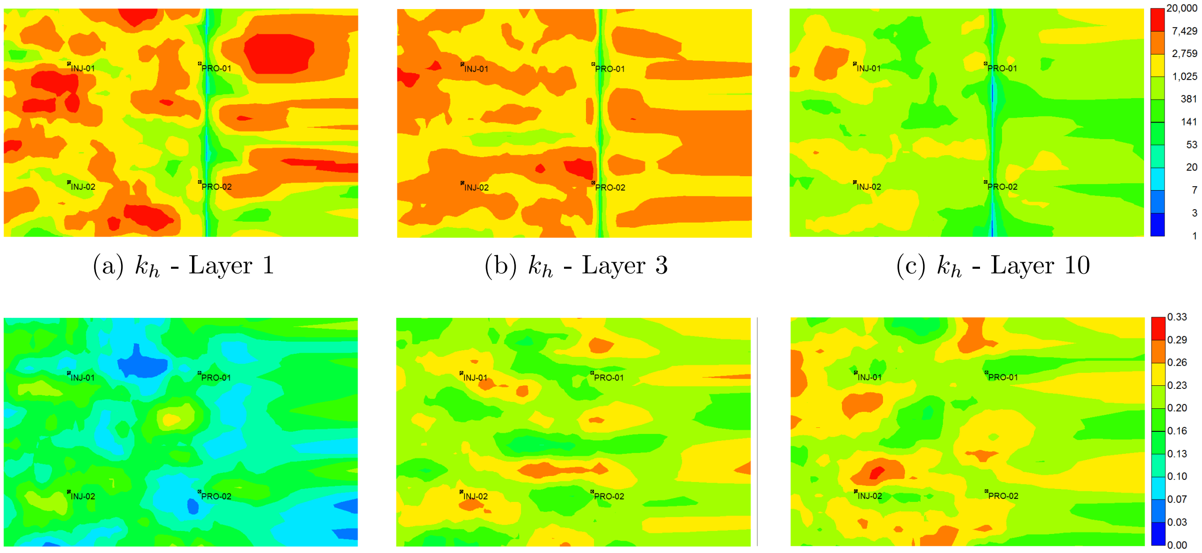
(b) Initial oil saturation

Figure 5.19: Top structure and initial oil saturation of the Oseberg model. The tar mat is highlighted with a blue line.

Table 5.3: Petrophysical property parameters, Oseberg model.

Parameters	Etive		Rannoch		Oseberg		
	Mean	Variance	Mean	Variance	Mean top	Mean bottom	Variance
$\ln(k)$	7.5	1.2	2.1	1.8	7.8	6.3	0.4
$\ln(k_z)$	6.3	1.8	0.15	2.2	6.4	4.4	0.8
ϕ	0.14	0.002	0.1	0.001	0.22	0.22	0.001

We put two gas injectors under rate control in the gas cap and two oil producers under pressure control in the oil zone. All four wells are initially vertical and are set to fully penetrate all 10 reservoir layers. The initial locations of the two producers are set close to the tar mat to delay gas breakthrough based on engineering experience. The two injectors are initially placed in the middle of the gas cap. Intuitively, the locations of the gas injectors, as long as they are in the gas cap, may not affect the reservoir development very much because the gas density is very low compared to the densities of oil and water and the reservoir permeabilities are high and fairly uniform with the exception of the tar mat. Due to the low permeability of the tar mat, it seems likely that the optimal locations of the producers are just above the tar mat. The initial well locations are shown in both Figs. 5.20 and 5.21. Injectors are under rate control with fixed injection rate of 100 MMscf/day. Producers are under BHP control with specified bottomhole pressure of 2,000 psi. The maximum water cut and gas oil ratio for a producer are set to 0.98 and 561 MMscf/STB, respectively, and a producer is shut in if these values are exceeded. Also an upper limit of 100,000 STB/day is set for the oil production rate of each producer and a producer switches to constant oil rate control if this value is exceeded. The maximum injection BHP is set to be 6,000 psi, and an injector switches to bottomhole pressure control if this value is exceeded. A minimum distance of 500 ft between any pair of the wells is enforced. During the optimization, the bounds on the well trajectory variables of the injectors are set to be the boundaries of the initial gas cap (the first 31 gridblocks along x direction) while the bounds on the well trajectory variables of the producers are set to be the boundaries of the initial oil zone (the 32th to 47th gridblocks along x direction). We parameterize all four wells as slanted wells using the coordinates of their end points and consider both deterministic optimization and bi-objective optimization in which, there are 24 optimization variables in total. Similar to the channelized reservoir, all the trajectory variables are scaled to the domain of [0,1] and the StoSAG perturbations are sampled from Gaussian distributions $\mathcal{N}(0, \sigma_i^2)$ so that the perturbation of each parameter on the magnitude of σ_i approximately corresponds to one gridblock. For a design variable corresponding to the y-coordinate, σ_i is set equal to $\frac{1}{27}$; for a design variable corresponding to the x-coordinate, σ_i is set equal to $\frac{1}{31}$ for the injectors and $\frac{1}{16}$ for the producers; for a design variable corresponding to z coordinate, σ_i is set equal to 0.1. In the GPS implementation, a fixed mesh size, Δ_i, is used where Δ_i is set equal to σ_i. In this example, we set the oil price equal to \$50.0/STB, the water disposal cost is \$5.0/STB, the gas injection cost is \$1.50/Mscf, the gas disposal cost is zero, and the annual discount rate is 10.0%.



(a) k_h - Layer 1

(b) k_h - Layer 3

(c) k_h - Layer 10

(d) ϕ - Layer 1

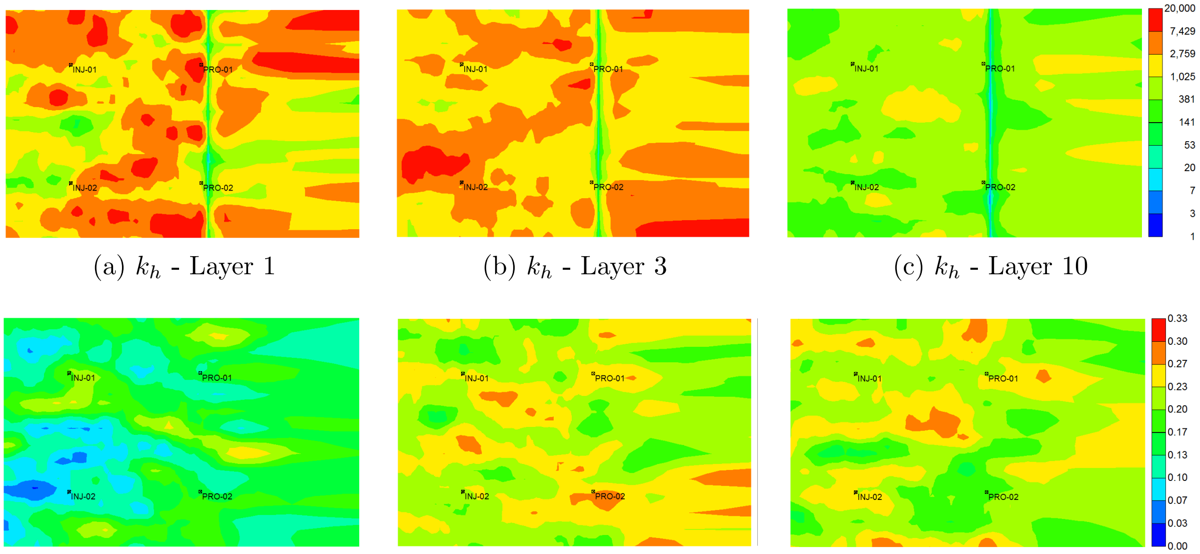
(e) ϕ - Layer 3

(f) ϕ - Layer 10

Figure 5.20: Horizontal permeability (mD, upper row) and porosity field (bottom row) for realization #1, Oseberg.

5.5.1 Deterministic optimization using realization #1

We firstly investigated deterministic optimization using realization #1. Fig. 5.22 shows the NPV versus number of simulations using StoSAG (blue line) and GPS (red line) respectively. From Fig. 5.22, StoSAG improved the NPV significantly, by approximately 12.44% within 500 simulation runs and the subsequent GPS method improved the NPV by another 7% within 200 more simulations. It is worthwhile to mention that due to the roughness of the objective function for the well placement optimization problem and the existence of the gradient estimation error in StoSAG, the NPV curve may consists of multiple plateaus where low-quality search directions are obtained. Moreover, when the StoSAG



(a) k_h - Layer 1 (b) k_h - Layer 3 (c) k_h - Layer 10

(d) ϕ - Layer 1 (e) ϕ - Layer 3 (f) ϕ - Layer 10

Figure 5.21: Horizontal permeability (mD, upper row) and porosity field (bottom row) for realization #8, Oseberg.

algorithm is terminated after 5 consecutive re-samples without obtaining an uphill direction (see Algorithm 2), it is often useful to apply a follow-up GPS step.

