T H E U N I V E R S I T Y O F T U L S A

THE GRADUATE SCHOOL

ROBUST LIFE-CYCLE PRODUCTION OPTIMIZATION WITH STATE CONSTRAINTS

by
Zhe Liu

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

T H E U N I V E R S I T Y O F T U L S A

THE GRADUATE SCHOOL

ROBUST LIFE-CYCLE PRODUCTION OPTIMIZATION WITH STATE CONSTRAINTS

by
Zhe Liu

A THESIS

APPROVED FOR THE DISCIPLINE OF

PETROLEUM ENGINEERING

By Thesis Committee

Albert C. Reynolds, Chair
Mustafa Onur
Bill Coberly
Ram Mohan
Evren Ozbayogul

COPYRIGHT STATEMENT

ABSTRACT

Zhe Liu (Doctor of Philosophy in Petroleum Engineering)

Robust Life-Cycle Production Optimization with State Constraints

Directed by Albert C. Reynolds

158 pp., Chapter 5: Conclusions

(1912 words)

In the development of a reservoir, designing a set of optimal well controls (bottom-hole pressures or rates) for injection and production that maximize a specified cost (objective) function is critically important because changing the well controls changes the underground flooding pattern (streamlines) and significantly impacts the cost function. As is done here, the cost or objective function is typically the net present value (NPV) of production over the life of the reservoir but can also be the cumulative oil production over the reservoir life. The design (optimization) variables are either the bottom-hole pressure (BHP) or a rate (oil, gas, water or total liquid) at each well on a set of specified control steps (time intervals) which are obtained by partitioning the total time of the reservoir life. The length of control steps need not be uniform and may vary from well to well. In most examples considered, we consider the most standard case where injection wells are water injection wells and the injection rates on the control steps are the control (optimization) variables. One principal challenge of the well control production optimization problem is how to handle the nonlinear state constraints efficiently in the optimization process. In fact, despite the hundreds of papers that exist on well controls optimization, almost none have properly considered well state constraints where these state constraints may include an upper bound on the wellbore pressure at a rate-controlled injection well, a bound on the liquid rate, water rate, gas rate, producing GOR or producing WOR at a BHP-controlled producing well or field constraints

typically imposed by the processing capability of facilities, for example, a bound on the field liquid rate, field gas rate and field water rate. Most papers on well control optimization simply impose state constraints using the internal heuristics of the commercial reservoir simulator used but as shown in this work, this procedure can produce a highly suboptimal value of the NPV of life-cycle production. One difficulty in dealing with constraints properly within the framework of mathematical optimization theory using standard techniques such as sequential quadratic programming (SQP), which is used here, is that we need to be able to compute gradients of constraint functions efficiently. With a simulator enhanced with the capability to compute the adjoint solutions needed to compute the gradients of the constraint functions, this is feasible particularly with deterministic optimization where optimization is done based on a single model. However, to account for geological uncertainty, one generally performs robust optimization instead of deterministic optimization. With robust optimization, one estimates the well controls that maximize the expected value of the life-cycle NPV of production where the expected value is approximated by its standard estimator, namely, the average NPV of production where the average is over a set of N_e realizations of the reservoir model selected to represent the uncertainty in the reservoir model. With robust optimization, all constraints should be satisfied at every control step for every realization of the reservoir model, and if this is done in the straightforward way, then a separate gradient of each constraint function must be computed for each of the N_e reservoir models which imposes a high computational cost on the gradient-based optimization process. One can reduce the cost by, for example, lumping all constraints into one but then one invariably ends up with optimal controls that result in a significant constraint violation for some realizations of the reservoir model. In this work, we provide a novel, efficient computational procedure that effectively ensures that all constraints are satisfied for all reservoir models with at most an extremely small constraint violation but does not require explicit enforcement of all constraints on all reservoir models and thus significantly enhances computational efficiency. For the common situation where the commercial reservoir simulator used does not have the necessary adjoint solution capability, we replace adjoint-gradients

in SQP by stochastic simplex approximate gradients (StoSAG). This procedure introduces severe difficulty in enforcing state constraints due to inaccuracies in stochastic gradients of the constraint functions. Because these gradients define the feasibility region in SQP, if they are inaccurate, the feasible region estimated using standard StoSAG may not contain any point close to an optimal point. We develop procedures for damping and smoothing the stochastic derivatives so that SQP with stochastic gradients can be successfully applied to solve the robust life-cycle optimal well control problem with state constraints. To the best of our knowledge, our work represents the first successful solution of this optimization problem uses stochastic gradients. Finally, it is important to note that within the SQP framework, constraints are enforced using the filter method.

With robust optimization, there is no control over downside risk. To understand this, let u^* be the optimal well controls obtained from robust life-cycle production optimization, and let m_k denote the specific reservoir model that gives the lowest NPV value at u^* among all N_e realizations of the reservoir model that are used to characterize geological uncertainty. Then, even though the average life-cycle NPV at u^* may be large, if the true reservoir model is similar to m_k, then operating the reservoir at u^* may yield such a low NPV that development of the reservoir is not commercially viable. Thus, even if the optimization of the average life-cycle NPV of production is the primary goal, reducing the downside risk may be an extremely important secondary objective of the operator. In addition, optimizing life-cycle NPV sometimes yield estimates of optimal controls that yield unacceptably low production rates during the short-term (1-5 years) which is generally unacceptable as operators have production targets. Thus, maximizing short-term production is another important secondary or tertiary objective. As the short-term production optimization one might choose to maximize the expected value of the cumulative oil production over a specified short-term period, but here, we use as the short-term objective as the maximization of the average NPV on a specified short-time time interval. Often, but not always, the objectives in multi-objective optimization problems are in conflict. For example, if operators want to achieve a short-term NPV greater than a specified value, one may have to accept a lower value of

average life-cycle NPV than can be obtained by robust optimization of only the NPV over the reservoir life. When objectives are in conflict, the standard procedure is to construct a trade-off curve, i.e., a Pareto front, which allows one to select a solution that gives an acceptable value of all of the objective functions. This can be done using a weighted-sum method or a normal boundary intersection (NBI) method, but here we develop a modified lexicographic method to attempt to obtain a set of Pareto optimal solutions. In the examples considered, the three objectives are (i) maximize average life-cycle NPV. (ii) maximize average short-term NPV and (iii) maximize the minimum NPV of the set of realizations. In our examples, objective (i) is always considered to be the primary objective. With bi-objective optimization, either objective (ii) or (iii) is considered as the secondary objective. With tri-objective optimization, both (ii) and (iii) are considered as secondary objectives. With the modified lexicographic method, we first maximize the expectation (average-value) of the life-cycle NPV by itself to obtain the corresponding optimum vector of well controls, u^*, and record the short-term NPV and minimum NPV over the set of realization at u^*. Then, we maximize the live-cycle NPV again but with state constraints added. With tri-objective optimization, the state constraints are that the short term NPV be greater than to a constant times the short term NPV at u^* and the minimum NPV be greater than a constant times the minimum NPV evaluated at u^*. Both constants should be greater than unity and by increasing each constant, we can increase the value of the corresponding secondary objective function, but possibly at the expense of decreasing the value of life-cycle NPV below the value obtained at u^*. By changing the constants, we can try to find Pareto optimal points. Multi-objective optimization is done using stochastic gradients (StoSAG) in the SQP-filter method. The methodology can also be applied to enforce standard state constraints such as upper bounds on field production rates. The performances of the standard lexicographic method and modified lexicographic method are compared on a bi-objective optimization problems with average short-term and life-cycle NPV as the two objectives and it is found that the main advantage of the modified lexicographic method has over the the standard lexicographic method is that it allows the generation of potential Pareto optimal points which

are uniformly spaced in the values of the second objective that one wishes to improve by bi-objective optimization.

If the reservoir simulation used for reservoir management has sufficient adjoint capability to produce all gradients needed to solve the production optimization problem with state constraints that is of interest, then it is natural to expect that SQP optimization based on adjoint gradients will yield the most computationally efficient and robust optimization procedure. Somewhat surprisingly, we show that one can develop a significantly more computationally efficient procedure by replacing the adjoint-enhanced reservoir simulator by a proxy model and optimizing the proxy. Our methodology achieves computational efficiency by generating a set of output values of the cost and constraint functions and their associated derivative values by running the numerical model (reservoir simulator) for a broad set of input design variables (well controls) and then using the set of input/output data to train a proxy model to replace the numerical model when computing values of cost and constraint functions and their derivatives during iterations of sequential quadratic programming (SQP). During SQP optimization, the values of cost/constraint functions and their derivatives are computed from the proxy functions; no reservoir simulation runs are performed. The derivation of the equations for computing the proxy-based model that uses both function and gradient information is similar to that of least squares support vector regression (LS-SVR). However. this method is referred to as gradient-enhanced support vector regression (GE-SVR) because, unlike LS-SVR, the method uses derivative information, not just function values, to train the proxy. Similar to LS-SVR, improved (higher) estimated optimal NPV values may be obtained by using iterative resampling (IR). With IR, after each proxy-based optimization, one evaluates the cost and constraint functions and their derivatives at the estimated optimal controls using reservoir simulator output, and then adds this new input/output information to the training set to update the proxy models for predicting NPV and constraints. Using the updated proxies, one applies SQP optimization again. IR continues until the simulator and proxy evaluated at the latest estimate of the optimal well controls give the same value of NPV within a specified percentage tolerance and the average

constraints evaluated by reservoir simulator at the latest optimal well controls within some tolerance. Our results indicate that proxy-based optimization with iterative resampling may require up to an order of magnitude less computational time than is required using the adjoint-capable reservoir simulator for evaluation of the NPV and constraint functions and their derivatives at each stage of the SQP algorithm. We also provide some comparison of the optimization results generated using both function and gradient information to build the proxy to results obtained when the proxy is trained using only function information, i.e., to results generated with an LS—SVR proxy and find that GE-SVR is roughly an order of magnitude more computationally efficient than LS-SVR but also provides a better approximation of a complex cost function surface so that is possible to locate multiple optima in cases where LS-SVR fails to identify the multiple optima.

ACKNOWLEDGEMENTS

First, I would like to give my deepest appreciation to my advisor, Dr. Albert Reynolds. He provided many insightful inputs for this work and he is very patient with my poorly written papers and reports. He also helped me achieved personally. It is my utmost pleasure to have him as my Ph.D. advisor.

I would like to thank everyone in my committee, Dr. Mustafa Onur, Dr.Bill Coberly, Dr. Ram Mohan and Dr. Evren Ozbayogul for their inputs to improve the quality of this work. I also appreciate the member companies of the TUPREP as well as the Bellwether Fellowship for the financial support provided for this work.

I also appreciate all the help and supports I received from my colleagues and friends during my Ph.D. program. I am always deeply grateful to my parents, Zhixiang Liu and Chunhong Yang, for their unconditionally love and support throughout my life.

TABLE OF CONTENTS

COPYRIGHT iii

ABSTRACT iv

ACKNOWLEDGEMENTS x

TABLE OF CONTENTS xii

LIST OF TABLES xiv

LIST OF FIGURES xxi

CHAPTER 1: **INTRODUCTION** 1
- 1.1 **Literature Review on Production Optimization** 4
- 1.2 **Literature Review on Multi-Objective Optimization** 10
- 1.3 **Literature Review on Proxy-based Optimization** 14
- 1.4 **Dissertation Outline** 17

CHAPTER 2: **AN SQP FILTER ALGORITHM WITH A MODIFIED STOCHASTIC GRADIENT FOR ROBUST LIFE-CYCLE OPTIMIZATION PROBLEMS WITH NONLINEAR STATE CONSTRAINTS** 19
- 2.1 **Introduction** 19
- 2.2 **Methodology** 20
  - 2.2.1 *Optimization Problem* 20
  - 2.2.2 *Stochastic Gradient, StoSAG* 23
  - 2.2.3 *Improving the Stochastic Gradient* 26
  - 2.2.4 *Sequential Quadratic Programming Filter (SQP-filter)* 43
  - 2.2.5 *Min-Max Scheme* 55
- 2.3 **Computational Results** 57
  - 2.3.1 *Example 1: Single Reservoir Model* 58
  - 2.3.2 *Example 2: Brugge Field* 69
- 2.4 **Comments** 80

CHAPTER 3: **A MIN-MAX SQP-FILTER METHOD FOR NONLINEAR CONSTRAINED MULTI-OBJECTIVE ROBUST OPTIMIZATION PROBLEMS WITH STOSAG** 82
- 3.1 **Introduction** 82

3.2 **Methodology** . . . 82
3.2.1 *Multi-Objective Optimization Problem* . . . 82
3.2.2 *Lexicographic method* . . . 84
3.3 **Computational Results** . . . 89

CHAPTER 4: **ROBUST LIFE-CYCLE PRODUCTION OPTIMIZATION WITH GRADIENT-ENHANCED SUPPORT VECTOR REGRESSION** 109
4.1 **Introduction** . . . 109
4.2 **Methodology** . . . 110
4.2.1 *GE-SVR Model* . . . 110
4.2.2 *Proxy-based Optimization* . . . 119
4.3 **Computational Results** . . . 124
4.3.1 *Example 1* . . . 125
4.3.2 *Example 2* . . . 139
4.3.3 *Example 3* . . . 149

CHAPTER 5: **DISCUSSION AND CONCLUSIONS** 165

BIBLIOGRAPHY . . . 171

LIST OF TABLES

2.1 Statistical frequency of angles between adjoint gradient and different stochastic gradients (gradients of NPV) 43

2.2 The summary of constrained deterministic optimization with different bound constraints handling schemes on the cases with FLR$\leqslant$12,000 and FLR$\leqslant$10,000. 69

2.3 The summary of constrained robust optimization with different schemes; note nonlinear constraints are not satisfied for all realizations with expected value enforcement of constraints. 77

3.1 Summary of constrained bi-objective optimization with different lexicographic schemes. 97

3.2 Summary of the settings of constrained single objective, bi-objective and tri-objective optimization with modified lexicographic method. 104

3.3 Summary of computational results of constrained single objective, bi-objective and tri-objective optimization with modified lexicographic method. 104

4.1 Summary of the results for proxy-based optimization with iterative sampling on Rosenbrock function using 10 initial training samples 128

4.2 Summary of the 9 local minimums of the multi-minimum function within [-1, 1] $\times$ [-1, 1] 132

4.3 Nine initial guesses, estimate of optimal point for each initial guess, and the local minimum of the multi-minimum function that is closest to each optimal point generated by each application of iterative sampling with no perturbation and LS-SVR as the proxy model. 134

4.4 Nine initial guesses, estimate of optimal point for each initial guess, and the local minimum of the multi-minimum function that is closest to each optimal point generated by each application of iterative sampling with perturbation size of $\sigma_{rs} = 0.1$ and LS-SVR as the proxy model. 135
4.5 Nine initial guesses, estimate of optimal point for each initial guess, and the local minimum of the multi-minimum function that is closest to each optimal point generated by each application of iterative sampling with no perturbation and GE-SVR as the proxy model. 136
4.6 Nine initial guesses, estimate of optimal point for each initial guess, and the local minimum of the multi-minimum function that is closest to each optimal point generated by each application of iterative sampling with perturbation size of $\sigma_{rs} = 0.1$ and GE-SVR as the proxy model. 137
4.7 Summary the computational efficiency of the iterative sampling algorithm on the multi-minimum function with GE-SVR/LS-SVR as the proxy model with/without the perturbation scheme. 138
4.8 The summary of bound constraints. 141
4.9 Comparison of bound constrained optimization using reservoir simulator-based optimization with LS-SVR and GE-SVR proxy-based optimization without iterative sampling scheme. 148
4.10 The summary of inequality nonlinear state constraints for the proxy-based production optimization problem with life-cycle NPV as the objective function .150
4.11 Summary of iterative sampling steps of proxy-based optimization for deterministic constrained production optimization 158
4.12 Comparison of the performance of simulator-based and proxy-based optimizations on the deterministic production optimization problem with state constraints. (Example 3 part I) 159

LIST OF FIGURES

2.1 Comparison of adjoint and stochastic gradients of q_o of well P1 at $11th$ control step 30

2.2 Comparison of adjoint and stochastic derivatives of q_o of well P1 at 11th control step. 35

2.3 Comparison of adjoint and stochastic gradients of FLR at 11th control step with respect to control. 36

2.4 Comparison of adjoint and stochastic gradients of FWR at 11th control step with respect to all controls. 36

2.5 Comparison of adjoint and stochastic gradients of FGR at 11th control step with respect to all controls. 37

2.6 Comparison of adjoint and stochastic gradients of FLR at 20th control step with respect to all controls. 37

2.7 Comparison of adjoint and stochastic gradients of FWR at 20th control step with respect to all controls. 38

2.8 Comparison of adjoint and stochastic gradients of FGR at 20th control step with respect to all controls. 38

2.9 Comparison of adjoint and stochastic gradients of negative NPV with respect to all controls. 39

2.10 Estimation of the quality of gradient; x axis is the angle between the adjoint and stochastic gradient ($\circ$); y axis represemts the frequency of each stochastic gradient. 39

2.11 The acceptance criteria for the filter method, red point correspond to $u = u^{\ell}$, after Nocedal and Wright [67] 51
2.12 Log permeability of fields in all layers. 61
2.13 The optimal well controls; SQP-filter workflow. 62
2.14 BHP at injection well at the optimal well controls; SQP-filter workflow. . . . 63
2.15 Gas production rates at the optimal controls; SQP-filter workflow. 63
2.16 Water production rates at the optimal controls; SQP-filter workflow. 64
2.17 FLR and FGR at the optimal controls; SQP-filter workflow. 65
2.18 NPV versus number of iterations. 65
2.19 The optimal well controls; SQP-filter workflow. (stricter constraints) 67
2.20 Gas production rates at the optimal controls; SQP-filter workflow. (stricter constraints) 68
2.21 Water production rates at the optimal controls; SQP-filter workflow. (stricter constraints) 68
2.22 FLR and FGR at the optimal controls; SQP-filter workflow. (stricter constraints) 69
2.23 Log permeability distribution of selected realizations. 70
2.24 Top structure of Brugge case. 72
2.25 Comparison of the producer optimal controls of expected value and min-max schemes. 73
2.26 Comparison of the injector optimal controls of expected value and min-max schemes. 73
2.27 The FWR and FLR at optimal SQP-filter controls with min-max scheme. . . 74
2.28 The FWR and FLR at optimal SQP-filter controls with expected value scheme. 74
2.29 Optimal CDF of NPV corresponding to expected value scheme and min-max scheme. 75
2.30 Comparison of the producer optimal controls of min-max and bound constrained optimization without enforcing state constraints. 77

2.31 Comparison of the injector optimal controls of min-max and bound constrained optimization without enforcing state constraints. 78
2.32 The FWR and FLR at optimal controls of bound-constrained optimization without enforcing state constraints. 78
2.33 The FWR and FLR at optimal controls of bound-constrained optimization with nonlinear state constraints enforced by the reservoir simulator heuristically. 79
2.34 Comparison CDF of the NPV at the"optimal" controls; blue represents StoSAG with only bound constraints, red represents the modified results after enforcing constraints heuristically, yellow is from min-max scheme. 79
3.1 Comparison of optimal production BHP for single objective optimization (left) and bi-objective optimization with short-term NPV as the state constraints using modified lexicographic method (right) 90
3.2 Comparison of optimal water injection rates for single objective optimization (left) and bi-objective optimization with short-term NPV as the state constraints using modified lexicographic method (right) 91
3.3 Comparison of FWR at the optimum of single objective optimization (left) and at the optimum of bi-objective optimization (right) solved with modified lexicographic method 92
3.4 Comparison of FLR at the optimum of single objective optimization (left) and at the optimum of bi-objective optimization (right) solved with modified lexicographic method 93
3.5 CDF's of short-term NPV (left) and life-cycle NPV (right); **blue**: results from initial well controls; **red**: results from single objective life-cycle optimization; **yellow:**: results from modified lexicographic bi-objective optimization with state constraint on average negative short-term NPV. 93
3.6 Comparison of optimal production BHP for short-term and life-cycle NPV bi-objective optimization with standard lexicographic scheme (left) and modified lexicographic scheme (right) 95

3.7 Comparison of optimal water injection rates for short-term and life-cycle NPV bi-objective optimization with standard lexicographic scheme (left) and modified lexicographic scheme (right) 95
3.8 CDF's of short (left) and long (right) terms NPV's from the two bi-objective optimization schemes at the estimated optimal controls. **red**: standard lexicographic method; short-term NPV as primary objective, life-cycle NPV as a constraint; **blue**: modified lexicographic method; life-cycle NPV as primary objective, short-term NPV as a constraint. 98
3.9 Comparison of optimal production BHP for single objective optimization (left) and tri-objective optimization with short-term NPV and minimum risk as the state constraints (right) using modified lexicographic method 98
3.10 Comparison of optimal water injection rates for single objective optimization (left) and tri-objective optimization with short-term NPV and minimum risk as the state constraints (right) using modified lexicographic method 99
3.11 Comparison of FWR at the optimum of single objective optimization (left) and at the optimum of tri-objective optimization (right) solved with modified lexicographic method 100
3.12 Comparison of FLR at the optimum of single objective optimization (left) and at the optimum of tri-objective optimization (right) solved with modified lexicographic method 101
3.13 CDF's of short-term NPV (left) and life-cycle NPV (right); **blue**: estimates from single objective life-cycle optimization; **red**: from modified lexicographic bi-objective optimization with state constraint on negative average (minimum average) short term NPV; **yellow:** from modified lexicographic bi-objective optimization with state constraint on negative worst-case NPV; **purple:** from modified lexicographic tri-objective optimization with state constraints on both negative average (minimum average) short term NPV and negative worst-case NPV. 103

3.14 Pareto fronts of short-term NPV v.s. life-cycle NPV Blue: short-term NPV constraint varies, no constraint on risk J_{risk}. Black: $J_{risk} \leq -4.5 \times 10^9$, short-term NPV constraint varies. Red: $J_{risk} \leq -4.6 \times 10^9$, short-term NPV constraint varies. 107
3.15 Pareto fronts of minimum NPV v.s. life-cycle minimum NPV constraint varies, no constraint on short-term NPV. 108
4.1 The workflow of proxy-based optimization with iterative sampling. 122
4.2 Response surface of Rosenbrock function with 16 sample points. 126
4.3 Response surfaces of Rosenbrock function with 10 sample points. 127
4.4 Response surface of Rosenbrock function with iterative sampling scheme approximated by LS-SVR. 129
4.5 Response surface of Rosenbrock function with iterative sampling scheme approximated by GE-SVR. 130
4.6 Response surfaces of multi-minimum function with 12 sample points. 131
4.7 Iterative sampling process starting at the point (0, 0.5) with LS-SVR and GE-SVR as the proxy models and with and without the perturbation scheme. 133
4.8 The Log permeability of each geological layer. 141
4.9 Comparison of Adjoint gradient (red) with proxy predicted gradient (blue) at the center of control space with different hyperparameters 142
4.10 The change of 5-fold cross validation error of GE-SVR versus different σ. . . 145
4.11 The change of 5-fold cross validation error of LS-SVR versus different σ as the proxy. 146
4.12 Life-cycle NPV at each iteration of the gradient-based optimization for proxy-based optimization workflow with LS-SVR and GE-SVR results compared to the value of the true simulator prediction. 147
4.13 Comparison approximated scaled gradient of life-cycle NPV by the GE-SVR model at the 1st and 40th iterations for deterministic production optimization with bound constraints. 148

4.14 Comparison of the optimal BHPs for producers generated by simulator-based life-cycle production optimization (blue lines) and proxy-based optimization (red lines) for a deterministic case with state constraints. 151
4.15 Comparison of the optimal water injection rates generated by simulator-based optimization (blue lines) and proxy-based optimization (red lines) for deterministic life-cycle production optimization with state constraints. 151
4.16 Comparison of the oil production rates of each producer under the optimal controls generated by simulation-based life-cycle production optimization (blue lines) and proxy-based optimization (red lines) for the deterministic production optimization with state constraints. 152
4.17 Comparison of the BHP at each injector under the optimal controls generated by simulation-based optimization (blue lines) the proxy-based optimization (red lines) for the deterministic production optimization with state constraints. 153
4.18 Comparison of the FLR, FWR and FGR under the optimal controls generated by simulation-based optimization (blue lines) the proxy-based optimization (red lines) for the deterministic production optimization with state constraints. 153
4.19 Comparison of adjoint gradient of life-cycle NPV with respect to well controls (red) with proxy predicted gradient of life-cycle NPV (blue) at different iterative sampling steps 155
4.20 The Log permeability of each geological layer for realization 1 160
4.21 The Log permeability of each geological layer for realization 2 160
4.22 The optimal BHP at producers generated from proxy-based constrained robust production optimization. 160
4.23 The optimal water injection rates for injectors generated form the proxy-based robust production optimization. 161
4.24 The FLR, FWR and FGR under the optimal controls generated by the proxy-based optimization for the robust production optimization with state constraints. 163

4.25 Comparison of CDF of the robust life-cycle NPV generated by operating under the initial well controls (blue) and optimal well control (red). 164

CHAPTER 1

INTRODUCTION

As a crucial step in the closed-loop reservoir management framework, life-cycle production optimization is defined as maximizing/minimizing a predefined objective (cost) function as a function of the design variables over the remaining life of the reservoir [5, 45, 12]. In cases of interest to us, the objective function is evaluated from the output of a run of the reservoir simulator that generates the solution of the discretized equations that define an initial-boundary-value problem for an evolutionary nonlinear system of partial differential equations. The specific application considered here is the optimal well control problem where the primary cost function is the average life-cycle net-present-value (NPV) of production where the average is over a set of realizations of the reservoir model that characterize the uncertainty in the reservoir model. This average NPV is sometimes referred to the expected value of expectation of the NPV because the average NPV is the standard estimator to the expected NPV. As discussed throughout this chapter, a large number of papers have been written about the optimal well control problem. What is novel about this work is the development of a set of techniques that allow the incorporation of nonlinear state constraints into the optimal well problem in a robust and computationally efficient way. State constraints are those that require output from the forward model (in our case, a reservoir simulator) in order to evaluate the constraint function to determine whether the associated constraint is satisfied. Typical state constraints are the upper bound on a well's producing gas-oil ratio (GOR) (or on the gas rate) or an upper bound on the producing water oil-ratio (WOR) (or on the water rate). The capability of processing facilities typically mandate nonlinear state constraints in the form of upper bounds on the field water production rate, the field water injection water, the field gas production rate or the total field liquid production rate.

There may also exist an upper bound on the field water (or gas) injection rate. This bound represents a linear constraint if all injection well controls are water rates (or gas rates) but is a state constraint if the injection well controls are the bottom hole pressures.

For solution of the constrained robust optimal well control problem introduced above, we apply sequential quadratic programming (SQP) with constraints enforced using the filter method. However, methods such as the augmented Lagrangian method [13] could be used. These two methods require the computation of the NPV and constraint functions. If the commercial simulator in use for reservoir management is capable of computing the gradient of the NPV function and the gradient of all constraints with the adjoint procedure, then common thinking suggests that a gradient-based algorithm such as SQP may well be the most computationally efficient and reliable procedure for the solution of the constrained optimal well control problem. However, there is a caveat, namely, when performing robust optimization, each constraint must be enforced on every reservoir model at every control step, and the number of adjoint solutions required significantly increases the computational cost. Lumping of constants can be done along the lines of [75, 47] or constraints can be averaged as in [61]. However, when constraints are lumped or averaged, then at the estimated optimal well controls, one invariably finds that some constraints are violated for some reservoir models, and in some cases, significantly violated. In such a situation, the estimated optimum well controls not be implementable in practice. A major contribution of our work is the development of a min-max scheme, in which at each iteration of SQP, each constraint at each control step is enforced on only a single model, the one that gives the maximum violation of that constraint at that particular control step. Although there is no theoretical proof that this mix-max scheme ensures that every constraint is satisfied for every reservoir model, computational results suggest that this is effectively the case, i.e., only negligible constraints violations occur for any reservoir model when the min-max scheme is utilized in the SQP-filter optimization process. Returning to our comment that SQP using adjoint gradients, might be expected to be the most computationally efficient method, we actually find that we can achieve up to an order of magnitude greater computational efficiency by replacing

the adjoint-capable simulator with a proxy model which is trained with input-output data from the simulator where the input consists of input training samples of the vector of well controls and corresponding output training samples consist of the resulting values of NPV and constraint functions and their derivatives. By using the so-called gradient-enhanced support vector regression proxy (GE-SVR), periodically updated by iterative resampling, we can obtain a solution to the optimal well control problem much faster than is obtained by using the adjoint-capable reservoir simulator to evaluate the NPV and constraint functions and their derivatives at each iteration of the SQP algorithm. Optimization of the proxy is still done using SQP with the filter method employed to satisfy constraints. The development of the GE-SVR proxy approach is our second major contribution to the solution of the constrained optimal well control problem.

We know of no commercial reservoir simulator that had the capability of computing all the adjoint solutions needed to compute the gradients of all the common state constraint functions mentioned above. Thus, many recent papers have focused on using stochastic gradients to solve optimal well control problems [19, 86, 23, 24, 30, 33, 11, 59]. However, to the best of our knowledge, no paper has applied SQP where the gradients of constraints were directly used to determine the feasible region in SQP. Many papers incorporate only bound constraints, often by truncation, when solving the optimal well control problem and ignore state constraints. One can of course insert the optimal controls obtained from optimization algorithm directly into the reservoir simulator and use the internal heuristics of the simulator to change the controls so all constraints are satisfied but such a procedure introduces two main issues. First, the resulting well control solution is suboptimal which is illustrated in Chapter 2, and secondly, the modified controls produced by simulator heuristics would typically be different for each reservoir model, whereas, in robust optimization, the goal is to obtain a single set of controls applicable to all reservoir models. The third major contribution of this work is the development of a procedure to compute more reliable stochastic gradients of constraint functions so that these gradients can define a reasonable feasible region for the SQP algorithm.

In the development of a reservoir, operating under the optimal control generated by optimizing only average life cycle NPV will ignore some other important factors which are also important for decision making. For example, at the optimal well control vector, the lowest life-cycle NPV of the values computed from the set of realizations of the reservoir may be extremely small. In this circumstance, If the true reservoir model is similar to the worst-case scenario, then operating the reservoir at the optimal controls found by optimizing only average life-cycle NPV may yield an realized NPV that is so small that the development of the reservoir is not commercially viable. Thus, even if the optimization of the average life-cycle NPV is the primary objective, lowering the downside risk may be an extremely important secondary objective for operators to consider. In addition, at some scenarios, the robust production optimization with the life-cycle NPV as the objective function generates the optimal controls that yield unacceptably low production rates during the short-term (1-5 years), which is generally impractical as operators have production targets. Thus, maximizing short-term production is another important objective in the context of reservoir development. In this work, all these three commonly used conflicting objectives which are average life-cycle NPV, average short-term NPV and the worst-case NPV are considered. The fourth major contribution of this work is the development of a modified lexicographic method for robust multi-objective optimization with nonlinear state constraints using the modified StoSAG gradient.

1.1 Literature Review on Production Optimization

It is commonly thought that gradient-based optimization algorithms are the most robust and computationally efficient when the objective (cost) function is differentiable [70, 69, 44, 57] subject to the caveat that a computationally efficient way is available to compute an accurate gradient, i.e., the adjoint solution is available and carefully implemented [2, 92, 88, 87, 52, 51, 6, 74, 27, 26, 16, 44, 85]. For the nonlinearly constrained optimal well control problems considered here, if the commercial reservoir simulator that is employed does not have adjoint capability, a practise developed over the last decades is to a use gradient-based

algorithms which utilize an approximate gradients [19, 86, 30, 33, 11]. Although derivative free algorithms [35, 21, 80, 42] could also be used, there are far too computational inefficient for use in large-scale production optimization problems.

The principle problem of interest here is the minimization (or maximization) of a differentiable cost (objective) function subject to bound and nonlinear state constraints where the forward model that must be run to evaluate the cost function and state constraints is a numerical model. In our case, the numerical model is a reservoir simulator. Although there are other problems of interest, the specific application considered here refers to the optimal well control problem where the cost function is the net-present value of production over the remaining reservoir life and the design or optimization variables are the controls (bottom-hole pressure (BHP) or rate) at each well at each specified control step. In the specific examples considered, the nonlinear state constraints are mainly facility constraints on field production rates although other nonlinear state constraints can easily be incorporated, and, in fact, we consider bounds on BHP at rate-controlled water injection wells. In this scenario, BHP represents a state constraint. As it is important to consider geological uncertainty, we do robust optimization where the geological uncertainty is represented by an ensemble of plausible reservoir models. Optimizing the average NPV on a set of realizations of the reservoir model is referred to as robust optimization. When optimization is based on a single reservoir model, the process is referred to as deterministic optimization [84, 19, 17, 30]. The production optimization problem considered is of interest because it represents the second step of closed-loop reservoir management [45].

As mentioned previously, gradient-based methods are highly efficient for the optimization of a differentiable cost function [2, 92, 87, 52, 51, 48, 70, 44] when the forward model is a reservoir simulator which has adjoint capability. However, commercial simulators commonly used by oil and gas producing companies have limited or no adjoint capability for general optimal control problems with nonlinear state constraints. In fact, generating extremely accurate approximations of the gradient using the adjoint method can be challenging [85]. Here, we seek a solution procedure for the nonlinearly-constrained, well-control

optimization problem when the adjoint solution is not available. For large-scale optimization problems, true derivative free methods such as the genetic algorithm [35] and particle swarm optimization [42] or methods based on quadratic interpolation polynomials [21, 93] are not computationally feasible so the natural path is to employ algorithms derived on the assumption of the availability of the gradient of the objective function but replace the gradient, sensitivity matrix, or individual derivatives involved in the gradient by approximations. Approximating individual derivatives by finite-difference is tenuous because finding an appropriate perturbation size to employ in the finite-difference approximation can be problematic and even when it is not, generating finite-difference approximations of all individual derivatives involved in the gradient is not computationally feasible for high-dimensional problems even with methods designed to enhance computational efficiency; see Yan et al. [89] and the references therein. A superior alternative to using finite-difference approximations of derivatives is to use a stochastic approximation of this true gradient, that is, apply an algorithm such as the simultaneous-perturbation stochastic approximation (SPSA) algorithm [79, 46, 86, 50] or ensemble-based optimization [62, 19, 18, 86, 30, 33, 11]. As shown by Do and Reynolds [24], SPSA and EnOpt are effectively theoretically equivalent and strongly related to an approximate simplex gradient. Moreover, Fonseca et al [33] show that, based on theoretical arguments, the stochastic simplex approximate gradient (StoSAG), which was introduced in Fonseca et al. [30], is expected to give a better approximation of the true gradient of the objective function for robust optimization if the geological uncertainty is significant than the approximation obtained with EnOpt [19]. Thus, in the optimization algorithms employed for the optimal well control problem considered here, we use StoSAG to approximate the true gradient.

Another plausible approach to generate an approximate solution of optimization problems where the forward model is a reservoir simulator is to build a proxy model for the reservoir simulator and to use the proxy model to replace the simulator in the evaluation of the objective function at each iteration of the optimization algorithm. Guo and Reynolds [37] built a support vector regression (SVR) proxy model based on reservoir simulation runs

but they did not consider using gradient information from the underlying reservoir simulator to train the proxy model. Moreover, Guo and Reynolds [37] did not consider nonlinear state constraints. In this work, we provide the details for the development of a support vector regression proxy model to replace the reservoir simulator where we use both function and derivative information to train the proxy and also train proxies to evaluate the state constraint functions and their derivatives.

Although work on life-cycle production optimization using a quasi-Newton procedure has been considered [77, 14, 16, 4, 39], virtually all work using stochastic or ensemble-based derivatives have used a steepest ascent (or descent) algorithm. One exception to this is the paper of Dehdari and Oliver [23] who proposed an SQP optimization workflow based on an ensemble-based approach to estimate the gradients. Although, that workflow is designed to solve robust constrained production optimization problems, they only apply deterministic production optimization for a single realization of the Brugge case. The control variables are the total liquid production rates of each ICV (inflow control valve) and the constraints of their work are total liquid rates, minimum allowable pressures and maximum allowable water cuts of each well. However, the minimum allowable pressures and maximum allowable water cuts which are nonlinear state constraints, are enforced heuristically directly by the numerical simulator which, as we show later, can yield highly suboptimal results. Thus, in their SQP framework, only linear constraints are considered directly. Note for their problem. the total liquid production rate of a well a linear constraint because the total liquid rate is a linear combination of the liquid production rates of the completion intervals of wells, which are the well controls. They implemented a spatial localization scheme proposed by Chen and Oliver [20] to improve the quality of the EnOpt gradient. The basic idea of the Chen and Oliver [20] work is to use a distance-based localization function to retain the cross-covariance of controls at neighboring wells with the NPV of any specific well while neglecting the weaker cross-covariance with controls of more distance wells. The distance-based function defined in their work, is a unit-step function with a critical distance.

Liu et al. [60] compared the performance of the SQP algorithm with the augmented

Lagrangian algorithm for optimization problems with nonlinear state constraints, using a reservoir simulator with the capability to compute adjoint gradients. They found that SQP has better performance in terms of constraint handling and convergence efficiency. However, we observed in later unpublished work. that the difference in performance of the two algorithms was largely due the the fact that the augmented Lagrangian algorithm involves heuristic choices of some parameters and adjustment of these parameters, especially the penalty factor. Once the parameters of the augmented Lagrangian method have been properly tuned, the SQP and augmented Lagrangian algorithms have a similar performance.

Sarma et al. [76] proposed a technique for implementation of the nonlinear constraints into the optimization process by converting the inequality constraints into a single equality constraint and a set of bounds on the slack variable. However, two set of adjoint equations are solved at each iteration of the optimization algorithm which can be computationally expensive. Drosos Kourouni [47] applied an SQP framework to solve the deterministic optimization problem where they used a simulator with adjoint capability to compute the gradient, i.e., they used the "adjoint gradient." In all examples they considered, the controls at producers and injectors are BHP's, gas is the injected fluid and the only nonlinear state constraints are bounds on the gas injection rate at each injector and possibly bounds on the production rate at each producer. They do not consider nonlinear state constraints in the form of field rates which is the focus of our paper. To achieve computational efficiency, Drosos Kourouni et al. [47] do not enforce individual constraints but instead lump constraints. For example if the bound on the gas injection rate at each injector is $q_{\max}$ and $q_{j,n}$ denotes the rate at the j injector, then the correct constraint specification is $q_{j,n} \leq q_{\max}$ for $j = 1, 2, \cdots N_w$ and $n = 1, 2, \cdots N$ where N_w is the number of wells and N is the number of time steps. However, instead of enforcing these individual constraints, Kourounis et al. enforce only the single lumped constraint,

$$\max\{q_{j,n} \,|\, j = 1, 2, \cdots N_w,\, n = 1, 2, \cdots N\} \leq q_{\max}. \tag{1.1}$$

(Because the left side of Eq. 1.1 is non-differentiable, the authors approximate Eq. 1.1 by a smooth function where the transformed constraint is a nonlinear function of original constraints which is already nonlinear.) Within the SQP framework, they use the ℓ_1 merit function to handle constraints. However, they also consider a second procedure, a heuristic one, in which they only consider bounds on the BHP controls within the SQP framework and let the simulator impose the rate constraints directly in the forward run of the simulator. In this heuristic approach, if a rate at a specific well at a specific time step exceeds its maximum, this specific rate is set equal to its maximum for that well at that specific time step, so the well becomes rate controlled instead of BHP controlled. Generally, they find that treating all constraints within the SQP formalism tends to give higher values of the cost function (cumulative oil production) than the heuristic procedure for small problems where as the opposite is true for large problems.

Unlike Dehdari and Oliver [23], we consider nonlinear state constraints within the standard SQP framework which avoids the issue of being trapped at a suboptimal result due to the enforcement of constraints by the reservoir simulator [68]. Moreover, instead of combining the objective function and constraint function using a single penalty function, the filter method [29, 67] is implemented to promote global convergence in our work. The filter method does not require that the user specify any penalty parameters and allows a certain amount of non-monotonicity because a step and new estimate of the optimization variables are accepted if the updated variables result in a reduction in either the objective function or the constraint violation.

Unfortunately, we found that the straightforward implementation of SQP with stochastic gradients with nonlinear state constraints often fails to give stochastic gradients of sufficient accuracy to ensure that SQP converges to a result that is close to optimal. Throughout, the accuracy of a stochastic gradient refers to how well it approximates the direction of the true gradient. The failure of SQP for optimization problems with nonlinear state constraints when using StoSAG to approximate gradients apparently arises because we need an accurate approximation of both the gradients of the constraints and the gradient of the NPV objec-

tive function to ensure that there is a feasible solution of the quadratic subproblem [67]. In particular, the SQP linearization of the constraints using stochastic gradients can generate a highly inaccurate feasible region unless the stochastic gradients of constraint functions provide a reasonable approximation of the true gradient. Perhaps, this is one of the reasons that nonlinear state constraints have generally not been implemented when using stochastic gradients. In any case, in this work we develop procedures to modify the stochastic gradients from StoSAG to attain stochastic gradients of sufficient accuracy so that these gradients can be used with the SQP method to solve the robust life-cycle production optimization problem with nonlinear state constraints where the filter methods is used to enforce the constraints.

1.2 Literature Review on Multi-Objective Optimization

In a decision making process, we often encounter situations where we have conflicting goals, criteria or objectives. Mathematically, these problems are described as follows: find a design vector (vector of optimization variables) which maximizes (or minimizes) each function in the set of functions which define the conflicting criteria or objectives. As the problem is untenable in the sense that, in general, it is not possible to find a single design vector which maximizes all objective functions, the solution to a multi-objective optimization problem is defined as the Pareto front which represents a trade-off curve, or more generally, a trade-off hypersurface. Given the Pareto front (surface), the decision maker can choose a point on the surface based on how large a decrease in each individual objective functions is acceptable in order to avoid making the values of the primary objective function(s) too small.

Many algorithms have been developed to solve multi-objective optimization problems. Generally these algorithms can be divided into two categories: derivative-based (or gradient-based) algorithms and derivative-free algorithms. The popular derivative-free algorithms include genetic-based multiobjective optimization algorithms such as the strength Pareto evolutionary algorithm-II (SPEA II) and the non-dominated sorting genetic algorithm-II (NSGA II) [96]. Reyes-Sierra and Coello [73] provide a review of multiobjective particle swarm algorithms. The major issue associated with any evolutionary algorithm is that the

computational cost becomes infeasible when the dimension of the vector of optimization variables exceeds a hundred or so and evaluation of the objective function at each iteration requires the run of a complex numerical model, in our case, a reservoir simulator. Thus, for a large-scale optimal well control problem, where we need to evaluate our objective function through using output from a reservoir simulation run, evolutionary multi-objective optimization algorithms are computationally infeasible. As opposed to evolutionary algorithms, gradient-based optimization methods, are computationally efficient for optimization of a differentiable objective function. Popular multi-objective optimization methods, such as the lexicographic method [64], the weighted sum method [34, 91] and the normal boundary intersection (NBI) method [22] can be implemented via the use of the gradient. The weighted sum method requires users to assign a different weight to each objective function; then all objective functions are summed with their corresponding weights to obtain an aggregate function. By maximizing the aggregate function, we can obtain a solution on the Pareto front. We can obtain different points on the Pareto front by assigning a different set of weights to each objective function. However, with the weighted sum method, points on the convex part of the Pareto front cannot be obtained. The NBI method can be viewed as the weighted sum method with additional constraints. These additional constraints guarantee that the whole Pareto front can be obtained by using the NBI method. The lexicographic method requires a user to order the relative importance of each objective function. Then, the priory objective function is optimized first; afterwards, the second most important objective function is optimized with an additional constraint which requires that a certain percentage (99%, for example) of the optimal value for the most important objective function is achieved; we repeat this process until the least important objective function is optimized. One common feature of the weighted sum method, the NBI method and the lexicographic method is that they all transform the original multiobjective optimization problem to a series of single objective optimization problems.

In the development and management of a reservoir, many people may be involved and different people may emphasize different objectives. For instant, reservoir engineers may

want to design strategies to maximize the estimated ultimate recovery (EUR) or maximize the life-cycle net present value (NPV) of production, while project managers may put more emphasis on the minimization of the development risk. From the operator's point of view, it may be desirable to maximize NPV or cumulative oil production over a relatively short time, e.g., the next one, two or a few years. Maximizing the long-term/short-term NPV and minimizing the risk are three potentially conflicting goals, and multi-objective techniques can be applied to obtain the optimal solutions as defined by the Pareto front.

Van Essen et al. [83] and Chen et al. [15] found that it is sometimes possible to improve the short-term NPV significantly by allowing a small decrease in the long-term NPV. In these papers, the gradient is computed by the adjoint method. A method that is conceptually considered in both papers proceeds as follows: first life-cycle optimization is performed and then one tries to increase the associated short-term NPV significantly without significantly decreasing the optimal life-cycle NPV. An alternate strategy in the attempt to maximize both long-term and short-term NPV was proposed by Chen et al. [15]; in this procedure, the life-cycle constrained optimization problem is solved first, and then the short-term NPV is optimized subject to the additional constraint that the long-term NPV is greater than or equal to the optimal NPV obtained by the life-cycle production optimization. Chen et al. actually allow a small decrease in the long-term NPV when trying to increase short-term NPV and the constrained production optimization problem with the long-term NPV as one of state constraints is solved by the augmented Lagrangian method.