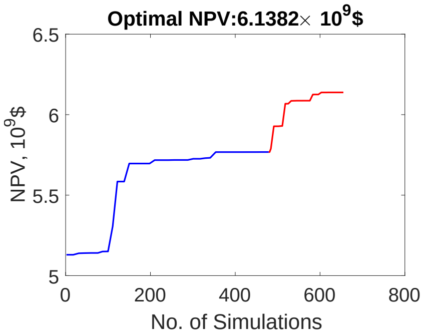
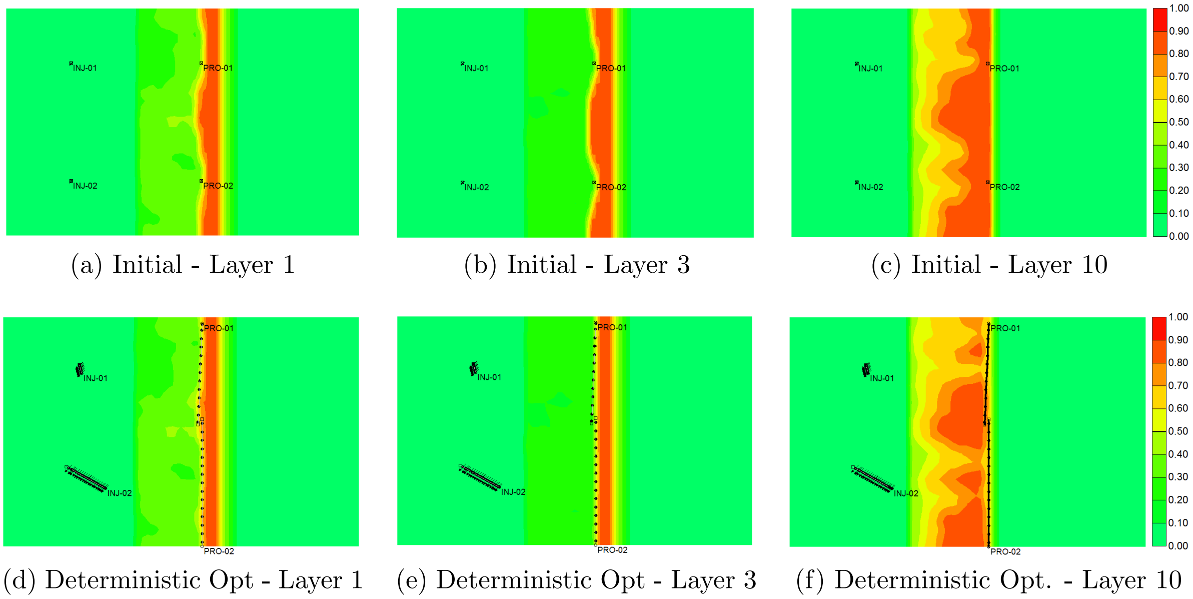


Figure 5.22: NPV versus number of simulations obtained for deterministic optimization of slanted well trajectories using realization #1, Oseberg. The blue line represents the StoSAG algorithm and the red line represents the GPS algorithm.

Fig. 5.23 shows the oil saturation after 3,900 days of production for the initial and optimal well placement using realization #1. In Figs. 5.23, the squares, circles and black dots represent the projections of the entry points, exit points and perforations of a well respectively; the black solid lines represent the projections of the perforations in the current layer. After deterministic optimization, (i) the two producers which are initially fully-penetrating vertical wells become horizontal in the 10th (bottom) layer along the tar mat because of the high and fairly uniform permeability with a lack of high permeable conduits, and (ii) both injectors still lay in the gas cap where INJ-01 perforates from layer 3 to layer 10 and INJ-02 perforates from layer 1 to layer 10. The minimum distance between these four wells are 511 ft which satisfies the minimum distance constraint of 500 ft. However, the trajectories indicate that effectively, we need only one long horizontal well whose length is almost equal to the reservoir width in the y direction. As we can see from Fig. 5.23, the gas flooding front is quite stable for the reservoir with a significant dip which indicates the geological uncertainty does not make a big influence in the flooding process. Similar to the channelized reservoir model, the minimum well spacing constraints are rarely violated during the optimization procedure since maximizing the NPV tends to avoid wells intersecting with each other.



(a) Initial - Layer 1 (b) Initial - Layer 3 (c) Initial - Layer 10

(d) Deterministic Opt - Layer 1 (e) Deterministic Opt - Layer 3 (f) Deterministic Opt. - Layer 10

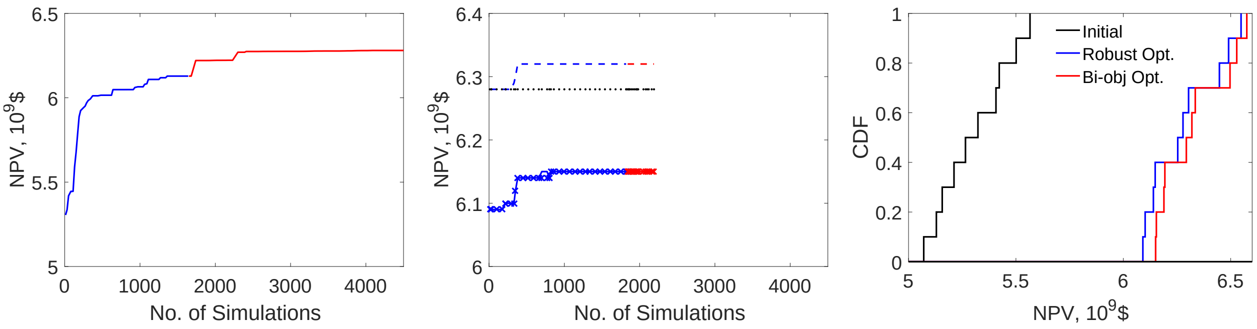
Figure 5.23: Oil saturation after 3,900 days of production obtained with initial and optimal well locations, deterministic optimization using realization #1, Oseberg.

5.5.2 Bi-objective optimization using 10 realizations

Fig. 5.24 shows a summary of the bi-objective well trajectory optimization where the average NPV and the minimum NPV of 10 realizations are maximized, subject to minimum well spacing constraints using StoSAG followed by GPS. In primary optimization (robust optimization), the average NPV over 10 realizations, J_E, is improved from \$5.31 × 10^9 to

\$6.28 × 10^9, by approximately 18%. The primary optimization is terminated after 4,500 simulations where GPS converged with the fixed mesh size. A secondary optimization is then carried out where the minimum NPV is maximized subject to an extra constraint that the average NPV is no less than 99% of the optimal NPV obtained in the primary optimization, or equivalently \$6.22 × 10^9. $N_{pert} = 10$ perturbations are used to estimate $d_{\hat{J}_R}$ and one perturbation per realization is used to estimate $\boldsymbol{d}_{c_J}$ in the secondary optimization, see the discussion of Algorithm 12. Fig. 5.24(b) shows the changes in the augmented Lagrangian Function (L_A), the minimum NPV (J_R) and the average NPV (J_E) with respect to the number of simulation runs. From Fig. 5.24(b), we see that in this example the minimum NPV is only increased from \$6.09 × 10^9 to \$6.14 × 10^9 (i.e., by approximately 0.8%) and the average NPV is also increased from \$6.28 × 10^6 to \$6.32 × 10^9 (i.e., by approximately 0.6%). Fig. 5.25 shows the indices of the model with lowest NPV value and the number of violated constraints during the secondary optimization process. Again, the secondary optimization rarely suffers from the non-differentiable issue and the risk model changes from realization #2 to #1. Fig. 5.14(c) shows the cumulative distribution function (cdf) of NPV's of the 10 realizations obtained with the well placement of the initial guess and the optimal solutions of the bi-objective optimization. From Fig. 5.14(c), both robust optimization and the secondary optimization shifted the cdf of NPV to the right. It is worthwhile to mention that after bi-objective optimization, the realization with the lowest NPV is realization #1 whose NPV is \$6.14 × 10^9, which is equal to the optimal NPV obtained for the deterministic optimization of realization #1.

Fig. 5.26 shows the oil saturation after 3,900 days of production for realization #1 (upper row) and #7 (bottom row) obtained with the initial guess and optimal solutions after robust optimization and secondary optimization. After robust optimization, (i) PRO-01 perforates both layer 9 and layer 10 and PRO-02 perforates only layer 10, and (ii) INJ-01 perforates from layer 2 to layer 4 and INJ-02 perforates from layer 2 to layer 10. After bi-objective optimization, (i) both producers perforate only layer 10, and (ii) INJ-01 perforates from layer 3 and layer 4 and INJ-02 perforates layer 3 to layer 10. Realization #7 has the highest NPV among all 10 realizations after both robust optimization and bi-objective optimization. Considering the locations of the gas injectors do not affect the reservoir development much, the major change of the optimal well trajectories is that, PRO-01, which penetrates layers 9 and 10 after robust optimization, now only penetrates layer 10.



(a) Robust Opt.

(b) Secondary Opt.

(c) CDF's

Figure 5.24: Bi-objective well placement optimization using 10 realizations, Oseberg reservoir. (a) Average NPV over 10 realizations versus number of simulation runs in primary optimization using StoSAG (blue) and GPS (red); (b) the augmented Lagrangian function, L_A, minimum NPV, J_R, and average NPV, J_E, versus number of simulation runs in secondary optimization using StoSAG (blue) and GPS (red) where L_A is represented by marker x, J_R is represented by the solid line, J_E is represented by the dashed line and $J_E(\boldsymbol{v}_E^*)$ is represented by the dotted line; (c) Cdf's of NPV for the initial well placements and the optimal well placement after primary and secondary optimization.

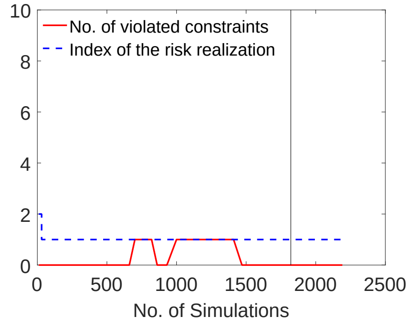
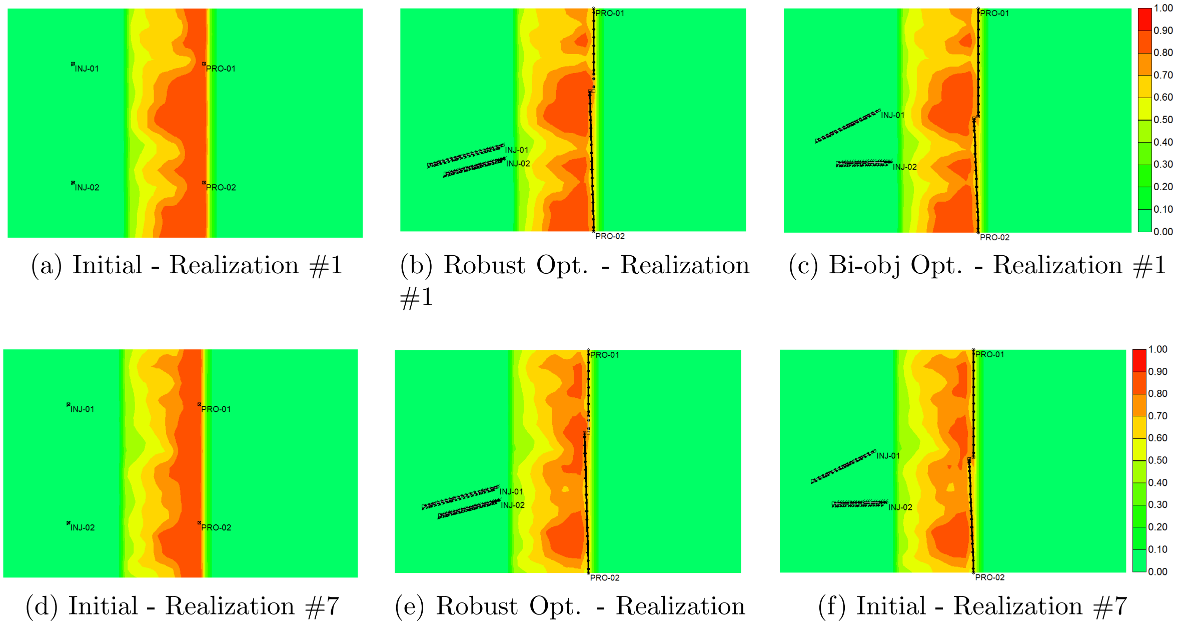


Figure 5.25: Number of violated constraints and indicies of the risk realization during secondary optimization of the bi-objective optimization of slanted well trajectories, Oseberg.



(a) Initial - Realization #1

(b) Robust Opt. - Realization #1

(c) Bi-obj Opt. - Realization #1

(d) Initial - Realization #7

(e) Robust Opt. - Realization #7

(f) Initial - Realization #7

Figure 5.26: Oil saturation in layer 10 after 3,900 days of production obtained with initial well locations, and optimal well locations after robust optimization and bi-objective optimization, for realization #1 and #7, Oseberg.

5.6 Summary and Discussion

In this chapter, we proposed parameterizations for both slanted wells and multi-segmented wells, and investigated the applicability of StoSAG and GPS for deterministic and robust well placement optimization considering the minimum well spacing constraints and the well length constraints. The augmented Lagrangian method is adopted to handle the nonlinear constraints. The necessity of using complicated well trajectory is also investigated based on a box-shaped channelized reservoir. When geological uncertainty exists, StoSAG and GPS are also coupled with lexicographic method to maximize the average NPV and maximize the minimum NPV of the ensemble of realizations representing the uncertainty. An efficient implementation of the StoSAG is also developed to reduce the computational cost of the secondary optimization. The effectiveness of the bi-objective optimization is validated on one water-flooding example, the channelized reservoir, and one gas-flooding example, the Oseberg example.

CHAPTER 6

DISCUSSION AND CONCLUSIONS

In this work, we developed and applied a hybrid algorithm which combines the Genetic Algorithm (GA), stochastic simplex approximate gradient (StoSAG), and General Pattern Search (GPS) to address three problems of interest in field development optimization:

- (i) selecting wells from a given set of potential paths and determining their types (injectors or producers), drilling order and locations,

- (ii) determining the well status (drill or not-to-drill), types (injectors or producers) and locations for both deterministic and robust optimization,

- (iii) and considering bi-objective well placement optimization using a complicated well trajectory parameterization.

In virtually all cases, well control optimization is done subsequent to the optimization of the other design variables.

For the first problem, this study revealed that (i) GA can effectively solve the problem of selecting a fixed number of wells from a given set of potential locations, moreover, for this problem GA gives higher optimal NPV compared to a gradient-based method by [25] and a StoSAG method using priority parameterization proposed by [44]; (ii) sequential optimization where the well locations and types are optimized in the first stage followed by a second stage of drilling order optimization gives a value of NPV that is comparable to or higher than the NPV generated with simultaneous optimization of well locations, types and drilling order; (iii) the optimization of well locations, types and drilling order with fixed BHP's followed by the well control optimization gives the highest optimal NPV if we operate injectors at their maximum BHP and producers at their minimum BHP when optimizing the well locations, types and drilling order.

For the second problem considered, we developed an iterative sequential algorithm (ItSA) where we alternate optimizing well status and types using GA with pure binary encoding and optimizing locations of wells using StoSAG or GPS if a StoSAG step fails to improve the NPV, and a sequential algorithm (SA) where we optimize well status, types and locations using GA first and then optimize the locations of wells drilled using StoSAG followed by GPS. ItSA and SA result in comparable values of the optimal NPV, but of these two algorithms, ItSAG converges approximately 20% faster for deterministic optimization. The performance of both algorithms are compared with that of Genocop III and the results of deterministic optimization indicate that Genocop III costs similar amount of simulation runs as SA, but gives the worst optimal NPV (6% and 7% lower than ItSA and SA for the PUNQ model respectively, and 1.5% and 3.3% lower than ItSA and SA for the Brugge model respectively). The ItSA coupled with an adaptive subset model selection procedure can efficiently optimize the well status, types and locations given a large number of realizations of reservoir models. In robust optimization where the computation cost becomes expensive, results indicate that it is more computationally efficient to eliminate StoSAG from the hybrid optimization algorithm when estimating the optimal value of well status, type and location design variables, i.e., simply combine GA and GPS in the hybrid algorithm.

For the third problem considered, we proposed a parameterization for slanted and mult-segmented well trajectories and found that for box-shaped reservoirs, there is not a significant gain in NPV after optimization by parameterizing wells as multi-segmented wells as opposed to slanted wells and that well placement optimization using GPS mehtod can still improve the NPV significantly after using the StoSAG method. We also found that the lexicographic method combined with a modified StoSAG/GPS algorithm can significantly reduce the downside risk with negligible deterioration in the the expected (average) NPV that is obtained by only maximizing the average NPV of life-cycle production.

BIBLIOGRAPHY

[1] A. A. Awotunde. Inclusion of well schedule and project life in well placement optimization. In *SPE Nigeria Annual International Conference and Exhibition*. Society of Petroleum Engineers, 2014.

[2] A. A. Awotunde. On the joint optimization of well placement and control. In *SPE Saudi Arabia Section Technical Symposium and Exhibition*. Society of Petroleum Engineers, 2014.

[3] A. A. Awotunde. Generalized field-development optimization with well-control zonation. *Computational Geosciences*, 20(1):213–230, 2016.

[4] J. E. Baker. Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the second international conference on genetic algorithms*, volume 206, pages 14–21, 1987.

[5] W. Bangerth, H. Klie, M. Wheeler, P. Stoffa, and M. Sen. On optimization algorithms for the reservoir oil well placement problem. *Computational Geosciences*, 10(3):303–319, 2006.

[6] B. Beckner and X. Song. Field development planning using simulated annealing-optimal economic well scheduling and placement. In *SPE annual technical conference and exhibition*. Society of Petroleum Engineers, 1995.

[7] M. C. Bellout, D. E. Ciaurri, L. J. Durlofsky, B. Foss, and J. Kleppe. Joint optimization of oil well placement and controls. *Computational Geosciences*, 16(4):1061–1079, 2012.

[8] A. C. Bittencourt and R. N. Horne. Reservoir development and design optimization. In *SPE annual technical conference and exhibition*. Society of Petroleum Engineers, 1997.

[9] B. Chen and A. C. Reynolds. Ensemble-based optimization of the water-alternating-gas-injection process. *SPE Journal*, 2015.

[10] C. Chen. *Adjoint-gradient-based production optimization with the augmented Lagrangian method.* PhD. thesis, University of Tulsa, Tulsa, Oklahoma, USA, 2011.

[11] C. Chen, G. Li, and A. Reynolds. Robust constrained optimization of short and long-term npv for closed-loop reservoir management. spe 141314. *Proceeding SPE RSS, Houston, TX, USA, 21–23 February*, 2011.

[12] Y. Chen, D. S. Olive, and D. Zhang. Efficient ensemble-based closed-loop production optimization. In *Proceedings of the SPE Improved Oil Recovery Symposium*, number SPE 112873, 2008.

[13] Y. Chen and D. Oliver. Ensemble-based closed-loop applied to Brugge field. In *Proceedings of the SPE Reservoir Simulation Symposium*, number SPE 118926, 2009.

[14] D. E. Ciaurri, T. Mukerji, and L. J. Durlofsky. Derivative-free optimization for oil field operations. In *Computational Optimization and Applications in Engineering and Industry*, pages 19–55. Springer, 2011.

[15] A. R. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to derivative-free optimization*, volume 8. Siam, 2009.

[16] R. J. de Moraes, R.-M. Fonseca, M. A. Helici, A. W. Heemink, and J. D. Jansen. An efficient robust optimization workflow using multiscale simulation and stochastic gradients. *Journal of Petroleum Science and Engineering*, 172:247–258, 2019.

[17] S. Ding, R. Lu, Y. Xi, S. Wang, and Y. Wu. Well placement optimization using direct mapping of productivity potential and threshold value of productivity potential management strategy. *Computers & Chemical Engineering*, 121:327–337, 2019.

[18] Y. Ding. Optimization of well placement using evolutionary methods. In *Europec/EAGE Conference and Exhibition*. Society of Petroleum Engineers, 2008.