Isebor and Durlofsky [41] generated the Pareto front for maximizing both the expectation of the NPV and the worst plausible NPV (downside risk) by using an implementation of a hybrid pattern search and particle swarm algorithm. In their paper, they considered a general field development problem where the well locations, the well controls and the number of wells are the optimization variables. Because the number of wells is a discrete variable, gradient-based methods are not directly applicable. Since Isebor and Durlofsky [41] do not utilize gradient information, in one example, it requires on the order of 800,000 simulation runs to obtain the Pareto front; thus the procedure would be completely impractical for

realistic problems.

Liu and Reynolds [54] developed and implemented the weighted sum and normal boundary intersection (NBI) methods to solve biobjective optimization problem where the two objectives are to maximize the expected value of the life-cycle NPV and minimize the standard deviation of the NPV over the ensemble of geological realizations. They observed that the use of standard deviation as the risk measure is not wise because the reduction in risk/uncertainty is mainly achieved by reducing the largest possible NPV. Thus, Liu and Reynolds [55] applied the constrained weighted sum and constrained NBI methods to maximize the expected value of NPV and reduce the risk by maximizing the worst NPV; in this work, they also considered the presence of nonlinear field constraints. In the work of Liu and Reynolds [56], the lexicographic method also was used to maximize expectation and maximize the NPV of the worst-case scenario (or minimize the standard deviation). Liu and Reynolds [53] solved the a tri-objective optimization problem with the expected life-cycle NPV, expected short-term NPV and downside risk as three conflicting objectives. The derivatives of those three objective functions are computed by the adjoint method and only bound constraints are considered in that work. For the methods utilized in these papers of Liu and Reynolds, a gradient-based algorithm with the gradients computed by the adjoint method is used for the optimization. While gradient-based methods significantly enhance computational efficiency, adjoint solutions are not generally available in commercial simulators.

Recently, in the petroleum engineering literature, ensemble-based optimization has been applied to biobjective optimization [32] [10]. The methods effectively all produce a stochastic approximation of the gradient or preconditioned gradient of the objective function. The stochastic gradient does not require a reservoir simulator with adjoint capability and simply uses the reservoir simulator as a black-box. In Chen [10], the life-cycle NPV and downside risk are considered as two conflicting constraints where the downside risk is characterize as conditional value at risk (CVaR) and worst case NPV, respectively. None of papers cited in this section used stochastic gradients to compute nonlinear state constraints

within the robust production optimization process.

In this work, we develop a robust multi-objective optimization algorithm based on the use of StoSAG gradients to help decision makers balance the trade-off between long-term/short-term interests and downside risk of a reservoir development with geological uncertainty. The min-max scheme is implemented to ensure that all nonlinear state constraints are satisfied at each control step at each realization.

1.3 Literature Review on Proxy-based Optimization

Depending on the scale of a problem, a single forward reservoir simulation run may take a few minutes to a dozen or more hours to finish. Therefore solving the optimization problem based on a full-physics reservoir simulator is a computationally demanding process, where the optimization algorithm may require hundreds to thousands of forward runs especially when nonlinear state constraints are involved [53]. In recent studies, many researchers proposed the use of lower fidelity and fast proxy models to replace the high fidelity and full-physics reservoir simulator. Significant computational resources can be saved if appropriate proxy models are selected to replace the reservoir simulator in the production optimization process. The most popular low-fidelity models in the production optimization community can generally be classified into three types: reduced-order models, physics-based data-driven models and learning-based data-driven models.

The basic idea of the reduced-order model is to generate low-order models using snapshots from a forward reservoir simulation runs. The reduced-order model can be used in place of a reservoir simulator in optimization algorithms to significantly reduce the computational time required to estimate an optimal solution. van Doren et al. [82] proposed proper orthogonal decomposition (POD) to reduce the order of the reservoir simulation problem; also see [8]. He and Durlofsky [40] enhanced the performance of the reduced-order model by introducing information from the Jacobian matrix. Although incorporating Jacobian information in the reduced-order model can improve its accuracy, the development of these reduced-order models requires a simulator which allows the user to access the necessary

Jacobian information which is generally not available from commercial reservoir simulator. Moreover, for a complex multi-phase flow model e.g., a compositional model, the reappearance and disappearance of phases may lead to many switches of the primary variables, which can lead to failure of the reduced-order model.

Typically, a physics-based data-driven model is formulated to preserve some of the reservoir physics which are related directly to the physics of flow and transport. The parameters involved in such models are obtained directly by history matching so the development of the model does not require a detailed geological model or access to the high fidelity reservoir simulator. One issue associated with this feature of data-driven models is the requirement that a large number of production data are available. Thus, for a green field, it may not be feasible to use the data-driven model for production optimization. Moreover, extending the physical-based data-driven model to problem with complex physics, e.g., a thermal three-phase compositional model, is a difficult process. Physics-based data driven models include [90], INSIM [94] and INSIM-FT [37, 39]. These specific models only apply for the two-dimensional two-phase flow of water and oil. The INSIM-FT has been extended to three-dimensional flow with gravitational effects included by Guo and Reynolds [38], but still only applies for water-oil systems.

Recently, machine learning-based proxy models have been applied to solve the production optimization problem [49, 9, 37]. Nonlinear regression enables us to build a proxy model as a computationally efficient forward model to replace a full-scale reservoir simulation model. The training data for a machine learning-based proxy utilizes data generated from the full-physics, full-scale reservoir model. Thus, compared to the physic-based data-driven models discussed above, the machine learning-based proxy is easy to extend to any type of flow and transport scenario for which a simulator is available to generate training output from training input. Moreover, with properly selected feature space and training points, the machine learning-based proxy model can be implemented to optimize well controls even for a green field.

A linear regression proxy model, which is the simplest commonly used machine learn-

ing algorithm, generates an approximate model which is a linear combination of explicitly defined vectors in feature space (e.g., polynomials, sine and cosine functions) with corresponding coefficients. These methods have the same issue that commonly exists for interpolation-based methods, i.e., the issue of data overfitting especially if the training outputs are significantly corrupted with noise. As we train our proxy to reservoir simulator output, not measurements, overfitting should not be a significant issue for a life-cycle production optimization problem.

In this work, our focus in the improvement of the least-squares support vector regression (LS-SVR) proxy model. The basic idea of LS-SVR is to transform the input data from the original space into a higher dimensional feature space, where the output data has a linear relationship with the transformed variables in feature space. The parameters defining the linear relationship between the output and the feature space variables are obtained by solving an optimization problem that minimizes a weighted sum of the LS-SVR model complexity and the deviation of the predicted response from the "true" response. However, the transformation into the feature space is usually not performed in an explicit way due to computational infeasibility [81]. Instead, a kernel function that satisfies Mercer's condition [66] is introduced to convert the optimization problem from a primal space into a dual space, which makes the problem solvable. Guo and Reynolds [37] were the first to apply the LS-SVR with iterative sampling to solve robust production optimization problems. Since, the LS-SVR provides an accurate approximation near training points, iterative sampling generates an LS-SVR proxy model that is highly accurate in the neighborhood of the optimal controls. Iterative sampling involves a sequence of applications of the optimization algorithm. In each optimization step, the reservoir simulator is replaced by a proxy which is used to generate the value of the cost/constraint functions and their derivatives at each iteration of the optimization algorithm. Upon convergence, the optimal well controls and corresponding values of cost/constraint functions and their derivatives, which must be obtained by a single run of the reservoir simulator, represent a new training sample which is added to the previous training set. Using the new training set, a updated proxy is generated and used to do a new

optimization run. The process of proxy updating and optimization is repeated until the results stabilize, which means that the reservoir simulator and the proxy produce virtually the same value of the objective function at the estimated optimal values of the design variables, e.g., the well controls. The basic idea of iterative sampling is to improve the reliability of the proxy near the optimal solution as the sequence of optimizations improve the estimate of the optimum.

In this work, we develop a least-squares gradient-enhanced support vector (GE-SVR) proxy that is trained to match both derivative and function data output from a reservoir simulator with adjoint capability. Unlike standard gradient-based optimization algorithms, which only incorporate gradient information from at most two previous iteration steps (e.g. BFGS and LBFGS), GE-SVR can utilize the gradient information at all training points and reuse this training information when starting from different initial guesses to try to find multiple local optimums for multi-modal functions. Moreover, once the initial proxy is obtained, evaluation of the proxy is far more computationally efficient than evaluating the response using a complex forward model, i.e., the reservoir simulator. Because of the previous two statements, the GE-SVR workflow with iterative sampling is expected to provide a robust and more computationally efficient algorithm than results when each iteration of the optimization algorithm requires running the reservoir simulator to evaluate outputs and their gradients with the adjoint gradient.

1.4 Dissertation Outline

This dissertation contains five chapters that proceed as follows:

In Chapter 2, we first define the objective function and state constraints in the petroleum engineering context. Then, two numerical schemes to improve the quality of the stochastic gradient are motivated and discussed and verified by numerical experiments. The formulation of the SQP-filter algorithm that we use is introduced in the next section. Next, we introduce a min-max scheme for enforcing nonlinear state constraints, which is more computationally efficient than explicitly including all constraints for all reservoir models di-

rectly in the definition of the optimization problem. Two numerical examples are designed to illustrate the robustness and efficiency of our SQP-filter framework and superiority of the improved StoSAG gradient over the standard StoSAG gradient.

In Chapter 3, we introduce the lexicographic method as a standard framework to solve the multi-objective optimization problem. Then, we introduce our modified lexicographic method to solve the constrained multi-objective optimization problem. In this chapter, stochastic gradients are used in the SQP filter method with a min-max scheme which reduces the computational effort necessary to enforce state constraints on every reservoir model at every control step. The Brugge case is tested with our constrained multi-objective optimization framework and for this example, we solve both bi-objective and tri-objective optimization problems.

In Chapter 4, we first discuss the least squares support vector regression (LS-SVR) proxy model. Then, we introduce the formulation and implementation of the gradient-enhanced support vector regression (GE-SVR) proxy model, which, unlike the LS-SVR proxy, is trained to match both function and derivative information output from a reservoir simulator with adjoint capability. Thereafter, the iterative sampling scheme are discussed. Finally, computational results for three examples are presented and discussed.

We summarize the main conclusion of this research in Chapter 5. In Chapter 5, conclusions are presented on a chapter by chapter basis.

CHAPTER 2

AN SQP FILTER ALGORITHM WITH A MODIFIED STOCHASTIC GRADIENT FOR ROBUST LIFE-CYCLE OPTIMIZATION PROBLEMS WITH NONLINEAR STATE CONSTRAINTS

2.1 Introduction

Solving a large-scale optimization problem with nonlinear state constraints is challenging when adjoint gradients are not available for computing the derivatives needed in the basic optimization algorithm employed. Here, we present a methodology for the solution of an optimization problem with nonlinear and linear constraints where the true gradients that cannot be computed analytically are approximated by ensemble-based stochastic gradients based on an improved stochastic simplex approximate gradient (StoSAG). Our discussion is focused on the application of our procedure to waterflooding optimization where the optimization variables are the well controls (pressures or rates) at specified time intervals (control steps) and the cost function is the average life-cycle net present value (NPV) of production where the average is over an ensemble of realizations of the reservoir model that characterize the geological uncertainty in the reservoir model. The optimization algorithm used for solving the constrained optimization problem is sequential quadratic programming (SQP) with constraints enforced using the filter method. We introduce modifications to StoSAG that improve its fidelity, i.e., the improvements give a more accurate approximation to the true gradient (assumed here to equal the gradient computed with the adjoint method) than the approximation obtained using the original StoSAG algorithm. The modification to StoSAG vastly improve the performance of the optimization algorithm; in fact, we show that if the basic StoSAG is applied without the improvements, then SQP may yield a highly suboptimal

result for optimization problems with nonlinear state constraints. For robust optimization, each constraint should be satisfied for every reservoir model which is highly computationally intensive. However, we demonstrate that the computationally viable alternative of letting the reservoir simulation enforce the nonlinear state constraints using its internal heuristics yields significantly inferior results. Thus, we develop an alternative procedure for handling nonlinear state constraints, which avoids explicit enforcement of nonlinear constraints for each reservoir model yet yields results where, at convergence any constraint violation for any model is extremely small.

2.2 Methodology

2.2.1 Optimization Problem

The production optimization problem considered here refers to the estimation of the optimal wells controls on predefined control steps that minimize the negative net present value (NPV) of life-cycle production subject to bound constraints and to operational constraints which represent nonlinear state constraints. Note that minimizing the negative NPV is equivalent to maximizing the NPV.

We let u be the N_u-dimensional column vector which contains all production and injection well controls. Notationally, $u = [u_1, ..., u_{N_u}]^T = [u^{1,1}, u^{2,1}...u^{N_c,1}, ..., u^{N_c,N_{\text{well}}}]^T$ where $u^{i,j}$ denotes the control at well j at the ith control step, N_c is the number of control steps and N_{well} is the total number of wells with their controls subject to optimization. Examples of well controls are the specified water injection rate of injectors, liquid or oil production rate of producers, or the bottom hole pressure of any type of well. Throughout, m denotes the vector of reservoir parameters, e.g., permeabilities, porosity and net-to-gross of each grid block, depths of fluid contacts or parameters in power-law relative permeability functions of a reservoir simulation model.

For a three-phase flow reservoir under waterflooding, the negative of the NPV func-

tional for any reservoir model m and any control vector u can be defined by

$$J(u,m) = -\sum_{n=1}^{N_\mathrm{t}} \left\{ \frac{\Delta t_n}{(1+b)^{\frac{t_n}{365}}} \left[\sum_{j=1}^{\mathrm{P}} \left(r_\mathrm{o} \cdot \bar{q}^n_{\mathrm{o},j}(u,m) + r_\mathrm{g} \cdot \bar{q}^n_{\mathrm{g},j}(u,m) - r_\mathrm{w} \cdot \bar{q}^n_{\mathrm{w},j}(u,m) \right) \right] \right\} - \sum_{n=1}^{N_\mathrm{t}} \left\{ \frac{\Delta t_n}{(1+b)^{\frac{t_n}{365}}} \left[\sum_{j=1}^{\mathrm{I}} \left(r_\mathrm{wi} \cdot \bar{q}^n_{\mathrm{wi},j}(u,m) \right) \right] \right\}. \quad (2.1)$$

Note we minimize the negative NPV which is equivalent to maximizing the NPV. The time at the end of the nth simulation time step is denoted by t_n; Δt_n is the size of nth time step; and N_t is the total number of time steps. P and I respectively denote the number of production and injection wells; $\bar{q}^n_{\mathrm{o},j}$ (STB/D), $\bar{q}^n_{\mathrm{w},j}$ (STB/D) and $\bar{q}^n_{\mathrm{g},j}$ (Mcf/D), respectively, denote the average oil, water and gas production rate of the jth well over the nth time step, whereas $\bar{q}^n_{\mathrm{wi},j}$ (STB/D) is the average water injection rate at the jth injection well over the nth time step; r_o (\$/STB) denotes the oil price; r_g (\$/Mcf) denotes the gas price; r_w (\$/STB) is the disposal cost of produced water; r_wi (\$/STB) is the water injection cost and b is the annual discount rate.

Normally, we wish to consider geological uncertainty, and then we minimize the average negative NPV of an ensemble of samples of a stochastic reservoir model, i.e., we consider robust optimization. For robust optimization, the objective function that is minimized is given by

$$\bar{J}(u) \equiv \frac{1}{N_\mathrm{e}} \sum_{k=1}^{N_\mathrm{e}} J(u, m_k) \approx E\left[J(u,m)\right], \quad (2.2)$$

where E denotes the expected value. Here, N_e represents the number of reservoir models used to characterize geological uncertainty, and the different reservoir models (set of vectors of model parameters) are given by m_j, $j = 1, 2, \ldots, N_\mathrm{e}$. Note that one evaluation of $\bar{J}(u)$ requires N_e reservoir simulation runs; thus, each evaluation of the objective function for robust optimization is computationally expensive if N_e is large. While various procedures [78, 63, 58, 59] have been proposed to select a subset of the N_e models for optimization in order to improve computational efficiency, such options are not explored here.

For robust optimization, the generic nonlinear constrained optimization problem is defined as

$$\underset{u\in\mathbb{R}^{N_u}}{\text{minimize}} \quad \bar{J}(u), \tag{2.3a}$$

$$\text{subject to} \quad u_i^{\text{low}} \leq \ u_i \ \leq \ u_i^{\text{up}}, \ i = 1, 2, \ldots, N_u, \tag{2.3b}$$

$$c_i(u, m_j) \geqslant 0 \qquad , \ i = 1, 2, \ldots, N_{\text{ic}} \ \ j = 1, 2, \ldots, N_{\text{e}}. \tag{2.3c}$$

For the specific computational examples presented here, we consider the most common well control case where the injection wells operate under rate control and the producers operate under pressure control. In this case, natural state constraints are facility-imposed upper bounds on the field liquid rate (FLR), the field gas rate (FGR), the field water rate (FWR) and bounds on the pressure at each rate-controlled water injector well. By the same techniques developed here, one could impose bounds on the producing water oil ratio and gas-oil ratio either on a well by well basis or on a field basis. We also consider linear bound constraints on the pressure of BHP-controlled producer wells. We do not include equality constraints in the description of the general constrained optimization problem simply because dealing with equality constraints requires no nontrivial modification of the SQP algorithm. For problems of interest to us, we commonly only have inequality constraints with the exception of the requirement of voidage replacement which is an equality constraint. It is of course possible to also have nonlinear constraints that do not involve the state of the system but those are much easier to enforce, as well as much less common, and are not considered here. Throughout, u_i^{up} and u_i^{low} denote the upper bound and lower bound of the ith control variable, $c_i(u)$ denotes the ith inequality constraint and N_{ic} is the number of nonlinear inequality constraints for one realization of reservoir model. It is important to note that $c_i(u)$ may be either a linear constraint or a nonlinear state constraint. In our implementation, each bound constraint on a control variable can either be converted to two linear inequality constraints of the form $c_i(u) \geqslant 0$ or simply enforced by truncation.

2.2.2 Stochastic Gradient, StoSAG

In order to solve the constrained robust optimization problem defined in Eq. 2.3, the SQP-filter algorithm is implemented where the gradients necessary to implement SGP are stochastic gradients given by StoSAG [33]. As we will see obtaining a sufficiently accurate gradient is necessary to obtain an estimated optimum that is close to a true optimum. To mitigate the convergence difficulty encountered when using SQP with stochastic gradients generated with standard StoSAG to solve the optimal control problem with nonlinear state constraints, we introduce two modifications that improve the accuracy of the stochastic gradient. To introduce the first modification, we first note that the NPV of life-cycle production optimization is a linear combination of the phase rates at all control steps. Thus, the gradient of the NPV functional with respect to the controls is a linear combination of the gradients of the individual flow rates at each control step. The derivatives involved in any one of these gradients represent the derivative of a flow rate at a particular control step with respect to all control variables. However, the derivative of a flow rate at control step i with respect to a control variable at a later control step theoretically and physically must be zero. Thus, after computing the stochastic gradient of a flow rate at a particular control step, we set the derivatives of the rate with respect to control variables pertaining to later control steps equal to zero. This modification of the StoSAG gradient is referred to as truncation. The second improvement to the StoSAG gradient, which is referred to as temporal damping, is motivated by our intuition, supported by observations, that the derivative of a phase rate at a specific well at control step i with respect to the control at control step $i-j$ usually is smaller than its derivative with respect to the control at control step $i-k$ if $0 \leq j < k$. As the magnitude of stochastic gradient does not generally follow this expected behavior, it is desirable to apply a damping factor to improve the fidelity of the stochastic gradient. Here, we derive a semi-theoretical procedure to modify the stochastic derivatives by damping to obtain modified derivatives which more closely follow the expected behavior of derivatives. The theoretical derivation, however, is based on single-phase flow and, in this sense, is a heuristic rather than a theoretical result. However, we illustrate that truncation and tempo-

ral damping yield more accurate stochastic gradients of the phase rates involved in the state constraints than those computed with StoSAG and the improved accuracy significantly improves the performance of SQP for solving the constrained optimization problems of interest here.

Throughout, the N_u dimensional column vector u^ℓ is the estimate of the vector of optimal controls at the ℓth iteration of the steepest descent optimization algorithm where the gradient in the algorithm is replaced by the stochastic simplex gradient discussed later. In the first part of this section the stochastic gradient with respect to u is denoted by $\nabla_{u,\text{sto}}$. However, after the procedure is clear, we simply use ∇ to denote the stochastic gradient with respect to u. First note that from Eq. 2.2, the stochastic gradient with respect to u of the average NPV at u^ℓ is given by

$$\nabla_{u,\text{sto}}\bar{J}(u^\ell) = \frac{1}{N_\text{e}}\sum_{k=1}^{N_\text{e}} \nabla_{u,\text{sto}} J(u^\ell, m_k) = \frac{1}{N_\text{e}}\sum_{k=1}^{N_\text{e}} \nabla_{u,\text{sto}} J_k^\ell \tag{2.4}$$

where the last equality simply introduces the notation $J_k^\ell = J(u^\ell, m_k)$. Thus, to compute the stochastic gradient $\nabla_{u,\text{sto}}\bar{J}(u^\ell)$, we simply compute the stochastic gradients of the J_k^ℓ's and average them.

With StoSAG, stochastic perturbations of u^ℓ are obtained as

$$\widehat{u}_{i,k}^\ell \sim N(u^\ell, C_U) \tag{2.5}$$

for $i = 1, 2, \cdots N_p$, $k = 1, 2, \cdots N_e$, where N_p denotes the number of perturbations, N_e is the number of models used to represent geological uncertainty and C_U is an assumed prior matrix for the controls introduced to promote the temporal smoothness of controls and their perturbations at each individual well as discussed at the end of this subsection. Note that $\widehat{u}_{i,k}^\ell$ represents the ith perturbation of u^ℓ associated with the kth geological realization. If N_e is on the order 10 to 15 or greater, we typically only use one to three perturbations per model [33] whereas, if $N_e = 1$ (geological uncertainty not considered), we generally choose

the value of N_p between 10 and 20.

The $N_u \times N_p$ matrix used to compute $\nabla_{u,\text{sto}} J_k^\ell$ is denoted by ΔU_k^ℓ and is defined as

$$\Delta U_k^\ell = [\widehat{u}_{1,k}^\ell - u^\ell \quad \widehat{u}_{2,k}^\ell - u^\ell \quad \cdots \widehat{u}_{N_\text{p},k}^\ell - u^\ell], \tag{2.6}$$

for $k = 1, 2, ..N_\text{e}$, where N_p and N_e respectively denote the number of perturbations and number of realizations in the ensemble. Note for each m_k, a different set of perturbations is used. It is possible to use a different number of perturbations for each m_k but we do not pursue that scenario here. For each reservoir model m_k, the perturbation of u^ℓ to $\widehat{u}_{i,k}^\ell$ results in a change in $J_k^\ell = J(u^\ell, m_k)$ given by

$$\delta J_{i,k}^\ell = J(\widehat{u}_{i,k}^\ell, m_k) - J_k^\ell. \tag{2.7}$$

For each k, $k = 1, 2, \cdots N_e$, the N_p dimensional column vector, ΔJ_k^ℓ , is then given by the N_p dimensional column vector defined by

$$\Delta J_k^\ell = [\delta J_{1,k}^\ell \quad \delta J_{2,k}^\ell \quad \cdots \quad \delta J_{N_p,k}^\ell]^T. \tag{2.8}$$

As derived in [33], the basic formulation of the StoSAG approximation of the gradient of $J_k^\ell = J(u^\ell, m_k)$ is given by

$$\nabla_{u,sto} J_k^\ell = (\Delta U_k^\ell)^+ \Delta J_k^\ell, \tag{2.9}$$

where the superscript + on a matrix denotes the Moore-Penrose pseudo-inverse [36]. From Eq. 2.4, it follows that the stochastic gradient of $\bar{J}^\ell = \bar{J}(u^\ell)$ is given by

$$\nabla_{u,sto} \bar{J}^\ell = \frac{1}{N_\text{e}} \sum_{k=1}^{N_\text{e}} ((\Delta U_k^\ell)^+ \Delta J_k^\ell), \tag{2.10}$$

Note $\nabla_{u,sto} \bar{J}^\ell$ is the approximation of $\nabla_u \bar{J}^\ell$ used in a gradient-based algorithm. As shown in [33], the expectation of $\nabla_{u,sto} \bar{J}^\ell$ is equal to the true gradient $\nabla \bar{J}(u^\ell)$ plus a bias term which is $\mathcal{O}(\max_{i,k}\{\| \widehat{u}_{i,k}^\ell - u^\ell \|\})$. Also note that $N_\text{p} \times N_\text{e}$ forward simulations are required

to estimate the stochastic gradient of $\bar{J}^{\ell}$ with the StoSAG method. For robust optimization, we do three perturbations per model in our computational examples.

The $N_u \times N_u$ matrix C_U introduced in Eq. 2.5 is a preconditioning covariance matrix which is generally designed to generate control variables with temporal smoothness [19, 50]. As C_U enhances the temporal smoothness of controls on a well by well basis, C_U is chosen to be a block diagonal matrix [33] consisting of N_w matrices where N_w is the number of wells. The matrix that appears as the k diagonal block of C_U is denoted by C_U^k and its dimension is $N_c^k \times N_c^k$ where N_c^k is the number of controls, or control steps, applied at well k. In the example presented here, the correlation length of control steps does not vary from well to well. The entries of the C_U^k's can be generated from any covariance function. In the examples presented here, we use a spherical covariance function so the entry in the ith row and jth column of C_U^k is given by

$$C_u^k(i,j) = \begin{cases} \delta_{u_k}^2 (1 - \frac{3|i-j|}{2L_k} + \frac{3|i-j|^3}{2L_k^3}) & , |i-j| < L_k \\ 0 & , |i-j| \geqslant L_k. \end{cases}, \tag{2.11}$$

where L_k is the correlation length for controls at well k and δ_{u_k} effectively defines the perturbation size used to compute the stochastic gradient. In the examples, we use the same correlation length at each well. It is also assumed that the same number of controls steps are used at each well although it is not necessary to do so [68]. The perturbation size, δ_{u_k}, could also be varied from control step to control step and from iteration to iteration of the optimization algorithm based on the magnitude of individual controls. However, in the example problems, the perturbation size is constant. Typically, we set δ_{u_k} equal to 0.001 of the range of control variables. It is important to note that because Eq. 2.9 approximate the gradient directly instead of C_U times the gradient, sampling perturbations from Eq. 2.5 results in smooth correlated perturbations but does not guarantee the final controls are smooth.

2.2.3 Improving the Stochastic Gradient

From this point on all gradients that appear in equations are stochastic gradients

with respect to u so the notation $\nabla_{u,sto}$ is simply replaced by ∇. The ideas for improving the quality of approximation of the gradient given by StoSAG are based on computing ΔJ_k^ℓ for each J_k^ℓ as the appropriate linear combination of the stochastic gradients of all the rates that appear in the definition of $J_k^\ell = J(u^\ell, m_k)$ and in the state constraints. These stochastic gradients of rates with respect to u are also used to generate the gradients needed to enforce the state constraints. From Eq. 2.1, we see that the $\nabla_{u,sto} J_k^\ell$ term on the right side of Eq. 2.9 can be written as

$$\nabla J_k^\ell = -\sum_{n=1}^{N_\mathrm{t}} \left\{ \frac{\Delta t_n}{(1+b)^{\frac{t_n}{365}}} \left[\sum_{j=1}^{\mathrm{P}} \left(r_\mathrm{o} \nabla \bar{q}_{\mathrm{o},j}^n(u^\ell, m_k) + r_\mathrm{g} \nabla \bar{q}_{\mathrm{g},j}^n(u^\ell, m_k) - r_\mathrm{w} \nabla \bar{q}_{\mathrm{w},j}^n(u^\ell, m_k) \right) \right] \right\} - \sum_{n=1}^{N_\mathrm{t}} \left\{ \frac{\Delta t_n}{(1+b)^{\frac{t_n}{365}}} \left[\sum_{j=1}^{\mathrm{I}} \left(r_\mathrm{wi} \nabla \bar{q}_{\mathrm{wi},j}^n(u^\ell, m_k) \right) \right] \right\}, \quad (2.12)$$

for $k = 1, 2, \cdots N_e$. Thus, to compute the StoSAG approximation of the gradient, we simply need to apply repeatedly the formula for StoSAG to compute the gradient of each individual rate term. For example, the stochastic simplex gradient of the average oil rate at production well j evaluated at model m_k at the n control step with respect to all the control variables is denoted by $\nabla \bar{q}_{\mathrm{o},j}^n(u^\ell, m_k)$ and can be computed by

$$\nabla \bar{q}_{\mathrm{o},j}^n(u^\ell, m_k) = [(\Delta U_k^\ell)^+](\Delta Q_{\mathrm{o,j}}^n)_k^\ell, \quad (2.13)$$

for $k = 1, 2, \cdots N_e$, $j = 1, 2, \cdots P$ and $n = 1, 2, \cdots N_t$. The matrix of perturbations, ΔU_k^ℓ, is defined in Eq. 2.6 and the matrix of resulting changes in $\bar{q}_{\mathrm{o},j}^n$ for reservoir model m_k is denoted by $(\Delta Q_{\mathrm{o,j}}^n)_k^\ell$ and defined by

$$(\Delta Q_{\mathrm{o,j}}^n)_k^\ell = [\bar{q}_{\mathrm{o},j}^n(\widehat{u}_{1,k}^\ell, m_k) - \bar{q}_{\mathrm{o},j}^n(u^\ell, m_k), \quad \bar{q}_{\mathrm{o},j}^n(\widehat{u}_{2,k}^\ell, m_k) - \bar{q}_{\mathrm{o},j}^n(u^\ell, m_k) \quad , \cdots, \quad \bar{q}_{\mathrm{o},j}^n(\widehat{u}_{N_p,k}^\ell, m_k) - \bar{q}_{\mathrm{o},j}^n(u^\ell, m_k)]. \quad (2.14)$$

for $k = 1, 2, \cdots N_e$, $j = 1, 2, \cdots P$ and $n = 1, 2, \cdots N_t$. The obvious analogues of Eq. 2.13 can be used to compute the stochastic gradients, $\nabla \bar{q}_{o,j}^n(u^\ell, m_k)$, $\nabla \bar{q}_{g,j}^n(u^\ell, m_k)$, $\nabla \bar{q}_{w,j}^n(u^\ell, m_k)$

for $j = 1, 2, \cdots P$ and $\nabla \bar{q}_{wi,}(u^\ell, m_k)$ for $j = 1, 2, \cdots I$ whenever water injection wells are not rate controlled. However, if the water injection rates are the control variables, the gradients $\nabla \bar{q}_{wi,}(u^\ell, m_k)$ can be computed analytically. Using the stochastic gradients of these rates in Eq. 2.12, one can calculate the stochastic gradients ∇J_k^ℓ for $k = 1, 2, \cdots N_e$, and, using these last results from Eq. 2.12 in Eq. 2.4 gives the stochastic gradient $\nabla \bar{J}(u^\ell)$. Equally importantly, the gradients of rates can be used to compute the gradient of nonlinear state constraints that involve rates. As to compute the gradient of bottom hole pressure (BHP) with respect to water production rates for the rate-controlled water injection well, we switch the $(\Delta Q_{\mathrm{o,j}}^n)_k^\ell$ to $(\Delta P_{\mathrm{j}}^n)_k^\ell$ where $(\Delta P_{\mathrm{j}}^n)_k^\ell$ is defined by

$$(\Delta P_{\mathrm{w,j}}^n)_k^\ell = [\bar{p}_{\mathrm{w},j}^n(\widehat{u}_{1,k}^\ell, m_k) - \bar{p}_{\mathrm{w},j}^n(u^\ell, m_k), \quad \bar{p}_{\mathrm{w},j}^n(\widehat{u}_{2,k}^\ell, m_k) - \bar{p}_{\mathrm{w},j}^n(u^\ell, m_k) \quad , \cdots , \quad \bar{p}_{\mathrm{w},j}^n(\widehat{u}_{N_p,k}^\ell, m_k) - \bar{p}_{\mathrm{w},j}^n(u^\ell, m_k)]. \tag{2.15}$$

for $k = 1, 2, \cdots N_e$, $j = 1, 2, \cdots I$ and $n = 1, 2, \cdots N_t$. Therefore, the stochastic simplex gradient of the BHP at a rate-controlled injection well j evaluated at model m_k at the n control step with respect to all the control variables is denoted by $\nabla \bar{p}_{\mathrm{w},j}^n(u^\ell, m_k)$ and can be computed by

$$\nabla \bar{p}_{\mathrm{o},j}^n(u^\ell, m_k) = [(\Delta U_k^\ell)^+](\Delta P_{\mathrm{o,j}}^n)_k^\ell, \tag{2.16}$$

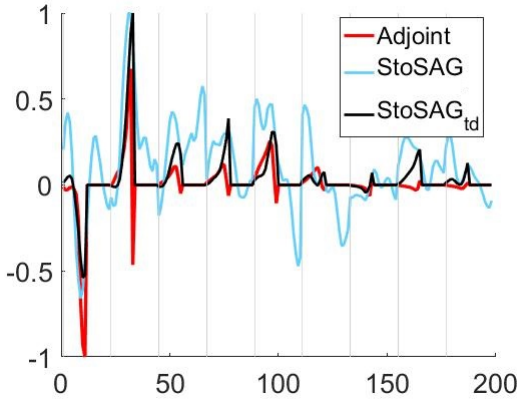
for $k = 1, 2, \cdots N_e$, $j = 1, 2, \cdots I$ and $n = 1, 2, \cdots N_t$.

In computing these stochastic gradients for the two examples considered later in this chapter, three perturbations are used for each model m_k in the standard procedure for computing the stochastic simplex gradient; see [24], [33]. However, as discussed previously, we show that it is possible to generate a better quality stochastic gradient that improves the performance of SQP for nonlinear state constrained optimal well control problems by first modifying the stochastic gradients of the rate terms before using them to compute the ∇J_k^ℓ.

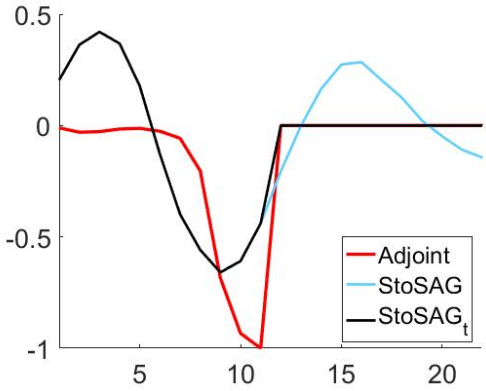
Based on the physics/mathematics of the problem, the average well rates over a reservoir simulator time step from t_{n-1} to t_n will not change if any control on a future control interval, $I_m = (t_{c,m-1}, t_{c,m}]$, is perturbed, i.e. if u_m^ℓ is any specific control defined

on the control step $I_m = (t_{c,m-1}, t_{c,m}]$ where $t_{c,m-1} > t_n$, then the correct derivative of all average rates defined on $(t_{n-1}, t_n]$ with respect to u_m^ℓ must be equal to zero. However, the StoSAG formula of Eq. 2.13 typically yields a nonzero value of such a derivative. Thus, to improve the accuracy of the stochastic derivatives, we simply truncate these values to zero. Below, we show how this affects the StoSAG derivatives by comparing StoSAG derivatives to those computed from the adjoint method. The adjoint gradient is treated as a benchmark because of its relatively high accuracy.

Next, we compare the accuracy of the standard StoSAG and our modifications of StoSAG with the gradient computed using the adjoint method. The basic reservoir model used in this comparison is the model considered in the first optimization example considered latter. For this reservoir model, there are 9 wells drilled in the reservoir and each well has 22 control steps, so the total number of control variables is $9 \times 22 = 198$. All production well controls are BHPs and all water injectors operate under injection rate controls. Fig. 2.1a compares standard StoSAG (light blue) derivatives of the average oil rate of well P1 at the 11th control step with respect to all the control variables with those derivatives computed by the adjoint method and with those computed by both truncation and damping (StoSAG$_{td}$) where discussion of the damping mechanism is postponed until later. Here, the StoSAG gradient is generated with 20 perturbations using a perturbation size δ_u in Eq. 2.11 equal 0.001 of the range of well control interval (lower bound to upper bound) and the control correlation length is 5 control steps. The range for each water injection control is $[0,\ 500]$, so the perturbation size of 0.5 (STB/d) is used to compute stochastic derivatives with respect to injection rate controls. Bounds on pressure at producers are given by $[1000,\ 4600]$, so perturbation size of 3.6 psi is used to compute derivatives with respect to pressure. Each integer value on the x-axis of Fig. 2.1a corresponds to the well control of a certain well at a certain time step. The thin vertical gray lines in Fig. 2.1a, as well as in similar figures presented later, separate the controls on a well by well basis. For example, the first 22 values, which are to the left of the first vertical gray line, pertain to the derivative of the oil rate of well P1 at the 11th control step with respect to all controls of producer P1. The



(a) Adjoint and stochastic derivatives of q_o at P1 at $11th$ control step with respect to all controls.



(b) Adjoint and stochastic derivatives of q_o at P1 at $11th$ control step with respect to its own controls.

Figure 2.1: Comparison of adjoint and stochastic gradients of q_o of well P1 at $11th$ control step

derivatives corresponding to each gradient are from the scaled gradient where the scaled gradient of q_o at the 11th control step is defined by $\nabla_u q_o / ||abs(\nabla_u q_o)||_\infty$. Thus, in absolute value, the maximum value of the derivatives included in each of those three gradients is equal to 1. This means the gradient comparison is really a comparison of directions rather than magnitudes which is appropriate because as long as we obtain a good search direction, a step in that direction is controlled by the step size. The derivatives of the oil rate of well P1 at 11th control interval with respect to its own controls obtained from the adjoint gradient, the StoSAG gradient and StoSAG gradient with truncation (StoSAG$_t$) are compared in Fig. 2.1b. (Results with both truncation and damping are not shown in this figure.) By only observing the gradient of oil rate of P1 at the 11th control step with respect to its own controls in Fig. 2.1, we can clearly see that after the 11th control step, all values of the adjoint gradient are zero. However, the StoSAG gradient produces nonzero values for the gradient components which are corrected to zero by truncation. However, the truncated StoSAG gradient does not give an accurate approximation of the derivative of P1's control at 11th control with respect to its controls at earlier time steps. Thus, truncation by itself (black curve) is not enough to give a good search direction but truncation plus damping (StoSAG$_{td}$) gives a direction in reasonable agreement with the adjoint gradient.

Our second idea for improving the quality of the gradient is referred to as temporal-damping. Temporal damping is motivated by our observations on how the magnitudes of the derivatives of the flow rate at a particular time at any well with respect to control variables depend on the control step on which the control variables apply. Roughly speaking, the magnitude of derivative of the average rate of well w on the nth control step with respect to the control of the same well at control step j decreases as $n-j$ increases. To make this as clear as possible, we consider the case where we are interested in the gradient of the average oil rate for control step n at well j with respect to all the controls of well j, i.e., the gradient of $\bar{q}^n_{o,j}$ with respect to the controls at well j, where, here and only here, we let $\widehat{u}^j_k$, $k = 1, 2, \cdots, N_c$, be the controls of well j for the N_c control steps. Then the objective is to estimate $\partial \bar{q}^n_{o,j}/\partial \widehat{u}^j_k$, $k = 1, 2, \cdots, n$ by damping, where $\partial \bar{q}^n_{o,j}/\partial \widehat{u}^j_k$ for $k = n+1, \cdots, N_c$ are automatically set equal to zero by truncation. Our observation is that the "baseline" gradient computed from the adjoint method typically tends to the following relationship:

$$\Big|\frac{\partial \bar{q}^n_{o,j}}{\partial \widehat{u}^j_n}\Big| > \Big|\frac{\partial \bar{q}^n_{o,j}}{\partial \widehat{u}^j_{n-1}}\Big| > \cdots > \Big|\frac{\partial \bar{q}^n_{o,j})}{\partial \widehat{u}^j_n}\Big|. \tag{2.17}$$

Eq. 2.17 applies at any well for any average rate on a control step. Thus, the general result is that the magnitude of the derivatives of an average flow rate (oil, water or gas) at a specific control step with respect to its own controls reach the maximum value at the specific control step and decrease in magnitude as the "times" of control steps track back in time from the specific control step. Next, we present the motivation of our temporal damping scheme. The motivation is theoretically-based but is derived only for single-phase flow for a single vertical well produced at a sequence of constant rates. Thus, there is no theoretical guarantee that it applies to multiphase flow problems even though our example computations suggest that it roughly applies for multiphase flow problems. In fact, though we believe damping can significantly improve the quality of the gradient, it is likely that there exist better damping procedures than the one presented here. In fact, part of the motivation for presenting the scheme is to encourage other researchers to investigate ways

to improve the quality of stochastic gradients.