[19] S. T. Do and A. C. Reynolds. Theoretical connections between optimization algorithms based on an approximate gradient. *Computational Geosciences*, 17(6):959–973, 2013.

[20] L. Durlofsky, I. Aitokhuehi, V. Artus, B. Yeten, and K. Aziz. Optimization of advanced well type and performance. In *ECMOR IX-9th European Conference on the Mathematics of Oil Recovery*, 2004.

[21] A. A. Emerick, E. Silva, B. Messer, L. Almeida, D. Szwarcman, M. Pacheco, and M.Vellasco. Well placement optimization using a genetic algorithm with nonlinear constraints. In *Proceedings of the SPE Reservoir Simulation Symposium*, number SPE 118808, 2009.

[22] G. M. V. Essen, P. M. J. V. den Hof, and J. D. Jansen. Hierarchical long-term and short-term production optimization. *SPE Journal*, 16(1):191–199, March 2011.

[23] R. Fletcher, S. Leyffer, and P. L. Toint. A brief history of filter methods. *Preprint ANL/MCS-P1372-0906, Argonne National Laboratory, Mathematics and Computer Science Division*, page 36, 2006.

[24] R. R.-M. Fonseca, B. Chen, J. D. Jansen, and A. Reynolds. A stochastic simplex approximate gradient StoSAG for optimization under uncertainty. *International Journal for Numerical Methods in Engineering*, 109(13):1756–1776, 2017.

[25] F. Forouzanfar, W. E. Poquioma, and A. C. Reynolds. Simultaneous and sequential estimation of optimal placement and controls of wells using a covariance matrix adaptation algorithm. *SPE Journal*, 2015.

[26] F. Forouzanfar and A. Reynolds. Joint optimization of number of wells, well locations and controls using a gradient-based algorithm. *Chemical Engineering Research and Design*, 92(7):1315–1328, 2014.

[27] F. Forouzanfar and A. C. Reynolds. Well-placement optimization using a derivative-free method. *Journal of Petroleum Science and Engineering*, 109:96–116, 2013.

[28] F. Forouzanfar, E. D. Rossa, R. Russo, and A. C. Reynolds. Life-cycle production optimization of an oil field with an adjoint-based gradient approach. *Journal of Petroleum Science and Engineering (online first)*, 2013.

[29] D. E. Goldberg and R. Lingle. Alleles, loci, and the traveling salesman problem. In *Proceedings of an international conference on genetic algorithms and their applications*, volume 154, pages 154–159. Lawrence Erlbaum, Hillsdale, NJ, 1985.

[30] R. Hanea, P. Casanova, F. H. Wilschut, and R. Fonseca. Well trajectory optimization constrained to structural uncertainties. In *SPE Reservoir Simulation Conference*. Society of Petroleum Engineers, 2017.

[31] R. Hanea, R. Fonseca, C. Pettan, M. Iwajomo, K. Skjerve, L. Hustoft, A. Chitu, and F. Wilschut. Decision maturation using ensemble based robust optimization for field development planning. In *ECMOR XV-15th European Conference on the Mathematics of Oil Recovery*, 2016.

[32] T. Humphries and R. Haynes. Joint optimization of well placement and control for nonconventional well types. *Journal of Petroleum Science and Engineering*, 126:242–253, 2015.

[33] T. D. Humphries, R. D. Haynes, and L. A. James. Simultaneous and sequential approaches to joint optimization of well placement and control. *Computational Geosciences*, 18(3-4):433–448, 2014.

[34] O. J. Isebor, L. J. Durlofsky, and D. E. Ciaurri. A derivative-free methodology with local and global search for the constrained joint optimization of well locations and controls. *Computational Geosciences*, 18(3-4):463–482, 2014.

[35] O. J. Isebor, D. Echeverría Ciaurri, and L. J. Durlofsky. Generalized field-development optimization with derivative-free procedures. *SPE Journal*, 19(05):891–908, 2014.

[36] J. D. Jansen, D. R. Brouwer, G. Naevdal, and C. P. J. W. van Kruijsdijk. Closed-loop reservoir management. *First Break*, 23:43–48, 2005.

[37] M. Jesmani, B. Jafarpour, M. Bellout, R. Hanea, and B. Foss. Application of simultaneous perturbation stochastic approximation to well placement optimization under uncertainty. In *ECMOR XV-15th European Conference on the Mathematics of Oil Recovery*, 2016.

[38] Y. Jiang, H. Kautz, and B. Selman. Solving problems with hard and soft constraints using a stochastic algorithm for max-sat. In *1st International Joint Workshop on Artificial Intelligence and Operations Research*, page 20, 1995.

[39] A. Karim, M. Ghazali, A. Raub, and M. R. Bin. Optimizing development strategy and maximizing field economic recovery through simulation opportunity index. In *SPE Reservoir Characterisation and Simulation Conference and Exhibition*. Society of Petroleum Engineers, 2011.

[40] B. Kayode, M. R. Yaacob, and F. A. Abdullah. Connected reservoir regions map created from time-lapse pressure data shows similarity to other reservoir quality maps in a heterogeneous carbonate reservoir. In *International Petroleum Technology Conference*. International Petroleum Technology Conference, 2019.

[41] R. A. Khan and A. A. Awotunde. Determination of vertical/horizontal well type from generalized field development optimization. *Journal of Petroleum Science and Engineering*, 162:652–665, 2018.

[42] J. Kraaijevanger, P. Egberts, J. Valstar, and H. Buurman. Optimal waterflood design using the adjoint method. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 2007.

[43] J. Lee, C. Park, J. M. Kang, and C. K. Jeong. Horizontal well design incorporated with interwell interference, drilling location, and trajectory for the recovery optimization. In *SPE/EAGE Reservoir Characterization & Simulation Conference*, 2009.

[44] O. Leeuwenburgh, A. Chitu, R. Nair, P. Egberts, L. Ghazaryan, T. Feng, and L. Hustoft. Ensemble-based methods for well drilling sequence and time optimization under uncertainty. In *ECMOR XV-15th European Conference on the Mathematics of Oil Recovery*, 2016.

[45] O. Leeuwenburgh, P. J. Egberts, and O. A. Abbink. Ensemble methods for reservoir life-cycle optimization and well placement. In *SPE/DGS Saudi Arabia Section Technical Symposium and Exhibition*. Society of Petroleum Engineers, 2010.

[46] L. Li and B. Jafarpour. A variable-control well placement optimization for improved reservoir development. *Computational Geosciences*, 16(4):871–889, 2012.

[47] L. Li, B. Jafarpour, and M. R. Mohammad-Khaninezhad. A simultaneous perturbation stochastic approximation algorithm for coupled well placement and control optimization under geologic uncertainty. *Computational Geosciences*, 17(1):167–188, 2013.

[48] X. Liu. *Multiobjective optimization of a waterflood with gradient-based algorithms*. The University of Tulsa, 2016.

[49] X. Liu and A. C. Reynolds. Gradient-based multi-objective optimization with applications to waterflooding optimization. *Computational Geosciences*, pages 1–17, 2014.

[50] X. Liu and A. C. Reynolds. Multiobjective optimization for maximizing expectation and minimizing uncertainty or risk with application to optimal well control. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 2015.

[51] F. G. Lobo and D. E. Goldberg. The parameter-less genetic algorithm in practice. *Information Sciences*, 167(1-4):217–232, 2004.

[52] R. Lu, F. Forouzanfar, and A. Reynolds. Bi-objective optimization of well placement and controls using StoSAG. In *SPE Reservoir Simulation Conference*. Society of Petroleum Engineers, 2017.

[53] R. Lu, F. Forouzanfar, and A. C. Reynolds. An efficient adaptive algorithm for robust control optimization using StoSAG. *Journal of Petroleum Science and Engineering*, 159:314–330, 2017.

[54] R. Lu and A. C. Reynolds. Joint optimization of well locations, types, drilling order and controls given a set of potential drilling paths. In *SPE Reservoir Simulation Conference*. Society of Petroleum Engineers, 2019.

[55] I. Manual. Version x-2009.10. *Computer Modelling Group (CMG), Calgary, Canada*, 2009.

[56] C. Maschio, L. Nakajima, and D. J. Schiozer. Production strategy optimization using genetic algorithm and quality map. In *Europec/EAGE Conference and Exhibition*. Society of Petroleum Engineers, 2008.

[57] Z. Michalewicz and G. Nazhiyath. Genocop III: A co-evolutionary algorithm for numerical optimization problems with nonlinear constraints. In *Proceedings of 1995 IEEE International Conference on Evolutionary Computation*, volume 2, pages 647–651. IEEE, 1995.

[58] R. Mitchell and S. Miska. *Fundamentals of drilling engineering*. Society of Petroleum Engineers, 2011.

[59] G. Montes, P. Bartolome, and A. L. Udias. The use of genetic algorithms in well placement optimization. In *SPE Latin American and Caribbean petroleum engineering conference*. Society of Petroleum Engineers, 2001.

[60] A. N. Morales, H. Nasrabadi, and D. Zhu. A new modified genetic algorithm for well placement optimization under geological uncertainties. In *SPE EUROPEC/EAGE annual conference and exhibition*. Society of Petroleum Engineers, 2011.

[61] H. Mühlenbein and D. Schlierkamp-Voosen. Predictive models for the breeder genetic algorithm I. continuous parameter optimization. *Evolutionary computation*, 1(1):25–49, 1993.

[62] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 2006.

[63] D. F. Oliveira and A. C. Reynolds. Hierarchical multiscale methods for life-cycle-production optimization: A field case study. *SPE Journal*, 20(05):896–907, 2015.

[64] D. S. Oliver. Incorporation of transient pressure data into reservoir characterization. *In Situ*, 18(3):243–275, 1994.