According to the superposition theorem for single phase flow, the pressure drop at a vertical well due to a sequence of constant production rates q_1, $q_2, \cdots, q_n$ is given by

$$\begin{aligned} p_i - p_{\text{wf}}(t) &= q_1 p_u(t-0) + (q_2 - q_1) p_u(t - t_1) + (q_3 - q_2) p_u(t - t_2) + \\ &\quad \cdots + (q_n - q_{n-1}) p_u(t - t_{n-1}) \\ &= q_1[p_u(t-0) - p_u(t-t_1)] + q_2[p_u(t-t_1) - p_u(t-t_2)] + \\ &\quad \cdots + q_{n-1}[p_u(t-t_{n-2}) - p_u(t-t_{n-1})] + q_n p_u(t - t_{n-1}), \end{aligned} \tag{2.18}$$

for $t > t_{n-1}$, where p_i is the initial pressure of the reservoir; $p_u(t)$ represents the unit-rate constant-drawdown pressure change at the time t; q_n indicates the liquid production rate at the nth control interval (nth control step). In the following derivation, we define $\Delta t_j = t - t_j$, $j = 1, \cdots, n-1$ and use the fact that $p_u(t - t_j)$ is equal to $p_u(\Delta t_j)$. It is assumed that vertical flow is negligible so from the viewpoint of a reservoir simulation model, the well is assumed only penetrating a single gridblock. Let $p_b(t)$ represent the average pressure of this gridblock at any time $t \geq t_n$. Then, by adding $(p_b(t) - p_i)$ to both sides of Eq. 2.18, it follows that

$$\begin{aligned} p_b(t) - p_{\text{wf}}(t) =& q_1[p_u(\Delta t) - p_u(\Delta t_1)] + \cdots + q_j[p_u(\Delta t_{j-1}] - p_u(\Delta t_j)] \\ &+ \cdots + q_{n-1}[p_u(\Delta t_{n-2}) - p_u(\Delta t_{n-1})] + q_n p_u(\Delta t_{n-1}) + [p_b(t) - p_i]. \end{aligned} \tag{2.19}$$

In oil-field units, Peaceman's well index [71] in the $x - y$ coordinate system for single-phase flow is defined as

$$WI = \frac{\sqrt{k_x k_y} h}{141.2\mu[\ln(r_o/r_w) + s]}, \tag{2.20}$$

where k_x and k_y in md are the well's gridblock permeabilities in the x and y directions, respectively; h is the reservoir thickness in ft; μ is the liquid viscosity in cp; r_o and r_w in ft denote the equivalent radius and wellbore radius, respectively. Although Peaceman's well index is derived for single-phase flow, effectively, it is applied to compute the flow rate of multiphase flow by multiplying by the phase relative permeability, i.e., to compute the flow

rate of phase ph, the pressure drop is multiplied by a "multiphase flow well index," which is defined by

$$WI_{ph} = \frac{\sqrt{k_x k_y}h}{[141.2\ln(r_o/r_w) + s]} \frac{k_{r,ph}}{\mu_{ph}} = WI\lambda_{ph}, \tag{2.21}$$

where $k_{r,ph}$ and λ_{ph}, respectively, represent the relative permeability and mobility of phase ph. Note this well index is no longer constant as it changes with saturation. The well index for computing the total liquid flow rate at time t is denoted by $WI_{\ell,t}$ and is defined by

$$WI_{\ell,t} = \frac{\sqrt{k_x k_y}h}{[141.2\ln(r_o/r_w) + s]} \left(\frac{k_{r,o}}{\mu_o} + \frac{k_{r,w}}{\mu_w}\right) = WI\lambda_{\ell}, \tag{2.22}$$

where λ_ℓ denotes the total liquid mobility (oil plus water). The total liquid production rate at time t based on the multiphase flow well index is given by

$$q_\ell(t) = \frac{\sqrt{k_x k_y}h}{1421.2\ln(r_o/r_w) + s}\Big(\frac{k_{ro}}{\mu_o} + \frac{k_{rw}}{\mu_w}\Big)[p_b(t) - p_{\rm wf}(t)] = WI_{\ell,t}[p_b(t) - p_{\rm wf}(t)]. \tag{2.23}$$

Now, we simply assume that Eq. 2.19 approximately applies when the total liquid rate is used in place of the single-phase flow rate. Under this assumption, using Eq. 2.19 in Peaceman's well model, we obtain

$$\begin{aligned} q_\ell(t) =& WI_{\ell,t}[p_b(t) - p_{\rm wf}(t)] = WI_{\ell,t}\{(q_{\ell,1}[p_u(\Delta t) - p_u(\Delta t_1)] + \cdots + q_{\ell,j}(t_j)[p_u(\Delta t_{j-1}) - p_u(\Delta t_j)] \\ & \cdots + q_{\ell,n-1}(p_u(\Delta t_{n-2}) - p_u(\Delta t_{n-1}) + q_{\ell,n}p_u(\Delta t_{n-1}) + (p_b(t) - p_i)\}, \end{aligned} \tag{2.24}$$

where $q_{\ell,j}$ denotes the average total liquid production rate for the jth control step. Note in our case, the producers are under BHP control so the well rates of producers vary within the control interval. However, Eq 2.24 is derived based on the constant production rates assumption, i.e., assumes that over all control intervals the production rate of the well is constant. We assume a BHP controlled well with an average production rate $\bar{q}_i$ on the ith control interval would have a similar pressure profile as a constant rate controlled well whose control rate q_i is equal to $\bar{q}_i$. The average rate at ith control step is computed by $\bar{q}_i = \frac{Q_i - Q_{i-1}}{\Delta t_i}$, where the Q_i and Q_{i-1} denote the cumulative production of a producer at the end of the

ith and (i-1)th control step, respectively; Δt_i denote the length of the ith control step. In Eq. 2.24, we approximate $q_\ell(t)$ by the average rate over the control step, i.e., by $q_{\ell,n} \equiv \bar{q}_\ell^n$ and also replace $WI_{\ell,t}$ for any t such that $t_{j-1} < t \leq t_j$ by its average over the control step from t_{j-1} to t_j, where this average well index is denoted by $\overline{WI}_{\ell,j}$. We let $p_{b,j}$ and $p_{\text{wf},j}$, respectively, represent the average pressure of a grid block which contains any specific well and the bottom hole pressure (BHP) at the jth control step. Modifying Eq. 2.24 using this notation and then using Eq. 2.23 to replace each average flow rate in the modified version of Eq. 2.24, it follows that the average liquid flow rate on the control interval $(t_{n-1}, t_n]$ can be approximated by

$$\begin{aligned} \bar{q}_\ell^n = q_{\ell,n} =& \overline{WI}_{\ell,n}\{\overline{WI}_{\ell,1}(p_{b,1} - p_{\text{wf},1})[p_u(\Delta t) - p_u(\Delta t_1)] + \cdots \\ &+ \overline{WI}_{\ell,j}(p_{b,j} - p_{\text{wf,j}})[p_u(\Delta t_{j-1}) - p_u(\Delta t_j)] + \cdots \\ &+ \overline{WI}_{\ell,n}(p_{b,n} - p_{\text{wf,n}})p_u(\Delta t_{n-1}\} + \overline{WI}_{\ell,n}(p_{b,t} - p_i). \end{aligned} \tag{2.25}$$

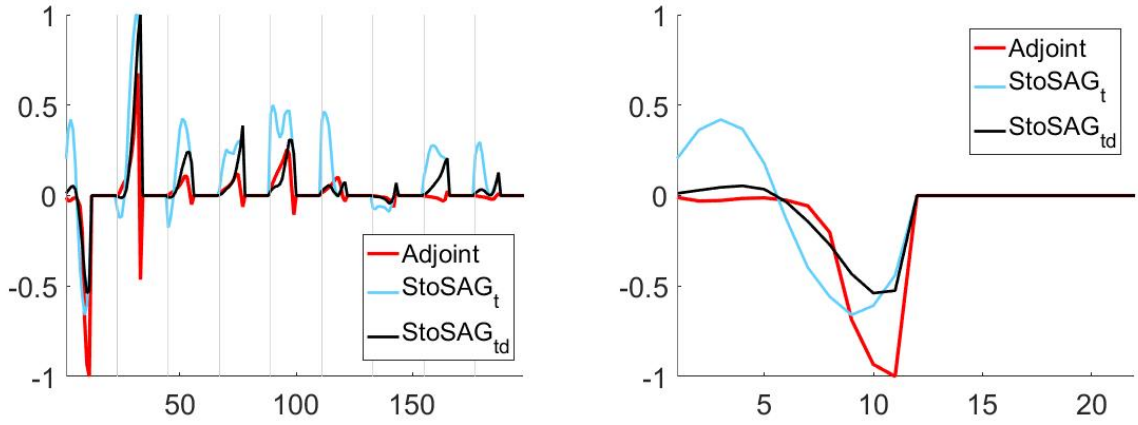
Then, by taking the derivative of Eq. 2.25 with respect to any $p_{\text{wf},j}$, we obtain

$$\frac{\partial q_{\ell,n}}{\partial p_{\text{wf},j}} = -\overline{WI}_{\ell,n}\overline{WI}_{\ell,j}[p_u(\Delta t_{j-1}) - p_u(\Delta t_j)], \tag{2.26}$$

for $j = 1, 2, \cdots n$. Eq. 2.26 for the derivative is relevant for pressure controlled wells and assumes that the dependence of $\overline{WI}_{\ell,n}$ and $\overline{WI}_{\ell,j}$ on $p_{\text{wf},j}$ can be neglected. We treat the value of $\frac{\partial q_{\ell,n}}{\partial p_{\text{wf},n}}$ as the reference value. As discussed previously, we expect the magnitude of $\partial q_{\ell,n}/\partial p_{\text{wf},j}$ to be the largest when $j = n$. Thus, the damping factor is defined by

$$D_{n,j} = \frac{\frac{\partial \bar{q}_{\ell,n}}{\partial p_{\text{wf},j}}}{\frac{\partial \bar{q}_{\ell,n}}{\partial p_{\text{wf},n}}} = \frac{\overline{WI}_{\ell,j}[p_u(\Delta t_{j-1}) - p_u(\Delta t_j)]}{\overline{WI}_{\ell,n}[p_u(\Delta t_{n-1}) - p_u(\Delta t_n)]}. \tag{2.27}$$

Eqs. 2.25 and 2.27 are derived by analogy with single phase flow, but are applied to damp the derivatives of phase flow rates. To make the damping procedure as clear as possible, we introduce some concise notation. We let $p_{\text{wf}}^{i,k}$ denote the pressure control of well k at control step i and let the average phase rate of phase ph at a specific producer, producer j at the



(a) Adjoint and stochastic derivatives of q_o at P1 at $11th$ control step with respect to all controls.

(b) Adjoint and stochastic gradients of q_o at P1 at $11th$ control step with respect to its controls.

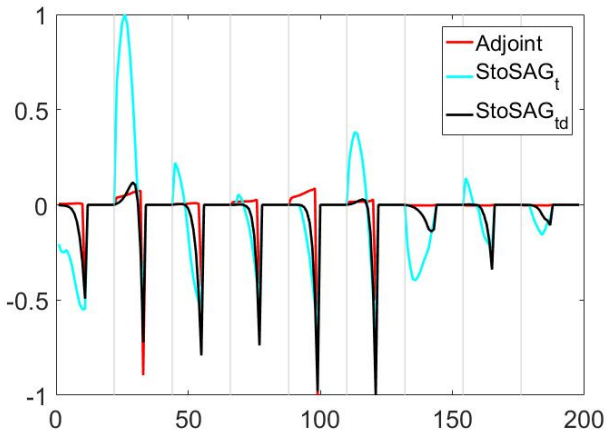
Figure 2.2: Comparison of adjoint and stochastic derivatives of q_o of well P1 at 11th control step.

specific control step n be denoted by $\vec{q}_{ph}^{n,j}$ with derivatives

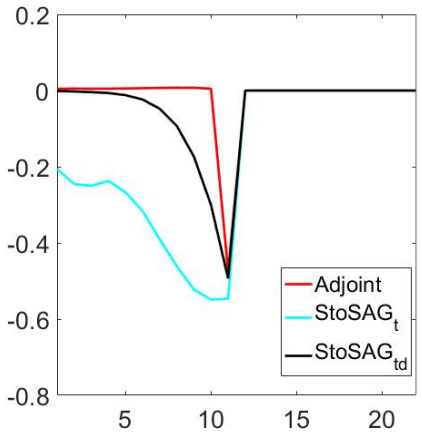
$$\frac{\partial \vec{q}_{ph}^{n,j}}{\partial \vec{p}_{\text{wf}}^{i,k}}, \quad i = 1, 2, \cdots, n, \quad k = 1, 2, \cdots, N_w, \tag{2.28}$$

for all $n = 1, 2, 3, \cdots, N_c$; recall that if $i > n$, the derivative of such phase rate with respect to well controls will be truncated to zero. After computing the derivatives in Eq 2.28 by the standard StoSAG procedure they are damped by multiplying by $D_{n,i}$ for $i = 1, 2, \cdots, n$ regardless of whether $k = j$ or $k \neq j$.

The derivative obtained with the adjoint gradient, the StoSAG$_t$ gradient and the StoSAG gradient with truncation and temporal damping (StoSAG$_{td}$) are compared in Fig. 2.2 where each gradient is normalized by dividing it by its infinity norm. Fig. 2.2a presents the various derivatives of q_o at well P1 at the 11th control step with respect to controls at all wells at all control steps. Fig. 2.2b presents the various derivatives involved in the gradient of q_o at P1 at 11th control step with respect to only its own control variables. The results of Fig. 2.2 strongly suggest that the direction of StoSAG$_{td}$ gives a significantly superior approximation of the direction of the adjoint derivatives than does StoSAG$_t$. Figs 2.3, 2.4 and 2.5, respectively, compare the adjoint derivatives of the field liquid production rates (FLR), the field water

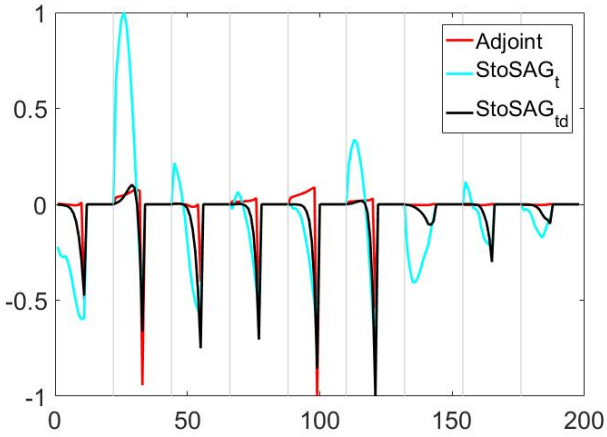


(a) Comparison of the derivatives of the FLR at 11th control step with respect to all controls (Adjoint & StoSAG$_t$ & StoSAG$_{td}$).

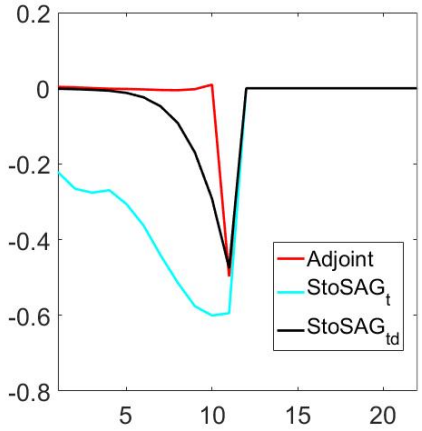


(b) Comparison of the derivatives of the FLR at 11th control step with respect to well P1 controls (Adjoint & StoSAG$_t$ & StoSAG$_{td}$).

Figure 2.3: Comparison of adjoint and stochastic gradients of FLR at 11th control step with respect to control.

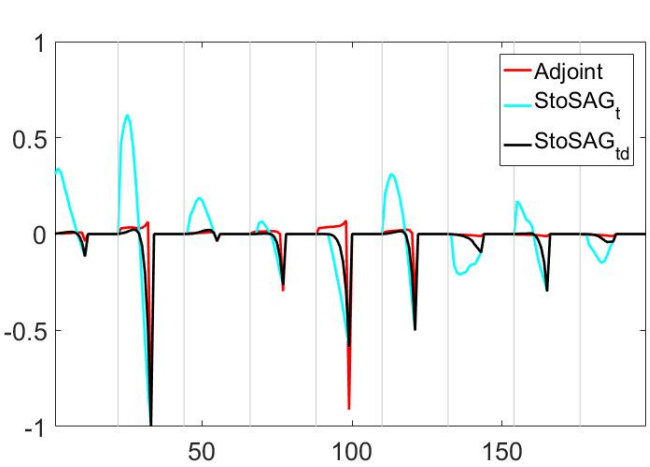


(a) Comparison of the derivatives of the FWR at 11th control step with respect to all controls (Adjoint & StoSAG$_t$ & StoSAG$_{td}$).

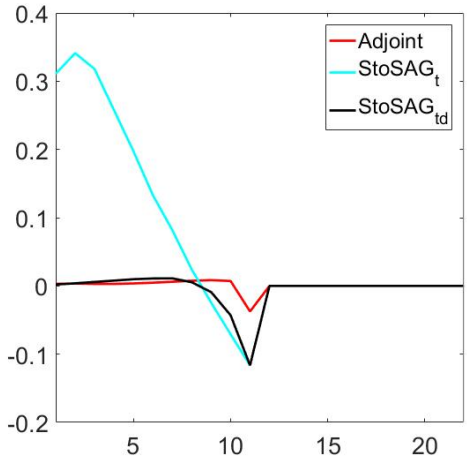


(b) Comparison of the derivatives of the FWR at 11th control step with respect to well P1 controls (Adjoint & StoSAG$_t$ & StoSAG$_{td}$).

Figure 2.4: Comparison of adjoint and stochastic gradients of FWR at 11th control step with respect to all controls.

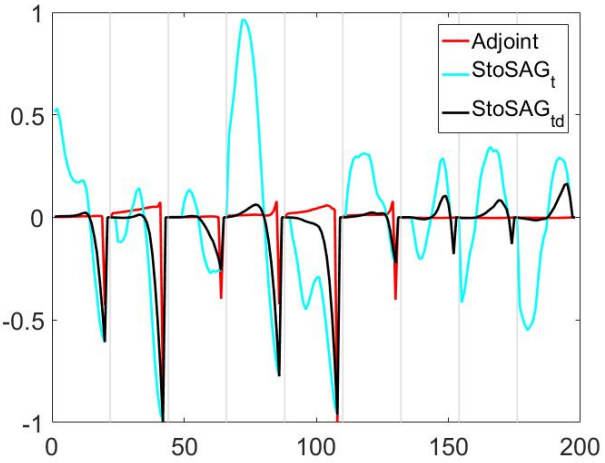


(a) Comparison of the derivatives of the FGR at 11th control step with respect to all controls (Adjoint & StoSAG$_t$ & StoSAG$_{td}$).

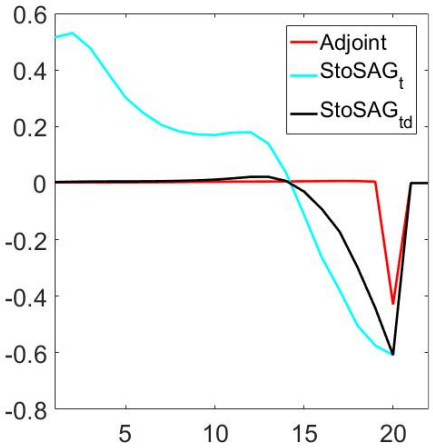


(b) Comparison of the derivatives of the FGR at 11th control step with respect to well P1 controls (Adjoint & StoSAG$_t$ & StoSAG$_{td}$).

Figure 2.5: Comparison of adjoint and stochastic gradients of FGR at 11th control step with respect to all controls.

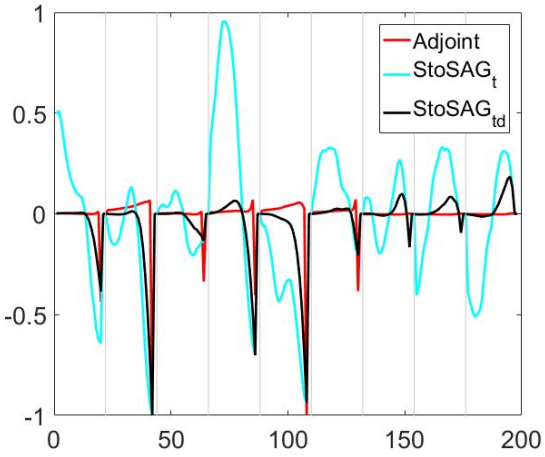


(a) Comparison of the derivatives of the FLR at 20th control step with respect to all controls (Adjoint & StoSAG$_t$ & StoSAG$_{td}$).

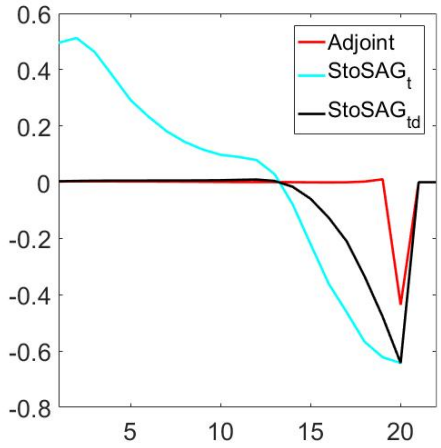


(b) Comparison of the derivatives of the FLR at 20th control step with respect to well P1 controls (Adjoint & StoSAG$_t$ & StoSAG$_{td}$).

Figure 2.6: Comparison of adjoint and stochastic gradients of FLR at 20th control step with respect to all controls.

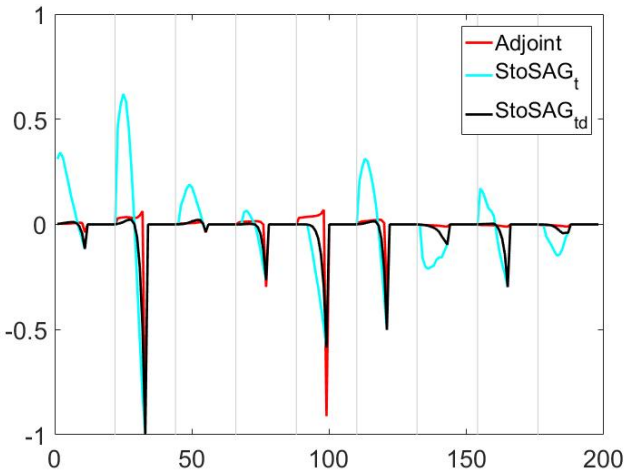


(a) Comparison of the derivatives of the FWR at 20th control step with respect to all controls (Adjoint & StoSAG$_t$ & StoSAG$_{td}$).

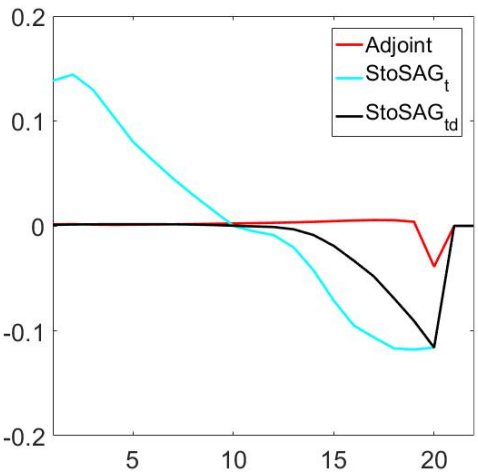


(b) Comparison of the derivatives of the FWR at 20th control step with respect to well P1 controls (Adjoint & StoSAG$_t$ & StoSAG$_{td}$).

Figure 2.7: Comparison of adjoint and stochastic gradients of FWR at 20th control step with respect to all controls.

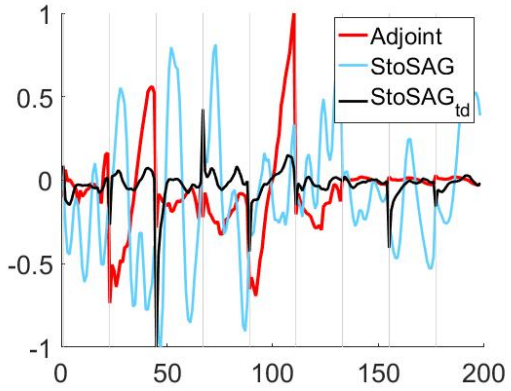


(a) Comparison of the derivatives of the FGR at 20th control step with respect to all controls (Adjoint & StoSAG$_t$ & StoSAG$_{td}$).

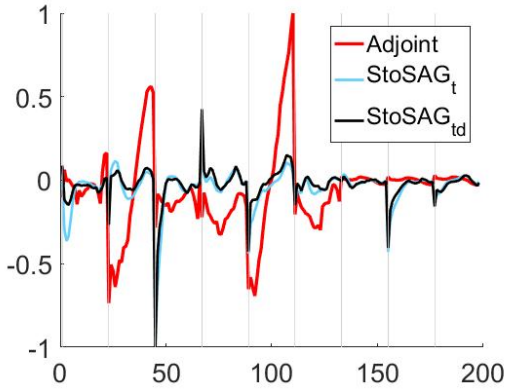


(b) Comparison of the derivatives of the FGR at 20th control step with respect to well P1 controls (Adjoint & StoSAG$_t$ & StoSAG$_{td}$).

Figure 2.8: Comparison of adjoint and stochastic gradients of FGR at 20th control step with respect to all controls.

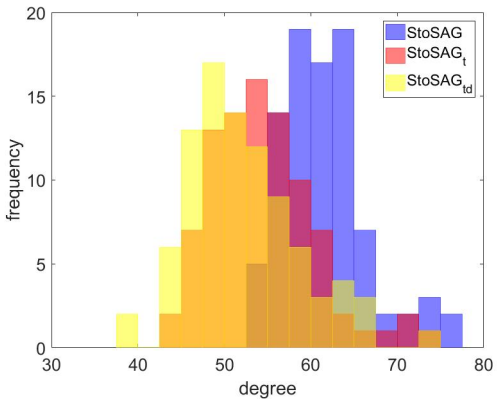


(a) Comparison of the derivatives of the negative NPV with respect to all controls (Adjoint & StoSAG & StoSAG$_{td}$).

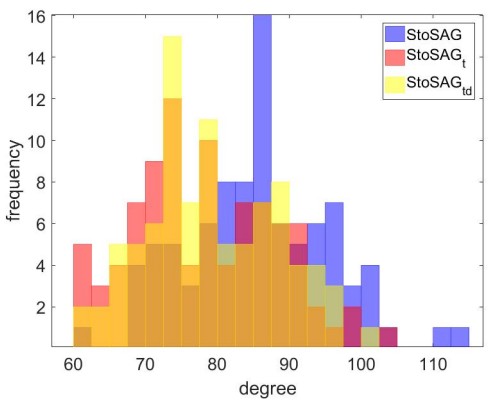


(b) Comparison of the derivatives of the negative NPV with respect to all controls (Adjoint & StoSAG$_t$ & StoSAG$_{td}$).

Figure 2.9: Comparison of adjoint and stochastic gradients of negative NPV with respect to all controls.



(a) Comparison of the quality of the gradient of the oil production rate of the P1 at the $11th$ control step.



(b) Comparison of the quality of the gradient of the negative NPV based on 90 stochastic gradients.

Figure 2.10: Estimation of the quality of gradient; x axis is the angle between the adjoint and stochastic gradient ($\circ$); y axis represemts the frequency of each stochastic gradient.

production rates (FWR) and the field gas production rate (FGR) at control step 11 with respect to all well controls with the corresponding derivatives generated by StoSAG$_t$ and StoSAG$_{td}$. In figs 2.6, 2.7 and 2.8, respectively, we compare the adjoint derivative (red lines) of the field liquid production rates (FLR) and field water production rates (FWR) and the field gas production rates (FGR), respectively, at control step 20 with respect to all well controls with the corresponding derivatives generated by StoSAG$_t$ (light blue lines) and StoSAG$_{td}$ (black lines). The results indicate the scheme with both truncation and damping, StoSAG$_{td}$, generates the most accurate gradient of the field production rates. Although the modified StoSAG derivative obtained by truncation and damping gives better estimates of the adjoint derivative of field rates than are obtained by truncation only, this conclusion only weakly extends to the derivatives of NPV. Fig. 2.9b indicates that the stochastic gradient of the negative NPV obtained with both truncation and damping is not much different from the one obtained by truncation only. Note Fig. 2.9 indicates that both modified stochastic gradients give a fairly poor estimate of the adjoint gradient of the true negative NPV. At the end of this subsection, we give a plausible explanation of why our modification of the gradient does not significantly improve the gradient of NPV but does improve the gradients of state constraints. It is important to note that obtaining an accurate derivative of phase rates is critically important for computing the gradients of nonlinear constraints because these gradients determine the feasible region for constrained minimization of the negative NPV using the sequntial quadratic programming algorithm. Thus, improving the quality of the rate derivatives that appear in constraints is critically important.

It is important to emphasize again that a stochastic gradient does not need to give the correct magnitude of the gradient to be useful in optimization. What is important is that a stochastic gradient of a function to be minimized provides a downhill direction in which one can obtain a reasonable decrease in the function to be minimized. Note that a stochastic gradient provides a downhill direction if the angle between the stochastic gradient and the true gradient is less than 90°. (When a stochastic gradient at an iteration of steepest descent does not give a downhill search direction, it is often possible to generate a downhill

direction by simply generating new perturbations and recomputing the stochastic gradient.) In order to evaluate the quality of the gradient, similar to the results of Fig. 2.2, we consider the gradient of the oil rate of well P1 at the 11th control step. However, here we compute 90 stochastic gradients where each stochastic gradient is computed using 20 perturbations with perturbation size (δ_u in Eq. 2.11) equal to 0.001 times the range of the control variables. Then, we compute the angles between each of the 90 stochastic gradients and the adjoint gradient where this angle is computed by

$$\theta = \arccos\Big(\frac{(\nabla_{u,ad} q_{o,1}^{11} \cdot \nabla_{u,sto} q_{o,1}^{11})}{||\nabla_{u,ad} q_{o,1}^{11}||_2 \cdot ||\nabla_{u,sto} q_{o,1}^{11}||_2}\Big), \tag{2.29}$$

where $\nabla_{u,ad} q_{o,1}^{11}$ and $\nabla_{u,sto} q_{o,1}^{11}$, respectively, denote the adjoint gradient and the stochastic gradient with respect to the complete vector of controls and $||\cdot||_2$ represents the two-norm of the gradient. The histogram of the angles in degrees between the stochastic gradient and the adjoint gradient based on the 90 angles computed from the 90 stochastic gradients is shown in Fig. 2.10. From the results of Fig. 2.10a, it is clear that the distribution of the angles based on StoSAG$_{td}$ shows a distinct shift to the left compared with other stochastic gradients, i.e., on average, StoSAG$_{td}$ provides a direction closer to that of the true gradient than those directions obtained with the other stochastic gradients. Moreover, the angle between the truncated gradient (StoSAG$_t$) and the adjoint gradient is generally smaller than the angle between normal StoSAG and the adjoint gradient. Although not shown, the gradient of injection rates and the gradient of oil and water rates at other producers show results that are qualitatively similar to the results of Fig. 2.10a. Thus, we conclude that by combining the zero truncation and temporal damping schemes, we improve the quality of gradients of the production phase rates. The gradients of rates determine directly the gradients for state constraints on total field rates. Fig. 2.10b compares the quality of the three stochastic gradients of the negative NPV functional in Eq. 2.12. Again the angle between the adjoint gradient and the three stochastic gradients are compared based on the same 90 perturbations used to compare the quality of the oil rate in Fig. 2.10a. Again the two

modified stochastic gradients tend to give a search direction closer to the search direction represented by the adjoint gradient than does the unmodified StoSAG but the improvement in direction is far more modest than the improvement in the gradient of the oil rate. If the angle between stochastic gradient and adjoint gradient is smaller than 90 degrees, the stochastic gradient provides a downhill direction in the minimization process. The results of Table 2.1 indicate that StoSAG$_t$ and StoSAG$_{td}$ provide a higher quality gradient of NPV than the original StoSAG algorithm. However, the quality of StoSAG$_{td}$ is not superior to be quality of StoSAG$_t$ for computing the gradient of NPV. As the gradient of the negative NPV is a linear combination of individual rates at individual control steps, one might expect that quality of the stochastic gradient of negative NPV would directly reflect the quality of the gradient of a phase rate, e.g., that the histograms of angles of Fig. 2.10b and Fig. 2.10a would be more similar. However, the difference between the angle of the stochastic gradient of a specific flow rate at a specific control step and the adjoint gradient may be different at each control step and for each rate so when we form the negative NPV by combining all the associated rate gradients, the histogram of angles between the adjoint gradient and any particular stochastic gradient may be quite different than a histogram of angles between rate gradients. Moreover, when we form the linear combination of phase rates of different magnitudes to compute the gradient of NPV, we cannot scale each gradient by its infinity norm before combining the gradients. Thus, even if the stochastic gradient of each and every flow rate at each and every control step is the same direction as the corresponding adjoint gradient, unless the magnitude of each stochastic gradient is the same as its adjoint-gradient counterpart, the linear combination of the stochastic gradients may give a significantly different search direction than the linear combination of the adjoint-gradient counterparts. On the other hand, nonlinear state constraints are applied on each control step and do not involve combining rates at different control steps; thus, the accuracy of the gradients of constraints on field rates more closely reflects the accuracy of gradients of flow rates of a single well at single control step.

On the basis of the results of Table 2.1 for our SQP-filter framework, we apply only

	60-70°	70-80°	80-90°	>90°
StoSAG	5	19	42	24
StoSAG$_t$	18	35	25	12
StoSAG$_{td}$	14	39	25	12

Table 2.1: Statistical frequency of angles between adjoint gradient and different stochastic gradients (gradients of NPV)

the gradient-truncation scheme to improve the NPV gradient but use both truncation and damping when computing the stochastic derivatives of rates that are involved in state constraints.

2.2.4 Sequential Quadratic Programming Filter (SQP-filter)

Sequential quadratic programming (SQP) is a optimization algorithm which is designed to solve the general constraints optimization problem. The optimization problem of interest in this work can be written as

$$\underset{u\in\mathbb{R}^{\mathrm{N}u}}{\text{minimize}} \quad \bar{J}(u), \tag{2.30a}$$

$$\text{subject to} \quad c_i(u) \geq 0, \;\; i = 1, 2, \cdots, \hat{N}_{\mathrm{ic}}. \tag{2.30b}$$

where the total number of inequality constraints, $\hat{N}_{ic}$ includes enforcing on all realizations the original inequality constraints plus the original bound constraints which have now been converted to inequality constraints. Thus, $\hat{N}_{ic} = N_{ic} \times N_e + N_{lc}$. Recall that N_{ic} is the total number of nonlinear inequality state constraints which have to be enforced on all N_e realizations; N_{lc} denote the number of bound constraints which are converted to linear inequality constraints when applying the SQP algorithm. Any bound constraints which are not converted to general inequality constraints are handled by the truncation. More details on the handling of bound constraints within SQP are discussed later. Throughout, we let λ_{c}

denote the vector of Lagrangian multipliers for inequality constraints i.e.,

$$\lambda_c = [\lambda_{c,1} \quad \lambda_{c,2} \quad ... \quad \lambda_{c,\hat{N}_{ic}}]^T, \tag{2.31}$$

The Lagrangian function of the constrained problem (Eq. 2.30) is given by

$$L(u, \lambda_{\mathrm{c}}) = \bar{J}(u) - \lambda_c^T c = \bar{J}(u) - \sum_{i=1}^{\hat{N}_{ic}} \lambda_{\mathrm{c},i} c_i, \tag{2.32}$$

where $c = [c_1, c_2, \cdots c_{\hat{N}_{ic}}^T]$ is the vector of constraint functions. A solution of the problem in Eq. 2.30 must satisfy the first order optimality conditions referred to as the Karush-Kuhn-Tucker (KKT) conditions [67], which are given by

$$\nabla_u L(u, \lambda_{\mathrm{c}}) = 0, \tag{2.33a}$$

$$c_i(u) \geqslant 0 \qquad , i = 1, 2, \ldots, \hat{N}_{ic}, \tag{2.33b}$$

$$\lambda_{\mathrm{c,i}} \geqslant 0 \qquad , i = 1, 2, \ldots, \hat{N}_{ic}, \tag{2.33c}$$

$$\lambda_{\mathrm{c},i} c_i(u) = 0 \qquad , i = 1, 2, \ldots, \hat{N}_{ic}. \tag{2.33d}$$

In the SQP framework, the set of KKT conditions are solved in order to find the optimal solution of the constrained optimization problem. The basic idea of SQP is to approximate each KKT condition (Eq. 2.33) with its first order Taylor series expansion and replace the KKT conditions with linearized approximations. Using the linear approximations of the KKT conditions at the $(\ell+1)$th iteration, the optimality conditions are approximated

by

$$\nabla_u L(u^{\ell+1}, \lambda_c^{\ell+1}) \approx \nabla_u L(u^\ell, \lambda_c^\ell) + \nabla_u^2 L(u^\ell, \lambda_c^\ell)\delta u^\ell + \nabla_{u,\lambda_c}^2 L(u^\ell, \lambda_c^\ell)\delta\lambda_c^\ell = 0, \tag{2.34a}$$

$$c_i(u^{\ell+1}) \approx c_i(u^\ell) + \delta u^{\ell T}\nabla_u c_i(u^\ell) \geqslant 0, \qquad i = 1, 2, \ldots, \widehat{N}_{ic}, \tag{2.34b}$$

$$\lambda_{\mathrm{c}} \geqslant 0. \tag{2.34c}$$

$$\lambda_{\mathrm{c,i}}[c_i(u^\ell) + \delta u^{\ell T}\nabla_u c_i(u^\ell)] = 0 \qquad i = 1, 2, \ldots, \widehat{N}_{ic}, \tag{2.34d}$$

where δu^ℓ and $\delta\lambda_c$, respectively, denote the change in u and the change in λ_c from the ℓth to $(\ell+1)$th iteration step i.e., $u^{\ell+1} = u^\ell + \delta u^\ell$ and $\lambda_c^{\ell+1} = \lambda_c^\ell + \delta\lambda_c$. From Eq. 2.32, it follows that $\nabla_u L(u^\ell, \lambda^\ell)$ is given by

$$\nabla_u L(u^\ell, \lambda_c^\ell) = \nabla_u \bar{J}(u^\ell) - (\lambda_c^\ell)^T A_c^\ell = \nabla_u \bar{J}^\ell - (A_c^\ell)^T \lambda_c^\ell. \tag{2.35}$$

where A_c denotes the Jacobian matrices of the inequality constraints and is given by

$$A_c = [\nabla c_1, \quad \nabla c_2, \quad \ldots \quad \nabla c_{\widehat{N}_{ic}}]^T. \tag{2.36}$$

Note that in the following derivation, the gradients of the objective function $\bar{J}$ with respect to control variables u at the ℓth iteration is written as $\nabla_u \bar{J}^\ell$. Here, $Y^\ell \equiv \nabla_u^2 L(u^\ell, \lambda^\ell)$ denotes the Hessian of the Lagrange function, which usually is expensive to estimate. Therefore, the BFGS algorithm [1, 67] is implemented to approximate the Hessian of the Lagrange function. With BFGS, the Hessian matrix $Y^{\ell+1}$ is approximated with both the current gradient information $(\nabla_u \bar{J}^{\ell+1}, A_c^{\ell+1})$ and the previous gradient information $(\nabla_u \bar{J}^\ell, A_c^\ell)$. The initial guess of the approximate Hessian matrix, Y^0, is defined as the $N_u \times N_u$ identity matrix. The updating formula of BFGS is given by

$$Y^{\ell+1} = Y^\ell + \frac{\gamma^\ell(\gamma^\ell)^T}{\delta u^{\ell T}\gamma_\ell} - \frac{Y^\ell \delta u^\ell \delta u^{\ell T}(Y^\ell)^T}{\delta u^{\ell T} Y^\ell \delta u^\ell}, \tag{2.37}$$

where

$$\begin{aligned}\gamma^{\ell} &= \nabla_u L(u^{\ell+1}, \lambda^{\ell+1}) - \nabla_u L(u^{\ell}, \lambda^{\ell+1}) \\ &= (\nabla_u \bar{J}^{\ell+1} - \nabla_u \bar{J}^{\ell}) - (A_c^{\ell+1} - A_c^{\ell})^T \lambda^{\ell+1},\end{aligned} \tag{2.38}$$

By taking the gradient of Eq. 2.35 with respect to λ_c^{ℓ}, we find

$$\nabla^2_{u,\lambda_c} L(u^{\ell}, \lambda_c^{\ell}) = -(A_c^{\ell})^T. \tag{2.39}$$

By inserting Eqs. 2.35 and 2.39 into Eq. 2.34a, it follows that

$$\nabla_u \bar{J}^{\ell} - (A_c^{\ell})^T \lambda_c^{\ell} + \nabla_u^2 L(u^{\ell}, \lambda_c^{\ell}) \delta u^{\ell} - (A_c^{\ell})^T \delta\lambda_c^{\ell} = 0. \tag{2.40}$$

Introducing the notation, $Y^{\ell} \equiv \nabla_u^2 L(u^{\ell}, \lambda^{\ell})$ into Eq. 2.40 and rearranging the resulting equation gives

$$Y^{\ell}\delta u^{\ell} + \nabla_u \bar{J}^{\ell} - (A_c^{\ell})^T(\lambda_{\mathrm{c}}^{\ell} + \delta\lambda_{\mathrm{c}}^{\ell}) = Y^{\ell}\delta u^{\ell} + \nabla_u \bar{J}^{\ell} - (A_c^{\ell})^T \lambda_{\mathrm{c}}^{\ell+1} = 0. \tag{2.41}$$

We can rewrite the linearized KKT conditions of Eq. 2.34 as:

$$Y^{\ell}\delta u^{\ell} + \nabla_u \bar{J}^{\ell} - (A_c^{\ell})^T \lambda_{\mathrm{c}}^{\ell+1} = 0, \tag{2.42a}$$

$$A_c \delta u^{\ell} \geqslant -c, \tag{2.42b}$$

$$\lambda_{\mathrm{c}}^{\ell} \geqslant 0. \tag{2.42c}$$

$$(\lambda_{\mathrm{c,i}}^{\ell})(A_{c,i}^{\ell}\delta u^{\ell} + c_i) = 0, \quad i = 1, 2, \ldots, \hat{N}_{ic}, \tag{2.42d}$$

where $A_{c,i} = \nabla c_i^T$, $i = 1, 2, \cdots, \hat{N}_{ic}$. The above linearized approximate KKT conditions are the exact KKT conditions of the following quadratic programming (QP) problem, which is solved at every optimization iteration:

$$\underset{\delta u \in \mathbb{R}^{N_u}}{\text{minimize}} \quad \frac{1}{2}\delta u^{\ell T} Y^{\ell} \delta u^{\ell} + \delta u^{\ell T} \nabla_u \bar{J}(u^{\ell}), \tag{2.43a}$$

$$\text{subject to:} \quad A_c \delta u^{\ell} \geqslant -c^{\ell}. \tag{2.43b}$$

This QP problem can either be solved by active-set or interior point methods [67]. We do not discuss the details of solving this QP problem, since there are many algorithms and free-license software available to solve it [67]. To be specific, the Matlab code used to solve QP problem is `quadprog`. Therefore, the code to solve the ℓth QP problem should be : `quadprog`$(Y^{\ell},\ \nabla_u \bar{J}^{\ell},\ -A_c^{\ell},\ c^{\ell})$. The solution of this QP problem, δu^{ℓ}, is the vector used to update the vector of control variables, u^{ℓ}, to $u^{\ell+1} = u^{\ell} + \delta u^{\ell}$.

Note that enforcing all constraints for all models is computationally expensive because we have to approximate the gradient of $N_{ic} \times N_e$ state constraints for robust optimization. To be specific, $\hat{N}_{ic} = N_{ic} \times N_e + N_{lc}$ where N_{lc} is the the number of bound constraints which are selected to be handled by SQP algorithm. If all the bound constraints are enforced by the SQP-filter algorithm as a general inequality constraints then $N_{lc} = 2N_u$. If the upper bound of the i entry of the control vector u are handled by SQP algorithm that upper bound constraint can be represented as $c_{\text{up,i}} = u_i^{\text{up}} - u_i \geq 0$. The lower bound of the ith entry of the control vector u can be converted to the inequality constraint given by $c_{\text{low,i}} = u_i - u_i^{\text{low}} \geq 0$. The corresponding gradient of the upper bound constraint of the ith variable can be written as

$$\nabla_u c_{\text{up,i}} = \nabla_u (u_i^{\text{up}} - u_i) = [0...0 \quad -1 \quad 0....0]^T. \tag{2.44}$$

Note that the gradient vector $\nabla_u c_{\text{up,i}}$ is a $N_u \times 1$ vector, which has all elements equal to zero except the ith element which is equal to -1. Similarly, the gradient of the lower bound constraint of the ith control variable is given by

$$\nabla_u c_{\text{low,i}} = \nabla_u (u_i - u_i^{\text{low}}) = [0...0 \quad 1 \quad 0....0]^T \tag{2.45}$$

where the gradient vector $\nabla_u c_{\text{low,i}}$ has all elements equal to zero except the ith element which is equal to 1. The Jacobian matrices of the inequality constraints A_c should incorporate the gradient information of bound constraints which are considered as a general inequality constraints in SQP. For instance, if we consider the upper bound and lower bound of the ith control variable u_i as general inequality constraints for all i, $i = 1, 2, \cdots N_u$ and j, $j = 1, 2, \cdots N_e$ then the Jacobian matrix of all inequality constraints can be written as

$$A_c = [\nabla c_{1,1},\, \nabla c_{2,1},\, \cdots \nabla c_{N_{ic},1},\, \cdots \nabla c_{1,j},\, \nabla c_{2,j},\, \cdots \nabla c_{N_{ic},j}, \cdots \nabla c_{1,N_e},\, \nabla c_{2,N_e},\, \cdots \nabla c_{N_{ic},N_e}$$
$$\nabla c_{\text{up},1},\, \nabla c_{\text{low},1,}\, \cdots \nabla c_{\text{up,i}},\, \nabla c_{\text{low,i}},\, \cdots \nabla c_{\text{up},N_u},\, \nabla c_{\text{low},N_u}]^T. \quad (2.46)$$

Unfortunately, the dimension of A_c is $\hat{N}_{ic} \times N_u$ where $\hat{N}_{ic} = N_{ic} \times N_e + 2N_u$ when all bound constraints are converted to general inequality constraints. Thus, A_c matrix can be an extremely large dense matrix if there are even a handful of constraints that need to be enforced on a large number of control steps for several realizations of the reservoir model.

The second option for handling bound constraints is to simply apply truncation. To be specific, if the ith entry of the updated control vector at iteration ℓ is such that u_i^ℓ is greater than u_i^{up}, we set $u_i^\ell = u_i^{\text{up}}$ directly. If ith entry of the updated control vector at iteration ℓ is such that u_i^ℓ is less than u_i^{low}, we set $u_i^\ell = u_i^{\text{low}}$. Compared to the truncation scheme, handling the bound constraints by converting them to general inequality constraints is more theoretically sound and reduces the likelihood of being trapped at a local minimum when the optimal solution is close to its bound constraints. However, converting the bound constraints to general inequality constraints will increase the number of inequality constraints and lead the SQP problem to be computationally expensive when the number of control variables N_u of the production optimization problem is large. Furthermore, for some bound constraints, there is nothing to gain by converting them to general inequality constraints. For example, the upper bound on BHP for the BHP controlled producers should be handled by truncation when the upper bound of the BHP is set equal to or slightly less that the initial reservoir pressure, since during production optimization if this bound is exceeded the well could not

produce. Similarly, lower bounds on water injection rates are set as zero and it is physically impossible for an injection rate to be less than zero at any control step so it makes sense to enforce a zero lower bound on a well's water injection rate at a time step equal to zero. On the other hand, if the lower bound on BHP for one or more producing wells is set relatively high, then the actual optimal controls at some control steps may be close to the lower bound. In this scenario, it is appropriate to convert the lower bound constraints on BHP to inequality constraints and incorporate their gradients directly into the Jacobian of inequality constraints (see Eq. 2.46) in the application of SQP.

As mentioned previously, to enforce every constraint for every realization of the reservoir model at every control step is generally not computationally feasible. Thus, we consider alternatives that are more computationally efficient. The first option we try is to require that each constraint only be satisfied on average, i.e., we replace Eq. 2.3c by

$$\bar{c}_i(u) = \frac{1}{N_e}\sum_{j=1}^{N_e} c(u, m_j) \geqslant 0 \quad i = 1, 2, \ldots, N_{ic}. \tag{2.47}$$

where the bound constraints under this averaging constraints enforcement scenario is handled by truncation. In this case, in Eqs 2.32 through 2.43 , $\hat{N}_{ic}$ is replace by N_{ic} and c of Eq 2.32 is replaced by

$$c = [\bar{c}_1(u) \quad \bar{c}_2(u) \quad \ldots \quad \bar{c}_{N_{ic}}(u)]^T. \tag{2.48}$$

In SQP, we implement the filter method to determine whether we accept or reject a proposed update (δu^ℓ) to the estimate of the optimal well controls at the ℓth iteration step. The filter method has been applied before for constrained well control in conjunction with derivative free algorithms [43]. The filter method effectively views constrained optimization as a bi-objective minimization of the objective function $\bar{J}(u)$ and the constraint violation $\bar{\theta}(u)$. For the general case of Eq, 2.46 where we convert all bound constraints to inequality constraints and explicitly enforce all constraints for all realizations, the (average) constraint

violation function is defined by

$$\bar{\theta}(u) = \frac{1}{N_{\mathrm{e}}} \sum_{j=1}^{N_{\mathrm{e}}} \Big[\sum_{i=1}^{N_{ic}} c_i^+(m_j, u) \Big] + \sum_{c=1}^{N_{lc}} c_c^+(u), \tag{2.49}$$

where

$$c_i^+(m_j, u) = |\min_{1 \leqslant i \leqslant N_{ic}} \{c_i(m_j, u), 0\}|. \tag{2.50}$$

and

$$c_c^+(u) = |\min_{1 \leqslant i \leqslant N_{lc}} \{c_c(u), 0\}|. \tag{2.51}$$

Note N_{lc} denote the number of bound constraints which are converted to the general inequality constraints and $c_c^+(u)$ are not dependent on the realization of reservoir model m_j. For the case under consideration where we only enforce that the constraints are satisfied on average and bound constraints are enforced by truncation, i.e., require that Eq. 2.47 holds Eq. 2.49 is replaced by

$$\bar{\theta}(u) = \sum_{i=1}^{N_{ic}} \bar{c}_i^+(u), \tag{2.52}$$

where

$$\bar{c}_i^+(u) = |\min\{\bar{c}_i(u), 0\}| \text{ for } i = 1, 2, \cdots N_{ic}, \tag{2.53}$$

where $\bar{c}_i$ is defined in Eq. 2.47.

Unlike penalty methods, which combine both cost function and constraints into a single minimization problem, the filter method keeps the constraints and objective function separate. Let the filter F, be a set of previously generated pairs of $(\bar{\theta}_k, \bar{J}_k)$, say for $k = 1, 2, \cdots n_\ell$. A proposed new iterate $u^{\ell+1}$ is accepted by the filter criteria whenever it satisfies either

$$\bar{\theta}(u^{\ell+1}) \leqslant (1 - \gamma_\theta)\bar{\theta}_k \quad \text{or} \quad \bar{J}(u^{\ell+1}) \leqslant \bar{J}_k - \gamma_J \bar{\theta}(u^{\ell+1}) \quad , \tag{2.54}$$

for all $(\bar{\theta}_k, \bar{J}_k) \in F$. Here γ_θ and γ_J are defined parameters with small positive values. The first inequality in Eq. 2.54 enforces a sufficient reduction in the average value of the constraint violation $\bar{\theta}$. The dependence on $\bar{\theta}(u^\ell)$ is it $u^{\ell+1}$ here of is it u^ℓ in Eq. 2.53. in the

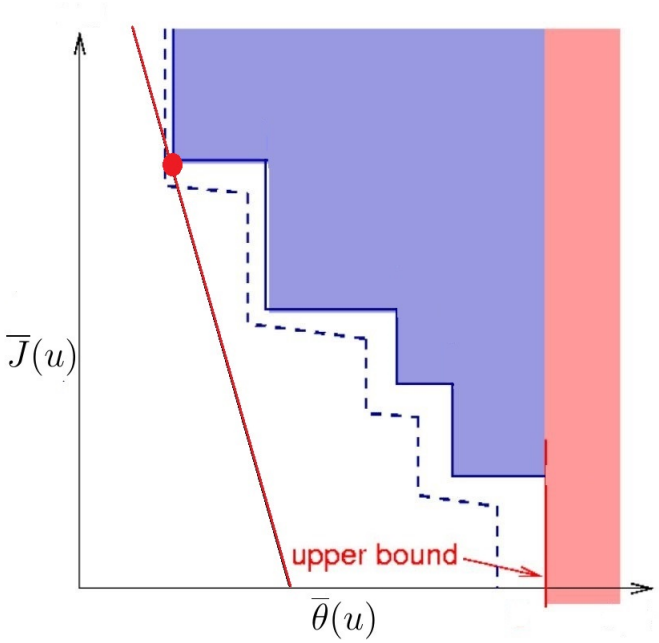


Figure 2.11: The acceptance criteria for the filter method, red point correspond to $u = u^{\ell}$, after Nocedal and Wright [67]

second inequality is a technical requirement to enable convergence to be proven [28]. These inequalities avoid acceptance of an iterate $u^{\ell+1}$ if $(\bar{\theta}(u^{\ell+1}), \bar{J}(u^{\ell+1})$ is very close to a point that is already a point in F, i.e., points that fall in the empty envelope (the region between the dashed line and solid color patches) in Fig. 2.11 are not accepted. More specifically, the algorithm does not accept iterates that fall on the right hand side of the envelop (dashed line). After a new pair $(\bar{\theta}_{\ell+1}, \bar{J}_{\ell+1}) \equiv (\bar{\theta}(u^{\ell+1}), \bar{J}(u^{\ell+1}))$ has been accepted, we delete all old pairs in the filter, $(\bar{\theta}_k, \bar{J}_k)$'s in F, for which $\bar{\theta}_k > \bar{\theta}(u^{\ell+1})$ and $\bar{J}_k > \bar{J}(u^{\ell+1})$, i.e., delete points from the filter that are dominated by the point, $(\bar{\theta}(u^{\ell+1}), \bar{J}(u^{\ell+1}))$.

Compared with the ℓ_1 merit function or other penalty methods, the filter method is a moderate criteria which allows the acceptance of more proposed updates. As shown in Fig 2.11, any update whose corresponding objective function and overall constraint violation is located to the left of dashed curve will be accepted by the filter algorithm as new estimate of the optimal u. In Fig. 2.11, the red dot represents the current iteration u^{ℓ}. As shown in Nocedal and Wright [67], the penalty method would only accept $u^{\ell+1}$ as the new estimate of the optimum if the point $(\bar{\theta}(u^{\ell+1}, \bar{J}(u^{\ell+1}))$ falls to the left of the red line in Fig. 2.11. Thus,

it is clear that the filter method makes it easier to accept an updated optimum than does the penalty method. Thus, the SQP-filter should be more suitable to our situation than a penalty method since both gradients with respect to the objective function and nonlinear constraints are approximated via a stochastic gradient approximation scheme, where a high rejection frequency in proposed updates could lead to the termination of the optimization algorithm at a highly suboptimal solution.

For the optimization algorithm based the stochastic gradient, instead of updating on the original well control vector u, the optimization algorithm are required to update on the normalized well control vector x, since it is hard to compute the stochastic gradient correctly in magnitude. A [0, 1] scaling is used to normalize the entries of the vector of well controls, which is a commonly used normalization scheme for production optimization problem with stochastic gradient [31, 12, 33]. Recall from the definition of u, which is defined in Section 2.2.1, that u denotes the vector of all N_{u} controls which includes the controls of every well at every time step. Let u_i be the ith component of u with upper bound u_i^{up} and lower bound u_i^{low}, then the scaled version, x, of u is formed by mapping u_i to x_i given by

$$x_i = \frac{u_i - u_i^{\mathrm{low}}}{u_i^{\mathrm{up}} - u_i^{\mathrm{low}}}. \tag{2.55}$$

Moreover, the x_i also can be scaled back to u_i as

$$u_i = x_i(u_i^{\mathrm{up}} - u_i^{\mathrm{low}}) + u_i^{\mathrm{low}}. \tag{2.56}$$

Note that after the [0, 1] scaling, any entry of well control vector x_i which the corresponding original entry u_i is within upper bound u_i^{up} and lower bound u_i^{low} will be bounded within in the range of [0, 1]. Therefore, [0, 1] will be the control interval for the normalized control vector x. In the context of production optimization with stochastic gradient, the update direction of the normalized control vector should also be normalized to a reasonable range which ensures the update direction for normalized vector δx neither very big to have that most entries of vector x be bounded at the bound 1 or 0 nor very small to have negligible

effect on the objective function. Therefore, update vector of the normalized control vector at the ℓth iteration δx^ℓ is defined as

$$\delta x^\ell = \rho^\ell \frac{\delta u^\ell}{||abs(\delta u^\ell)||_\infty}, \tag{2.57}$$

where δu is the solution of the QP problem defined in Eq. 2.43; ρ^ℓ denote the updating step size of the normalized vector at the ℓ iteration. After normalizing of δx^ℓ as the Eq. 2.57 above, the maximum and minimum entries in δx^ℓ should be bounded within the $[-\rho^\ell, \rho^\ell]$. For instance, if $\rho^\ell = 0.2$, the maximum allowable update step size at the ℓth iteration is the 20% of the control interval, since the normalized control vector x is bounded within [0, 1].

The summary of the SQP-filter algorithm follows:

Algorithm 2.1: SQP-filter.

1. Set ℓ=0. Select initial values of the control variables, u^0 and normalize it to x^0 by Eq. 2.55. Set the maximum number of iterations, n_{maxiter}, e.g., $n_{\text{maxiter}} = 100$. Set the initial Hessian matrix Y^ℓ to the identity matrix I_{N_u}. Set values of γ_θ and γ_J in Eq. 2.54, e.g., $\gamma_\theta = 0.01$ and $\gamma_J = 0.02$. Set the value of the convergence criteria parameter ϵ_x, e.g., ϵ_x=0.0001. Here we use normalized search direction δx and initialize δx^0 as a random vector to make sure that $||abs(\delta x^0)||_\infty > \epsilon_x$. Set the maximum allowable step size $\rho_{\max} = 0.2$ and we also set initial step size $\rho^0 = 0.2$.

2.
   - **IF** $||abs(\delta x^\ell)||_\infty < \epsilon_x$ **OR** $\ell \geq n_{maxiter}$
     - **Terminate the algorithm.**
   - **ELSE**
     - Approximate the gradient of the objective function $\bar{J}$ with respect to the control variables, $\nabla \bar{J}^\ell$ by StoSAG$_t$; compute the Hessian matrix Y^ℓ by the BFGS algorithm; compute the vector of constraints c^ℓ and the gradient of inequality constraints with respect to control variables, A_c^ℓ, by StoSAG$_{td}$.

3. Solve the QP problem of Eq. 2.43 for the search direction to obtain δu^ℓ. Normalize the update search direction as $\delta x^\ell = \rho^\ell \frac{\delta u^\ell}{||abs(\delta u^\ell||_\infty)}$.

4. **IF** QP solution is infeasible **THEN**

   - Enter restoration phase and return to normal SQP when $x^{\ell+1}$ is found whose corresponding QP is feasible. (see [29, 67] for discussion of the restoration phase) After finding the update, δx^{ℓ}, from the restoration phase, then go back to step 2.

5. **ELSE**

   - Set $x^{\ell+1} = x^{\ell} + \delta x^{\ell}$ and scaling $x^{\ell+1}$ back to $u^{\ell+1}$ by Eq. 2.56. Evaluate the filter pair $(\bar{J}^{\ell+1}, \bar{\theta}^{\ell+1})$ at $u^{\ell+1}$, where $\bar{\theta}^{\ell+1} = \bar{\theta}(u^{\ell+1})$, and $\bar{J}^{\ell+1} = \bar{J}(u^{\ell} + 1)$.
   - **IF** $(\bar{J}^{\ell+1}, \bar{\theta}^{\ell+1})$ is acceptable to the filter i.e., if Eq. 2.54 holds, **THEN**
     - Accept the normalized control vector $x^{\ell+1}$ and original control vector $u^{\ell+1}$ and add $(\bar{J}^{\ell+1}, \bar{\theta}^{\ell+1})$ to the filter.
     - Remove points dominated by $(\bar{J}^{\ell+1}, \bar{\theta}^{\ell+1})$ from the filter.
     - Increase the updating step as $\rho^{\ell+1} = \max(2\rho^{\ell}, \rho_{\max})$.
     - $\ell = \ell + 1$ and go back to step 2.
   - **ELSE**
   - set n_p=0, and $\rho^{\ell,0} = \rho^{\ell}$.
     - **REPEAT** cut the updating step as $\rho^{\ell,n_p} = 0.5 * \rho^{\ell,n_p}$.
     - Solve the QP problem for the search direction $\delta u^{\ell,n_p}$ and normalize $\delta x^{\ell} = x\rho^{\ell,n_p} \frac{\delta u^{\ell}}{||abs(\delta u^{\ell})||_{\infty}}$.
     - Set $x^{\ell+1,n_p} = x^{\ell} + \delta x^{\ell,n_p}$ and scale $x^{\ell+1,n_p}$ back to $u^{\ell+1,n_p}$, Evaluate the filter pair $(\bar{J}^{\ell+1,n_p}, \bar{\theta}^{\ell+1,n_p})$ at $u^{\ell+1,n_p}$.
     - **IF** $(\bar{J}^{\ell+1,n_p}, \bar{\theta}^{\ell+1,n_p})$ is acceptable to the filter **THEN**
       * Accept $x^{\ell+1,n_p}$ and $u^{\ell+1,n_p}$. Add $(\bar{J}^{\ell+1,n_p}, \bar{\theta}^{\ell+1,n_p})$ to the filter.
       * Remove all points dominated by $(\bar{J}^{\ell+1,n_p}, \bar{\theta}^{\ell+1,n_p})$ from the filter.
       * Increase the updating step as $\rho^{\ell+1} = \max(2 * \rho^{\ell,np}, \rho_{\max})$.
       * $\ell = \ell + 1$ and go back to step 2.
     - **ELSE IF** $n_p > 3$
       * Accept $x^{\ell+1,n_p}$ and $u^{\ell+1,n_p}$ i.e., set $x^{\ell+1} = x^{\ell+1,n_p}$ and $u^{\ell+1} = u^{\ell+1,n_p}$.

* Reset the Hessian matrix $Y^{\ell+1}$ to the identity matrix I_{N_u}.

* $\ell = \ell + 1$ and go back to step 2.

– **ELSE**

* $n_p = n_p + 1$;

– **END IF**

• **END IF**

2.2.5 Min-Max Scheme

As another alternative to explicitly enforcing all constraints for every reservoir model at every time step, we propose here a min-max scheme to implicitly satisfy all constraints for each realization m_j. Instead of constraining only the average values of nonlinear state constraints Eq. 2.47, or enforcing every constraint for every realization Eq. 2.3, the min-max algorithm minimizes the maximum violation of each constraint. For the min-max, the optimization problem of Eqs.(2.3 is replaced by Eq. 2.58)

$$\underset{u\in\mathbb{R}^{N_u}}{\text{minimize}} \quad \bar{J}(u), \quad \text{average NPV} \tag{2.58a}$$

$$\text{subject to} \quad u_i^{\text{low}} \leq \ u_i \ \leq \ u_i^{\text{up}} \quad , i = 1, 2, \ldots, N_u, \tag{2.58b}$$

$$c_{i,j}(u) \geqslant 0 \quad , i = 1, 2, \ldots, N_{\text{ic}}, \tag{2.58c}$$

where $c_{i,j}$ denotes value the ith constraint of jth realization, where the ith constraint has its maximum violation when evaluated at the jth realization. This means that for each i, i =1,2,... N_{ic}, j is chosen such that

$$j = \underset{1\leqslant k\leqslant N_e}{\text{argmin}}\{c_i(u, m_k)\}. \tag{2.59}$$

By enforcing $c_{i,j} \geqslant 0$ at each iteration of the SQP-filter method, we simply minimize the maximum constraint violation. The index of j corresponding to each nonlinear constraint

may dynamically change from iteration to iteration as the optimization proceeds. There is no guarantee that enforcing each constraint on the model that has the maximum constraint violation will lead to the satisfaction of all constraints for all reservoir models. However, the min-max scheme will result in an estimated optimum such that all constraints are satisfied if the models that give the maximum constraint violations stabilize, i.e., stop changing from iteration to iteration. In any case, computations indicate that the use of the min-max scheme in our SQP filter optimization algorithm produces a solution such that the violation of any constraint for any of the N_e reservoir models is negligible.

When we only enforce that the constraints averaged over the set of reservoir models are nonnegative, $N_p \times N_e$ perturbations are used to compute the gradient of the average value of the nonlinear state constraint, N_p perturbations for each reservoir model. For robust optimization where only a few realizations, say 5-10, are used to represent geological uncertainty, we generally use three perturbations per reservoir model. However, for the mim-max scheme where the $c_{i,j}$ of Eq. 2.58c are applied individually rather than averaged, we found that using only three perturbations per constraint can sometimes lead to inaccuracies in the stochastic gradients that severely degrade the performance of the SQP-filter algorithm. Thus, we wish to increase the number of perturbations for realizations for which constraints are most significantly violated but to avoid computational inefficiency, we wish to minimize the number of additional perturbations. In order to find a reasonable balance between the computational cost and the accuracy of the gradients of the constraints in the min-max scheme, a voting method is proposed to allocate the number of perturbations for each realization dynamically.

First, we define the total number of extra perturbation to be used for constraint enforcement with the min-max scheme as N_{ep}. Then the number of perturbations for the jth realization is defined by:

$$N_{p,j} = N_p + \text{int}\Big(\frac{\sum_{i=1}^{N_{ic}} |\min_{1 \leqslant k \leqslant N_e} (c_i(m_k, u)\delta_{k,j}, 0)|}{\bar{\theta}(u)} N_{ep}\Big). \tag{2.60}$$

where $\delta_{k,j}$ is defined as

$$\delta_{k,j} = \begin{cases} 1, & \text{if } j = k \\ 0, & \text{otherwise} \end{cases} \tag{2.61}$$

where int is used to denote the greatest integer function and

$$\bar{\theta}(u) = \sum_{i=1}^{N_{\text{ic}}} |\min_{1 \leqslant k \leqslant N_e} (c_i(m_k, u), 0)|. \tag{2.62}$$

In our procedure, we set $N_{p,j} = N_p$, whenever $\overline{\theta}(u) = 0$, i.e., when all constraints are satisfied for every realization. This means that when all constraints are satisfied, we can simply use the same N_p perturbations of the vector of controls used to compute the stochastic gradient of the average NPV to compute the gradients of the constraints.

2.3 Computational Results

In this section, we present two examples to illustrate the performance of our SQP-filter optimization workflow using the modified StoSAG gradient for constrained production optimization. In the first example, we consider optimization for a single reservoir model (deterministic optimization) so $N_e = 1$ in Eq. 2.3 and Eq. 2.49. This example is intended to demonstrate the effectiveness of the modified StoSAG algorithm, and to illustrate that without modification, using StoSAG to compute all gradients may lead to complete failure of the SQP-filter algorithm when nonlinear state constraints are included. In the second example, the Brugge case, we consider geological uncertainty and compare the performance of the following three procedures for handling nonlinear state constraints; (1) enforce constraints only on an average basis, i.e., apply Eq. 2.47; (2) apply the min-max scheme to enforce constraints; and (3) ignore state constraints when doing optimization and after estimating the optimal controls, input them in the reservoir simulator and let the simulator apply its internal heuristics to enforce the constraints.

2.3.1 Example 1: Single Reservoir Model

In this example, a synthetic 3D, 3-phase reservoir simulation model with an orthogonal Cartesian grid system is built. The reservoir simulation model has a $20 \times 20 \times 8$ grid so there are 8 reservoir simulation layers. To test the optimization framework, we only perform the SQP-filter algorithm on a single geological realization. The permeability distributions for the layers are shown in Fig. 2.12. We assume this reservoir has a homogeneous porosity field with the porosity value of each reservoir gridblock as 0.2.The first layer is initially occupied by gas and the 8th layer contains only water. The initial pressure of the reservoir is 4,400 psi. The irreducible water saturation and residual oil saturation are both equal to 0.2. Nine vertical wells have been drilled and completed to develop this reservoir with the well locations shown in Fig. 2.12. Producers P2 and P5 are perforated in layers 4-7 and producers P1, P3, P4 and P6 are perforated in layers 6-7. All the injectors (I1, I2 and I3) are completed only in layer 8 and inject only water.

The length of each control step is set equal to 180 days. Thus, there are 22 control steps for each well and the total number of control variables to be estimated is $9 \times 22 = 198$. For producers, the control variables are their bottom hole pressures (BHPs) at 22 control steps and the controls of injectors are their water injection rates at control steps. The lower and upper bounds on BHP for each producer are 3,000 psi and 4,600 psi, respectively. The lower and upper bounds on water injection rates for each injector are 0 STB/d and 500 STB/d. Since the lower bound on BHP of producers are set relative high in this case, we expect that at the optimal controls, there are many producers at many control steps operating close or at the lower bound of the BHP. Therefore, the lower bound constraints for all producers are converted to general inequality constrains. Other bound constraints, such as the upper bound of the BHP for the producers and upper and lower bound of water injection rates for the injectors, are enforced by the truncation scheme. The general inequality constraints handled by SQP-filter algorithm and the scaling of each type of those constraints in this problem are summarized as follows:

1. P1-P6: $c = \frac{p_{\text{wf}} - 3000}{1000} \geqslant 0$,

2. I1-I3: $c = \frac{4600 - p_{\text{wf}}}{1000} \geqslant 0$,

3. FLR: $c = \frac{12{,}000 - FLR}{12{,}000} \geqslant 0$,

4. FGR: $c = \frac{40{,}000 - FGR}{40{,}000} \geqslant 0$.

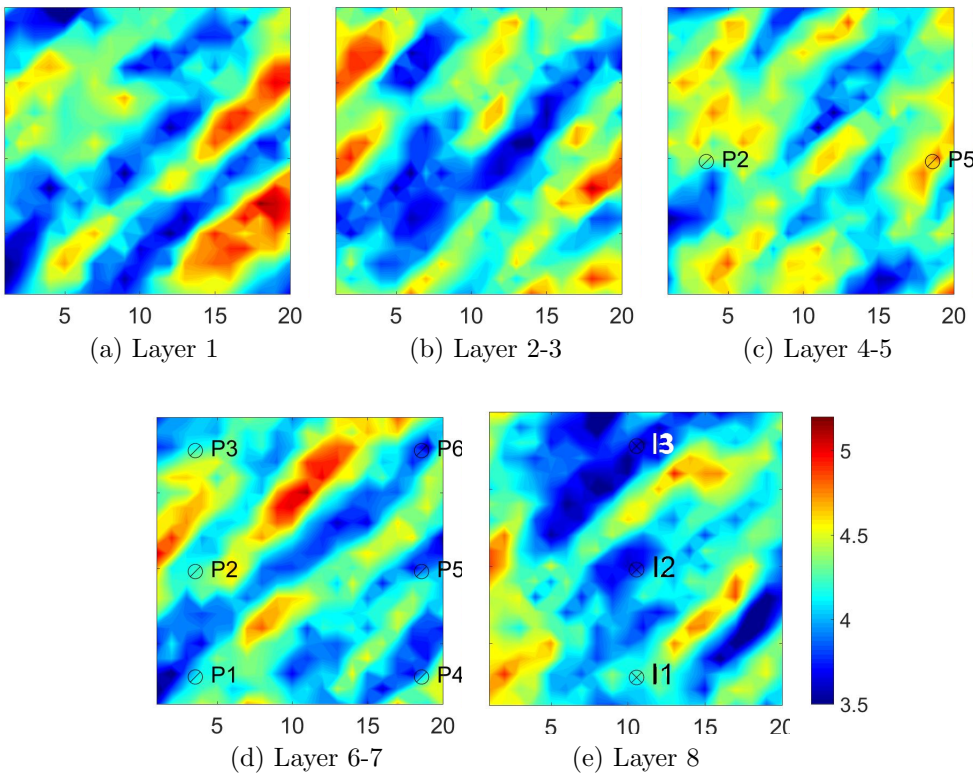
In the preceding two equations, the units of the FLR and FGR, respectively, are STB/day and Mscf/D. The lower bound of $3,000$ psi on BHP's at all six producers at all control steps represent $6 \times 22 = 132$ linear constraints. Since water injectors are rate controlled, the upper bounds on the BHP of each of the three injection wells at each of the 22 control steps represent $3 \times 22 = 66$ nonlinear state constraints. Finally, there are upper bounds on the field liquid rate (FLR) and field gas rate (FGR) at each of the 22 controls steps so the constraints on FLR and FGR represent $2 \times 22 = 44$ nonlinear state constraints. Therefore, the total number of constraints in the this example is $(6 + 3 + 2) \times 22 = 11 \times 22 = 242$.

Note that instead of scaling all the constraints to [0,1], we select slightly different scaling factors for the different type of constraints. Specifically, the scaling factors are set as 1000 for both pressure constraints. With this setting, the same pressure violation will give the same value of a pressure constraint violation; for example, if $p_{\text{wf}} = 2900$ psi at a producer and $p_{\text{wf}} = 4700$ psi at a injector, we have a 100 psi constraint violation in both cases and $c = -0.1$ in both cases. As the controls at producers are BHP's, the lower bound of 3,000 psi on p_{wf} is a linear constraint whereas the upper bound of 4,600 psi on p_{wf} at injection wells is a nonlinear state constraint since the injection wells are controlled by water injection rates. The other two constraints are nonlinear state constraints pertaining to the field liquid rate (FLR) and the field gas rate (FGR) at every control step. Specifically, we enforce a maximum field liquid production rate equal to 12,000 STB/day and a maximum field gas production rate equal to 40,000 Mcf/day. These type of constraints are facility constraints. Although the constraints should be enforced at each simulation time step, the number of simulation time steps can change at every simulation run, which would result in a change in the number of constraints at each iteration of the optimization algorithm. Therefore, in our problem, we enforce the nonlinear state constraints at every control step instead of at every

time step.

The number of perturbations used to approximate the stochastic gradient is set equal to 30. Since we solve a deterministic optimization problem in this example and we don't need the min-max scheme as proposed in Section 2.2.5. Therefore, the number of extra perturbations (N_{ep}) is set as 0 and the total number used to compute the stochastic gradient of both NPV and state constraint is 30. The perturbation size (δ_u in Eq. 2.11) is set as 0.1% of the control range of each control variable where the control range is defined as the difference between the upper bound and lower bound of each unscaled control variable. In order to impose a small amount of temporal smoothness on the well control perturbations, a temporal correlation matrix C_U is used as discussed previously where the correlation length is equal to the length of five control steps, i.e., equal to 900 days. More precisely, the correlation length is $L = 5$ in the covariance function of Eq. 2.11. For the NPV computations, the oil price is set to \$80/STB, the water disposal cost is \$3/STB, and water injection cost is set to \$2/ STB. The gas price is set to a low value of \$0.02/ Mcf, which assumes the gas must essentially be given away which is the case in some fields. The annual discount rate is 0.05. The initial guess for all BHP controls is equal to 3,500 psi and the initial guess for all water injection rates is equal to 100 STB/day.

Fig. 2.13 illustrates the optimal controls generated by the SQP-filter method. From the results, we can observe that most producers tend to restrict their production by maintaining a relatively high BHP during the first two years of production. After that the optimal BHP's of producers are lowered close to, or equal to, the lower bound on pressure of $3,000$ psi from around 540 days to 2160 days. After producing 2160 days, the BHPs of most producers at most control steps are close to, or equal to, the lower bound on BHP (3000 psi) which indicates that the lower bound constraints on BHP (converted to inequality constraints) are active. The water injection controls are also shown in Fig. 2.13. The results of Fig. 2.14 illustrate that the upper bound of 4,600 psi on p_{wf} (nonlinear state constraint) is active at injector I2 during the first control step but is not otherwise active. One can also note from the injection rate controls shown in Fig. 2.13, that most of the cumulative water injected is



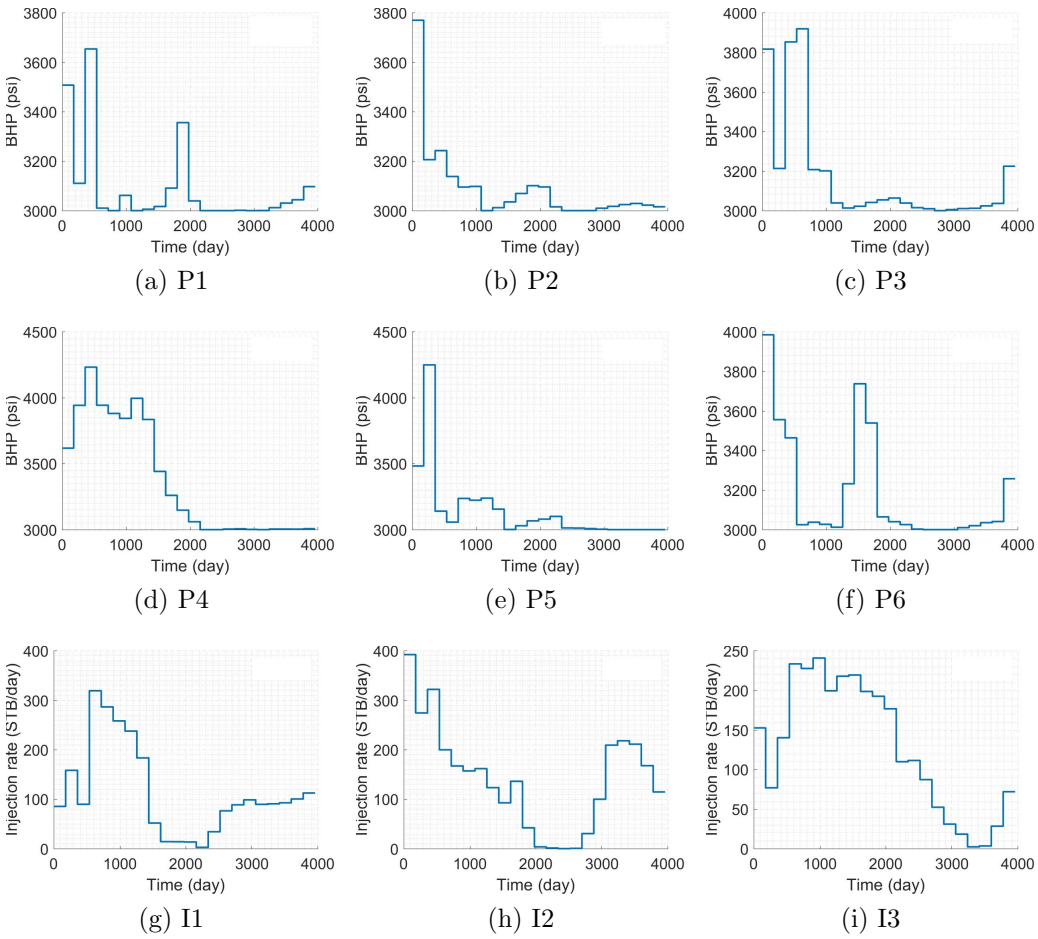
(a) Layer 1 (b) Layer 2-3 (c) Layer 4-5

(d) Layer 6-7 (e) Layer 8

Figure 2.12: Log permeability of fields in all layers.

injected during the first half of the reservoir life. The water and gas production rates of each well under the optimal controls is shown in Figs. 2.15 and 2.16, respectively.

Fig. 2.17 shows both the field liquid production rate (FLR) and field gas production rate (FGR) corresponding to the optimal controls. In Fig. 2.17b, we observe that the FGR is close to its upper bound during the first four control steps, i.e., the state constraints on FGR are active or nearly active so if BHP's at producers are lowered this constraint will be violated. This explains why all producers operate at high pressure at the first few control steps; see Fig. 2.13. Both Fig. 2.17a and 2.17b illustrate the ability of the SQP-filter to enforce the nonlinear state constraints. From the beginning to 900 days, the FGR constraint of 40,000 Mscf/D strongly influences the estimation of optimal controls. Then, from 900 to 2700 days, the upper bound of 12,000 STB/day on FLRs are active or nearly active and affect the estimation of optimal controls. The active FLR constraints at least partially explains why from 900 to 2700 days of the production period, all injectors tend to decrease their



(a) P1 (b) P2 (c) P3

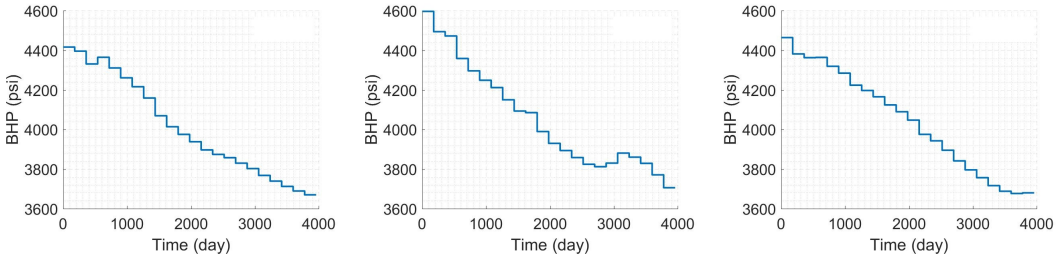
(d) P4 (e) P5 (f) P6

(g) I1 (h) I2 (i) I3

Figure 2.13: The optimal well controls; SQP-filter workflow.

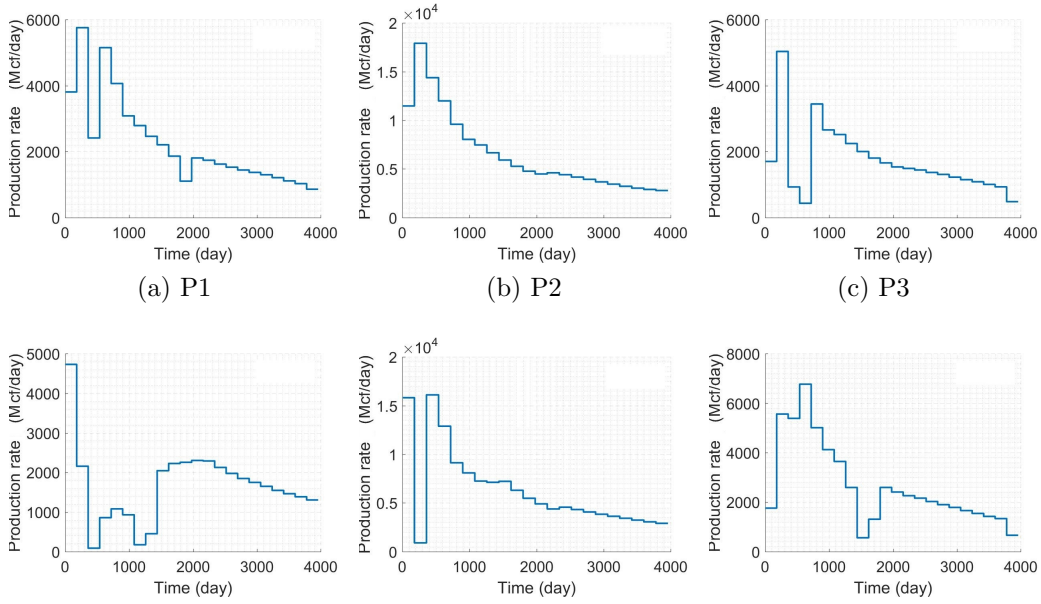
water injection rates to reduce FLRs.

The blue curve in Fig. 2.18, shows how the NPV changes from iteration to iteration of the SQP-filter algorithm using our modified StoSAG algorithm. The red curve of Fig. 2.18a shows the corresponding result obtained by direct implementation of the original StoSAG gradient into the SQP-filter algorithm; note that use of standard StoSAG results in premature termination of the algorithm at a highly sub-optimal solution. At each step of the SQP-filter, the QP problem defined in Eq. 2.43 is solved to compute the update direction of δu^{ℓ}. The feasible region of the QP subproblem is defined by the gradients of all constraints Eq. 2.43b. Recall that Fig. 2.9 illustrates that the standard StoSAG gradients of state constraints provide relatively poor approximations of the corresponding gradients computed with the adjoint method. Since the StoSAG gradients of nonlinear state constraints serve as linear



(a) I1 (b) I2 (c) I3

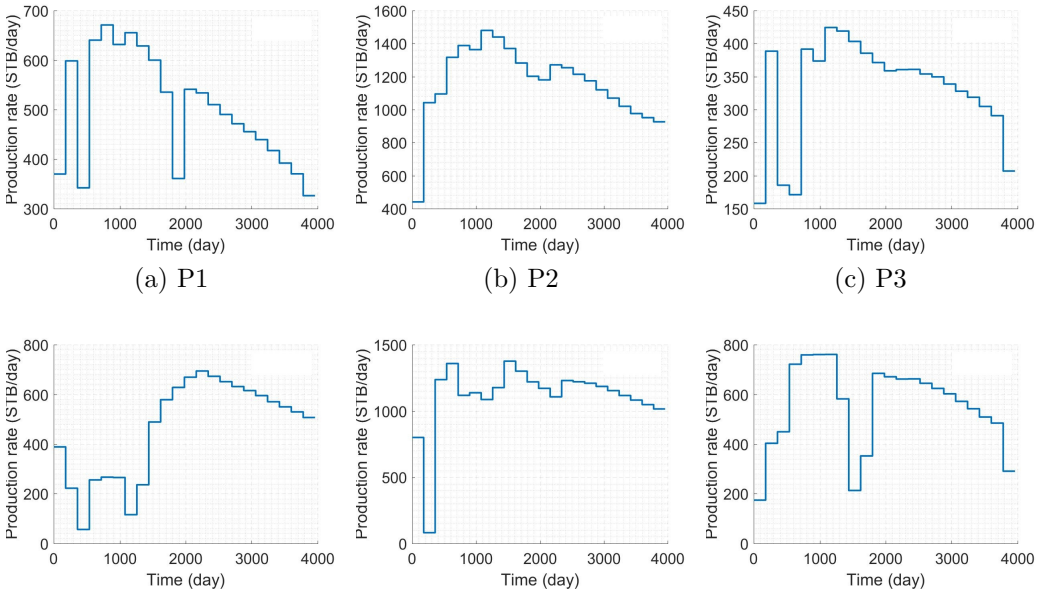
Figure 2.14: BHP at injection well at the optimal well controls; SQP-filter workflow.



(a) P1 (b) P2 (c) P3

(d) P4 (e) P5 (f) P6

Figure 2.15: Gas production rates at the optimal controls; SQP-filter workflow.

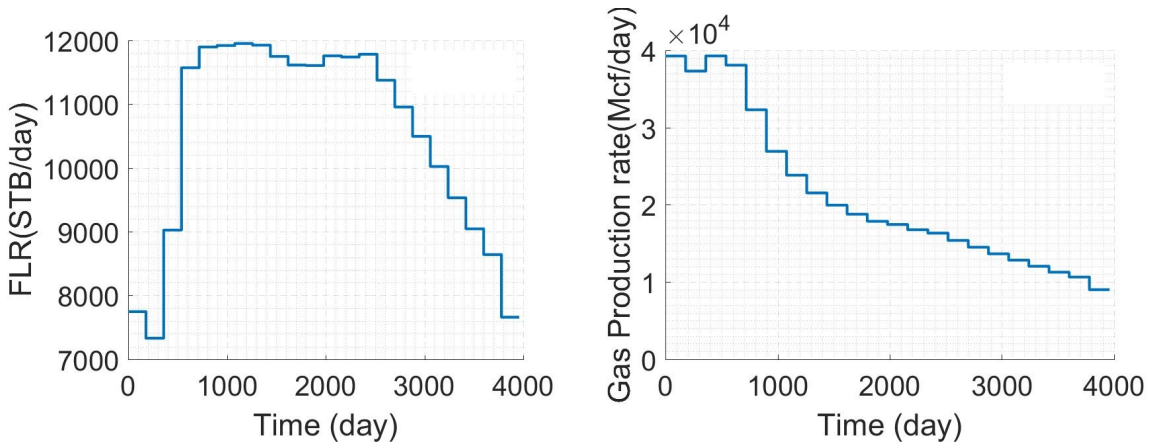


(a) P1 (b) P2 (c) P3

(d) P4 (e) P5 (f) P6

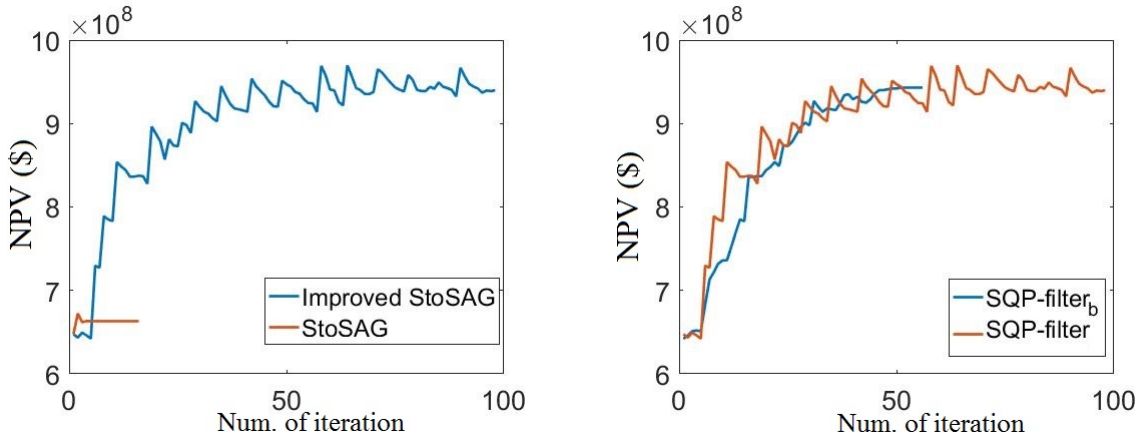
Figure 2.16: Water production rates at the optimal controls; SQP-filter workflow.

constraints for the QP problem, if they are of poor quality, it can easily result in a QP problem that is infeasible. To be specific, because of the poor quality of the standard StoSAG gradient, the restoration phase (see the 4th step of the SQP-filter algorithm presented previously) can continuously return a δx which makes the QP problem infeasible. When we repeat the cycle between step 2 of algorithm and the restoration phase (step 4), the step is repeatedly cut and the algorithm is terminated because $||abs(\delta x^{\ell})||_{\infty} \leq \epsilon_x$ even through we have not improved the objective function, and this is what causes the poor performance of Fig. 2.18a generated using the original StoSAG. However, if the gradients of nonlinear constraints are improved by truncation and temporal damping (StoSAG$_{td}$), the NPV increases albeit with some oscillations as shown in Fig. 2.18a. The oscillations result from the restoration of the constraint violations. To mitigate such oscillations, we can set an upper bound on the infeasibility of the filter. When the upper bound for the infeasibility is considered, a filter pair $(\bar{J}^{\ell+1}, \bar{\theta}^{\ell+1})$ will be rejected if $\bar{\theta}^{\ell+1} > \theta_{\max}$. Here, we set θ_{max}=2. From Fig 2.18b, we observe that the SQP-filter with this upper bound on $\bar{\theta}$ only takes 53 iterations to converge, where the SQP-filter without the bound requires 98 iterations to converge. As we mentioned previously, the number of perturbations to generate StoSAG gradient is set equal to 20.



(a) FLR (b) FGR

Figure 2.17: FLR and FGR at the optimal controls; SQP-filter workflow.



(a) Behavior of NPV with and without improved StoSAG gradient.

(b) Behavior NPV with and without an upper bound on $\bar{\theta}$;

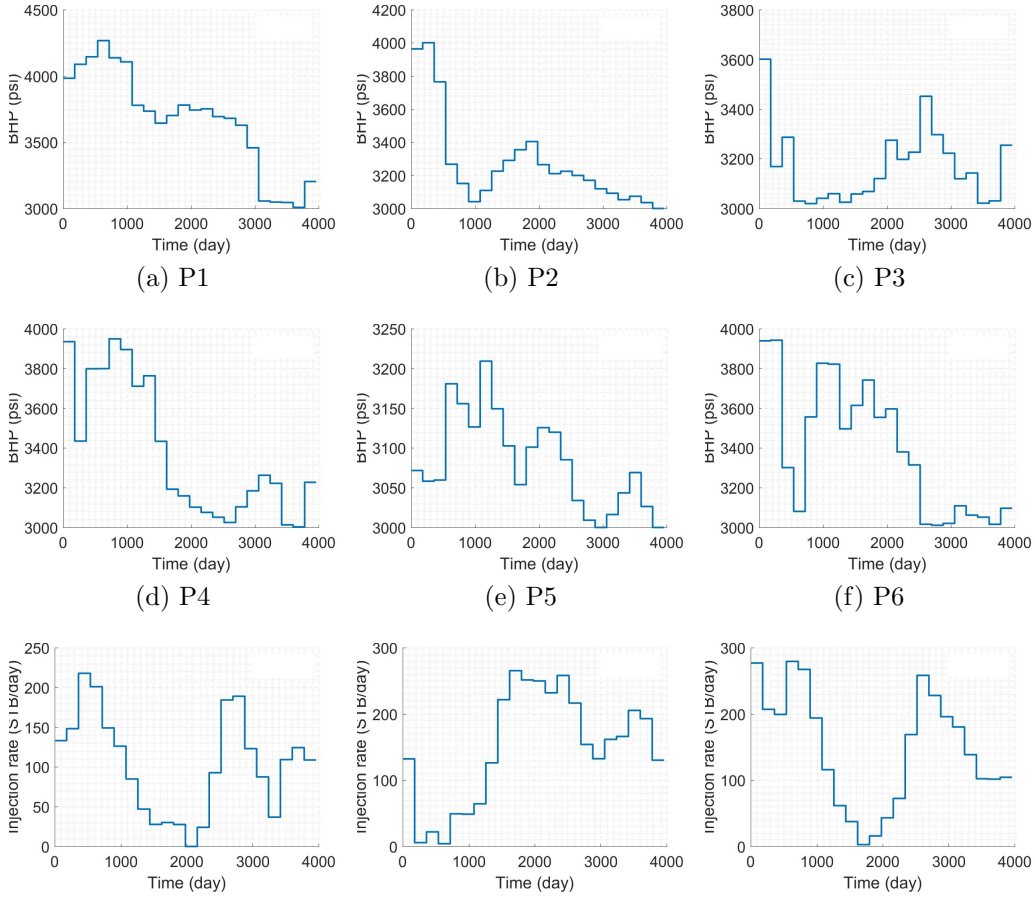
Figure 2.18: NPV versus number of iterations.