[65] J. E. Onwunalu and L. J. Durlofsky. Development and application of a new well pattern optimization algorithm for optimizing large scale field development. In *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers, 2009.

[66] J. E. Onwunalu and L. J. Durlofsky. Application of a particle swarm optimization algorithm for determining optimum well location and type. *Computational Geosciences*, 14(1):183–198, 2010.

[67] D. Orvosh and L. Davis. Shall we repair? genetic algorithms combinatorial optimization and feasibility constraints. In *Proceedings of the 5th International Conference on Genetic Algorithms*, page 650. Morgan Kaufmann Publishers Inc., 1993.

[68] D. Peaceman. Representation of a horizontal well in numerical reservoir simulation. *SPE Advanced Technology Series*, 1(01):7–16, 1993.

[69] D. W. Peaceman. Interpretation of well-block pressures in numerical reservoir simulation (includes associated paper 6988). *Society of Petroleum Engineers Journal*, 18(03):183–194, 1978.

[70] D. W. Peaceman. Interpretation of well-block pressures in numerical reservoir simulation with nonsquare grid blocks and anisotropic permeability. *Society of Petroleum Engineers Journal*, 23(03):531–543, 1983.

[71] L. Peters, R. Arts, G. Brouwer, C. Geel, S. Cullick, R. Lorentzen, Y. Chen, K. Dunlop, F. Vossepoel, R. Xu, P. Sarma, A. Alhuthali, and A. Reynolds. Results of the Brugge benchmark study for flooding optimisation and history matching. *SPE Reservoir Evaluation & Engineering*, 13(3):391–405, 2010.

[72] D. D. Salam, I. Gunardi, and A. Yasutra. Production optimization strategy using hybrid genetic algorithm. In *Abu Dhabi International Petroleum Exhibition and Conference*. Society of Petroleum Engineers, 2015.

[73] P. Sarma, K. Aziz, and L. J. Durlofsky. Implementation of adjoint solution for optimal control of smart wells. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 2005.

[74] P. Sarma and W. H. Chen. Efficient well placement optimization with gradient-based algorithms and adjoint models. In *Intelligent energy conference and exhibition*. Society of Petroleum Engineers, 2008.

[75] M. Schoenauer and Z. Michalewicz. Evolutionary computation at the edge of feasibility. In *International Conference on Parallel Problem Solving from Nature*, pages 245–254. Springer, 1996.

[76] M. G. Shirangi and L. J. Durlofsky. Closed-loop field development optimization under uncertainty. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 2015.

[77] N. Sibaweihi, A. A. Awotunde, A. S. Sultan, and H. Y. Al-Yousef. Sensitivity studies and stochastic optimization of CO_2 foam flooding. *Computational Geosciences*, 19(1):31–47, 2015.

[78] D. Simon. *Evolutionary optimization algorithms*. John Wiley & Sons, 2013.

[79] V. Torczon. On the convergence of pattern search algorithms. *SIAM Journal on optimization*, 7(1):1–25, 1997.

[80] R. R. Torrado, G. D. Paola, A. F. Perez, A. A. Fuenmayor, M. S. De Azevedo, and S. Embid. Optimize a WAG field development plan, use case of carbonate ultra-deep water reservoir. In *Proceedings of EUROPEC 2015, Madrid, Spain, 01-04 June*, number SPE 174344, 2015.

[81] S. Vlemmix, G. J. Joosten, R. Brouwer, and J. D. Jansen. Adjoint-based well trajectory optimization. In *EUROPEC/EAGE Conference and Exhibition*. Society of Petroleum Engineers, 2009.

[82] O. Volkov and M. Bellout. Gradient-based constrained well placement optimization. *Journal of Petroleum Science and Engineering*, 2018.

[83] C. Wang, G. Li, and A. C. Reynolds. Optimal well placement for production optimization. *SPE Journal*, 14(3):506–523, 2009.

[84] X. Wang, R. D. Haynes, and Q. Feng. Well control optimization using derivative-free algorithms and a multiscale approach. *arXiv preprint arXiv:1509.04693*, 2015.

[85] C. Yang, C. Card, L. X. Nghiem, and E. Fedutenko. Robust optimization of SAGD operations under geological uncertainties. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 2011.

[86] B. Yeten, L. J. Durlofsky, and K. Aziz. Optimization of nonconventional well type, location and trajectory. In *SPE annual technical conference and exhibition*. Society of Petroleum Engineers, 2002.

[87] M. Zandvliet, M. Handels, G. van Essen, R. Brouwer, and J. D. Jansen. Adjoint-based well placement optimization under production constraints. *SPE Journal*, 13(04):392–399, 2008.

[88] F. Zhang. *Automatic History Matching of Production Data for Large Scale Problems*. Ph.D. thesis, The University of Tulsa, Tulsa, Oklahoma, 2002.

[89] K. Zhang, G. Li, A. C. Reynolds, J. Yao, and L. Zhang. Optimal well placement using an adjoint gradient. *Journal of Petroleum Science and Engineering*, 73(3-4):220–226, 2010.

[90] Y. Zhang, R. Lu, F. Forouzanfar, and A. C. Reynolds. Well placement and control optimization for WAG/SAG processes using ensemble-based method. *Computers & Chemical Engineering*, 101:193–209, 2017.

APPENDIX A

MINIMUM WELL SPACING CONSTRAINT

The enforcement of the minimum well spacing constraint requires the calculation of the distance between two line segments in 3D. The procedure for performing this calculation is presented here. Consider two line segments, namely AB and CD, in 3D space. The coordinates of points A, B, C, D are denoted by $\boldsymbol{x}_A, \boldsymbol{x}_B, \boldsymbol{x}_C$ and $\boldsymbol{x}_D$, respectively.

Suppose that P is a point on the line through A and B, AB, then the coordinates $\boldsymbol{x}_P$ of point P can be written as

$$\boldsymbol{x}_P = \boldsymbol{x}_A + s_P \left(\boldsymbol{x}_B - \boldsymbol{x}_A\right). \tag{A.1}$$

If $0 \leqslant s_P \leqslant 1$, point P is on the line segment AB. If $s_P < 0$ or $s_P > 1$, point P is on the extended line of the line segment AB. Similarly, if Q is a point on the line through C and D, CD, then the coordinates $\boldsymbol{x}_Q$ of point Q can be written as

$$\boldsymbol{x}_Q = \boldsymbol{x}_C + s_Q \left(\boldsymbol{x}_D - \boldsymbol{x}_C\right). \tag{A.2}$$

Note Q is a point on the line segment CD if and only if $0 \leq s_Q \leq 1$. The distance between the points P and Q is

$$|PQ| = \|\boldsymbol{x}_P - \boldsymbol{x}_Q\|_2, \tag{A.3}$$

and the distance between the two lines is defined as the minimum value of $|PQ|$. Let

$$f\left(s_P, s_Q\right) = \|\boldsymbol{x}_P - \boldsymbol{x}_Q\|_2^2. \tag{A.4}$$

If $f\left(s_P, s_Q\right)$ is equal to the distance between the two lines AB and CD, then s_P and s_Q must satisfy the following two equations,

$$\begin{cases} \frac{\partial f\left(s_P, s_Q\right)}{\partial s_P} = 0, \\ \frac{\partial f\left(s_P, s_Q\right)}{\partial s_Q} = 0. \end{cases} \tag{A.5}$$

If the solution obtained for Eq. A.5 satisfies $0 \leqslant s_P \leqslant 1$ and $0 \leqslant s_Q \leqslant 1$, then the point P is on the line segment AB and the point Q is on the line segment CD, and the distance (calculated with Eq. A.3) between line AB and CD is the same as the distance between line segments AB and CD, i.e., shortest distance between a pair of points, P and Q with P on the line segment AB and Q on the line segment CD. Otherwise, we need to calculate the distance between the point A and the line segment CD, the distance between the point B and the line segment CD, the distance between the point C and the line segment AB and the distance between point D and the line segment AB. Then the distance between line segment AB and line segment CD is the smallest one of these four distances.

In the following, we briefly explain the computation of the distance of the point, P, to the line segment, AB, in 3D. The following equation defines the plane that passes through the point P and is perpendicular to the line AB.

$$(\boldsymbol{x}_B - \boldsymbol{x}_A)^T (\boldsymbol{x} - \boldsymbol{x}_P) = 0. \tag{A.6}$$

Let Q be the intersection point of the plane and line AB. It is obvious that PQ is perpendicular to AB, and the point Q is the perpendicular foot whose coordinates are given by

$$\boldsymbol{x}_Q = \boldsymbol{x}_A + s_Q \left(\boldsymbol{x}_B - \boldsymbol{x}_A\right). \tag{A.7}$$

Substituting Eq. A.7 into Eq. A.6 and solving the resulting equation for s_Q gives

$$s_Q = \frac{(\boldsymbol{x}_A - \boldsymbol{x}_B)^T (\boldsymbol{x}_A - \boldsymbol{x}_P)}{\|\boldsymbol{x}_A - \boldsymbol{x}_B\|_2^2}. \tag{A.8}$$

If $0 \leqslant s_Q \leqslant 1$, the point Q is on the line segment AB, and the length of line segment PQ is by definition the distance between the point P and the line segment AB. If $s_Q < 0$, the point Q is on the extended line of line segment BA, and the distance between the point P and the line segment AB is the distance between the point P and point A. If $s_Q > 1$, the point Q is on the extended line of line segment AB, and the distance between the point P and the line segment AB is the distance between the point P and point B.