Therefore, roughly $(98-53)\times 20 = 900$ simulation runs are saved by setting an upper bound for $\bar{\theta}$, i.e., by setting an appropriate upper bound for $\bar{\theta}$, the efficiency of the SQP-filter algorithm can be significantly improved. When the upper bound $\theta_{\max} = 2$ is incorporated, the optimization require a total of 1475 reservoir simulation runs. The final NPV from the SQP-filter algorithm with an upper bound on feasibility is $\$9.42 \times 10^8$ whereas, without the bound, the NPV is $\$9.39 \times 10^8$. Thus, in this case, setting maximum feasibility only slightly affects the ultimate NPV, but using the bound improves computational efficiency.

To examine the performance of the SQP-filter framework on constraint handling, we change the maximum allowable FLR from 12,000 STB/day to 10,000 STB/day and keep all

other constraints the same. The optimal controls obtained with the modified constraints are shown in Fig. 2.19. Compared with the optimal controls of the previous case (Fig. 2.13), the BHP controls of producers are generally higher in order to satisfy the stricter constraint on the FLR. Moreover, we can observe that compared to the original problem with the less strict constraint on the FLR, the water injection rates of injectors are lower. For completeness, Fig. 2.20 and Fig. 2.21, respectively, show the gas and water rates at producers at the optimal controls and Fig. 2.22 shows the FLR and FGR under the new constraints. Note that except for the first few control steps, the upper bound of 10,000 STB/D on the FLR has the dominant effect on the results. Specifically, from time zero to 900 days, the FGR serves as the active constraint whereas, from roughly 900 to 3780 days, the FLR becomes the active constraint which leads to restrictions on producer BHP's and water injection rates. Compared to the results of Fig. 2.17a, the time where the FLR constraint is active or nearly active is longer in Fig 2.22 because of the lower (stricter) constraint on the FLR.

To investigate the effect of converting bound constraints to general inequality constraints, we run the exact same case by SQP-filter without converting the bound constraints to inequality constraints i.e., the lower bound constraints at BHP of producers are also handled by the truncation scheme. Table 2.2 summarizes the results of constrained deterministic optimization with different bound constraints handling schemes on the cases with FLR$\leqslant$12,000 and FLR$\leqslant$10,000, respectively. The reason why we consider these two cases is that from Fig. 2.13 the many optimal BHPs at difference control steps of producers with FLR$\leqslant$12,000 tend to operate on the lower bound of BHP (3,000 psi). While in the case with FLR$\leqslant$10,000, the only a few optimal BHPs are operating around the lower bound of BHP. By comparing the first row and second row in the Table 2.2, we can observe that by converting lower bound on BHP pressure of producers to general inequality constraints, the SQP-filter algorithm can yield higher life-cycle NPV and require a fewer number of simulations when the optimal BHPs are operated close to their lower bounds (Fig. 2.13). However, when the optimal BHPs are operated far from their lower bounds (Fig. 2.19), we can observe that there are no significant differences between the converting bound to inequality constraints

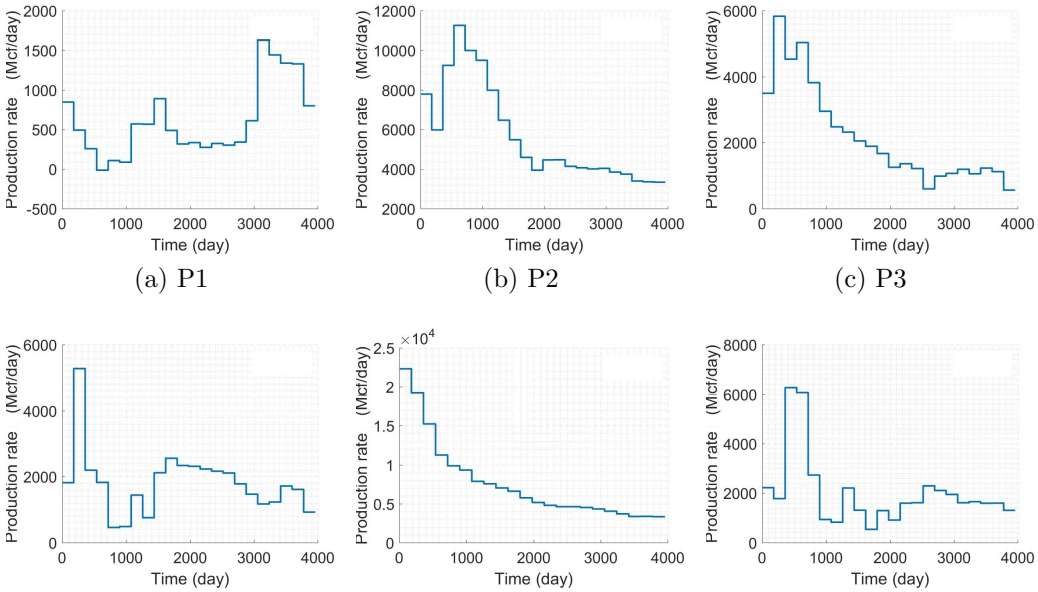


(a) P1 (b) P2 (c) P3

(d) P4 (e) P5 (f) P6

(g) I7 (h) I8 (i) I9

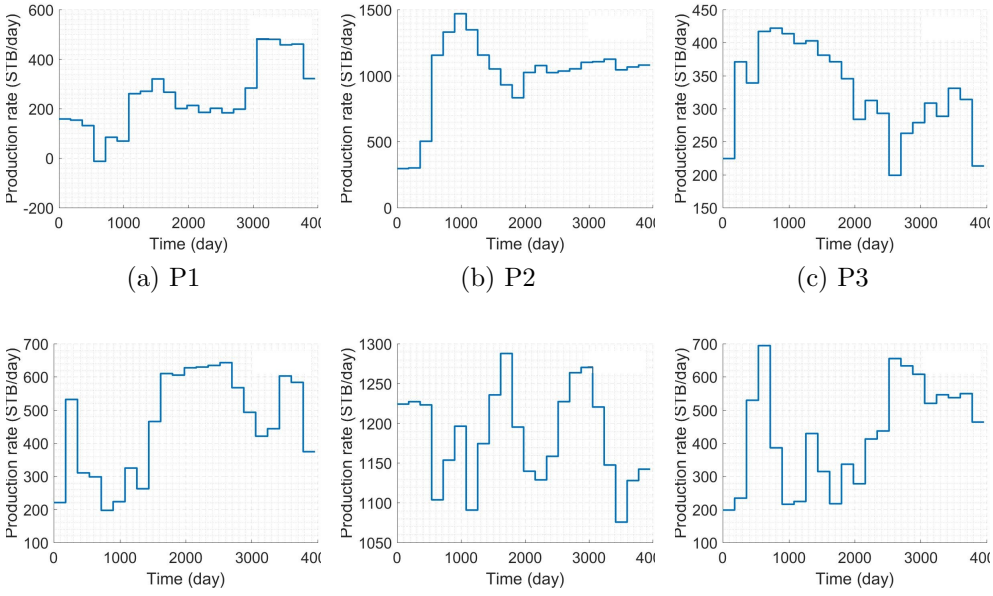
Figure 2.19: The optimal well controls; SQP-filter workflow. (stricter constraints)



(a) P1 (b) P2 (c) P3

(d) P4 (e) P5 (f) P6

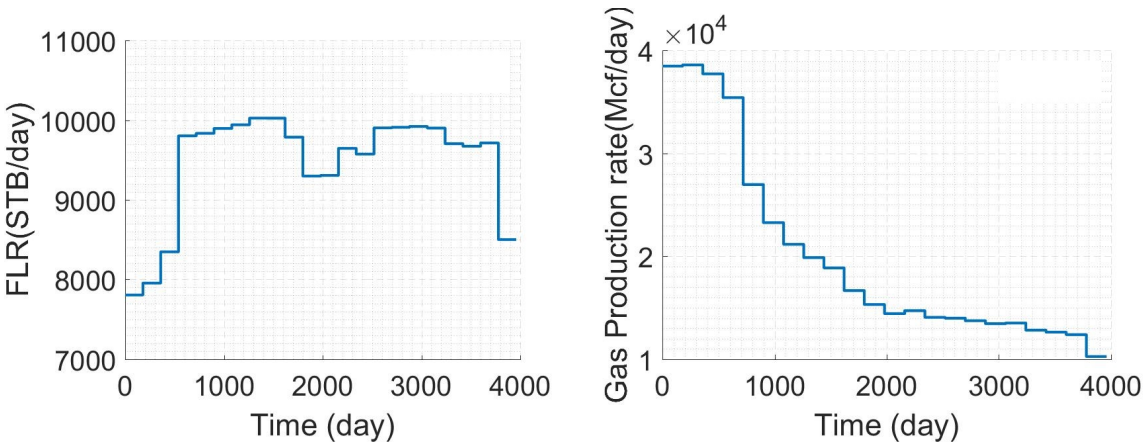
Figure 2.20: Gas production rates at the optimal controls; SQP-filter workflow. (stricter constraints)



(a) P1 (b) P2 (c) P3

(d) P4 (e) P5 (f) P6

Figure 2.21: Water production rates at the optimal controls; SQP-filter workflow. (stricter constraints)



(a) FLR (b) FGR

Figure 2.22: FLR and FGR at the optimal controls; SQP-filter workflow. (stricter constraints)

scheme and the truncation scheme in terms of optimal NPV and the number of simulations.

Scheme	Case Settings	Optimal NPV,10^8 USD	Num. of Simulations
Convert constraints	FLR$\leqslant$12,000	9.42	1160
Truncation	FLR$\leqslant$12,000	9.37	1320
Convert constraints	FLR$\leqslant$10,000	8.52	955
Truncation	FLR$\leqslant$10,000	8.50	950

Table 2.2: The summary of constrained deterministic optimization with different bound constraints handling schemes on the cases with FLR$\leqslant$12,000 and FLR$\leqslant$10,000.

2.3.2 Example 2: Brugge Field

In the second example, we apply the SQP-filter with the modified StoSAG gradient to estimate the well controls that maximize the NPV of life-cycle production. Unlike the first example, we do robust optimization using ten realizations of the reservoir model to represent uncertainty. The Brugge field is a synthetic reservoir model designed by TNO as an SPE comparative study for closed-loop optimization [72]. Although the organizers created 104 realizations conditioned on well logs, we select only 10 geological realizations to perform robust constrained optimization. The 10 realizations were selected by first running the reservoir simulator to compute the NPV of life-cycle production for each of the

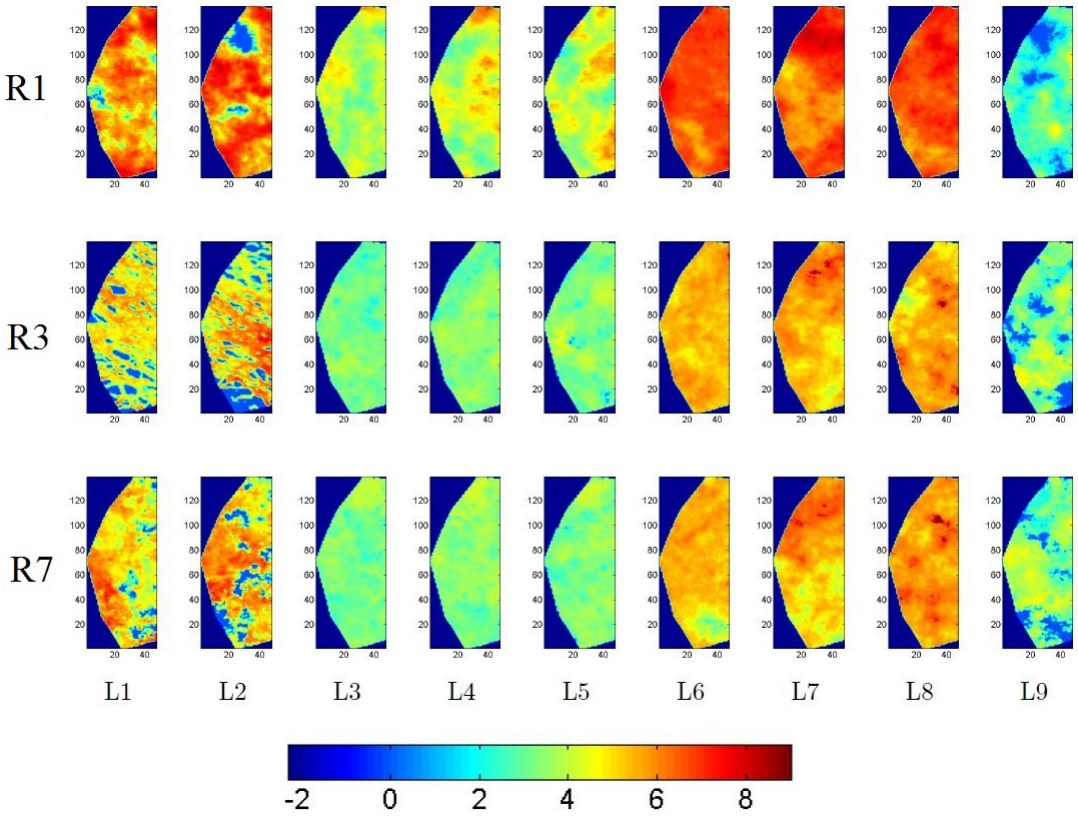


Figure 2.23: Log permeability distribution of selected realizations.

104 realizations of the Brugge model that were originally provided. Then we ranked the NPV's from smallest to largest, $NPV_1 \leq NPV_2 \leq \cdots NPV_{104}$ and selected the 10 models that correspond to NPV_{1+10j} for $j = 0, 1, \cdots 9$ to represent geological uncertainty. Fig. 2.23 displays the distributions of horizontal permeability of the realizations 1, 3 and 7. Ten injectors and 20 producers are drilled near the crest of the reservoir with the location of each well shown in Fig. 2.24. The control variables for producers are BHPs and for injectors are water injection rates. The minimum and maximum BHP for producers are set equal to 500 psi and 3,400 psi, respectively. The minimum and maximum water injection rate are 0 STB/d and 10,000 STB/d. All bound constraints are handled by truncation in this example. Since the total number well controls in this example is large, it would be very computationally expensive to handle bound constraints by converting them to general inequality constraints. The reservoir life is set equal to 22 years and the control interval for each well is one year. Therefore, the total number of control variables for this problem is $30 \times 22 = 660$. The economic parameters for this case are the same as for the previous case except that we have no gas production. The nonlinear state constraints in this problem are the maximum field liquid production rate (FLR) and the maximum field water production rate (FWR), where these constraints, which are applied on each control step, are given by

1. FLR: $c = \frac{40{,}000 - FLR}{40{,}000} \geqslant 0,$
2. FWR: $c = \frac{10{,}000 - FWR}{10{,}000} \geqslant 0,$

where rates are in STB/d. We compare three methods for handling constraints, namely, the min-max procedure, expectation scheme (enforcing the average of the nonlinear constraints to be satisfied) and using heuristic schemes applied internally in a simulator. As noted previously, at each iterative the nonlinear constraint over all realizations. All bound constraints are handled by truncation scheme in order to have consist comparison with the heuristic scheme. In the heuristic scheme, we solve the optimization problem with only bound constraints with the StoSAG gradient and steepest decent method [33] with bound constraints are handled by truncation scheme. Then, we input the optimal controls obtained

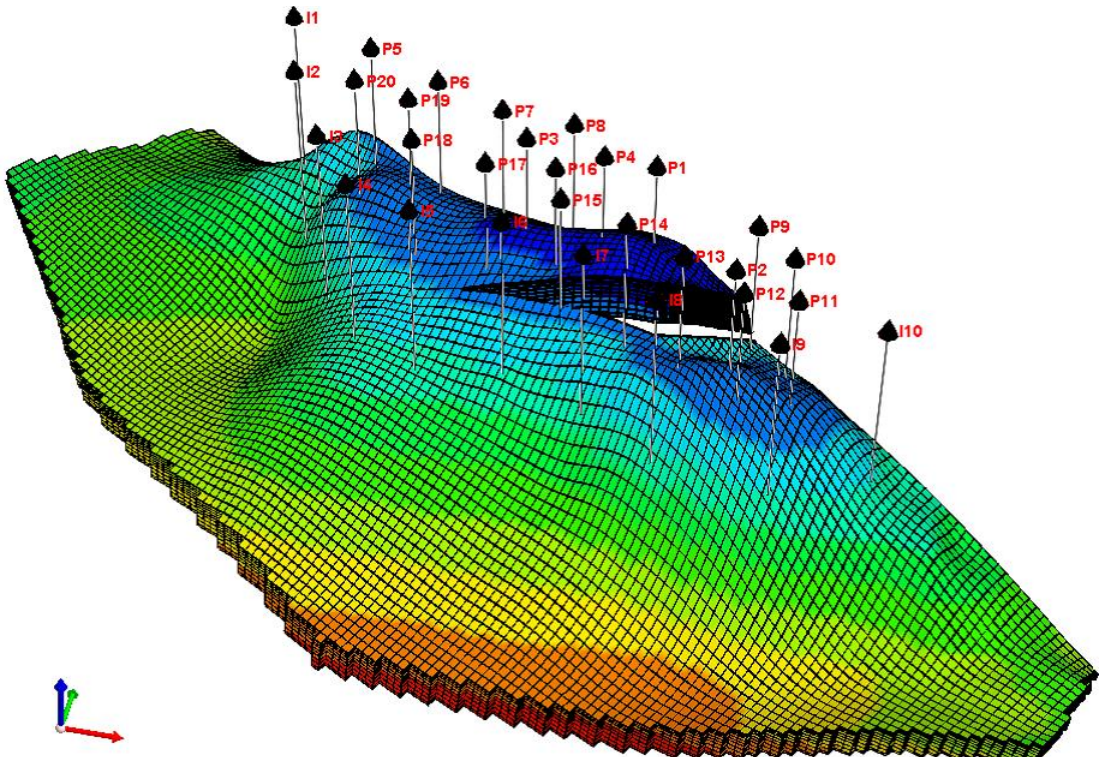
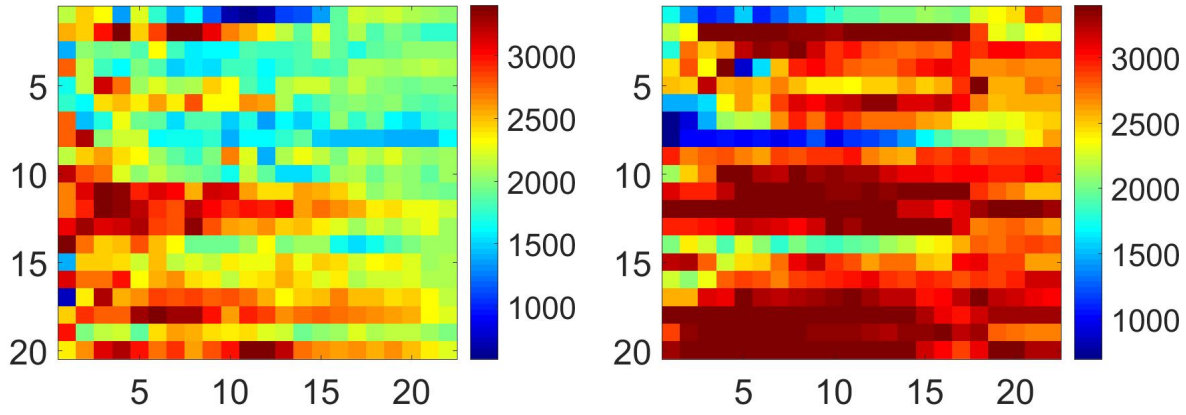


Figure 2.24: Top structure of Brugge case.

into the simulator and let the simulator itself enforce the nonlinear constraints on the FLR and FWR using the key word **GCONPROD** in **Eclipse 100** (Schlumberger Software Ltd., Houston, Texas, USA). Although there is many options for the **GCONPROD** keyword e.g., "RATE", "WELL" and "PLUG", we found only slight differences in the NPV of each realization generated by those different options when we inserted the optimal controls generated by optimization with only bound constraints into the simulator. Thus, we only show the results for the option "RATE" which, among the options, actually gives the highest expected NPV for this example when state constraints are enforced by the internal reservoir simulator heuristics.

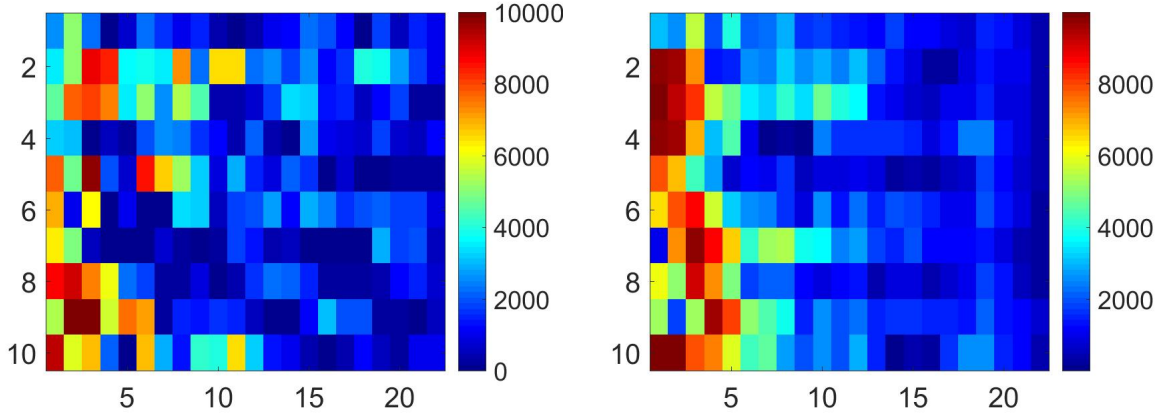
Figs. 2.25 and 2.26 compare the optimal controls of producers and injectors from the min-max scheme with the optimal controls obtained when only the expected (average) value of constraints is enforced. In these and similar figures, the abscissa represents the control step index and the ordinate indicates the index of producers or injectors. The optimal controls generated from the min-max and expected value schemes are quite different.



(a) Optimal BHPs of producers (min-max)

(b) Optimal BHPs of producers (expected)

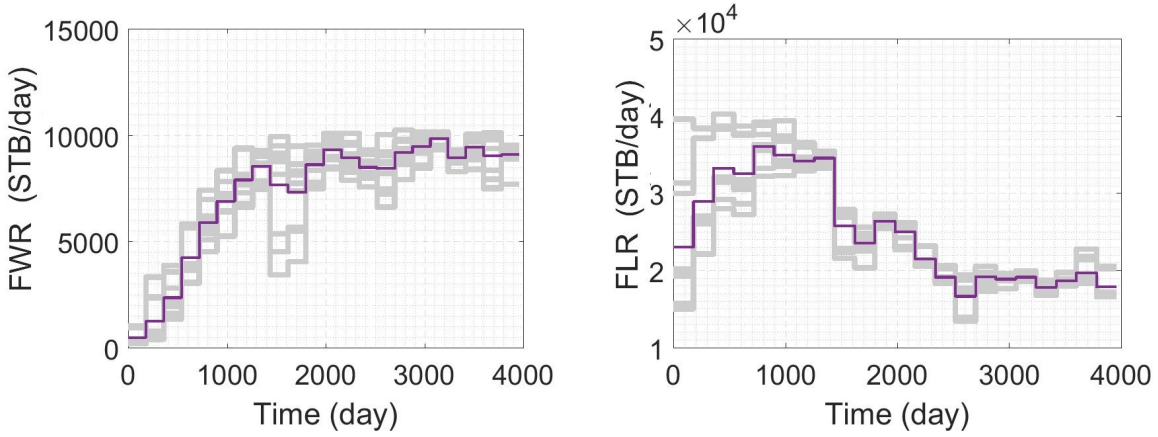
Figure 2.25: Comparison of the producer optimal controls of expected value and min-max schemes.



(a) Optimal waters injection rates of injectors (min-max)

(b) Optimal waters injection rates of injectors (expected)

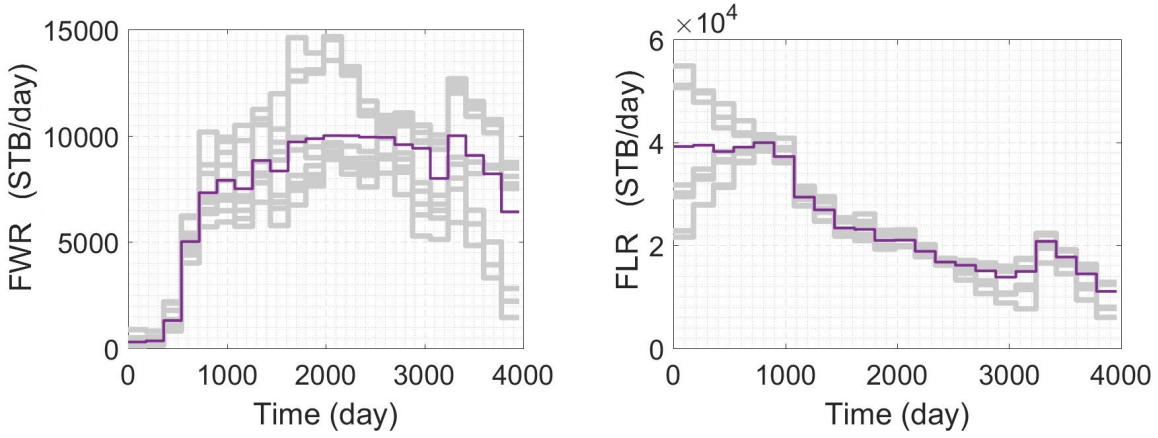
Figure 2.26: Comparison of the injector optimal controls of expected value and min-max schemes.



(a) Field water production rate under the optimal controls. (min-max)

(b) Field liquid production rate under the optimal controls. (min-max)

Figure 2.27: The FWR and FLR at optimal SQP-filter controls with min-max scheme.



(a) Field water production rate under the optimal controls (expected)

(b) Field liquid production rate under the optimal controls (expected)

Figure 2.28: The FWR and FLR at optimal SQP-filter controls with expected value scheme.

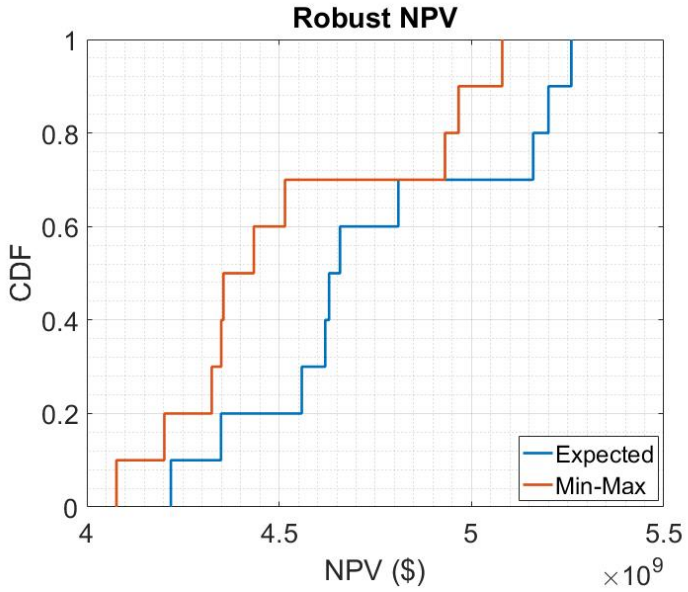
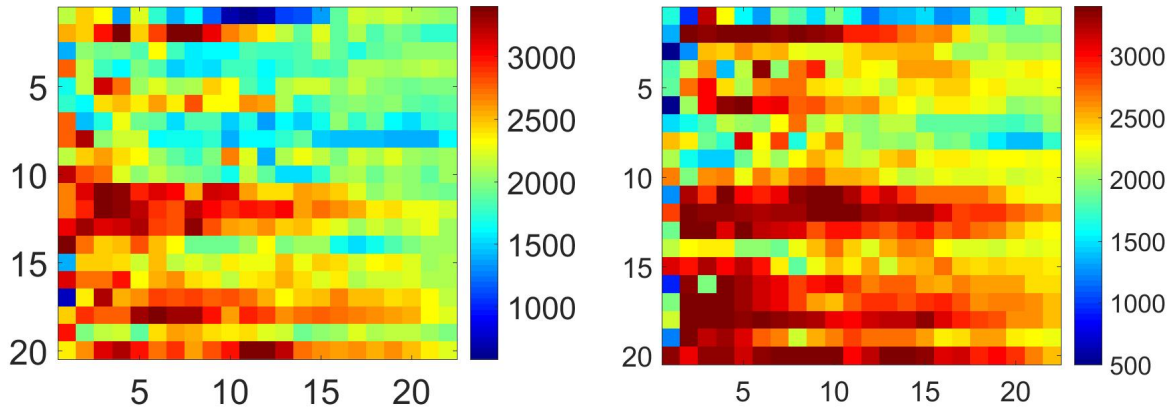


Figure 2.29: Optimal CDF of NPV corresponding to expected value scheme and min-max scheme.

Figs. 2.27 and 2.28 show the field water production rate (FWR) and field liquid production rate (FLR) under the estimated "optimal" controls generated by the min-max and expected value schemes respectively. In both of these constraint handling procedure, we use an upper bound of $\overline{\theta} = 1$ in the filter method; see the discussion of Fig. 2.18b. The purple line on each of those four plots represents the expected value (average value) of either FWR or FLR over the whole ensemble of reservoir models and the gray lines represent the FWR and FLR from the ten realizations. In the min-max results of Fig. 2.27, all the constraints are satisfied for every realization at the optimal controls. From 0-1000 days of production, the FLRs of some realizations become active constraints. From 1000 days to the end of production, the FWRs of some realizations become active constraints. When the constraints are only enforced on their average value over all realizations, the results of Fig. 2.28, show that though the average constraints are not violated, i.e., $E[\text{FLR}] \leq 40,000$ STB/d and $E[\text{FWR}] \leq 10,000$ STB/d, where E denotes expected or average value, some individual realizations violate the 10,000 STB/d constraint on FWR by almost 50%, at times around 2000 days. For some realizations, the FLR constraint is violated by around 25%, during, the first couple of control steps.

Fig. 2.29 compares the cumulative distribution function (CDF) of the NPV's of the individual realizations of the reservoir model at the optimal controls obtained with the min-max and the expected value schemes. The optimal NPVs of the expected value scheme are all larger than the NPVs generated by the min-max scheme and the expected NPV generated by the expected value scheme is larger than min-max scheme by 1.6×10^8 \$ as shown in Table 2.3. This is as expected because the expected value procedure allows some realizations to severely violate the constraints. However, if the true reservoir corresponds to a realization where the estimated optimal controls violate the nonlinear state constraints, the estimated optimal controls would have to be modified as the field constraints are imposed by the capacity of the facilities and cannot be violated. For such a case, the optimal controls produced by the expected value scheme become meaningless. On the other hand, the min-max scheme is a valid scheme to handle nonlinear constraints in robust optimization and eliminates any nonnegligible constraint violations.

Figs. 2.30 and 2.31 compare the optimal controls generated by nonlinear constrained robust optimization with the min-max scheme and the bound-constrained robust optimization using the standard StoSAG gradient without enforcing the state constraints. Generally, the producers operate at higher pressures and injectors at higher injection rates when only bound-constrained optimization is applied. Figs. 2.32 and 2.33, respectively, show the FWR and FLR of all realizations from bound-constrained optimization compared to the results obtained by inserting the results obtained from bound-constrained optimization into the simulator and allowing the simulator to heuristically satisfy the state constraints by modifying the controls obtained from bound-constrained optimization. In Fig. 2.32, both the FWR and FLR constraints are severely violated since during the bound constrained optimization process none of nonlinear state constraints are considered. After applying the nonlinear state constraints using simulator heuristics in Fig. 2.33, all realizations obey the nonlinear constraints strictly. However, enforcing constraints heuristically produces final controls that vary from realization to realization, so there is no way to determine which set of controls should be provided to the field operator. Fig. 2.34 compares the CDFs of the NPV at the



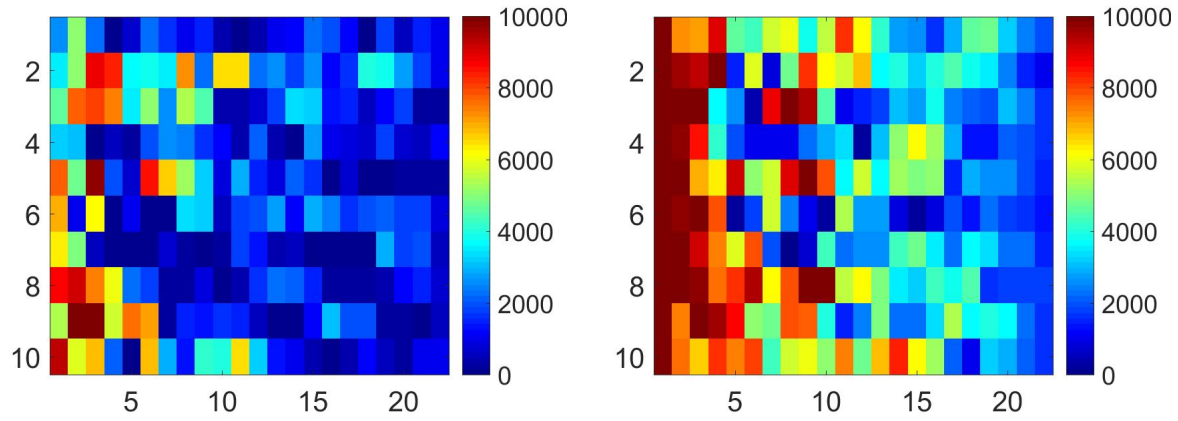
(a) Optimal BHPs of producers (min-max) (b) Optimal BHPs of producers (Bound constraints)

Figure 2.30: Comparison of the producer optimal controls of min-max and bound constrained optimization without enforcing state constraints.

optimal controls produced with different schemes. The blue line denotes the CDF at the optimal control generated by bound-constrained optimization with the StoSAG gradient and the red line denotes the resulting CDF after the nonlinear constraints are then enforced heuristically by the simulator. Note that there is a significant decrease in the NPV of each realization after enforcing the nonlinear constraint heuristically. Compared with the CDF of NPV generated with the min-max scheme, the CDF of NPV generated by enforcing constraints heuristically is much lower. In fact, the expected NPV obtained with the min-max scheme is higher than the one from the heuristic scheme by 7.5×10^8\$ when the nonlinear constraints are considered.

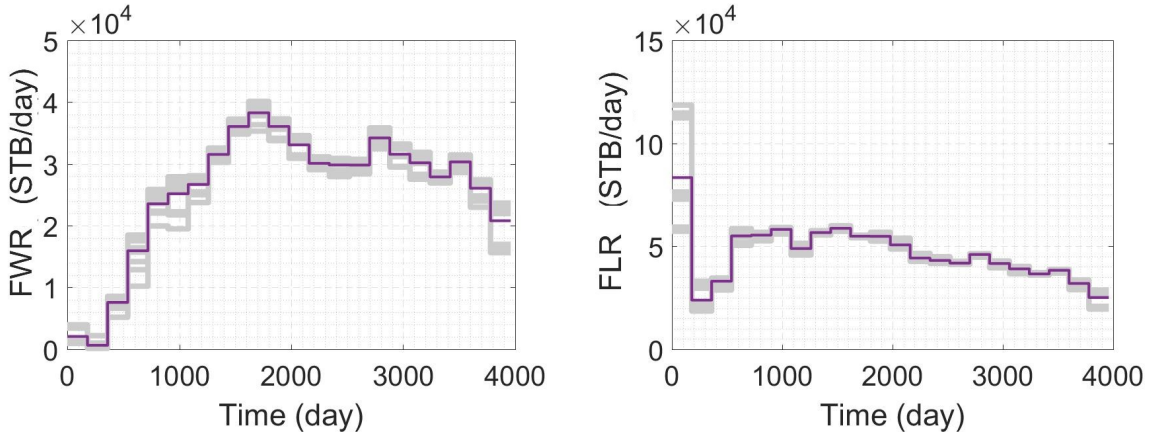
Scheme	Optimal NPV,10^9 USD	Num. of Simulations
Min-max	4.38	4530
Expected	4.54	3200
Heuristic	3.63	2060

Table 2.3: The summary of constrained robust optimization with different schemes; note nonlinear constraints are not satisfied for all realizations with expected value enforcement of constraints.



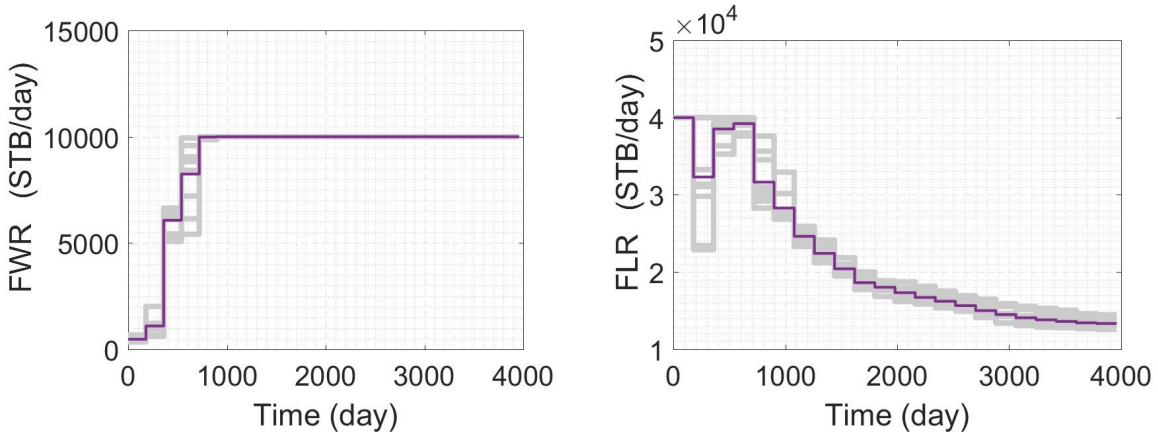
(a) Optimal injection rates of injectors (min-max) (b) Optimal injection rates of injectors (Bound constraints)

Figure 2.31: Comparison of the injector optimal controls of min-max and bound constrained optimization without enforcing state constraints.



(a) Field water production rate under the optimal controls. (b) Field liquid production rate under the optimal controls.

Figure 2.32: The FWR and FLR at optimal controls of bound-constrained optimization without enforcing state constraints.



(a) Field water production rate under the optimal controls.

(b) Field liquid production rate under the optimal controls.

Figure 2.33: The FWR and FLR at optimal controls of bound-constrained optimization with nonlinear state constraints enforced by the reservoir simulator heuristically.

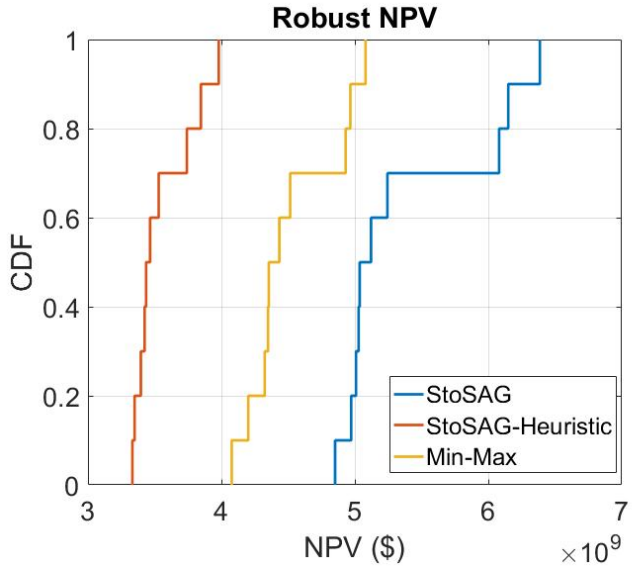


Figure 2.34: Comparison CDF of the NPV at the"optimal" controls; blue represents StoSAG with only bound constraints, red represents the modified results after enforcing constraints heuristically, yellow is from min-max scheme.

2.4 Comments

As mentioned in the introduction, Kourounis et al.[47] considered a SQP framework where all wells were BHP controlled and bounds on rates at each time step were the only nonlinear state constraints considered. They compared constraint handling within the SQP formalism and a heuristic method for the state constraints where if a well rate at a specific time step exceeds its specified maximum value (bound), the BHP control is abandoned and the simulator specifies the rate equal to its bound for that time step. For larger problems they concluded that the heuristic method for constraint handling results in a higher value of the estimated optimal objective function. As it may seem that this result is in conflict with our conclusion that enforcing constraints heuristically within a forward run of the simulator may yield highly suboptimal results, additional discussion seems desirable. The only nonlinear constraints considered by Kourounis et al. in their work are bounds on rates for BHP controlled injectors and producers. It is entirely possible that setting a rate equal to its bound at a time step at a well where the rate constraint is violated under BHP control yields results that are fairly close to optimal whereas if available simulator heuristics are used to enforced bounds on both control variables and field rates, we may obtain a highly suboptimal result as we have shown. Thus, then there may be no conflict between our conclusions and those of Kourounis et al. It is also possible that the Kourounis et al. results for both SQP constraint handling and heuristic treatment are both suboptimal but we do not claim this is the case. In addition, there are many differences between the two workflows besides the procedures for computing gradients. Specifically, Kourounis et al. lump state constraints, whereas we do not and then they approximate the nonlinear lumped constraints by an even more nonlinear approximation. Moreover, they use an ℓ_1 merit function to treat state constraints whereas we use a filter method. We think it is possible that the combination of lumping constraints, transforming the lumped constraints to a more nonlinear constraint function and using the ℓ_1 merit function may not be the best procedure for treating nonlinear state constraints. Early in our work with SQP, we tried implementing an ℓ_1 merit function and found that except for small problems, the overall SQP algorithm could sometimes be

trapped at a suboptimal local minimum.

CHAPTER 3

A MIN-MAX SQP-FILTER METHOD FOR NONLINEAR CONSTRAINED MULTI-OBJECTIVE ROBUST OPTIMIZATION PROBLEMS WITH STOSAG

3.1 Introduction

Multiobjective optimization aims to find the best trade-offs between conflicting objectives. In this Chapter, we investigate a framework to solve the constrained multi-objective optimization problem when the accurate analytical gradient is unavailable. Comprehensive work on this problem has been presented for the case when adjoint solutions for computing the necessary gradients can be computed with the reservoir simulation software in use; see Liu and Reynolds [53, 55, 56]. The objectives of interest to us are maximizing the life-cycle NPV, maximizing short term NPV and minimizing risk with geological uncertainty. The robust multi-objective optimization problem is solved with a modified lexicographic method proposed here where stochastic gradients are calculated with the modified StoSAG which was introduced in Chapter 2, and the customized SQP-filter algorithm proposed in Chapter 2 is applied for optimization. We apply this framework developed to solve a multiobjective optimization problem for the Brugge reservoir with state constraints.

3.2 Methodology

3.2.1 Multi-Objective Optimization Problem

Due to the volatility of the oil price, the longer the production life-cycle, the larger the financial risk. Therefore, to balance the long-term interest and short-term interest, maximization of both life-cycle NPV and short-term NPV are important objectives to consider.

The formulation of the negative life-cycle NPV is defined in Chapter 2; see Eq. 2.1. The negative short-term NPV for a given vector of well controls, u, and a given reservoir model, m, is denoted by $J_{s-NPV}(u,m)$ and is defined as

$$J_{s-NPV}(u,m) = -\sum_{n=1}^{N_{\text{short}}} \left\{ \frac{\Delta t_n}{(1+b)^{\frac{t_n}{365}}} \left[\sum_{j=1}^{\text{P}} \left(r_{\text{o}} \cdot \bar{q}^n_{\text{o},j}(u,m) + r_{\text{g}} \cdot \bar{q}^n_{\text{g},j}(u,m) - r_{\text{w}} \cdot \bar{q}^n_{\text{w},j}(u,m) \right) \right] \right\} - \sum_{n=1}^{N_{\text{t}}} \left\{ \frac{\Delta t_n}{(1+b)^{\frac{t_n}{365}}} \left[\sum_{j=1}^{\text{I}} \left(r_{\text{wi}} \cdot \bar{q}^n_{\text{wi},j}(u,m) \right) \right] \right\}. \quad (3.1)$$

Note we minimize the negative short-term NPV which is equivalent to maximize the short-term NPV. All the variables and parameters in Eq. 3.1 have the same meaning as those in Eq. 2.1 except, N_{short} denotes the number of time steps within the short-term period of interest. For robust multi-objective optimization, instead of minimizing the negative short-term NPV of a single reservoir model, we minimize the average negative short-term NPV where the average is over an ensemble of realizations. Thus, average negative shor-term NPV is defined as

$$\bar{J}_{s-NPV}(u) = \frac{1}{N_{\text{e}}} \sum_{k=1}^{N_{\text{e}}} J_{s-NPV}(u,m_k) \approx E\left[J_{s-NPV}(u,m) \right], \quad (3.2)$$

where E denoted expectation and N_e is the number of realizations of the reservoir modeled generated to represent the geological uncertainty.

Minimizing the development risk related with geological uncertainty is another common objective in the development of a reservoir. Commonly used risk measures in the oil and gas community include standard deviation or variance, the worst-case scenario (minimum NPV over the set of geological models), safety margin, mean-variance, value at risk (VaR) and conditional value at risk (CVaR), which is also known as the expected shortfall. Liu and Reynolds [54] found that standard deviation is not a good measure for risk minimization. Capolei et al. [7] also pointed out that, except for the worst-case scenario and CVaR, most of the risk measures are not satisfactory. Therefore, in this work, we consider the negative value of the worst-case scenario NPV as a measure of risk. The formulation of worst-case

scenario NPV given by

$$J_{\text{risk}}(u) = \max\{J_{NPV}(u, m_1) \cdots J_{NPV}(u, m_k) \cdots J_{NPV}(u, m_{N_e})\}, \tag{3.3}$$

where $J_{NPV}(u, m_k)$ denotes the life-cycle NPV of the kth realization under the well control u. Therefore, $J_{\text{risk}}(u)$ denotes the minimum value of NPV (maximum of negative NPV) over all realizations while operating under the control variables u.

The general multiobjective robust optimization problem with state inequality constraints and bound constraints is define as

$$\underset{u \in \mathbb{R}^{\text{N}u}}{\text{minimize}} \quad \{\bar{J}_{NPV}(u), \bar{J}_{s-NPV}(u), J_{\text{risk}}(u)\} \tag{3.4a}$$

$$\text{subject to} \quad u_i^{\text{low}} \leq u_i \leq u_i^{\text{up}}, \quad i = 1, 2, \ldots, N_{\text{u}}, \tag{3.4b}$$

$$c_i(u, m_j) \geqslant 0, \qquad i = 1, 2, \ldots, N_{\text{ic}}, \quad j = 1, 2, \ldots, N_{\text{e}}, \tag{3.4c}$$

where $\bar{J}_{NPV}$ denotes the negative average life-cycle NPV over an ensemble of realizations; $\bar{J}_{s-NPV}$ denotes the average negative short-term average NPV over the ensemble of realizations on a time period significantly shorter than the reservoir life; J_{risk} denotes the risk in the robust reservoir development to be specific the negative value of the worst-case NPV. Eq. 3.4c indicates that all state constraints should be satisfied for every realization of the reservoir model.

3.2.2 Lexicographic method

Based on previous work [53], the lexicographic is one of most efficient way to solve the multi-objective optimization problem. The basic idea of the lexicographic method for bi-objective optimization is that we solve two optimization problems. The first problem solved is always a single objective optimization problem with standard state and bound constraints. The second optimization problem is also a single objective optimization but in this case, another state constraint in added in the form of a bound on the value the objective

function that is not directly optimized in this second step. In the case of maximization of objective functions, this added state constraint is a lower bound on an objective function. However, in the case where the optimization is formulated as a minimization problem, e.g., minimize a negative NPV, the added state constraint is a upper bound on the minimum of the objective functions. For a tri-objective optimization, a third optimization is performed where one of the objectives is optimized with two state constraints in the form of bounds on the other two objective functions. We first discuss the standard lexicographic method and then discuss the modified lexicographic method used in our computational examples.

For the standard lexicographic method, the first step is to determine a primary objective function. In this work, we select the average negative life-cycle NPV, where this average negative NPV is denoted here by $\bar{J}_{NPV}(u)$, i.e.,the notation $\bar{J}(u)$ in Eq. 2.3 is now replaced by the notation $\bar{J}_{NPV}(u)$. Then, at the first step of the standard lexicographic method we solve the single objective life-cycle robust production optimization problem with nonlinear state and bound constraints specified in Eq. 2.3; the solution of this problem is denoted as u^*. Then, the value of the negative average life-cycle NPV ($\bar{J}_{NPV}(u^*)$) at the vector of optimal controls, u^*, is recorded. Next, a value of γ_1 such that $0 < \gamma_1 \leq 1$ is selected. The value of γ_1 represents the decimal percentage of the increase in the negative life-cycle NPV (decrease in life-cycle NPV) the operator is willing to sacrifice in upper to improve the value of a secondary objective function beyond its value at u^*. Then, a secondary objective function is optimized with an additional state constraint in the form of $\bar{J}_{NPV}(u) \leq \gamma_1 \bar{J}_{NPV}(u^*)$. In the case where the second optimization problem is to optimize the short-term NPV, the second optimization problem that is solved is specified as

$$\underset{u \in \mathbb{R}^{N_u}}{\text{minimize}} \quad \bar{J}_{s-NPV}(u), \tag{3.5a}$$

$$\text{subject to} \quad \bar{J}_{NPV}(u) \leq \gamma_1 \bar{J}_{NPV}(u^*) \tag{3.5b}$$

$$u_i^{\text{low}} \leq \ u_i \ \leq \ u_i^{\text{up}}, \qquad i = 1, 2, \cdots, N_u, \tag{3.5c}$$

$$c_i(u, m_j) \geq 0, \qquad i = 1, 2, \cdots, N_{\text{ic}},\ j = 1, 2, \cdots, N_{\text{e}}. \tag{3.5d}$$

Note if $\gamma_1 = 0.99$ is selected, then the state constraint on $\bar{J}_{NPV}(u)$ indicates that we can tolerate at most a one percent increase in average negative life-cycle NPV, which is equivalent of being willing to tolerate at most a one percent decrease in average life cycle NPV, in order to decrease the value of negative short-term, or, equivalently, increase average short-term NPV above its value $-\bar{J}_{s-NPV}(u^*)$. In the tri-objective optimization case, a third optimization is performed. Let u^{**} denote the solution of Eq. 3.5 and record the values of the negative average life-cycle NPV ($\bar{J}_{NPV}(u^{**})$) and negative average short-term NPV ($\bar{J}_{s-NPV}(u^{**})$) at the new vector of optimal well control variables u^{**}. Next select values of of γ_1 and γ_2 such that $0 < \gamma_1 \leq 1$ and $0 < \gamma_2 \leq 1$. Then, formulate a new constrained production optimization problem as:

$$\underset{u \in \mathbb{R}^{N_u}}{\text{minimize}} \quad J_{\text{risk}}(u), \tag{3.6a}$$

$$\text{subject to} \quad \bar{J}_{NPV}(u) < \gamma_1 \bar{J}_{NPV}(u^{**}) \tag{3.6b}$$

$$\bar{J}_{s-NPV}(u) < \gamma_2 \bar{J}_{s-NPV}(u^{**}) \tag{3.6c}$$

$$u_i^{\text{low}} \leq \ u_i \ \leq \ u_i^{\text{up}}, \qquad i = 1, 2, \ldots, N_u, \tag{3.6d}$$

$$c_i(u, m_j) \geq 0, \qquad i = 1, 2, \cdots, N_{\text{ic}},\ j = 1, 2, \ldots, N_{\text{e}}, \tag{3.6e}$$

where the primary objective function for this problem is the negative value of the worst-case NPV and both average life-cycle NPV and short-term NPV are treated as the state constraints. We can of course, interchange the roles of the short term NPV and risk in the two preceding optimization problems. Note that above, we allow the possibility of selecting $\gamma_=1$ and $\gamma_2 = 1$ in the preceding bi-objective and tri-objective optimization steps because when the number of control steps is large, it is possible that after the first and second optimization steps, there remain sufficient degrees of freedom so that one can optimize a second, and perhaps even a third objective function without decreasing its previous optimal value; see Chen et al. [15] and Van Essen et al. [83].

Instead of using the standard lexicographic method, we implement a modified lexicographic method to solve the robust multi-objective optimization problem with the improved

StoSAG. Like the standard method, two optimization problems are solved for bi-objective optimization and three optimization problems are required to be solved when there are three objectives. In each optimization problem, the min-max scheme is applied to enforce standard state constraints (ones that do not involve a bound on one of the objective functions) using the filter method. Bound constraints are handled by truncation scheme in our work on multi-objective optimization. To be specific, if u_i, the ith entry in the vector of control variables, is larger than the upper bound of this control variable, u_i^{up}, at some optimization step, then we set $u_i = u_i^{\text{up}}$. Similarly, if the ith entry of the vector of control variables, u_i, is smaller than its specified lower bound, u_i^{low}, we set $u_i = u_i^{\text{low}}$. The first step of the modified lexicographic method is same as the standard lexicographic method which minimizes the average negative life-cycle NPV, $\bar{J}_{NPV}(u)$, with state and bound constraints to obtain the estimated optimal well controls, u^*. Then, we record the negative average short-term NPV ($\bar{J}_{s-NPV}(u^*)$) and the largest negative NPV among all realizations ($J_{\text{risk}}(u^*)$) at the optimal vector of well controls, u^*. Next values of γ_1 and γ_2 are selected such that $\gamma_1 > 1$ and $\gamma_2 > 1$ (corresponding to desired decrease in negative short-term NPV and desired decrease in the negative of the worst-case NPV, respectively). Then $\gamma_1 \bar{J}_{s-NPV}(u^*)$ and $\gamma_2 J_{\text{risk}}(u^*)$ are applied as added state constraints (upper bounds) on $\bar{J}_{s-NPV}(u)$ and $J_{\text{risk}}(u)$ in a new optimization problem where we again optimize $\bar{J}_{NPV}(u)$. Specifically, we solve the following problem:

$$\begin{aligned}
\underset{u\in\mathbb{R}^{N_u}}{\text{minimize}} \quad & \bar{J}_{NPV}(u) && && \text{(3.7a)}\\
\text{subject to} \quad & \bar{J}_{s-NPV}(u) < \gamma_1 \bar{J}_{s-NPV}(u^*) && && \text{(3.7b)}\\
& J_{\text{risk}}(u) < \gamma_2 J_{\text{risk}}(u^*) && && \text{(3.7c)}\\
& u_i^{\text{low}} \le \ u_i \ \le \ u_i^{\text{up}}, && i = 1, 2, \cdots, N_u, && \text{(3.7d)}\\
& c_i(u, m_j) \ge 0, && i = 1, 2, \cdots, N_{\text{ic}},\ j = 1, 2, \cdots, N_{\text{e}}, && \text{(3.7e)}
\end{aligned}$$

Note that we can actually increase the value of γ_1 and γ_2 gradually and solve the lexicographic problem repeatedly to attempt to generate a Pareto front (or surface) for the mulit-objective

optimization problem. Also note that for bi-objective optimization, one could delete either Eq. 3.7b or 3.7c from the problem of Eq. 3.7. One could of course also first solve Eq. 3.7 converted to a bi-objective optimization problem and then solve as a third optimization problem, Eq. 3.7. However, solving a bi-objective problem following by solving a tri-objective problem is more computationally expensive than skipping the bi-objective problem and simply solving the tri-objective optimization problem of Eq. 3.7.

To solve the problem in Eq. 3.7, the optimization algorithm needs to handle $N_{\text{ic}} \times N_e + 2$ state constraints, since the all standard state constraints (those involving the c_i's) are required to be satisfied for each realization. To reduce the number of constraints enforced in this problem, the min-max scheme proposed in Section 2.2.5 is applied. However, the min-max scheme is only applied to the standard state constraints, not to the two state constraints on objective functions, Eqs. 3.8b and 3.8c. The bound constraints are enforced by truncation. When applying the min-max scheme, the multi-objective robust constrained production optimization of Eq. 3.7 is redefined as

$$\begin{aligned}
\underset{u\in\mathbb{R}^{\mathrm{N}u}}{\text{minimize}} \quad & \bar{J}_{NPV}(u) && \text{(3.8a)}\\
\text{subject to} \quad & \bar{J}_{s-NPV}(u) < \gamma_1 \bar{J}_{s-NPV}(u^*) && \text{(3.8b)}\\
& J_{\text{risk}}(u) < \gamma_2 J_{\text{risk}}(u^*) && \text{(3.8c)}\\
& u_i^{\text{low}} \leq \ u_i \ \leq \ u_i^{\text{up}}, \quad i \ = \ 1, 2, \cdots, N_{\text{u}}, && \text{(3.8d)}\\
& c_{i,j}(u) \geq 0, \qquad i \ = \ 1, 2, \cdots, N_{\text{ic}}, && \text{(3.8e)}
\end{aligned}$$

where $c_{i,j}$ denotes the ith constraint function evaluated at the jth realization, where the maximum violation of the ith constraint occurs for realization j. This means that for each i, i =1,2,... N_{ic}, j is chosen such that

$$j = \underset{1\leqslant k\leqslant N_e}{\text{argmin}}\{c_i(u, m_k)\}. \tag{3.9}$$

By enforcing $c_{i,j} \geq 0$ at each iteration of the SQP-filter method, we are efficiently minimize

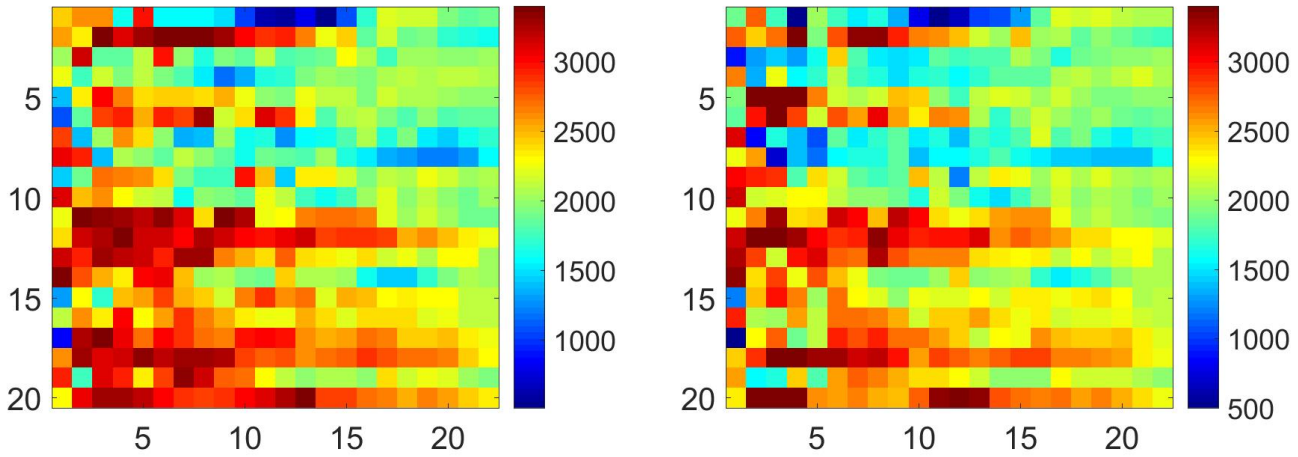
the maximum constraint violation among all realizations. The index j which corresponds to each nonlinear constraint may change from iteration to iteration as the optimization proceeds. By applying the min-max scheme, the number of state constraints to be enforced in the optimization process is reduced from $N_{\text{ic}} \times N_e + 2$ to $N_{\text{ic}} + 2$.

3.3 Computational Results

In this example, we apply the modified lexicographic method on the Brugge case to solve an robust multi-objective optimization problem with state constraints. The Brugge field is a synthetic reservoir model designed by TNO as an SPE comparative study for closed-loop optimization [72]. Although the organizers created 104 realizations conditioned on well logs, we select only 10 geological realizations to perform robust constrained optimization. Details about the Brugge case and how to select the 10 realizations are selected are discussed in the Chapter 2.

Ten injectors and twenty producers are drilled near the top crest of the reservoir with the location of each well shown in Fig. 2.24. The control variables for producers are BHPs and for injectors are water injection rates. The minimum and maximum BHP for producers are set equal to 500 psi and 3400 psi, respectively. The minimum and maximum water injection rate are 0 STB/d and 10,000 STB/d. The reservoir life is set equal to 22 years and the control interval for each well is one year. Therefore, the total number of control variables for the life-cycle optimization problem is $30 \times 22 = 660$. The time span to compute the short-term NPV is set as 5 years which means $N_{\text{short}} = 5$. For the NPV computations, the oil price is set to \$50/STB, the water disposal cost is \$3/STB, and water injection cost is set to \$2/ STB. The gas price is set to a low value of \$0.02/ Mcf, which assumes the gas must essentially be given away which is the case in some fields. The annual discount rate is 0.05. The nonlinear state constraints in this problem are the maximum field liquid production rate (FLR) and the maximum field water production rate (FWR), where the scaled form of these constraints, which are applied on each control step, are given by

1. FLR: $c = \frac{50{,}000 - FLR}{50{,}000} \geqslant 0$,



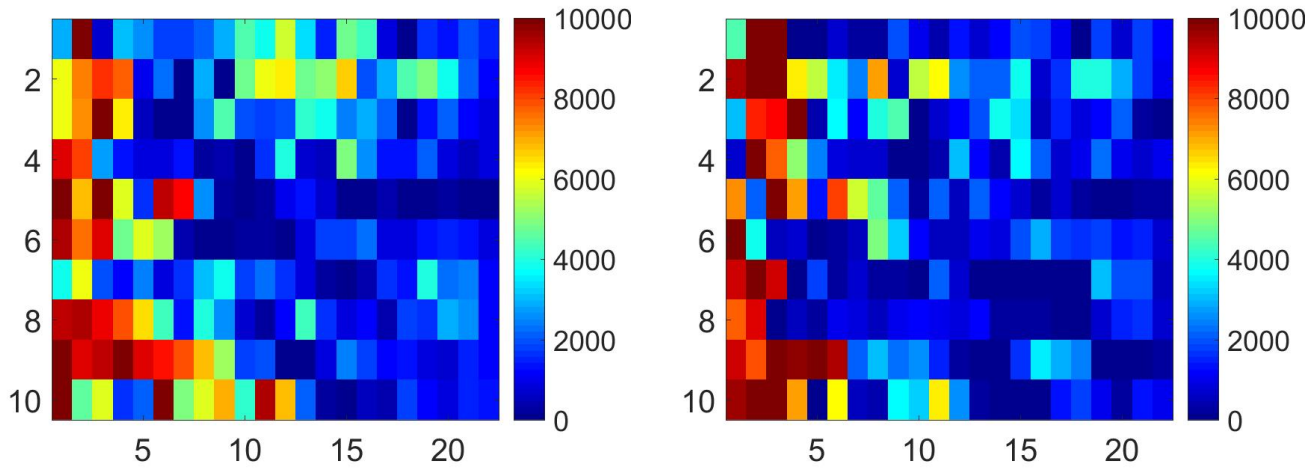
(a) Optimal BHPs from single objective life-cycle optimization.

(b) Optimal BHPs from bi-objective optimization with life-cycle & short-term NPV as objectives.

Figure 3.1: Comparison of optimal production BHP for single objective optimization (left) and bi-objective optimization with short-term NPV as the state constraints using modified lexicographic method (right)

2. FWR: $c = \frac{15{,}000 - FWR}{15{,}000} \geqslant 0.$

As discussed in section 3.2.2, with both the standard and modified lexicographic methods for multi-objective optimization, the first step involves the minimization of the average negative life-cycle NPV with only the original state and bound constraints imposed. For the Brugge example, we then apply the modified lexicographic method to solve a bi-objective optimization problem with the short-term NPV and life-cycle NPV as the objective functions. This means we solve Eq. 3.8 with the constraint on J_{risk} deleted but the constraint on short-term NPV is enforced as one of the state constraints. In the biobjective problem, we set $\gamma_1 = 1.23$. After the single objective optimization step, where the secondary objectives are ignored, the optimal u^* is such that $\bar{J}_{s-NPV}(u^*) = -2.44 \times 10^9$, so the constraint of Eq. 3.8b of the problem Eq. 3.8 is $\bar{J}_{s-NPV}(u) < \gamma_1 \bar{J}_{s-NPV}(u^*) = -3.0 \times 10^9$. Figs. 3.1 and 3.2 compare the optimal controls of producers and injectors from the single objective (minimize average negative NPV) constrained optimization and the results from solving the bi-objective optimization. In Figs. 3.1 and 3.2 and similar figures, the abscissa represents the control steps and the ordinate indicates the index of producers or injectors. From the results of Fig. 3.1 we



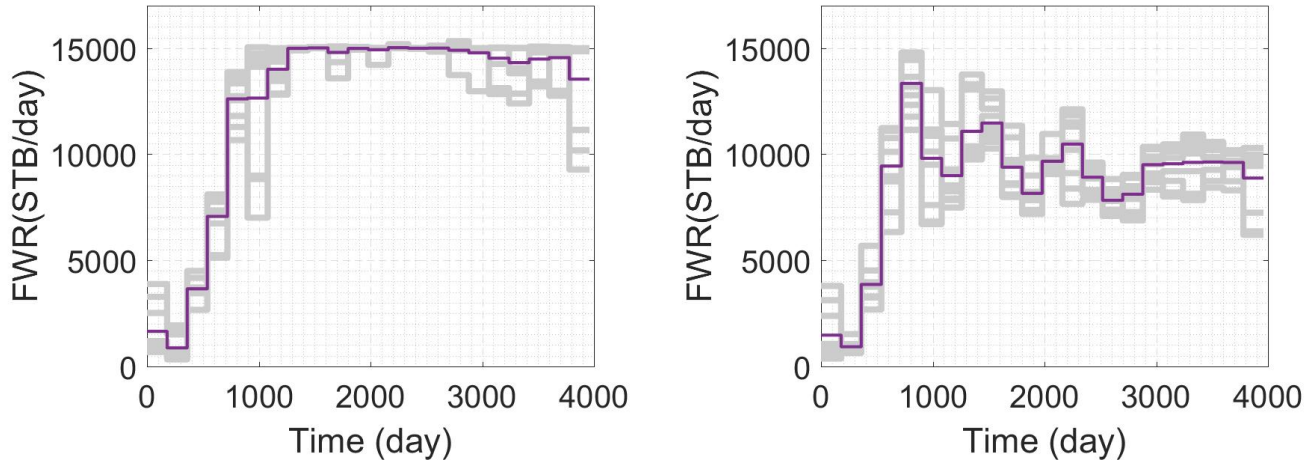
(a) Optimal injection rates from single objective life-cycle optimization

(b) Optimal injection rates from bi-objective optimization with life-cycle & short-term NPV as objectives.

Figure 3.2: Comparison of optimal water injection rates for single objective optimization (left) and bi-objective optimization with short-term NPV as the state constraints using modified lexicographic method (right)

can observe that compared with optimal BHP from the single objective life-cycle optimization in Fig. 3.1a, the optimal well BHP of each well at each control step generated from the bi-objective optimization tends to lower the BHP at the first 5 control steps in Fig. 3.1b since the time span of the short-term NPV is defined as 5 years. Fig. 3.2 compares the optimal water injection rates for the single objective and bi-objection optimal results. Fig. 3.2b illustrates that with bi-objective optimization, more water tends to be injected at wells 1, 7 and 10 in the first 5 years than is the case with the optimal injection rates obtained with single objective life-cycle constrained production optimization of Fig. 3.2a. This is not surprising as in the bi-objective optimization step, one is trying to increase the short-term NPV above the value obtained from single objective optimization. Using the well controls, u^*, obtained from single objective optimization. the following values are obtained: $\bar{J}_{NPV}(u^*) = -4.86 \times 10^9$ (average life-cycle NPV of $\$4.86 \times 10^9$); $\bar{J}_{s-NPV}(u^*) = -2.44 \times 10^9$ (average short-term NPV of $\$2.44 \times 10^9$). Using the optimum well controls u^{**} obtained from the bi-objective optimization solution gives $\bar{J}_{NPV}(u^{**}) = -4.96 \times 10^9$ and $\bar{J}_{s-NPV}(u^{**}) = -3.01 \times 10^9$.

Figs. 3.3 and 3.4 compare the field water production rates (FWR) and the field liquid production rate (FLR), respectively, computed from the optimization results of Figs. 3.1 and

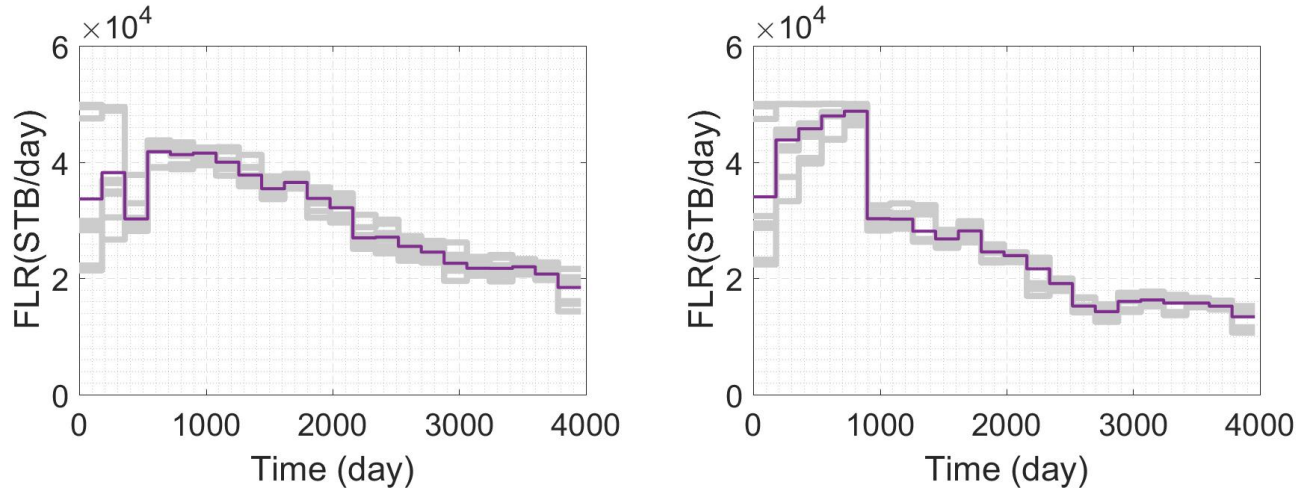


(a) FWR under the optimal controls of single objective optimization.

(b) FWR under the optimal controls of bi-objective optimization using modified lexicographic method with life-cycle NPV as primary objective and short-term NPV as second objective.

Figure 3.3: Comparison of FWR at the optimum of single objective optimization (left) and at the optimum of bi-objective optimization (right) solved with modified lexicographic method

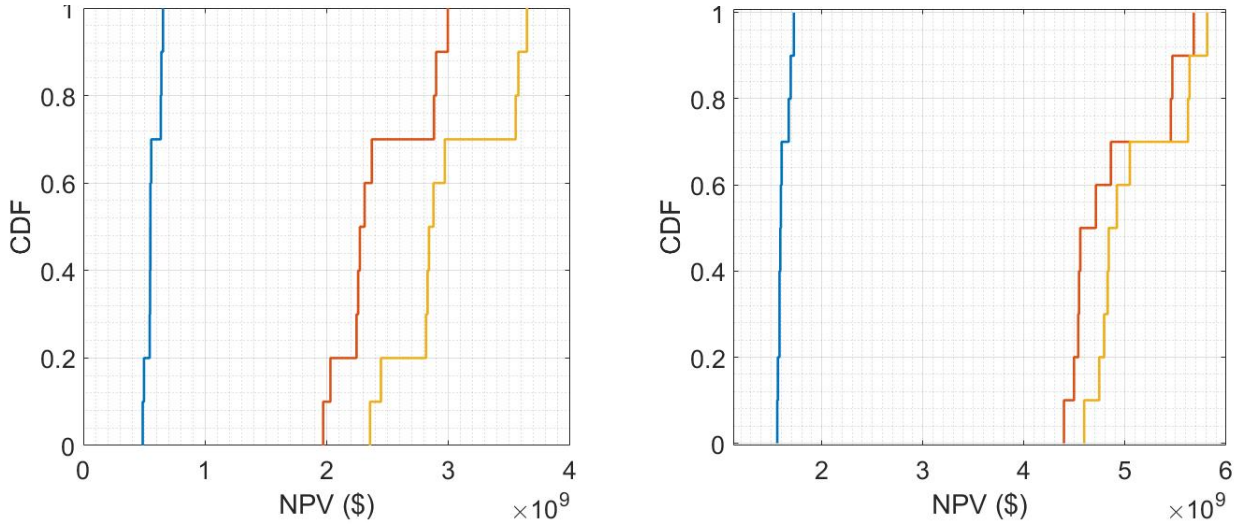
3.2, respectively . The purple line on each of those four plots represents the expected value (average value) of either FWR or FLR over the ten selected ensemble of reservoir models and the gray lines represent the FWR and FLR of the ten realizations. In Fig. 3.3, all field water production rates constraints are satisfied for every realization at the optimal controls for both cases. However, we can observe that bi-objective optimization with a constraint on the short-term negative NPV tends to generate lower average water production rates than that average FWR generated by the life-cycle optimization from the 1000th day of production until the end of the reservoir life. Fig 3.4 compares the field liquid production rates (FLR) under the different optimal controls (life-cycle optimization and bi-objective optimization). From the results, we can observe that compared with life-cycle optimization result in Fig. 3.4a, the bi-objective optimization result with short-term NPV as the secondary objective in Fig. 3.4b tends to yield a higher average FLR at the first 5 control steps. As we expected, the optimal controls computed by bi-objective optimization tend to increase FLR in the first five control steps by lowing the BHP illustrated in Fig. 3.1b to maximize the short-term NPV whose time span is 5 years.



(a) FWR under the optimal controls of single objective optimization.

(b) FLR under the optimal controls of bi-objective optimization using modified lexicographic method with life-cycle NPV as primary objective and short-term NPV as second objective

Figure 3.4: Comparison of FLR at the optimum of single objective optimization (left) and at the optimum of bi-objective optimization (right) solved with modified lexicographic method



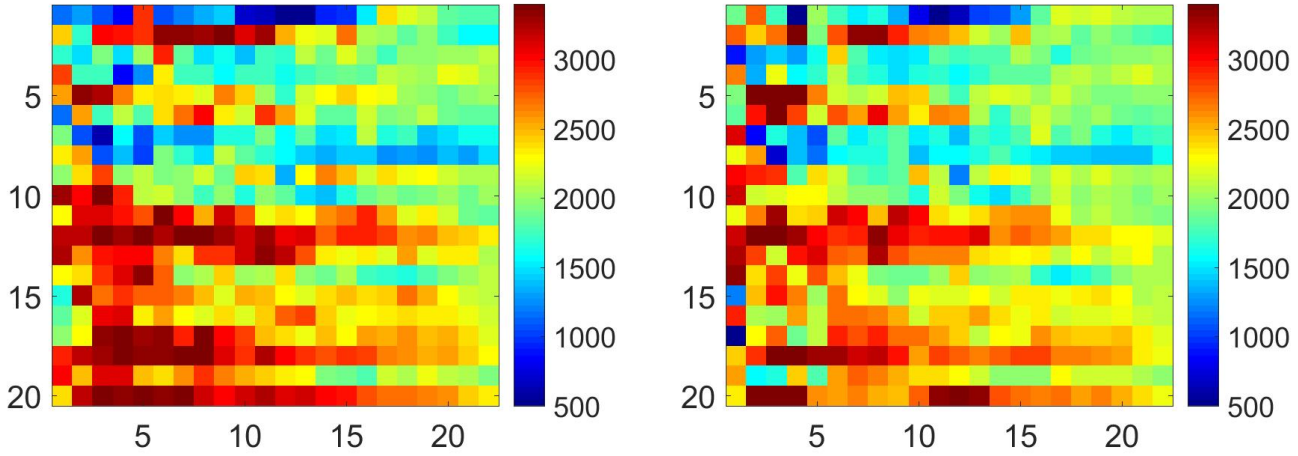
(a) CDF of the short-term NPV

(b) CDF of the life-cycle NPV

Figure 3.5: CDF's of short-term NPV (left) and life-cycle NPV (right);
blue: results from initial well controls;
red: results from single objective life-cycle optimization;
yellow:: results from modified lexicographic bi-objective optimization with state constraint on average negative short-term NPV.

Fig. 3.5 compares the cumulative distribution functions (CDFs) of the short-term NPV (left plot) and life-cycle NPV (right plot) at the optimal controls obtained with the single objective life-cycle constrained optimization and the CDF obtained with the subsequent bi-objective constrained optimization. The CDF of the NPV with initial controls in which the BHP for all producers at all control steps is equal to 2,300 psi and water injection rate for all injectors at all control steps is 1,000 STB/day are represented as blue lines in both Fig. 3.5a and 3.5b. The robust short-term and life-cycle NPV generated with the single objective of life-cycle NPV with no constraints on average negative short-term NPV are shown as as red lines whereas the yellow lines denote the short-term and life-cycle NPV results corresponding to the bi-objective optimal control results. First, it is obvious from the results that the initial well controls give highly sub-optimal results. As one might guess, the bi-objective solution gives a larger short-term NPV for each realization of the reservoir model than is obtained from the solution of the first life-cycle optimal solution. This is not surprising because the bi-objective optimization problem includes a state-constraint that effectively requires that the short-term NPV be 23% greater ($\gamma_1 = 1.23$ than the short-term NPV obtained by optimizing average negative life-cycle with no constraint on the value of short-term NPV. What is surprising is that bi-objective optimization gives a larger NPV for every realization of the reservoir model than is obtained by optimizing average negative NPV with no constraint on short-term NPV. This is surprising, as if the two objectives are in conflict, then one cannot expect to increase short-term NPV by solving the second optimization problem. However, it is not possible to conclude that the two objectives are in conflict. Instead it is likely that using StoSAG to solve the first single objective life-cycle production optimization problem resulted in a somewhat suboptimal solution of the single-objective optimization problem. In the second optimization solution where a constraint on short-term NPV is imposed, we again optimize life-cycle NPV and thus we are able to increase life-cycle NPV above its value obtained during the first optimization plot as can be seen by comparing the red and yellow curves of Fig. 3.5b.

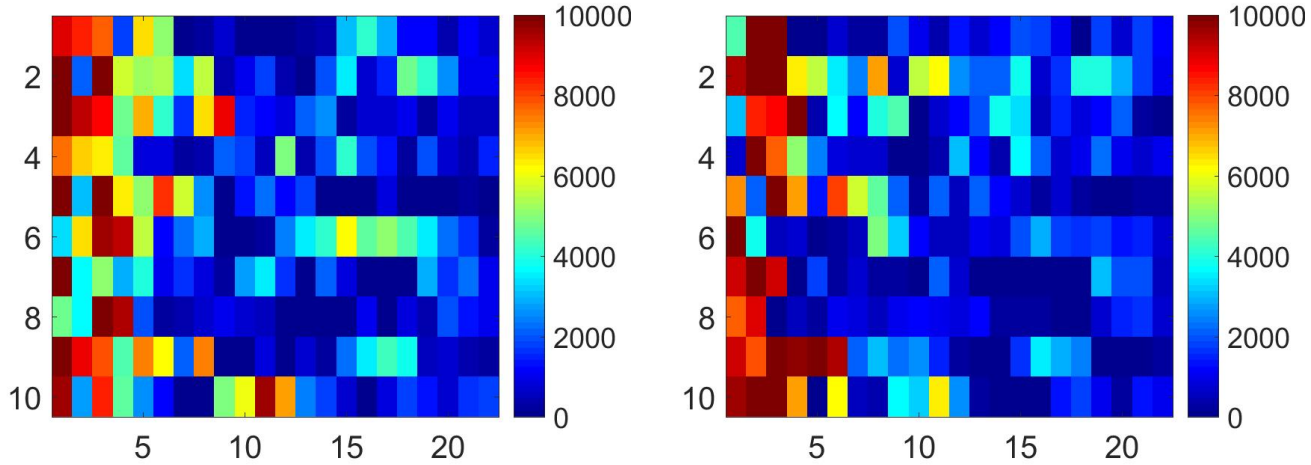
The preceding bi-objective optimization results were obtained with the modified lex-



(a) Optimal BHPs of bi-objective optimization using standard lexicographic scheme

(b) Optimal BHPs of bi-objective optimization using modified lexicographic scheme

Figure 3.6: Comparison of optimal production BHP for short-term and life-cycle NPV bi-objective optimization with standard lexicographic scheme (left) and modified lexicographic scheme (right)



(a) Optimal BHPs of bi-objective optimization using standard lexicographic scheme

(b) Optimal BHPs of bi-objective optimization using modified lexicographic scheme

Figure 3.7: Comparison of optimal water injection rates for short-term and life-cycle NPV bi-objective optimization with standard lexicographic scheme (left) and modified lexicographic scheme (right)

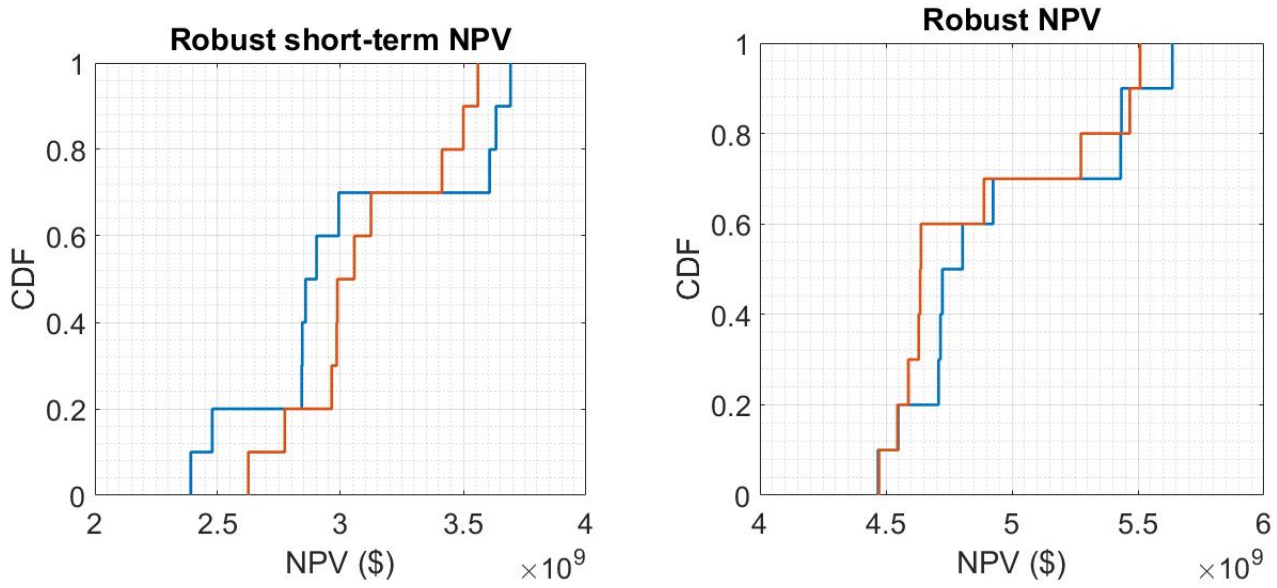
icographic method. In the following case, we keep all parameters settings of the example the same and compare the performance of the standard lexicographic and the modified lexicographic methods on the bi-objective optimization problem with life-cycle and short-term NPV as the two objectives. Recall that for the second optimization problem solved with the standard lexicographic method, we directly minimize the short-term NPV with a constraint imposed on the maximum life-cycle NPV obtained during the first step. As the life-cycle NPV cannot expect to be improved during the optimization of average negative short-term NPV, during this second optimization solution with the standard lexicographic method, it is unreasonable to set $\gamma_1 > 1$. Thus to try to keep $\bar{J}_{NPV}(u)$ as small as possible, or equivalently, average life-cycle NPV as large as possible, we set $\gamma_1 = 1$ in Eqs. 3.5b. Thus, the state constraint for the second optimization problem in the application of the standard lexicographic is set as $\bar{J}_{NPV}(u) \leq -4.86 \times 10^9$ USD. In the modified lexicographic scheme, which was applied previously (see Figs. 3.1 through 3.5 the primary objective is life-cycle NPV and short-term NPV is set as one of state constraint. The γ_1 for the modified lexicographic is set as 1.23. Therefore, the state constraint for the modified lexicographic is set as $\bar{J}_{s-NPV}(u) < -3.0 \times 10^9$ USD. Figs. 3.6 and 3.7, respectively, compare the optimal BHP controls at producers and optimal water injection rates at injectors computed by the standard lexicographic method (left) and modified lexicographic scheme (right). In Fig.3.6, the bi-objective optimization results generated with both the standard and modified lexicographic methods tend to lower the BHP at the first 5 control steps. From Fig. 3.7, we also observe that the optimal injection rates generated by standard and modified lexicographic methods both tend to inject water at large water injection rates during the first 5 years which coincides with the time period of the short-term NPV. Fig. 3.8 compares the CDF obtained for bi-objective optimization with the standard lexicographic method (blue lines) with the corresponding CDF obtained with the modified lexicographic methods (red lines). The CDFs of the short-term NPV are plotted on the left and the CDFs of life-cycle NPV are plotted on the right. Some statistics calculated from these results are summarized in Table 3.1. From the results of Table 3.1, we can observe that the two lexicographic methods produce very

similar results. However, the CDF results of Fig. 3.8a indicate that the standard deviation of the short-term NPV's obtained from the modified lexicographic method is much larger than the corresponding standard deviation computed from the standard lexicographic NPV results. This disadvantage of the modified lexicographic method is slightly ameliorated by the fact the modified lexicographic method gives a higher life-cycle NPV for virtually every realization of the reservoir model than does the standard lexicographic method. Also note that one can improve the results on the standard deviation by introducing as a third objective, the minimization of risk, i.e., solve the problem of Eq. 3.8. It is important to note that the main reason we prefer the modified lexicographic method is that by choosing uniformly spaced values of γ_1 and solving the associated set of bi-objective optimization problems with the modified lexicographic method, it is easy to find points on the Pareto front such that the values of the short-term average NPV are uniformly spaced. However, with the standard lexicographic method, it is challenge to find appropriate values of γ_1 to generate uniformly spaced points on the Pareto front; see Liu and Reynolds [55]. The generation of a Pareto front is discussed later in this chapter.

	$\bar{J}_{NPV}$	$\bar{J}_{s-NPV}$ (USD)	Constraint	Number of Sim.
Standard lexicographic	-4.92×10^9	-3.08×10^9	$\bar{J}_{NPV}(u) \leq -4.86 \times 10^9$	5400
Modified lexicographic	-4.96 ×10^9	-3.01 ×10^9	$\bar{J}_{s-NPV}(u) \leq -3 \times 10^9$	5600

Table 3.1: Summary of constrained bi-objective optimization with different lexicographic schemes.