APPENDIX B

MULTI-SEGMENTED WELL TRAJECTORY CALCULATION

In Chapter 5 of this dissertation, the trajectory of a multi-segmented well (e.g., well iw) is parameterized using Eq. 5.4, which is recited below

$$\boldsymbol{X}_{iw} = [x_0^{iw}, y_0^{iw}, z_0^{iw}, z_1^{iw}, l_1^{iw}, \theta_1^{iw}, z_2^{iw}, l_2^{iw}, \delta\theta_2^{iw}, \cdots, z_{N_{seg}}^{iw}, l_{N_{seg}}^{iw}, \delta\theta_{N_{seg}}^{iw}].$$

To obtain a multi-segmented well trajectory given an estimate of the variables in Eq. 5.4, we need to calculate the coordinates of the end points of each well segment. The calculation of the segment end points of the multi-segmented well iw with N_{seg} segments in Cartesian coordinate system is presented in Algorithm 13.

Algorithm 13 Calculation of the segment end points of the multi-segmented well *iw*

```
• For $i = 1 : N_{seg}$
    – Calculate the azimuth angle of the $i$th segment, $\theta_i^{iw}$. If $i > 1$, $\theta_i^{iw} = \theta_{i-1}^{iw} + \delta\theta_i^{iw}$.
    – Calculate $\delta z_i^{iw} = z_i^{iw} - z_{i-1}^{iw}$.
    – Calculate the horizontal departure of the $i$th segment using
          $HD_i^{iw} = \sqrt{(l_i^{iw})^2 - (\delta z_i^{iw})^2}$.
    – Calculate the $(x, y)$ coordinates of end point of $i$th segment using
          $x_i^{iw} = x_{i-1}^{iw} + HD_i^{iw} \cos(\theta_i^{iw})$,
          $y_i^{iw} = y_{i-1}^{iw} + HD_i^{iw} \sin(\theta_i^{iw})$.
    – Set $i = i + 1$.
• EndFor
```

To apply the optimal estimate of Eq. 5.4, the multi-segmented trajectory obtained with Algorithm 13 has to be smoothed in order to be drillable. In this dissertation, each pair of consecutive well segments of well *iw* is smoothed as a circular arc located in the plane of these two segments; see Fig. 5.2 for an illustration. The smoothing procedure of well *iw* is presented in Algorithm 14.

Algorithm 14 Smoothing of the multi-segmented well *iw*

```
• For $i = 1 : N_{seg} - 1$
    – Calculate the coordinates of the end points of the $i$th and $(i+1)$th segments.
    – Calculate the coordinates of the starting point, B, and the end point, D, of the
      smoothing arc, $\widehat{BD}$.
      If ($l_i^{iw} > l_{i+1}^{iw}$)
        ∗ $s = \frac{l_{i+1}^{iw}}{2l_i^{iw}}$,
        ∗ $x_B^{iw} = x_i^{iw} - s(x_i^{iw} - x_{i-1}^{iw})$, $y_B^{iw} = y_i^{iw} - s(y_i^{iw} - y_{i-1}^{iw})$, $z_B^{iw} = z_i^{iw} - s(z_i^{iw} - z_{i-1}^{iw})$,
        ∗ $x_D^{iw} = \frac{x_i^{iw}+x_{i+1}^{iw}}{2}$, $y_D^{iw} = \frac{y_i^{iw}+y_{i+1}^{iw}}{2}$, $z_D^{iw} = \frac{z_i^{iw}+z_{i+1}^{iw}}{2}$,
      else
        ∗ $s = \frac{l_i^{iw}}{2l_{i+1}^{iw}}$,
        ∗ $x_B^{iw} = \frac{x_i^{iw}+x_{i-1}^{iw}}{2}$, $y_B^{iw} = \frac{y_i^{iw}+y_{i-1}^{iw}}{2}$, $z_B^{iw} = \frac{z_i^{iw}+z_{i-1}^{iw}}{2}$,
        ∗ $x_D^{iw} = x_i^{iw} + s(x_{i+1}^{iw} - x_i^{iw})$, $y_D^{iw} = y_i^{iw} + s(y_{i+1}^{iw} - y_i^{iw})$, $z_D^{iw} = z_i^{iw} + s(z_{i+1}^{iw} - z_i^{iw})$.
    – Calculate the inclination angle of the $i$th and $(i+1)$th segments,
        ∗ $\varphi_i^{iw} = 90 - \arcsin\left(\frac{z_i^{iw} - z_{i-1}^{iw}}{l_i^{iw}}\right)$, $\varphi_{i+1}^{iw} = 90 - \arcsin\left(\frac{z_{i+1}^{iw} - z_i^{iw}}{l_{i+1}^{iw}}\right)$, $\delta\varphi_i^{iw} = \varphi_{i+1}^{iw} - \varphi_i^{iw}$.
    – Calculate the dogleg angle $\beta_i^{iw}$ and the radius of curvature $R_i^{iw}$ of the arc $\widehat{BD}$,
        ∗ $\beta_i^{iw} = 2\text{arcsin}\sqrt{\left(\sin\frac{\delta\varphi_i^{iw}}{2}\right)^2 + \sin\varphi_i^{iw}\sin(\varphi_i^{iw} + \delta\varphi_i^{iw})\left(\sin\frac{\delta\theta_i^{iw}}{2}\right)^2}$,
        ∗ $R_i^{iw} = \frac{\min(l_i^{iw},\ l_{i+1}^{iw})}{2\tan\frac{\beta_i^{iw}}{2}}$.
    – Calculate the dogleg severity,
        ∗ $DLS_i^{iw} = \frac{100}{R_i^{iw}}\frac{180}{\pi}$.
    – Set $i = i + 1$.
• EndFor
```

APPENDIX C

EQUIVALENT WELL INDEX CALCULATION

In this dissertation, well trajectories can be parameterized as vertical, slanted or multi-segmented. Each well can perforate many gridblocks. In CMG (the simulator used in this work to simulate the reservoir dynamics) as well as other commercial simulators, one can only define a perforation within one gridblock as a single line segment. Given the Cartesian coordinates of the entry point and exit point to a gridblock, the perforation length, the completion factor and the perforated gridblock indices, CMG can calculate an accurate productivity/injectivity index using a Peaceman type well model ([69, 70, 68]). In this section, we firstly provide a routine which identifies the grid cells crossed by each well and calculates the required information of each perforated cell for CMG based on an estimate of the design variables given in Eq. 5.1 or 5.4.

Note that for a multi-segmented well, there may be two segments of perforations in the gridblock containing the end point of a well segment (except the last well segment), which can not be specified in CMG. Thus the calculation of a one-segmented pseudo-perforation with equal productivity/injectivity index is required and we provide the calculation based on CMG [55] in Appendix C.2. In this dissertation, this routine is only developed for reservoirs with (transformed) Cartesian grids.

C.1 Calculation of the Inputs for CMG

If well iw is a slanted well, we can obtain the coordinates of the two end points of well iw directly from an estimate of the variables in Eq. 5.1. If well iw is a multi-segmented well, we can obtain the coordinates of the end points of the N_{seg} well segments of well iw

from an estimate of the variables in Eq. 5.4 using Appendix B. Consider a set of Cartesian grids given in Fig. C.1, we provide a routine which identifies the grid cells crossed by well iw and calculates the Cartesian coordinates of the entry and exit points, the perforation length and the completion factor of each perforation of well iw in Algorithm 15. Since a slanted well can be considered as a multi-segmented well with a single well segment, this routine is general for both slanted wells and multi-segmented wells. However, we emphasize that this routine is only developed for Cartesian grids illustrated in Fig. C.1.

In Algorithm 15, we denote the number of gridblocks of the reservoir along (x, y, z) directions respectively as N_{GridI}, N_{GridJ} and N_{GridK}. We define $\boldsymbol{x}^{yz}$ as a $(N_{\text{GridI}}+1)$-dimensional array where $x^{yz}_{i_g}$ is set equal to the x coordinate of the left surface of gridblock $(i_g, 1, 1)$, for $i_g = 1, 2, \cdots, N_{\text{GridI}}$ and $x^{yz}_{N_{\text{GridI}}+1}$ is set equal to the x coordinate of the right surface of gridblock $(N_{\text{GridI}}, 1, 1)$. Similarly, we define $\boldsymbol{y}^{xz}$ as a $(N_{\text{GridJ}}+1)$-dimensional array where $y^{xz}_{i_g}$ is set equal to the y coordinate of the back surface of gridblock $(1, i_g, 1)$ for $i_g = 1, 2, \cdots, N_{\text{GridJ}}$ and $y^{xz}_{N_{\text{GridJ}}+1}$ is set equal to the y coordinate of the front surface of gridblock $(1, N_{\text{GridJ}}, 1)$. We also define $\boldsymbol{z}^{xy}$ as a $(N_{\text{GridK}}+1)$-dimensional array where $z^{xy}_{i_g}$ is set equal to the z coordinate of the top surface of gridblock $(1, 1, i_g)$ for $i_g = 1, 2, \cdots, N_{\text{GridK}}$ and $z^{xy}_{N_{\text{GridK}}+1}$ is set equal to the z coordinate of the bottom surface of the $(1, 1, N_{\text{GridK}})$ gridblock. We denote the coordinates of the entry and exit points of the i_pth perforation of the ith segment of well iw respectively as $(x^{iw,0}_{i,i_p}, y^{iw,0}_{i,i_p}, z^{iw,0}_{i,i_p})$ and $(x^{iw,1}_{i,i_p}, y^{iw,1}_{i,i_p}, z^{iw,1}_{i,i_p})$. We also denote its perforation length and the completion factor respectively as l^{iw}_{i,i_p} and f^{iw}_{i,i_p}.

In the following, we describe the procedure to determine the (i, j, k) indicies of a perforated gridblock given the coordinates of a point. Take the point $(x^{iw}_i, y^{iw}_i, z^{iw}_i)$ as an example. If $x^{iw}_i < x^{yz}_{N_{\text{GridI}}+1}$, then the gridblock index along x-direction, GI^{iw}_i, is set equal to $\{i_g : x^{yz}_{i_g} \leq x^{iw}_i < x^{yz}_{i_g+1}\}^{N_{\text{GridI}}}_{i_g=1}$; otherwise, set $GI^{iw}_i = N_{\text{GridI}}$. If $y^{iw}_i < y^{xz}_{N_{\text{GridJ}}+1}$, then the gridblock index along y-direction, GJ^{iw}_i, is set equal to $\{i_g : y^{xz}_{i_g} \leq y^{iw}_i < y^{xz}_{i_g+1}\}^{N_{\text{GridJ}}}_{i_g=1}$; otherwise, set $GJ^{iw}_i = N_{\text{GridJ}}$. If $z^{iw}_i < z^{xy}_{N_{\text{GridK}}+1}$, then the gridblock index along z-direction, GK^{iw}_i, is set equal to $\{i_g : z^{xy}_{i_g} \leq z^{iw}_i < z^{xy}_{i_g+1}\}^{N_{\text{GridK}}}_{i_g=1}$; otherwise, set $GK^{iw}_i = N_{\text{GridK}}$.

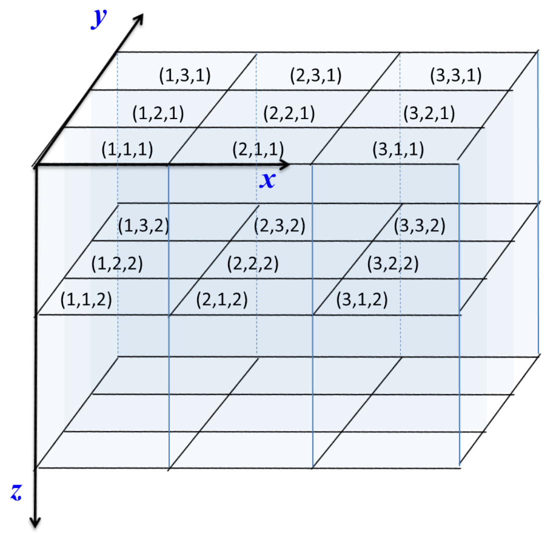


Figure C.1: Numbering of the Cartesian grids. Numbers in parenthesis on the top surface of each gridblock indicate the (i, j, k) gridblock indicies.

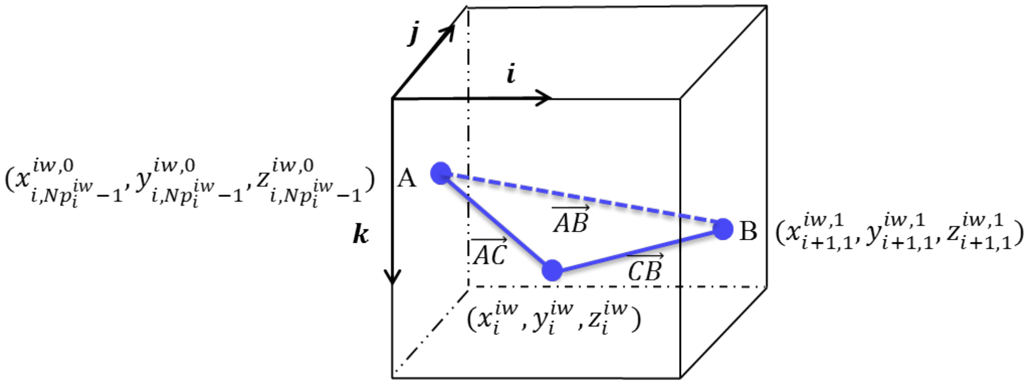
Algorithm 15 Calculation of the input parameters into CMG given the Cartesian coordinates of the segment end points of well iw

```
• For $i = 1 : N_{seg}$
    – Calculate the Cartesian coordinates of the two end points of segment $i$, i.e., $(x_{i-1}^{iw}, y_{i-1}^{iw}, z_{i-1}^{iw})$ and $(x_i^{iw}, y_i^{iw}, z_i^{iw})$, and their gridblock indcies, i.e., $(GI_{i-1}^{iw}, GJ_{i-1}^{iw}, GK_{i-1}^{iw})$ and $(GI_i^{iw}, GJ_i^{iw}, GK_i^{iw})$.
    – Define a $N_s^{trial}$-dimensional vector $\boldsymbol{s}$ where $N_s^{trial} = |GI_{i-1}^{iw} - GI_i^{iw}| + |GJ_{i-1}^{iw} - GJ_i^{iw}| + |GK_{i-1}^{iw} - GK_i^{iw}| + 3$. Set $m = 1$.
    – For $i_g = \min(GI_{i-1}^{iw}, GI_i^{iw}) \ : \ \max(GI_{i-1}^{iw}, GI_i^{iw})$
        ∗ If $x_i^{iw} \neq x_{i-1}^{iw}$, set $s_m = \frac{x_{i_g}^{yz} - x_{i-1}^{iw}}{x_i^{iw} - x_{i-1}^{iw}}$ and $m = m + 1$.
    – EndFor
    – For $j_g = \min(GJ_{i-1}^{iw}, GJ_i^{iw}) \ : \ \max(GJ_{i-1}^{iw}, GJ_i^{iw})$
        ∗ If $y_i^{iw} \neq y_{i-1}^{iw}$, set $s_m = \frac{y_{j_g}^{xz} - y_{i-1}^{iw}}{y_i^{iw} - y_{i-1}^{iw}}$ and $m = m + 1$.
    – EndFor
```

```
•
    – For $k_g = \min(GK_{i-1}^{iw}, GK_i^{iw})$ : $\max(GK_{i-1}^{iw}, GK_i^{iw})$
        ∗ If $z_i^{iw} \neq z_{i-1}^{iw}$, set $s_m = \frac{z_{k_g}^{xy} - z_{i-1}^{iw}}{z_i^{iw} - z_{i-1}^{iw}}$ and $m = m + 1$.
    – EndFor
    – Remove $s_i, i = m, m+1, \cdots, N_s^{trial}$ from $\boldsymbol{s}$ and remove repeated elements in $\boldsymbol{s}$.
      Define $A = \{s_i : s_i \leq 0 \text{ and } s_i \in \boldsymbol{s}\}$, $B = \{s_i : 0 < s_i < 1 \text{ and } s_i \in \boldsymbol{s}\}$ and
      $C = \{s_i : s_i \geq 1 \text{ and } s_i \in \boldsymbol{s}\}$. Denote the dimensionality of $B$ as $N_B$ and set
      $Np_i^{iw} = N_B + 2$.
    – Define a $Np_i^{iw}$-dimensional vector $\boldsymbol{c}^p$ where $c_1^p = \max\{A\}$, $c_{i+1}^p = B_i$, $i = 1, 2, \cdots, N_B$ and $c_{Np_i^{iw}}^p = \max\{C\}$. Sort $\boldsymbol{c}^p$ in ascending order.
    – For $i_p = 1$: $Np_i^{iw} - 1$
        ∗ Calculate the coordinates of the entry and exit points of the $i_p$th perforation
          using
              $(x_{i,i_p}^{iw,0}, y_{i,i_p}^{iw,0}, z_{i,i_p}^{iw,0}) = (x_{i-1}^{iw}, y_{i-1}^{iw}, z_{i-1}^{iw}) + c_{i_p}^p * \big\{(x_i^{iw}, y_i^{iw}, z_i^{iw}) - (x_{i-1}^{iw}, y_{i-1}^{iw}, z_{i-1}^{iw})\big\}$
              $(x_{i,i_p}^{iw,1}, y_{i,i_p}^{iw,1}, z_{i,i_p}^{iw,1}) = (x_{i-1}^{iw}, y_{i-1}^{iw}, z_{i-1}^{iw}) + c_{i_p+1}^p * \big\{(x_i^{iw}, y_i^{iw}, z_i^{iw}) - (x_{i-1}^{iw}, y_{i-1}^{iw}, z_{i-1}^{iw})\big\}$
        ∗ Calculate the perforation length using
              $l_{i,i_p}^{iw} = \Big(\min(c_{i_p+1}^p, 1) - \max(c_{i_p}^p, 0)\Big) * \sqrt{(x_i^{iw} - x_{i-1}^{iw})^2 + (y_i^{iw} - y_{i-1}^{iw})^2 + (z_i^{iw} - z_{i-1}^{iw})^2}$
        ∗ Set the completion factor $f_{i,i_p}^{iw} = 1$.
        ∗ Calculate the perforation gridblock indices based on the coordinates
          $(\frac{x_{i,i_p}^{iw,0} + x_{i,i_p}^{iw,1}}{2}, \frac{y_{i,i_p}^{iw,0} + y_{i,i_p}^{iw,1}}{2}, \frac{z_{i,i_p}^{iw,0} + z_{i,i_p}^{iw,1}}{2})$.
    – EndFor
    – If $i > 1$ and $c_{i,1}^p < 0$, re-calculate the required information for a pseudo-perforation
      in the gridblock of the first perforation of the $i$th well segment,
        ∗ reset the coordinates of the entry and exit points of the pseudo-perforation
          respectively as $(x_{i-1,Np_i^{iw}-1}^{iw,0}, y_{i-1,Np_i^{iw}-1}^{iw,0}, z_{i-1,Np_i^{iw}-1}^{iw,0})$ and $(x_{i,1}^{iw,1}, y_{i,1}^{iw,1}, z_{i,1}^{iw,1})$,
        ∗ re-calculate the length of the pseudo-perforation using
              $l_{i,i_p}^{iw} = \sqrt{(x_{i,1}^{iw,1} - x_{i-1,Np_i^{iw}-1}^{iw,0})^2 + (y_{i,1}^{iw,1} - y_{i-1,Np_i^{iw}-1}^{iw,0})^2 + (z_{i,1}^{iw,1} - z_{i-1,Np_i^{iw}-1}^{iw,0})^2}$,
        ∗ re-calculate the completion factor $f_{i,1}^{iw}$ of the pseudo-perforation using Ap-
          pendix C.2 which is discussed later.
• EndFor
```

C.2 Two-segmented Perforation within a Gridblock

When wells are parameterized as multi-segmented wells, it is common that two line segments of perforations exist in one simulation gird; see perforation $\overrightarrow{AC}$ and $\overrightarrow{CB}$ in Fig. C.2 as an example. Since in most of the commercial simulators, one can only define the perforation with one segment, a pseudo-perforation with equal injectivity/productivity is required. Thus, the calculation of the coordinates of the entry and exit point, the perforation length and the completion factor of this pseudo-perforation is presented here.



j
i
k
$(x^{iw,0}_{i,Np_i^{iw}-1}, y^{iw,0}_{i,Np_i^{iw}-1}, z^{iw,0}_{i,Np_i^{iw}-1})$
A
$\overrightarrow{AB}$
B
$(x^{iw,1}_{i+1,1}, y^{iw,1}_{i+1,1}, z^{iw,1}_{i+1,1})$
$\overrightarrow{AC}$
$\overrightarrow{CB}$
$(x_i^{iw}, y_i^{iw}, z_i^{iw})$

Figure C.2: Two-segmented perforation within one simulation gridblock.