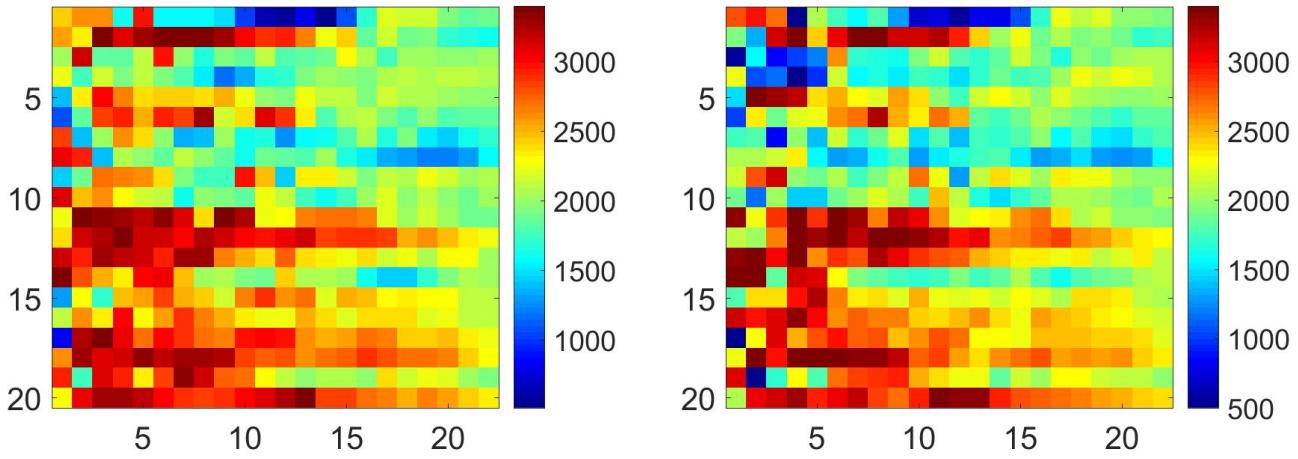
Figs. 3.9 and 3.10, respectively, compare the optimal controls of producers and injectors generated from single objective life-cycle constrained optimization and tri-objective optimization using the modified lexicographic method with both short-term NPV and downside risk used as additional nonlinear state constraints; see Eq. 3.8c. Recall that at the optimal solution of the single objective optimization (u^*) negative average life-cycle is given by NPV $\bar{J}_{NPV}(u^*) = -4.86 \times 10^9$, negative average short-term NPV is $\bar{J}_{s-NPV}(u^*) = -2.44 \times 10^9$ and the negative worst-case NPV is $\bar{J}_{\text{risk}}(u^*) = -4.35 \times 10^9$. Therefore, for tri-objective constrained robust optimization, the primary objective is average negative life-cycle NPV with



(a) CDF of the short-term NPV

(b) CDF of the life-cycle NPV

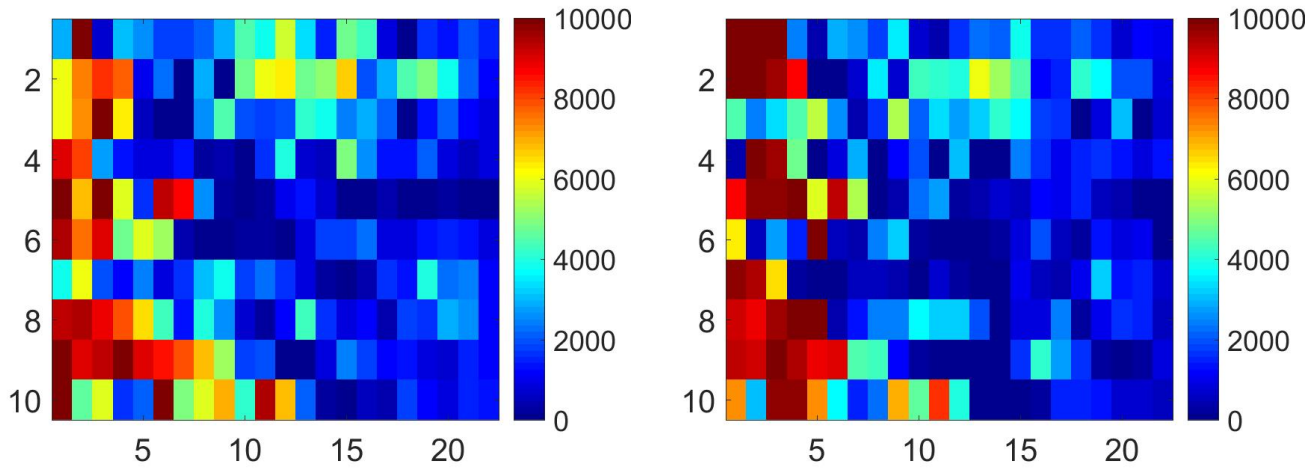
Figure 3.8: CDF's of short (left) and long (right) terms NPV's from the two bi-objective optimization schemes at the estimated optimal controls.
red: standard lexicographic method; short-term NPV as primary objective, life-cycle NPV as a constraint;
blue: modified lexicographic method; life-cycle NPV as primary objective, short-term NPV as a constraint.



(a) Optimal BHPs from single objective life-cycle optimization.

(b) Optimal BHPs of producers from tri-objective optimization

Figure 3.9: Comparison of optimal production BHP for single objective optimization (left) and tri-objective optimization with short-term NPV and minimum risk as the state constraints (right) using modified lexicographic method

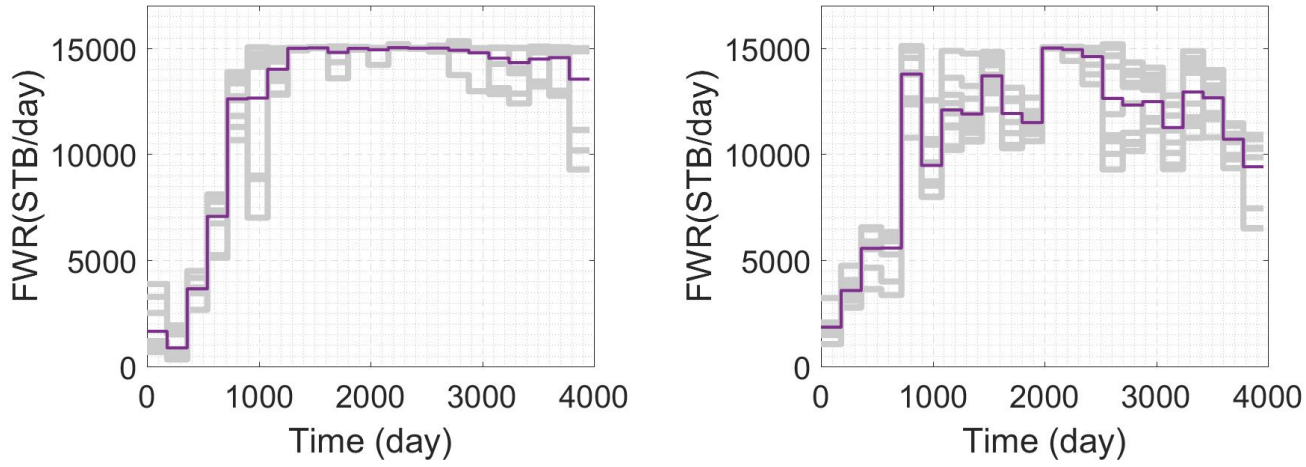


(a) Optimal waters injection rates of injectors from single objective life-cycle optimization.

(b) Optimal waters injection rates of injectors from tri-objective optimization

Figure 3.10: Comparison of optimal water injection rates for single objective optimization (left) and tri-objective optimization with short-term NPV and minimum risk as the state constraints (right) using modified lexicographic method

negative short-term average NPV constrained by $\bar{J}_{s-NPV}(u) \leq \gamma_1 \bar{J}_{s-NPV}(u^*) = -3.0 \times 10^9$ USD ($\gamma_1 = 1.23$) and the downside risk constrained by $J_{\text{risk}}(u) \leq \gamma_2 J_{\text{risk}}(u^*) = -4.6 \times 10^9$ USD ($\gamma_2 = 1.06$). From the results of Fig. 3.9, we can observe that, compared with optimal BHP's from single objective life-cycle optimization in Fig. 3.9a, the optimal BHP of each well at each control step generated from the tri-objective optimization tends to be lower the BHP during the first 5 control steps in Fig. 3.9b. This result makes sense because the time span of the short-term NPV is defined as 5 years and short-term NPV is one of objectives for the tri-objective optimization. Figs. 3.11 and 3.12 compare the field water production rates (FWR) and the field liquid production rate (FLR) at the single objective constrained optimization and tri-objective constrained optimization, respectively. The purple line on each of those four plots represents the expected value (average value) of either FWR or FLR over the ensemble of ten reservoir models and the ten gray lines represent FWR or FLR of the ten realizations. From Figs. 3.11 and 3.12 we can observe that the tri-objective optimization produces results such that all the state constraints are satisfied at each control step for every realization of the reservoir model. It is always important to note that constraints are satisfied since the min-max scheme is used to satisfy constraints within the filter



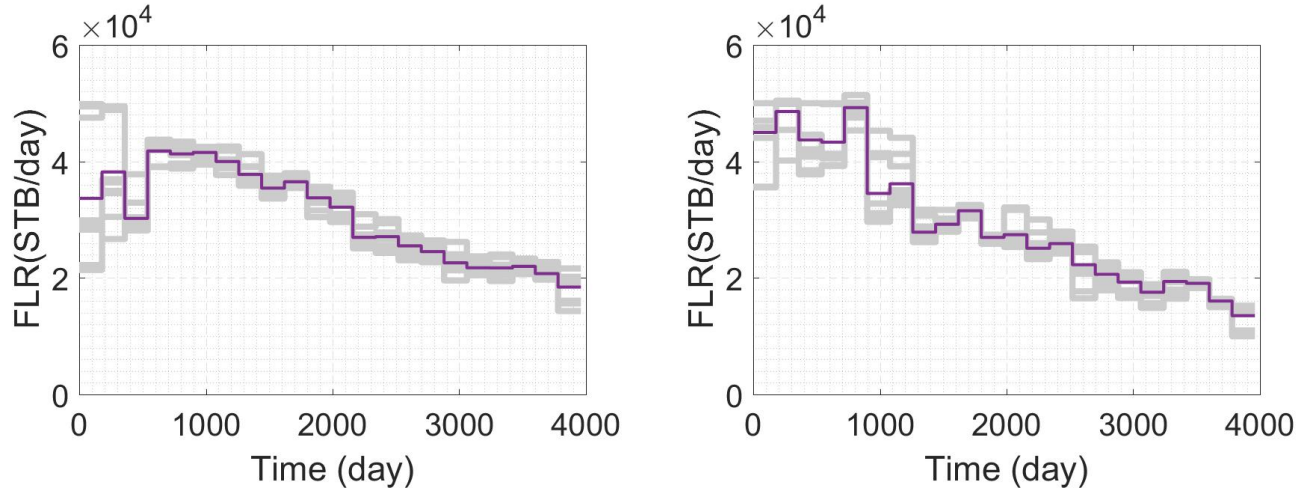
(a) FWR under the optimal controls of single objective optimization.

(b) FWR under the optimal controls of tri-objective optimization using modified lexicographic method with life-cycle NPV as primary objective, short-term NPV and minimum risk as the objectives enforced by state constraints

Figure 3.11: Comparison of FWR at the optimum of single objective optimization (left) and at the optimum of tri-objective optimization (right) solved with modified lexicographic method

method. Fig 3.12 compares the field liquid production rates (FLR) obtained for the optimal controls obtained with single objective robust life-cycle production optimization with those obtained with tri-objective optimization using the modified lexicographic method. Note that tri-objective optimization (Fig. 3.12b) tends to generate higher values of the average FLR and individual FLR's at the first 5 control steps than does single objective life cycle optimization because with tri-objective optimization we are forcing the average short-term NPV to be greater than a prescribed lower bound.

Fig. 3.13 compares the cumulative distribution functions (CDFs) of the short-term NPV on the left and life-cycle NPV on the right at the optimal controls obtained from single objective, bi-objective and tri-objective optimization. The bi-objective and tri-objective results are generated with the modified lexicographic procedure. In one of the bi-objective optimization problems, the average negative short-term NPV is set as the extra state constraint (Eq. 3.8b) and no state constraint on down-side risk is considered. The value of γ_1 in Eq. 3.8b in this bi-objective optimization problem is set as $\gamma_1 = 1.23$, which is equivalent to



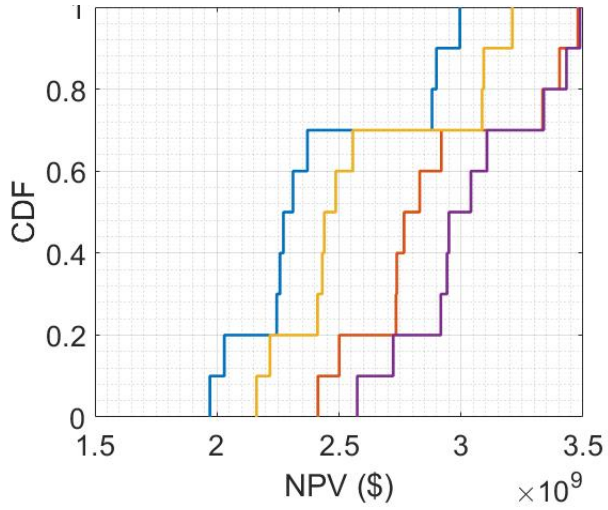
(a) FLR under the optimal controls of single objective optimization.

(b) FLR under the optimal controls of tri-objective optimization using modified lexicographic method with life-cycle NPV as primary objective, short-term NPV and minimum risk as the objectives enforced by state constraints

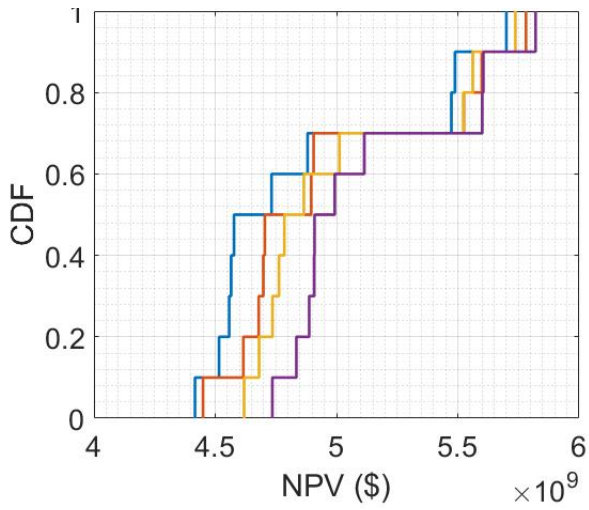
Figure 3.12: Comparison of FLR at the optimum of single objective optimization (left) and at the optimum of tri-objective optimization (right) solved with modified lexicographic method

$\bar{J}_{s-NPV}(u) \leq \gamma_1 \bar{J}_{s-NPV}(u^*) = -3.0 \times 10^9$. In the other bi-objective optimization problem, the negative worst-case NPV is set as the extra state constraint (Eq. 3.8c) to minimize the downside risk and no state constraint on average negative short-term NPV is considered. The value of γ_2 in Eq. 3.8c in this bi-objective optimization problem is set as $\gamma_2 = 1.06$, which is equivalent to $J_{\text{risk}}(u) \leq \gamma_2 J_{\text{risk}}(u^*) = -4.6 \times 10^9$ $(J_{\text{risk}}(u^*) = -4.35 \times 10^9)$. In Fig. 3.13, the blue lines denote the short-term and life-cycle NPV's evaluated at the optimal well controls generated with single objective life-cycle production optimization. The red lines denote the short-term and life-cycle NPV's computed with the optimal well controls obtained from bi-objective optimization with the average short and average long term NPV as the two objectives. The CDF's shown in yellow are the average short-term and average life-cycle NPV's estimated from the bi-objective optimization optimal result in which life-cycle NPV is set as the primary objective and downside risk is treated as a state constraint. The purple lines in Fig. 3.13 represent the CDFs of short-term and life-cycle NPV generated from the tri-objective optimization result with both short-term NPV and downside risk treated

as state constraints in the modified lexicographic method; see Eq. 3.8. The settings of the constraints for single, bi-objective and tri-objective optimization with the modified lexicographic method are summarized in Table 3.2. Under the modified lexicographic scheme, the primary objective function is always life-cycle NPV ($\bar{J}_{NPV}$) and the average negative short-term NPV and negative worst-case NPV are set as extra state constraints which are denoted as `State Constraint 1` and `State Constraint 2` in Table 3.2. Moreover, constraints values on average negative short-term NPV (`State Constraints 1`) and negative worst-case NPV (`State Constraints 2`) are also summarized in Table 3.2. Note that only the state constraints on the second and third objective functions are listed in this table, i.e., the state constraints enforced on the FWR and FLR are included in Table. 3.2. From Fig. 3.13, we can observe that the single objective optimization result has the lowest values of average short-term and average life-cycle NPV among all the optimization results. In fact, among all CDF's the lowest minimum NPV and lowest maximum NPV are the lowest for the CDF generated by minimizing the average negative life-cycle NPV. Comparing the Fig. 3.13a results for the short-term CDF for bi-objective optimization results where risk is the second objective (yellow line) with those where short-term NPV is the second objective (red line), we see that, as expected, the case obtained where short-term NPV is the second objective (red line) gives higher short-term NPV than the case is obtained where risk is the second objective (yellow line) in Fig. 3.13a. However, as also is expected, the case where the downside risk is the second objective (yellow lines) yields a higher value of the worst-case NPV than is obtained for the case where short-term NPV is the second objective (red line) in Fig. 3.13b. This choice of γ_1 and γ_2 also gives the perhaps unexpected result that, among the four CDF's for average short-term NPV and the four CDF's for average life-cycle NPV, the CDF's that gives highest individual NPV values are the two obtained from tri-objective optimization. Other choices for γ_1 and γ_2 will be explored later when we discuss Pareto fronts. Specific values of average life-cycle NPV, average short-term NPV and worst case NPV computed from single objective, bi-objective and tri-objective optimization results corresponding to the results of Fig. 3.13 are summarized in Table. 3.3. The number



(a) CDFs of short-term NPV



(b) CDFs of life-cycle NPV

Figure 3.13: CDF's of short-term NPV (left) and life-cycle NPV (right); **blue**: estimates from single objective life-cycle optimization; **red**: from modified lexicographic bi-objective optimization with state constraint on negative average (minimum average) short term NPV; **yellow:** from modified lexicographic bi-objective optimization with state constraint on negative worst-case NPV; **purple:** from modified lexicographic tri-objective optimization with state constraints on both negative average (minimum average) short term NPV and negative worst-case NPV.

of simulation runs (# of Sim.) in this table refers to the total number of simulation runs required to obtain the results including the 4300 simulation runs required to complete the single objective optimization of average negative life-cycle NPV.

The solution of a multi-objective optimization result is represented by a Pareto front [55, 53]. Each point on the Pareto front corresponds to one solution of a multi-objective optimization and the x coordinate and y coordinates of that point correspond to the values of distinct objective functions evaluated at that multi-objective optimum, respectively. Here, we let (x, y) denote the values of the two objective functions involved in the specified bi-objective optimization problem. Consider two points, $A = (x_1, y_1)$ and $B = (x_2, y_2)$. Point A is said to dominate point B if $x_1 > x_2$ and $y_1 \geq y_2$ or if $x_1 \geq x_2$ and $y_1 > y_2$ Otherwise B is not dominated by A. The Pareto front consist of all points that are not dominated by any other point. For tri-objective optimization, a similar relationship defines the Pareto surface; see Liu and Reynolds [53]. In this work, we attempt to generate the Pareto front for the

CDF color	Primary Objective	State Constraint 1	State Constraint 2
Blue	$\bar{J}_{NPV}$	-	-
Red	$\bar{J}_{NPV}$	$\bar{J}_{s-NPV} \leq -3.0 \times 10^9$	-
Yellow	$\bar{J}_{NPV}$	-	$J_{\text{risk}} \leq -4.6 \times 10^9$
Purple	$\bar{J}_{NPV}$	$\bar{J}_{s-NPV} \leq -3.0 \times 10^9$	$J_{\text{risk}} \leq -4.6 \times 10^9$

Table 3.2: Summary of the settings of constrained single objective, bi-objective and tri-objective optimization with modified lexicographic method.

Objective functions	$\bar{J}_{NPV}$ (USD)	J_{risk} (USD)	$\bar{J}_{s-NPV}$ (USD)	# of Sim.
$\bar{J}_{NPV}$	-4.86×10^9	-4.35×10^9	-2.44×10^9	4300
$\bar{J}_{NPV}$, $\bar{J}_{s-NPV}$	-4.96×10^9	-4.5×10^9	-3.01×10^9	5600
$\bar{J}_{NPV}$, J_{risk}	-4.94×10^9	-4.6×10^9	-2.6×10^9	5400
$\bar{J}_{NPV}$, $\bar{J}_{s-NPV}$, J_{risk}	-5.04 ×10^9	-4.64×10^9	-3.05×10^9	6200

Table 3.3: Summary of computational results of constrained single objective, bi-objective and tri-objective optimization with modified lexicographic method.

bi-objective optimization problem considered by gradually tightening the constraint value on the second objective function, either short-term NPV or worst-case NPV for the example under consideration where the modified lexicographic method is applied to to solve each of the sequence of optimization problems.

For the Brugge example under consideration, Fig. 3.14 represents three attempts to generate Pareto front or potential points on the Pareto surface. In all cases, the points on each of the three colored curve are generated by increasing the upper bound on the average negative short-term NPV. The point labeled 1 is the beginning point for all three curves and represents the results generated by simply minimizing average negative life-cycle NPV with no state constraint on average negative short-term NPV or on the negative worst-case NPV. As before, the optimal vector of controls obtained as the solution of this optimization problem is denoted by u^*. We first discuss the blue curve of Fig. 3.14 where the points 2, 3, 4

and 5, respectively, represent bi-objective optimization solutions with the constraint on the average negative short-term NPV given respectively, by $\bar{J}_{s-NPV} \leq -2.6 \times 10^9$ ($\gamma_1 = 1.06$), $\bar{J}_{s-NPV} \leq -2.8 \times 10^9$ ($\gamma_1 = 1.14$), $\bar{J}_{s-NPV} \leq -3.0 \times 10^9$ ($\gamma_1 = 1.23$), and $\bar{J}_{s-NPV} \leq -3.2 \times 10^9$ ($\gamma_1 = 1.31$). The first thing we note is that point 1 and the blue optimization solution 2 are dominated by points 3, 4 and 5 on the blue curve so the only points found that are on the Pareto front are blue points 3, 4 and 5. If we select the controls associated with blue point 3 as the well controls to be implemented in practice, we generate the highest average life-cycle NPV (4.97×10^9) obtained among the possibilities incorporated in optimal points 1, 2, 3, 4 and 5. However, if instead, we choose to implement the controls associated with blue point 5, we increase average life short-term NPV from the value 2.81×10^9 corresponding to point 3 to 3.07×10^9 USD (point 5) but this increase is achieved by inducing a very small decrease in the average life cycle NPV from 4.97×10^9 USD (point 3) to 4.95×10^9 USD corresponding to point 5. The black curve in Fig. 3.14 is generated by tri-objective optimization with the same four constraint values (γ_1 values) used to generate the blue points but with a constraint also enforced on the worst-case NPV, or, equivalently, on the upper bound of the maximum negative NPV; see Eq. 3.8c. The specific risk constraint used corresponds to using $\gamma_2 = 1.02$ in Eq. 3.8c. Since $J_{\text{risk}}(u^*) = 4.4 \times 10^9$ USD, Eq. 3.8c corresponds to $J_{\text{risk}} \leq \gamma_2 J_{\text{risk}}(u^*) = -4.5 \times 10^9$ USD or equivalently that the worst-case NPV is greater than or equal to 4.5×10^9 USD. The red curve is generated by the tri-objective optimization in the same way as the black curve except in this case the risk constraint is tightened to require that the worst-case NPV is greater than or equal to 4.6×10^9 USD. For the two tri-objective sets of optimization results, only two points obtained (points 4 and 5) are non-dominated points, i.e., are on the appropriate Pareto front.

Fig. 3.15 shows the results of attempts to generate a Pareto front by increasing the upper bound on the negative worst-case NPV where the two objectives are the average negative life-cycle NPV and the minimum NPV. The point labeled 1 is the beginning point and represents the result generated by simply minimizing average negative life-cycle NPV with no state constraints on the risk objective function. Points 2, 3, 4 and 5 on the curve in

Fig. 3.14, respectively, represent of the bi-objective optimization solutions with the average life-cycle NPV as the primary objective and a constraint on the worst-case NPV given respectively, by $\bar{J}_{\text{risk}} \leq -4.4 \times 10^9$ ($\gamma_2 = 1.01$), $\bar{J}_{\text{risk}} \leq -4.5 \times 10^9$ ($\gamma_1 = 1.03$), $\bar{J}_{\text{risk}} \leq -4.6 \times 10^9$ ($\gamma_1 = 1.06$), and $\bar{J}_{\text{risk}} \leq -4.7 \times 10^9$ ($\gamma_1 = 1.31$). We can observe that point 1, 2 and 3 are dominated by points 4 and 5 on the blue curve in Fig. 3.15 so the only points found that are on the Pareto front are blue points 4 and 5. If we select the controls associated with blue point 4 the well controls to be implemented in practice, we generate the highest average life-cycle NPV ($\$ 4.96 \times 10^9$) obtained among the possibilities incorporated in optimal points 1, 2, 3, 4 and 5. Compared with the optimal result at point 4, if we implement the controls associated with blue point 5, we could slightly increase the worst-case NPV from 4.6×10^9 USD to 4.61×10^9 USD at the negligible cost of decreasing the life-cycle NPV from 4.96×10^9 USD to 4.95×10^9 USD. Therefore, the average life-cycle NPV and worst-case NPV objectives conflict with each other at points 4 and 5. In Fig. 3.15, we see a similar performance at point 1, 2 and 3 as occurred in Fig. 3.14 which is the average life-cycle NPV increase as state constraint on worst-case NPV become stricter within some range. One plausible explanation of such results is that, because of the stochastic nature of the approximate gradients used in the single objective production optimization problem with the SQP-filter algorithm, the SQP-filter algorithm can be trapped into a local minimum easily even though this local optimum is not on the Pareto front of the multi-objective optimization problem. When the life-cycle average NPV is optimized again with a lower bound imposed as a state constraint on the average short-term NPV and/or the worst-case NPV, the gradients of these state constraints cause the modified lexicographic to escape the local minimum and and achieve a higher average life-cycle NPV. Once the lower bound imposed in the the state constraints become large enough, the two secondary objectives incorporated as state constraints act as if they are truly in conflict with the primary objective, the maximization of average life-cycle NPV. When this happens, the average life-cycle NPV obtained during the bi-objective or tri-objective optimization step is lower that the value obtained by optimizing only the average life-cycle NPV with no constraints on the other two objective functions and points that are

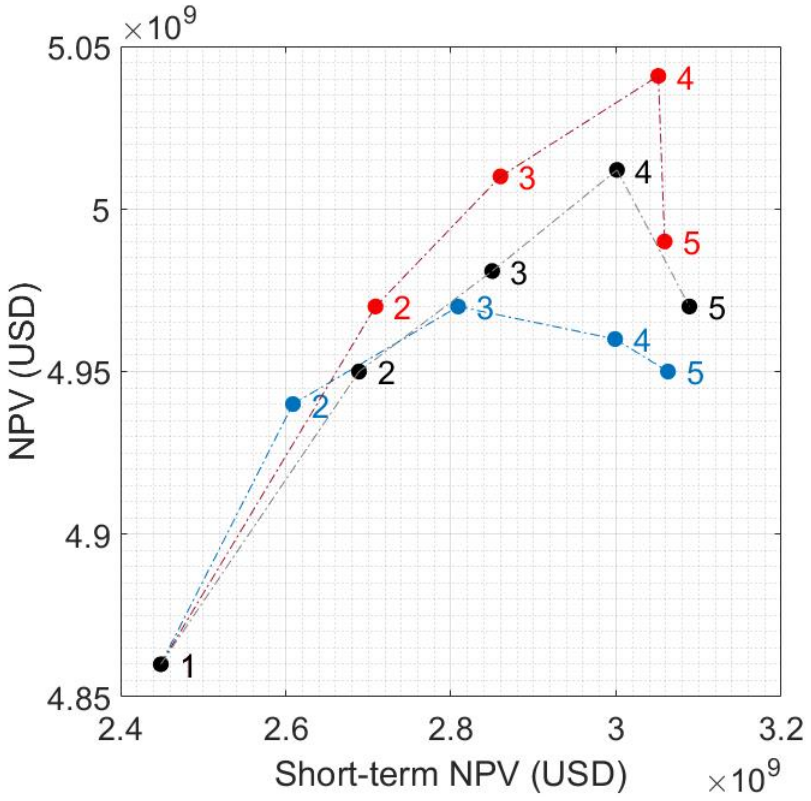


Figure 3.14: Pareto fronts of short-term NPV v.s. life-cycle NPV
Blue: short-term NPV constraint varies, no constraint on risk J_{risk}.
Black: $J_{risk} \leq -4.5 \times 10^9$, short-term NPV constraint varies.
Red: $J_{risk} \leq -4.6 \times 10^9$, short-term NPV constraint varies.

Pareto optimal can be obtained. Despite the aforementioned atypical behavior, the points generated in an attempt to find the Pareto front provides choices that enable the operational engineer to set controls based on his or her short-term and life-cycle production goals and risk tolerance.

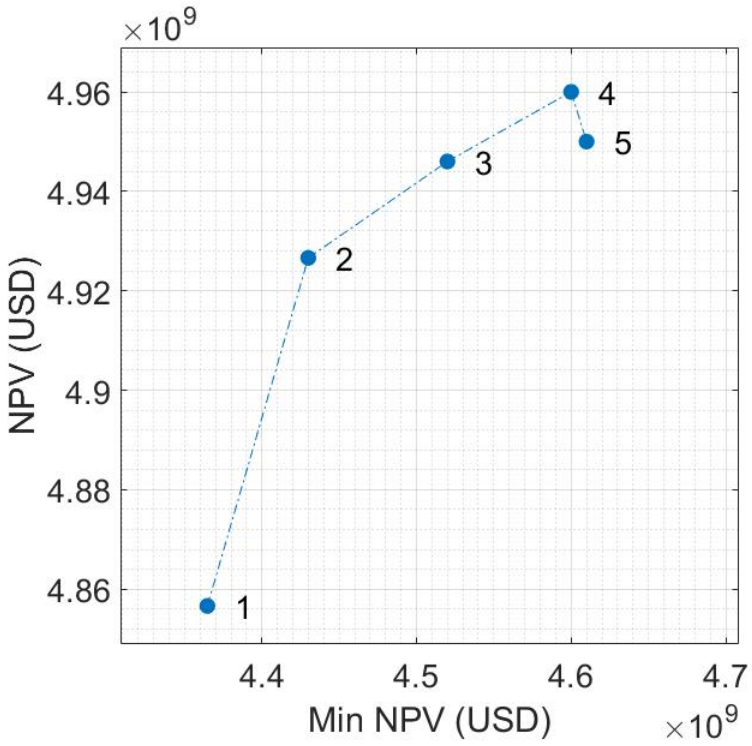


Figure 3.15: Pareto fronts of minimum NPV v.s. life-cycle minimum NPV constraint varies, no constraint on short-term NPV.

CHAPTER 4

ROBUST LIFE-CYCLE PRODUCTION OPTIMIZATION WITH GRADIENT-ENHANCED SUPPORT VECTOR REGRESSION

4.1 Introduction

In this chapter, we develop and employ a proxy-based optimization workflow to solve the robust life-cycle production optimization problem with state constraints which is described in Chapter 1. The work here considers only a single objective, namely, the minimization of the average NPV of life-cycle production where the average is over a set of realizations of the reservoir model that characterizes the geological uncertainty. However, as proxies are trained to reproduce the simulator prediction of constraint functions and the negative NPV as a function of the vector of well controls, u, the results obviously extend to the bi-objective case where either average negative short-term NPV or the minimization of downside risk is the second objective, and hence also extend to the tri-objective optimization problem considered previously. It is important to note that instead of training a proxy to predict the negative NPV function, $\bar{J}_(u, m)$, we train a unique proxy for objective function and each state constraints of each reservoir model m_k so that the associated proxies can predict the value of $J(u, m_k)$ and $c_i(u, m_k)$, $i = 1, 2, 3, \cdots, N_{ic}$ for any specified well control vector u. Our proxy-based optimization methodology achieves computational efficiency by generating a set of output values of the cost and constraints functions and their associated gradient values as a function of input values of u by running the reservoir simulator for a broad set of input variables (u's) and then, for each and every m_k, using the set of input/output data to train a proxy model to replace computationally expensive reservoir simulation runs when computing values of the cost and constraint functions and their derivatives during iterations of a gradient-based optimization algorithm (sequential

quadratic programming (SQP) with the filter method implemented for constraint handling). Computational results indicate that calculating values of cost/constraint functions and their derivatives using the proxy functions at iterations of the SQP method requires an order of magnitude less computational time than is required using the reservoir simulator model. Thus, the proxy-based optimization approach can enable the solution of large-scale production optimization problems where computational resources are not sufficient to provide a solution if the simulation model must be used for function and gradient evaluation at each iteration of the optimization algorithm. The methodology can be applied given any set of nonlinear state constraints. The proxy model used for optimization is a gradient-enhanced support vector (GE-SVR) regression model. We provide some comparison of the optimization results generated using both function and gradient information to build the proxy to results obtained when the proxy is trained using only function information, i.e., to results generated from least square support vector regression (LS—SVR) proxy. When values of gradients of the cost function (and possibly state constraints) are used to build a proxy, we refer to the proxy as a GE-SVR proxy even though a least-square procedure is also used to build the GE-SVR proxy. An iterative sampling scheme is also to improve the accuracy of proxy model around the optimal solution as the optimization proceeds. The computational results presented in this chapter illustrate that proxy-based optimization with the iterative sampling scheme solve the robust production optimization problem with state constraints efficiently when the reservoir simulator has the adjoint capability needed to generate the derivative information necessary to generate input-output samples for training the GE-SVR proxy.

4.2 Methodology

4.2.1 GE-SVR Model

In this section, we introduce the theoretical background of the GE-SVR model. Compared to the LS-SVR model, the GE-SVR enforces first order consistency by matching both

function value and gradient information at training points where LS-SVR uses only output function values. Given a number of training input samples, corresponding output response for the objective function and its gradient where the output responses are generated from the "true" model (the reservoir simulator in this work), the "GE-SVR" model "learns" how to map the input vector to output responses and generates a function to predict the output response of high-fidelity forward model for any given input. To be specific, for the production optimization problem, the input data is defined as the vector of scaled control variables of all wells at all control steps, x_k. A [0,1] scaling is used to normalize the entries of the vector of well controls, which is a standard procedure used for LS-SVR [81, 95, 37]. Recall from the definition of u, which is defined in Section 2.2.1, that u denotes the vector of all N_{u} controls which includes the controls of every well at every time step. Let u_ℓ be the ℓth component of u with upper bound u_ℓ^{up} and lower bound u_ℓ^{low}, then the scaled version, x, of u is formed by mapping u_ℓ to x_ℓ with the mapping given by

$$x_i = \frac{u_i - u_i^{\mathrm{low}}}{u_i^{\mathrm{up}} - u_i^{\mathrm{low}}}. \tag{4.1}$$

We let u^k, $k = 1, 2, \cdots, N_s$ be the input samples of the well control vector u and let x^k, $k = 1, 2, \cdots, N_s$ be the scaled version of u^k obtained from Eq. 4.1. The output response of the GE-SVR model for the input sample $x^k = x(u^k)$ is denoted by $\hat{y}^k, k = 1, 2, ...N_s$. Effectively, we train the GE-SVR model for each type of output, e.g., one proxy for $J(u, m_k)$ and $\nabla_u J(u, m_k)$, one proxy for each $c_i(u, m_k)$ and $\nabla_u c_i(u, m_k)$ for $i = 1, 2, \cdots, N_{ic}$ and $k = 1, 2, \cdots, N_e$, where N_{ic} is the number of inequality constraints and N_e is the number of realizations considered in order to account for geological uncertainty.

The SVR model maps an N_u-dimensional input vector x (corresponding to u) to the output response of the model, $\hat{y}$, by

$$\hat{y}(x) = w^T \cdot \phi(x) + b, \tag{4.2}$$

where $\phi(\cdot)$ is a nonlinear mapping of the input space to the higher dimensional feature space;

w is a vector of weights corresponding to the dimension of feature space; b is a scalar, which represents the bias term. Here $\hat{y}(x)$ in Eq. 4.2 denotes the SVR proxy used to approximate $y(x)$ output from reservoir simulator. One of the biggest advantages of the LS-SVR or GE-SVR model over linear regression or other machine learning methods is that the user does not need to specify the dimension of the feature space and does not need to explicitly define the mapping ϕ into the feature space. The model is built based on a kernel function which represents an inner product in the feature space. More details about the derivation and explanation of the LS-SVR for reservoir engineering applications can be found in Guo and Reynolds [37] and Emilio and Reynolds [25].

The gradient information in the training samples is generated by a reservoir simulator with the adjoint capability. An in-house 3D three-phase reservoir simulator [60] with adjoint capability for both history matching and production optimization is used as the true model to generate model output responses including gradient information. The r^{th} entry of the gradient of the proxy model with respect to the input control variables is defined as

$$D_r(x) \equiv \frac{\partial \hat{y}(x)}{\partial x_r} = w^T \cdot \frac{\partial \phi(x)}{\partial x_r} \tag{4.3}$$

where x_r denotes the r^{th} element of input vector x. For $k = 1, 2, \cdots, N_s$, where N_s is the number of training samples, u^k denotes the control vector which is input into the numerical model (reservoir simulator in this research) to generate the kth training data and x^k denotes the corresponding scaled training input vector. We let $y_k = y(u^k)$ and $D_r(x) \equiv \frac{\partial y(x)}{\partial x_r}$ i.e., $D_r(x)$ is the rth component of $\nabla_x y(x)$ computed from the adjoint solution of the numerical model (reservoir simulator in this research). Note that the numerical model uses the adjoint solution to compute $\nabla_u y(u)$ and then

$$D_r(x) = \frac{\partial y(x)}{\partial x_r} \equiv \frac{\partial y(x)}{\partial u_r}\frac{\partial u_r}{\partial x_r} = \frac{\partial y(x)}{\partial u_r}(u_r^{\text{up}} - u_r^{\text{low}}), \tag{4.4}$$

see Eq. 4.1. Similar to the LS-SVR procedure [37, 25], to construct the GE-SVR model given training sample $(x^k, y(x^k), \nabla_x y(x^k))$, $k = 1, 2, \cdots N_s$ we solve a convex quadratic optimiza-

tion defined by

$$\underset{w,b,e}{\text{minimize}} \quad \{J(w,b,e) \equiv \frac{1}{2}w^T w + \frac{1}{2}C_0 \quad \sum_{k=1}^{N_s}(e_k)^2 + \frac{1}{2}\sum_{k=1}^{N_s}\sum_{r=1}^{N_u} C_r[\hat{e}_r(x^k)]^2\}, \tag{4.5a}$$

$$\text{subject to} \quad e_k = y^k - w^T\phi(x^k) - b \qquad k = 1, 2, \ldots, N_\text{s}, \tag{4.5b}$$

$$\hat{e}_r(x^k) = D_r(x^k) - w^T\frac{\partial\phi(x^k)}{\partial x_r} \qquad k = 1, 2, \ldots, N_\text{s},\ r = 1, 2, \ldots, N_\text{u}, \tag{4.5c}$$

where the first term in Eq. 4.5a serves as a regulation term to prevent potential overfitting of the GE-SVR model; the second and third terms, respectively, denote the weighted data mismatch between the function training data and associated derivative training data, respectively, and the corresponding proxy predicted data. C_0 and C_r, $r = 1, 2, \cdots N_u$ are hyperparameters which define the weights on function mismatch and the derivative mismatch in Eq. 4.5a. Finally, e_k and $\hat{e}_r(x^k)$ in Eqs. 4.5b and 4.5c, respectively define the response and gradient mismatches between the training data and the proxy model as equality constraints.

To solve this constrained optimization problem, we let $\hat{e}_r^k \equiv \hat{e}_r(x^k)$ and define the Lagrangian function

$$\begin{aligned} L(w,b,e;\alpha,\beta) = \frac{1}{2}w^T w &+ \sum_{k=1}^{N_s}[\frac{1}{2}C_0(e_k)^2 + \alpha_k(y^k - w^T\phi(x^k) - b - e_k)] \\ &+ \sum_{k=1}^{N_s}\sum_{r=1}^{N_u}[\frac{1}{2}C_r(\hat{e}_r^k)^2 + \beta_{r,k}(D_r(x^k) - w^T\frac{\partial\phi(x^k)}{\partial x_r} - \hat{e}_r^k)]. \end{aligned}$$

where α_k and $\beta_{r,k}$, $k = 1, 2, \cdots, N_s$ and $r = 1, 2, \cdots, N_u$ are the Lagrange multipliers for the constraints given by Eqs. 4.5b and 4.5c; their values are obtained by minimizing $L(w,b,e;\alpha,\beta)$. More generally, according to the first order optimality condition [67], the

solution of the optimization problem should satisfy the following five equations:

$$\nabla_w L = w - \sum_{k=1}^{N_s} \alpha_k \phi(x^k) - \sum_{k=1}^{N_s} \sum_{r=1}^{N_u} \beta_{r,k} \frac{\partial \phi(x^k)}{\partial x_r} = 0, \tag{4.6a}$$

$$\frac{\partial L}{\partial b} = \sum_{k=1}^{N_s} \alpha_k = 0, \tag{4.6b}$$

$$\frac{\partial L}{\partial e_k} = C_0 e_k - \alpha_k = 0, \tag{4.6c}$$

$$\frac{\partial L}{\partial \hat{e}_r^k} = C_r \hat{e}_r^k - \beta_{r,k} = 0, \tag{4.6d}$$

$$\frac{\partial L}{\partial \alpha_k} = y_k - w^T \phi(x^k) - b - e_k = 0, \tag{4.6e}$$

$$\frac{\partial L}{\partial \beta_{r,k}} = D_r(x^k) - w^T \frac{\partial \phi(x^k)}{\partial x_r} - \hat{e}_r^k = 0. \tag{4.6f}$$

for $k = 1, 2, \cdots, N_s$ and $r = 1, 2, \cdots, N_u$. Rearranging Eq. 4.6 with the indices of summation k and r, respectively, replaced by j and m in Eq. 4.6b gives

$$w = \sum_{j=1}^{N_s} \alpha_j \phi(x^j) + \sum_{j=1}^{N_s} \sum_{m=1}^{N_u} \beta_{m,j} \frac{\partial \phi(x^j)}{\partial x_m}, \tag{4.7a}$$

$$\sum_{k=1}^{N_s} \alpha_k = 0, \tag{4.7b}$$

$$\alpha_j = C_0 e_k, \tag{4.7c}$$

$$\beta_{r,k} = C_r \hat{e}_r^k, \tag{4.7d}$$

$$w^T \phi(x^k) + b + e_k = y^k, \tag{4.7e}$$

$$w^T \frac{\partial \phi(x^k)}{\partial x_r} + \hat{e}_r^k = D_r(x^k), \tag{4.7f}$$

where w is a constant vector and Eqs. 4.7(c) through 4.7(f) apply for $k = 1, 2, \cdots N_s$ and $r = 1, 2, \cdots N_u$. By substituting Eq. 4.7a into Eq. 4.7e, we have

$$\sum_{j=1}^{N_s} \alpha_j \phi(x^j)^T \phi(x^k) + \sum_{j=1}^{N_s} \sum_{m=1}^{N_u} \beta_{m,j} \frac{\partial \phi(x^j)}{\partial x_m}^T \phi(x^k) + b + \frac{\alpha_k}{C_0} = y^k, \tag{4.8}$$

for $k = 1, 2, \cdots, N_s$. Note that for any kernel function $K(x, \hat{x})$ that satisfies Mercer's condi-

tions [66], can be written as the inner product of mapping ϕ such that

$$K(x, \hat{x}) = \phi(x)^T \phi(\hat{x}), \tag{4.9}$$

for any input vectors x and $\hat{x}$. The partial derivative of the kernel function is given by

$$\frac{\partial K(x, \hat{x})}{\partial x_m} = \frac{\partial \phi(x)}{\partial x_m}^T \phi(\hat{x}). \tag{4.10}$$

Therefore, Eq. 4.8 can be written as

$$\sum_{j=1}^{N_s} \alpha_j K(x^j, x^k) + \sum_{j=1}^{N_s} \sum_{m=1}^{N_u} \beta_{m,j} \frac{\partial K(x^j, x^k)}{\partial x_m} + b + \frac{\alpha_k}{C_0} = y^k. \tag{4.11}$$

By substituting Eqs. 4.7a and 4.7d into Eq. 4.7f, we have

$$\sum_{j=1}^{N_s} \alpha_j \phi(x^j)^T \frac{\partial \phi(x^k)}{\partial x_r} + \sum_{j=1}^{N_s} \sum_{m=1}^{N_u} \beta_{m,j} \frac{\partial \phi(x^j)}{\partial x_m}^T \frac{\partial \phi(x^k)}{\partial x_r} + \frac{\beta_{r,k}}{C_r} = D_r(x^k), \tag{4.12}$$

for $k = 1, 2, \cdots, N_s$ and $m = 1, 2, \cdots, N_u$. The second order derivative of the Gaussian kernel is given by

$$\frac{\partial K^2(x, \hat{x})}{\partial x_m \partial \hat{x}_r} = \frac{\partial \phi(x)}{\partial x_m}^T \frac{\partial \phi(\hat{x})}{\partial \hat{x}_r}. \tag{4.13}$$

Therefore, Eq. 4.12 can be written as

$$\sum_{j=1}^{N_s} \alpha_j \frac{\partial K(x^j, x^k)}{\partial \hat{x}_r} + \sum_{j=1}^{N_s} \sum_{m=1}^{N_u} \beta_{m,j} \frac{\partial K^2(x^j, x^k)}{\partial x_m \partial \hat{x}_r} + \frac{\beta_{r,k}}{C_r} = D_r(x^k). \tag{4.14}$$

for $k = 1, 2, \cdots N_s$ and $r = 1, 2, \cdots N_u$.