Take Fig. C.2 as an example where point C is the end point of the ith well segment of well iw, point A is the entry point of the last perforation of the ith well segment and point B is the exit point of the first perforation of $(i+1)$th well segment. Denote perforation $\overrightarrow{AC} = (x_i^{iw}, y_i^{iw}, z_i^{iw}) - (x^{iw,0}_{i,Np_i^{iw}-1}, y^{iw,0}_{i,Np_i^{iw}-1}, z^{iw,0}_{i,Np_i^{iw}-1})$ and perforation $\overrightarrow{CB} = (x^{iw,1}_{i+1,1}, y^{iw,1}_{i+1,1}, z^{iw,1}_{i+1,1}) - (x_i^{iw}, y_i^{iw}, z_i^{iw})$. A pseudo-perforation $\overrightarrow{AB}$ ($\overrightarrow{AB} = (x^{iw,1}_{i+1,1}, y^{iw,1}_{i+1,1}, z^{iw,1}_{i+1,1}) - (x^{iw,0}_{i,Np_i^{iw}-1}, y^{iw,0}_{i,Np_i^{iw}-1}, z^{iw,0}_{i,Np_i^{iw}-1})$) is required instead of the true perforations $\overrightarrow{AC}$ and $\overrightarrow{CB}$ so that the flow rate of phase p remains roughly the same, i.e.,

$$WI_{\overrightarrow{AB}} \lambda_p \Delta p_{\overrightarrow{AB}} = WI_{\overrightarrow{AC}} \lambda_p \Delta p_{\overrightarrow{AC}} + WI_{\overrightarrow{CB}} \lambda_p \Delta p_{\overrightarrow{CB}}, \tag{C.9}$$

where p denotes oil, water and gas; λ_p is the mobility of phase p within the perforated gridlock; $\Delta p_{\overrightarrow{AB}}$, $\Delta p_{\overrightarrow{AC}}$ and $\Delta p_{\overrightarrow{CB}}$ respectively represent the pressure draw-down of perforations $\overrightarrow{AB}$, $\overrightarrow{AC}$ and $\overrightarrow{CB}$, and $WI_{\overrightarrow{AB}}$, $WI_{\overrightarrow{AC}}$ and $WI_{\overrightarrow{CB}}$ respectively represent the well indices of perforations $\overrightarrow{AB}$, $\overrightarrow{AC}$ and $\overrightarrow{CB}$. Assume $\Delta p_{\overrightarrow{AB}} = \Delta p_{\overrightarrow{AC}} = \Delta p_{\overrightarrow{CB}}$, then Eq. C.9 reduces to

$$WI_{\overrightarrow{AB}} = WI_{\overrightarrow{AC}} + WI_{\overrightarrow{CB}}. \tag{C.10}$$

The well index of a perforation $\boldsymbol{l}$ (e.g., $\boldsymbol{l}$ can be $\overrightarrow{AC}$, $\overrightarrow{CB}$ or $\overrightarrow{AB}$) is given by

$$WI = \frac{\beta K(\boldsymbol{l}) h(\boldsymbol{l}) f_{comp}}{\ln(r_e(\boldsymbol{l})/r_w) + s} \tag{C.11}$$

where β is a coefficient involving the angular fraction and the unit conversion factor, $K(\boldsymbol{l})$ is the completion planar averaged permeability, $h(\boldsymbol{l})$ is the perforation length (i.e. 2-norm of the vector $\boldsymbol{l}$), f_{comp} is the completion factor, $r_e(\boldsymbol{l})$ is the the drainage radius in the direction of the wellbore $\boldsymbol{l}$, r_w is the well radius and s is the skin factor. Note for a true perforation (e.g., $\overrightarrow{AC}$ and $\overrightarrow{CB}$), f_{comp} is always one. However, for a pseudo-perforation (e.g., $\overrightarrow{AB}$), f_{comp} has to be adjusted so that Eq. C.10 is satisfied.

Denote the perforation $\overrightarrow{AC}$ as $\boldsymbol{l}_1$, $\overrightarrow{CB}$ as $\boldsymbol{l}_2$ and $\overrightarrow{AB}$ as $\boldsymbol{l}$. Substituting Eq. C.11 into Eq. C.10 gives

$$\frac{\beta K(\boldsymbol{l}) h(\boldsymbol{l}) f_{comp}(\boldsymbol{l})}{\ln(r_e(\boldsymbol{l})/r_w) + s} = \frac{\beta K(\boldsymbol{l}_1) h(\boldsymbol{l}_1)}{\ln(r_e(\boldsymbol{l}_1)/r_w) + s} + \frac{\beta K(\boldsymbol{l}_2) h(\boldsymbol{l}_2)}{\ln(r_e(\boldsymbol{l}_2)/r_w) + s}, \tag{C.12}$$

and then $f_{comp}(\boldsymbol{l})$ can be solved as,

$$f_{comp}(\boldsymbol{l}) = \left(\frac{K(\boldsymbol{l}_1) h(\boldsymbol{l}_1)}{\ln(r_e(\boldsymbol{l}_1)/r_w) + s} + \frac{K(\boldsymbol{l}_2) h(\boldsymbol{l}_2)}{\ln(r_e(\boldsymbol{l}_2)/r_w) + s} \right) \frac{\ln(r_e(\boldsymbol{l})/r_w) + s}{K(\boldsymbol{l}) h(\boldsymbol{l})}. \tag{C.13}$$

In the following, we will briefly discuss the calculation of the drainage radius and the completion planar averaged permeability when a perforation does not parallel one of the gridblock coordinate axes. Take perforation $\boldsymbol{l}$ as an example. According to CMG [55], $K(\boldsymbol{l})$ and $r_e(\boldsymbol{l})$ are obtained by interpolating K's and r_e's along the gridblock coordinate axes.

Denote $\boldsymbol{i}$, $\boldsymbol{j}$ and $\boldsymbol{k}$ as unit vectors pointing in the local (x, y, z) directions. When the wellbore is parallel to the $\boldsymbol{D}$ axis ($\boldsymbol{D}$ can be either $\boldsymbol{i}$, $\boldsymbol{j}$ and $\boldsymbol{k}$), the drainage radius $r_e(\boldsymbol{D})$ is calculated as

$$r_e(\boldsymbol{D}) = \alpha\sqrt{\frac{V}{h(\boldsymbol{D})}}, \tag{C.14}$$

where, α is a coefficient involving the geometric factor, the angular fraction and the unit conversion factor, V is the bulk volume of the perforated gridblock, $h(\boldsymbol{D})$ is the grid block thickness in the direction $\boldsymbol{D}$. Define $\theta_i = \arccos(\frac{\boldsymbol{l}\cdot\boldsymbol{i}}{\|\boldsymbol{l}\|_2})$, $\theta_j = \arccos(\frac{\boldsymbol{l}\cdot\boldsymbol{j}}{\|\boldsymbol{l}\|_2})$ and $\theta_k = \arccos(\frac{\boldsymbol{l}\cdot\boldsymbol{k}}{\|\boldsymbol{l}\|_2})$. Then the value of the drainage radius in the direction of the wellbore is interpolated as

$$r_e(\boldsymbol{l}) = \left(\begin{array}{c} r_e(\boldsymbol{i})\cos^2(\theta_i)\sin^2(\theta_j)\sin^2(\theta_k) + r_e(\boldsymbol{j})\cos^2(\theta_j)\sin^2(\theta_i)\sin^2(\theta_k) \\ +r_e(\boldsymbol{k})\cos^2(\theta_k)\sin^2(\theta_i)\sin^2(\theta_j) \end{array}\right)/S,$$

where, $S = \cos^2(\theta_i)\sin^2(\theta_j)\sin^2(\theta_k)+\cos^2(\theta_j)\sin^2(\theta_i)\sin^2(\theta_k)+\cos^2(\theta_k)\sin^2(\theta_i)\sin^2(\theta_j)$. Similarly, the completion planar averaged permeability K is calculated by

$$K(\boldsymbol{l}) = \left(\begin{array}{c} \sqrt{K_yK_z}\cos^2(\theta_i)\sin^2(\theta_j)\sin^2(\theta_k) + \sqrt{K_xK_z}\cos^2(\theta_j)\sin^2(\theta_i)\sin^2(\theta_k) \\ +\sqrt{K_xK_y}\cos^2(\theta_k)\sin^2(\theta_i)\sin^2(\theta_j), \end{array}\right)/S,$$

where K_x, K_y, K_z respectively represent the permeability of the perforated gridblock along the (x, y, z) directions.