Next, some notations are introduced. We define

$$\gamma = [\gamma_0^T \quad \gamma_1^T \quad ... \gamma_{N_u}^T]^T, \tag{4.15}$$

where

$$\gamma_0 = [\alpha_1 \quad \alpha_2 \quad ...\alpha_{N_s}]^T, \tag{4.16}$$

and

$$\gamma_r = [\beta_{r,1} \quad \beta_{r,2} \quad ...\beta_{r,N_s}]^T \tag{4.17}$$

for $r = 1, 2, \cdots N_u$. Then the vector of all training output is integrated into an $N_s + N_s \times N_u = N_s(N_u + 1)$ dimensional column vector Y which is defined by

$$Y = [\quad [y(x^1) \cdots y(x^{N_s})]^T \quad (\hat{D}_1)^T \quad ...(\hat{D}_{N_u})^T]^T, \tag{4.18}$$

where

$$\hat{D}_r = [D_r(x^1) \quad ...D_r(x^{N_s})]^T, \tag{4.19}$$

for $r = 1, 2, \cdots N_u$. With this notation, we can write the complete set of linear equations specified by Eqs. 4.11 and 4.14 in the following matrix form:

$$\begin{bmatrix} \Phi & \widetilde{e} \\ \widetilde{e}^T & 0 \end{bmatrix} \begin{bmatrix} \gamma \\ b \end{bmatrix} = \begin{bmatrix} Y \\ 0 \end{bmatrix}. \tag{4.20}$$

where

$$\widetilde{e} = [\overbrace{1...1}^{N_s} \overbrace{0 \quad 0...0 \quad 0}^{N_s \times N_u}]^T \tag{4.21}$$

The matrix Φ in Eq. 4.20 is $N_g \times N_g$ where $N_g = N_s(N_u + 1) + 1$. and is given by

$$\Phi = \begin{bmatrix} H_{00} & H_{11} & ... & H_{1N_u} \\ H_{11} & Q_{11} & ... & Q_{1N_u} \\ ... & ... & ... & ... \\ H_{N_u 1} & Q_{N_u 1} & ... & Q_{N_u N_u} \end{bmatrix}, \tag{4.22}$$

Where the submatrices in Eq. 4.22 are defined below. I_{N_s} denotes the $N_s \times N_s$ identity

matrix; where

$$H_{00} = H_{00}^{*} + \frac{1}{C_0} I_{N_s}, \tag{4.23}$$

where

$$H_{00}^{*} = \begin{bmatrix} K(x^1, x^1) & ... & K(x^{N_s}, x^1) \\ ... & ... & ... \\ K(x^1, x^{N_s}) & ... & K(x^{N_s}, x^{N_s}) \end{bmatrix}. \tag{4.24}$$

$$H_{1r} = \begin{bmatrix} \frac{\partial K(x^1, x^1)}{\partial x_r} & ... & \frac{\partial K(x^{N_s}, x^1)}{\partial x_r} \\ ... & ... & ... \\ \frac{\partial K(x^1, x^{N_s})}{\partial x_r} & ... & \frac{\partial K(x^{N_s}, x^{N_s})}{\partial x_r} \end{bmatrix} \quad \text{for} \quad r = 1, ..., N_u, \tag{4.25}$$

$$H_{r1} = \begin{bmatrix} \frac{\partial K(x^1, x^1)}{\partial \hat{x}_r} & ... & \frac{\partial K(x^{N_s}, x^1)}{\partial \hat{x}_r} \\ ... & ... & ... \\ \frac{\partial K(x^1, x^{N_s})}{\partial \hat{x}_r} & ... & \frac{\partial K(x^{N_s}, x^{N_s})}{\partial \hat{x}_r} \end{bmatrix} \quad \text{for} \quad r = 1, ..., N_u. \tag{4.26}$$

$$Q_{m,r} = \begin{cases} Q_{m,r}^{*} + \frac{1}{C_r} I_{N_u}, & \text{if } r = m \\ Q_{m,r}^{*}, & \text{otherwise} \end{cases} \tag{4.27}$$

for $r = 1, 2, \cdots, N_u$ and $m = 1, 2, \cdots, N_u$, where

$$Q_{m,r}^{*} = \begin{bmatrix} \frac{\partial K^2(x^1, x^1)}{\partial x_m \partial \hat{x}_r} & ... & \frac{\partial K^2(x^{N_s}, x^1)}{\partial x_m \partial \hat{x}_r} \\ ... & ... & ... \\ \frac{\partial K^2(x^1, x^{N_s})}{\partial x_m \partial \hat{x}_r} & ... & \frac{\partial K^2(x^{N_s}, x^{N_s})}{\partial x_m \partial \hat{x}_r} \end{bmatrix} \tag{4.28}$$

for $r = 1, 2, \cdots, N_u$ and $m = 1, 2, \cdots, N_u$. In this work, we choose the Gaussian kernel as our kernel function so that

$$K(x, \hat{x}) = \exp(-\frac{||x - \hat{x}||^2}{2\sigma^2}), \tag{4.29}$$

and the partial derivative of the Gaussian kernel is given by

$$\frac{\partial K(x, \hat{x})}{\partial x_r} = -\frac{1}{\sigma^2}(x_r - \hat{x}_r) K(x, \hat{x}), \tag{4.30}$$

and

$$\frac{\partial K(x,\hat{x})}{\partial \hat{x}_r} = \frac{1}{\sigma^2}(x_r - \hat{x}_r)K(x,\hat{x}). \tag{4.31}$$

The second order derivative of Gaussian kernel is given by:

$$\frac{\partial K^2(x,\hat{x})}{\partial x_m \partial \hat{x}_r} = -\frac{1}{\sigma^4}(x_r - \hat{x}_r)(x_m - \hat{x}_m)K(x,\hat{x}). \tag{4.32}$$

Note that the left hand side of the matrix in the linear system given by Eq. 4.20 only depends on input data, i,e, the x^k, $k = 1, 2, \ldots N_s$ and does not depend on the training output function $y(x)$. Therefore, by simply replacing the right hand side of Eq. 4.20 with the appropriate training output, one can generate a GE-SVR proxy model for any type of output, i.e. one GE-SVR proxy for the objective function and its gradient and one for each nonlinear state constraint and its derivatives. This effectively means that we a GE-SVR model trained to match/predict the NPV and its gradient and one trained to match/predict each nonlinear constraint function and its derivatives for each reservoir model m_ℓ, $\ell = 1, 2, \cdots N_e$ used to characterize reservoir model uncertainty.

By substituting Eq. 4.7a into Eq. 4.2, the prediction of the GE-SVR model at an unsampled point, x in the input space can be expressed as

$$\hat{y}(x) = \sum_{j=1}^{N_s} \alpha_j \phi(x^j)^T \phi(x) + \sum_{j=1}^{N_s}\sum_{m=1}^{N_u} \beta_{m,j} \frac{\partial \phi(x^j)}{\partial x_m}^T \phi(x) + b, \tag{4.33}$$

which from Eqs. 4.9 and 4.10 maybe is equivalent to

$$\hat{y}(x) = \sum_{j=1}^{N_s} \widehat{\alpha}_j K(x^j, x) + \sum_{j=1}^{N_s}\sum_{r=1}^{N_u} \widehat{\beta}_{r,j} \frac{\partial K(x^j, x)}{\partial \hat{x}_r} + b, \tag{4.34}$$

where $(\widehat{\alpha}, \widehat{\beta}, b)$ is the solution of linear system in Eq. 4.20. Similarly, the analytical gradient predicted by the GE-SVR model at an upsampled point, x in the input space can be expressed as

$$\frac{\partial \hat{y}(x)}{\partial x_r} = \sum_{j=1}^{N_s} \alpha_j \phi(x^j)^T \frac{\partial \phi(x)}{\partial x_r} + \sum_{j=1}^{N_s}\sum_{m=1}^{N_u} \beta_{m,j} \frac{\partial \phi(x^j)}{\partial x_m}^T \frac{\partial \phi(x)}{\partial x_r}, \tag{4.35}$$

which from Eqs. 4.10 and 4.13 maybe is equivalent to

$$\frac{\partial \hat{y}(x)}{\partial x_r} = \sum_{j=1}^{N_s} \widehat{\alpha}_j \frac{\partial K(x^j, x)}{\partial \hat{x}_r} + \sum_{j=1}^{N_s} \sum_{m=1}^{N_u} \widehat{\beta}_{m,j} \frac{\partial^2 K(x^j, x)}{\partial x_m \partial \hat{x}_r} + b. \tag{4.36}$$

4.2.2 Proxy-based Optimization

To construct a proxy model, the initial sampling of the input vector is generated by Latin hypercube sampling (LHS) [65], which is a common method for the design of experiments. The advantage of LHS is that the N_s sample points are distributed evenly but randomly in the design space. Guo and Reynolds [37] select N_s=200 samples to train the LS-SVR model for producing NPV. LS-SVR does not use any output related to derivative information in training the proxy model. On the other hand, we have often used only $N_s = 5$ initial samples to build a GE-SVR proxy model, but this number is increased during optimization by iterative sampling. Specifically, after training the proxy, the optimization problem is solved using the GE-SVR model in place of the reservoir simulator when minimizing the average negative life-cycle NPV using sequential quadratic programming (SQP) algorithm with the filter method incorporated to enforce state constraints [61]. Throughout the subscript proxy on an objective or constraint violation function means the function is evaluated at a result obtained from or during the proxy-based optimization process using the SQP-filter method. The SQP optimization algorithm with a given proxy model is terminated once

$$\Delta \bar{J}^{\ell}_{\text{proxy}} < \epsilon_J, \tag{4.37}$$

and

$$\bar{\theta}_{\text{proxy}}(x^{\ell}) < \epsilon_{\theta}, \tag{4.38}$$

where $\bar{\theta}$ is defined in Eq. 2.49 and

$$\Delta \bar{J}^{\ell}_{\text{proxy}} = \frac{\bar{J}_{\text{proxy}}(x^{\ell}) - \bar{J}_{\text{proxy}}(x^{\ell-1})}{\bar{J}_{\text{proxy}}(x^{\ell})}. \tag{4.39}$$

Eq. 4.37 specifies one convergence criterion based on the relative change in the objective

function from iteration $\ell - 1$ to iteration ℓ during proxy-based optimization with the SQP-filter method, and Eq. 4.38 specifies a second convergence criterion that must be satisfied for termination of proxy-based optimization with the current GE-SVR proxy. Clearly, ϵ_J is the convergence tolerance on the change in the objective function (average negative life-cycle NPV) during the iterations of the SQP-filter method, and ϵ_θ is the convergence tolerance for the average constraint violation of single realization for the proxy-based SQP-filter method. In Eqs. 4.38 and 4.39, x^ℓ is the estimate of the scaled control vector at the current iterate of the SQP algorithm, $x^{\ell-1}$ is estimated at the previous iteration. One should note of course that $\bar{J}(x^\ell) \equiv \bar{J}(u^\ell)$, where the relationship between the scaled vector of controls, x, and unscaled controls, u is given by Eq. 4.1. In our computational examples, we set $\epsilon_J = 10^{-5}$ and $\epsilon_\theta = 10^{-4}$. We set stricter convergence criteria in the proxy-based SQP-filter algorithm than what we proposed in Section 2.2.4, since function and gradient evaluations using the proxy model are much cheaper than using the reservoir simulator and StoSAG to compute the values of the objective function, state constraints and corresponding gradients as was done in Chapter 2. Therefore, proxy-based optimization can afford more SQP iterations than the simulator-based optimization. Furthermore, the stricter convergence criteria can lead proxy-based optimization to reach better results in terms of a lower values of the objective functions and a lower value of the average constraint violation.

As mentioned previously, we use iterative sampling to improve the GE-SVR (or LS-SVR) proxy after each SQP optimization. The basic idea of iterative sampling is as follows: after proxy-based SQP optimization converges for a given GE-SVR proxy, we insert the optimal design variables into the high-fidelity numerical model (reservoir simulator) and compare the output from the simulator with the proxy generated response. If the two responses are not in agreement or the output from the simulator does not satisfy the constraints, the optimal scaled controls and corresponding reservoir simulator output are added to the training set, new GE-SVR proxies are generated and proxy-based optimization is repeated with the new trained GE-SVR proxies. This cycle is repeated until there is reasonable agreement between the simulator and proxy output responses evaluated at the optimal well controls generated

from proxy-based SQP optimization and the constraints evaluated from the simulator output are satisfied. Throughout, the subscript sim on a NPV function or on a constraint violation function indicates the function is evaluated using output from the reservoir simulator. The two criteria we have commonly used to test convergence of the iterative sampling procedure are enumerated immediately below.

1. The maximum allowable relative difference ϵ_{p} between the objective function values predicted by the proxy model and the reservoir simulator must be in good agreement, i.e., must satisfy

$$\frac{|\bar{J}_{\mathrm{sim}}(u^*) - \bar{J}_{\mathrm{proxy}}(x^*)|}{|\bar{J}_{\mathrm{sim}}(u^*)|} \leq \epsilon_{\mathrm{p}}. \tag{4.40}$$

   In all examples presented, the default value of ϵ_{p} is set equal to 0.01 which is the default value used in our in-house software.

2. At the optimum of each SQP proxy-based optimizations within iterative sampling, we denote the optimal scaled control vector by x^* with the corresponding vector of unscaled optimal well controls denoted by u^*. Then the second convergence criterion, which is on the overall average constraint violation computed from the reservoir simulator is given by

$$\bar{\theta}_{\mathrm{sim}}(u^*) \leq \epsilon_{\mathrm{c}}, \tag{4.41}$$

   where $\bar{\theta}_{\mathrm{sim}}$ is defined by Eq. 2.49. Since θ^*_{sim} is a measure of the scaled constraints, the magnitude of θ^*_{sim} depends on the scaling factors of the state constraints. If state constraints in the form of bounds are scaled by dividing each of them by their bound as is the case for the example in subsection 4.3.3, then θ^*_{sim} should be set equal to 0.01.

If either of two preceding criterion is not satisfied, we add the input/output sample result generated from the reservoir simulator with the proxy-based optimal controls to the old training set, generate new GE-SVR proxies and repeat the proxy-based optimization workflow again to improve the performance of the updated GE-SVR proxy model. The iterative sampling scheme is terminated once both of the two preceding criteria are satisfied. The

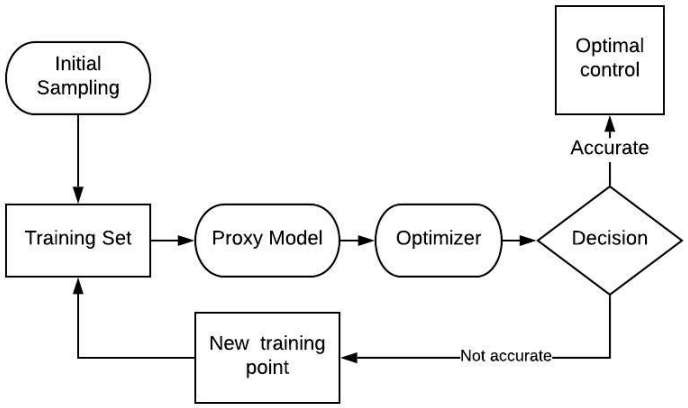


Figure 4.1: The workflow of proxy-based optimization with iterative sampling.

general workflow for proxy-based optimization is illustrated in Fig. 4.1 and the complete algorithm for proxy-based optimization with iterative sampling scheme is given immediately below.

1. Set the values ϵ_J and ϵ_θ as the criteria to stop the proxy-based SQP optimization. Set ϵ_p and ϵ_c as the criteria to stop the complete optimization process.

2. Specify the upper bound (u^{up}) and lower bound (u^{low}) for each component of the vector of design variables (control vector). Set the number of samples, N_s, to be used as the number samples in initial training sets.

3. Use LHS to generate N_s samples of the scaled input vector x, which is related to input vector u by Eq. 4.1. Generate the N_s sets of the output function $y(x)$ (e.g., life-cycle NPV) and its gradient with respect to x from the numerical model that we wish to replace with a proxy model in the optimization process. For the constrained optimization problem, the values of each nonlinear constraints and corresponding gradient values also need to be generated and used as training set outputs to generate proxies for the constraint functions and their derivatives. Set $n_\ell = 1$ and initialize $x^{*,0}$ as a vector with all elements equal to 0.5. Throughout, x^{*,n_ℓ} denotes the vector of scaled optimal controls obtained by proxy-based optimization using the n_ℓth GE-SVR proxies.

4. Train a GE-SVR proxy model to predict (estimate) simulator generated output values of $y(x)$ and its derivatives corresponding to inputs, and similarly train a proxy for each state constraint and its gradient.

5. Sample an initial guess as $x^{0,n_\ell} = x^{*,n_\ell-1} + \delta$, where δ is a $N_u \times 1$ vector whose elements are sampled from the normal distribution $N(0, \sigma_{rs}^2)$ where σ_{rs} is such that $0 \leq \sigma_{rs} \leq 0.1$; see the comment following the description of the algorithm. Apply truncation to bound x^{0,n_ℓ} within the interval $[0, 1]$. Apply SQP optimization with the GE-SVR model used to calculate output function and constraint function values and their derivatives. Let the scaled optimal well controls obtained be denoted by x^{*,n_ℓ} with corresponding unscaled optimal controls given by u^{*,n_ℓ}; see Eq. 4.1. In this SQP-filter proxy-based optimization, the min-max scheme proposed in Section 2.2.5 is applied to solve the robust optimization problem efficiently where the bound constraints are handled by truncation (see Section 2.2.4), and the SQP-filter algorithm is terminated when $\Delta\bar{J}_{\text{proxy}}(x^{*,n_\ell}) < \epsilon_J$ and $\bar{\theta}_{\text{proxy}}(x^{*,n_\ell}) < \epsilon_\theta$.

6. Insert the optimal control variables u^{*,n_ℓ} obtained in step 5 into the reservoir simulator to compute the objective function $\bar{J}_{\text{sim}}(u^{*,n_\ell})$, and overall average constraint violation $\bar{\theta}_{\text{sim}}(u^{*,n_\ell})$.

7. Compare the proxy-generated objective function value, $\bar{J}_{\text{proxy}}(x^{*,n_\ell})$, evaluated at the estimate of the scaled optimal x^{*,n_ℓ} to the corresponding value, $\bar{J}_{\text{sim}}(u^{*,n_\ell})$, computed from the reservoir simulator at the optimal u^{*,n_ℓ} corresponding to x^{*,n_ℓ}. and compute the average constraint violation $\bar{\theta}_{\text{sim}}(u^{*,n_\ell})$ using reservoir simulator output.

   - **IF** $|\bar{J}_{\text{sim}}(u^{*,n_\ell}) - \bar{J}_{\text{proxy}}(x^{*,n_\ell})|/\bar{J}_{\text{sim}}(u^{*,n_\ell}) \leq \epsilon_{\text{p}}$ **AND** $\bar{\theta}_{\text{sim}}(u^{*,n_\ell}) \leq \epsilon_c$
     - Terminate the iterative sampling algorithm.
   - **ELSE**
     - Add the new optimal values of the design variables and the corresponding responses obtained from the reservoir simulator into the old training set to

obtained a new training set. Retrain GE-SVR proxy models for the objective and state constraints using the updated training sets.

- Set $n_\ell = n_\ell + 1$ and go back to the step 5.

- **END IF**

Important comment. When starting with the next iterative sampling step, where we optimize with a new updated GE-SVR proxy, it is logical to start the SQP-filter for optimizing the new GE-SVR model using as the initial guess the estimated optimum obtained by minimizing the previous GE-SVR proxy. With this approach one set $\sigma_{rs} = 0$ in the normal distribution of step 5, or more directly, simply sets $\delta = 0$. However, we conducted some experiments to see if adding a small perturbation to the previous optimum to define the initial guess would yield a better approximation of the true objective function, avoid getting trapped at a bound, and generate a local minimum closer to a global optimum. However, we could not draw any general conclusions from our experiments, some of which are discussed later. As our intuition about the value of adding a small perturbation persists, the main reservoir model examples presented are based on using $\delta_{rs} = 0.1$. One could of course also decrease the value of σ_{rs} as n_ℓ increases. It is important to note, however, that setting the value of σ_{rs} much greater than 0.2 could cause each iterative sampling step to converge to a different point and might even make it impossible for multiple applications of the iterative sampling algorithm to find multiple local minimums by starting with a set of different initial guesses at the first optimization step

4.3 Computational Results

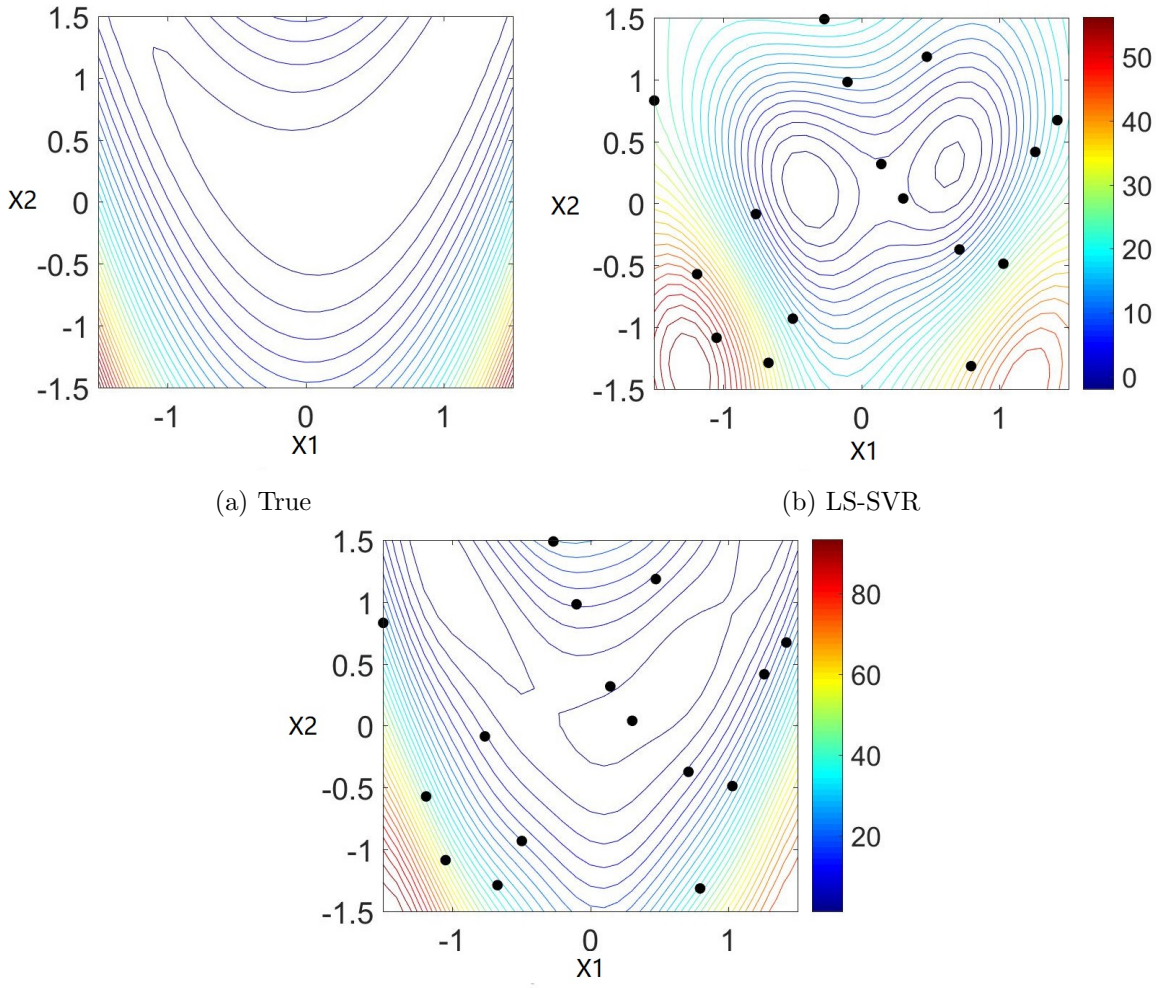
In this section, we illustrate the performance of our proxy-based optimization framework with GE-SVR as the proxy model. Two simple standard optimization test cases are considered in Example 1. Examples 2 and 3, respectively, consider life-cycle production optimization with only bound constraints and then with both bound and state constraints.

4.3.1 Example 1

In the first example, two distinct 2D functions which represent two standard optimization test cases are used to illustrate that GE-SVR provides a more accurate approximation of the true output response surface than does LS-SVR. The first validation function considered is the 2D Rosenbrock function, which is a commonly used benchmark function to test the performance of optimization algorithms. The second validation function considered is a 2D multi-minimal function, which has nine local minimums within the feasible region defined by the bounds on the two optimization variables. This example is pertinent because large-scale life-cycle production optimization problems have multiple local minimums (maximums). These 2D functions are also useful because the results are easy to visualize and validate.

Figs. 4.2 and 4.3 compare the true Rosenbrock response surface with the proxy-generated response surfaces of the LS-SVR and GE-SVR proxy models. For generating the LSV-proxies, we use $\sigma = 10$ and $C_0 = 1000$ and in generating the GE-SVR proxies, we use $\sigma = 10$, $C_0 = 1000$ and $C_r = 200$ for all r; see Eqs. 4.5 - 4.32. The location of sampling points obtained from LHS which are used to generate the response surfaces for both the LS-SVR and GE-SVR proxy models are shown as black dots in these and similar figures. The shape of response surfaces constructed by different models are represented by contour maps. From those results, we can observe that, compared with the LS-SVR result, the GE-SVR model better approximates the shape of the true response surface. In Fig. 4.2, the proxy models were generated using 16 training samples whereas the proxy models of Fig. 4.3 are constructed using only 10 sample points to investigate the effect of the number of sampling points on the accuracy of proxy modeling. For both the 10 and 16 sampling point cases, the GE-SVR proxy (Fig. 4.3) clearly gives better approximation of the shape of the true response surface than is obtained with the LS-SVR model.

It is important to note that the objective of using GE-SVR is to obtain with less computational cost estimates of local minimums that are as good or better than those obtained with LS-SVR. The true Rosenbrock function has only one local minimum which is equal

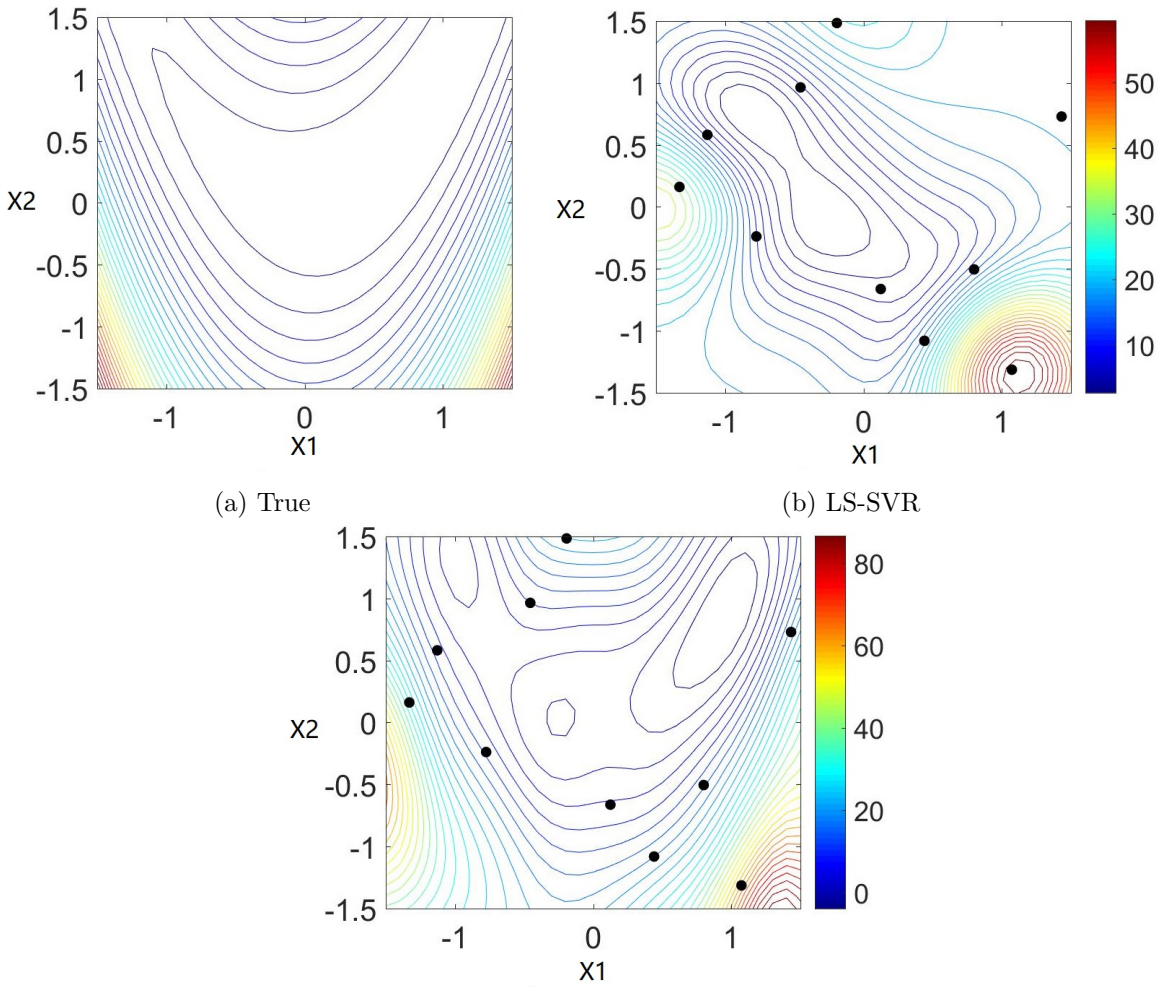


(a) True

(b) LS-SVR

(c) GE-SVR

Figure 4.2: Response surface of Rosenbrock function with 16 sample points.



(a) True

(b) LS-SVR

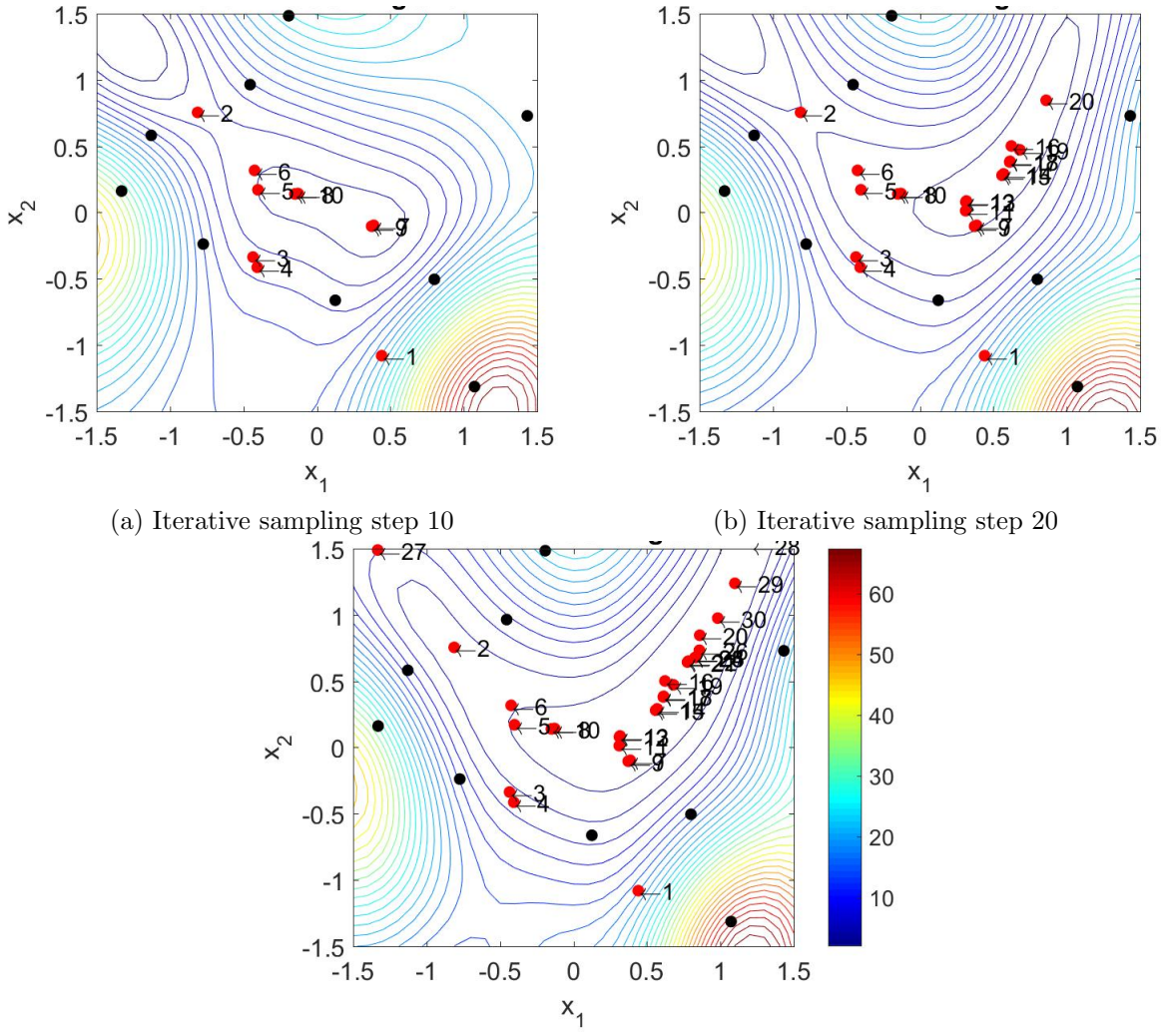
(c) GE-SVR

Figure 4.3: Response surfaces of Rosenbrock function with 10 sample points.

to zero and located at (x_1, x_2)=(1,1). The proxy-based optimization scheme with iterative sampling, which is summarized in Fig. 4.1 is applied to find the minimum of the Rosenbrock function where the first LS-SVR and GE-SVR proxy model are generated using the 10 training points shown as black dots in Figs. 4.4 and 4.5, respectively. These input training points were generated with LHS. In Figs. 4.4 and 4.5 and similar figures for the second 2D function considered, red dots denote the locations of estimated minimum points. Each red dot is labeled with a number which denotes the number of times gradient-based optimization has been applied to obtain the estimate of the minimum, i.e., the label j denotes the minimum obtained at the jth iterative sampling step by by applying SQP to minimize the proxy trained with the jth training set within the iterative sampling procedure. For this example, we set $\sigma_{rs} = 0.2$ in step 5 of the iterative sampling algorithm. The convergence criterion for the SQP algorithm is $\epsilon_J = 10^{-3}$ and the iterative sampling algorithm is terminated when $\epsilon_p = 10^{-2}$ (difference between proxy and true model optimum values). No constraints is considered in this example, therefore, the criteria involving the convergence tolerances ϵ_θ and ϵ_c do not apply for this example. Table 4.1 summarizes the proxy-based optimization results for the Rosenbrock function which has a minimum value of zero at $(1, 1)$. From the results in the table, we can observe that GE-SVR only takes 6 iterative sampling steps to reach the minimum while LS-SVR takes 30 iterative sampling steps. Figs. 4.4 and 4.5 also illustrate the change of the shapes in response surfaces which are approximated by LS-SVR and GE-SVR, respectively, in the iterative sampling process. As the algorithm is designed to do, we see that as iterative sampling proceeds, we generate successive proxies that better approximate the Rosenbrock function in the neighborhood of the true minimum.

Proxy model	Optimal value	Optimal x	Num. resampling steps
LS-SVR	1.63×10^{-3}	(0.982 0.976)	30
GE-SVR	1.14×10^{-4}	(0.99 0.98)	6

Table 4.1: Summary of the results for proxy-based optimization with iterative sampling on Rosenbrock function using 10 initial training samples

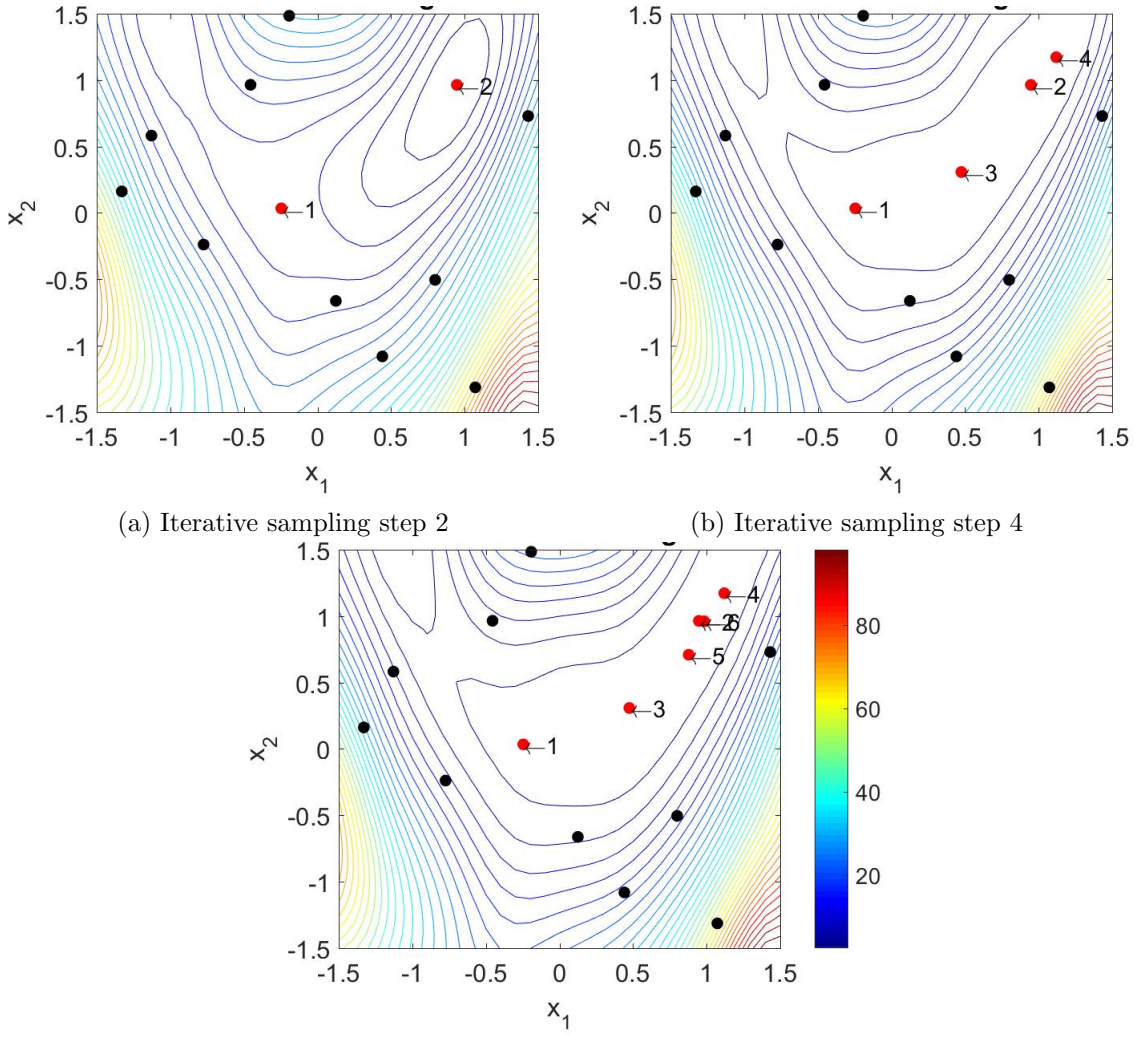


(a) Iterative sampling step 10

(b) Iterative sampling step 20

(c) Iterative sampling step 30

Figure 4.4: Response surface of Rosenbrock function with iterative sampling scheme approximated by LS-SVR.

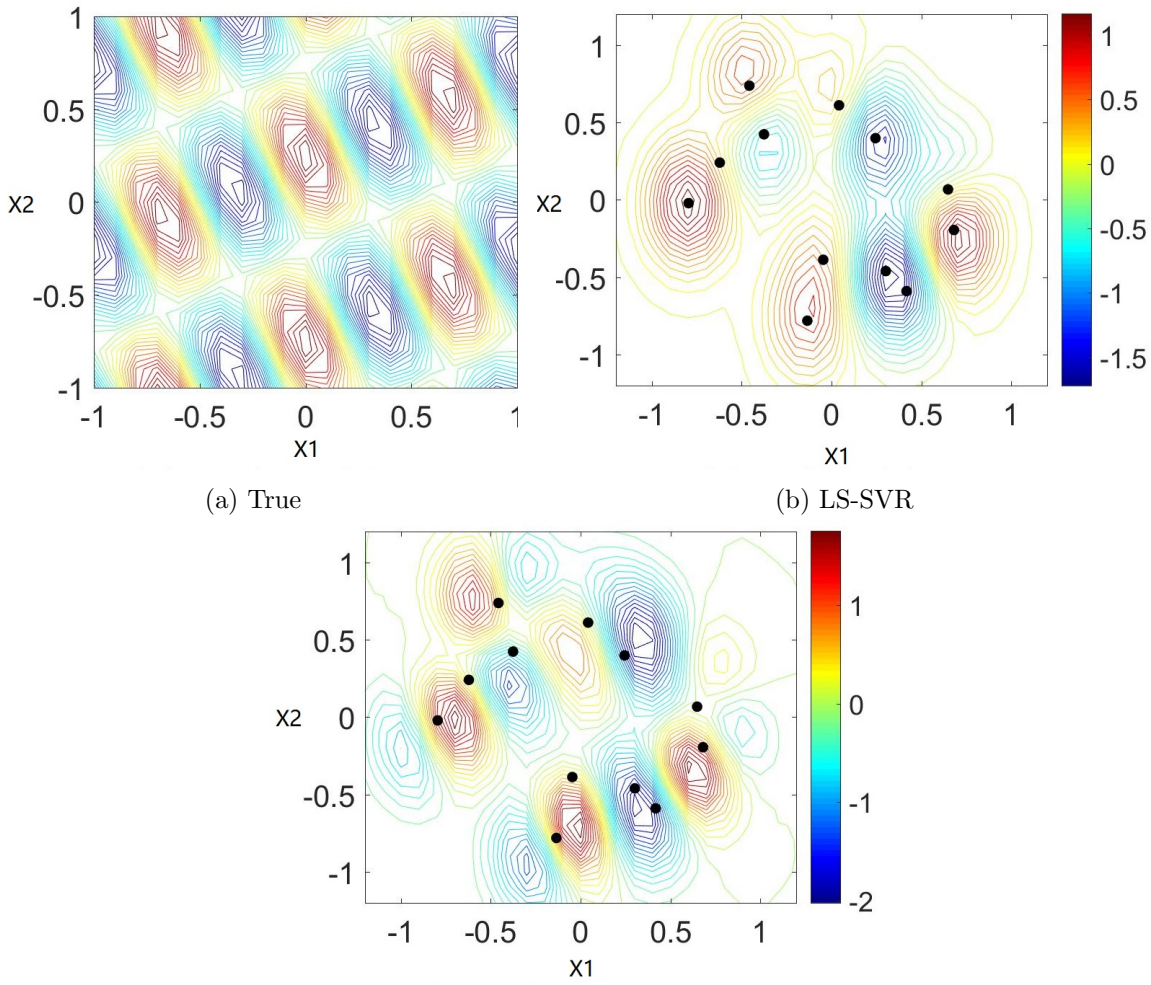


(a) Iterative sampling step 2

(b) Iterative sampling step 4

(c) Iterative sampling step 6

Figure 4.5: Response surface of Rosenbrock function with iterative sampling scheme approximated by GE-SVR.



(a) True

(b) LS-SVR

(c) GE-SVR

Figure 4.6: Response surfaces of multi-minimum function with 12 sample points.

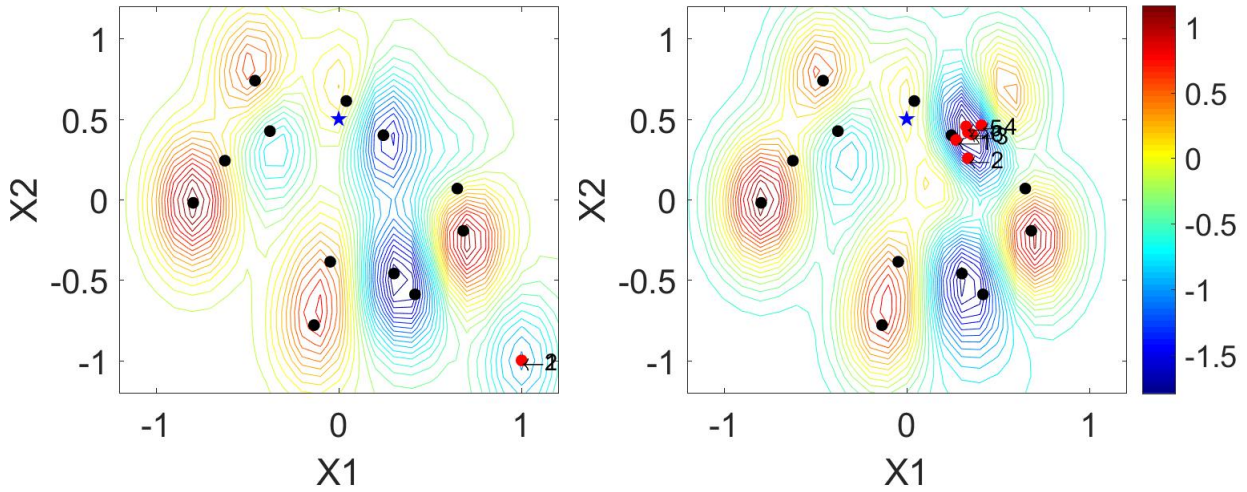
Location of local minimums	Corresponding function value
(-1, 0.75)	-2
(-1, -0.25)	-2
(-0.34, 1)	-1.84
(-0.34, 0.09)	-2
(-0.34, -0.91)	-2
(0.34, 0.42)	-2
(0.33, -0.58)	-2
(1, 0.75)	-2
(1, -0.25)	-2

Table 4.2: Summary of the 9 local minimums of the multi-minimum function within [-1, 1] × [-1, 1]

Then, the LS-SVR and GE-SVR are tested on a 2D function whose response surface have multiple local minimums. The equation of this multi-minimum function is defined as

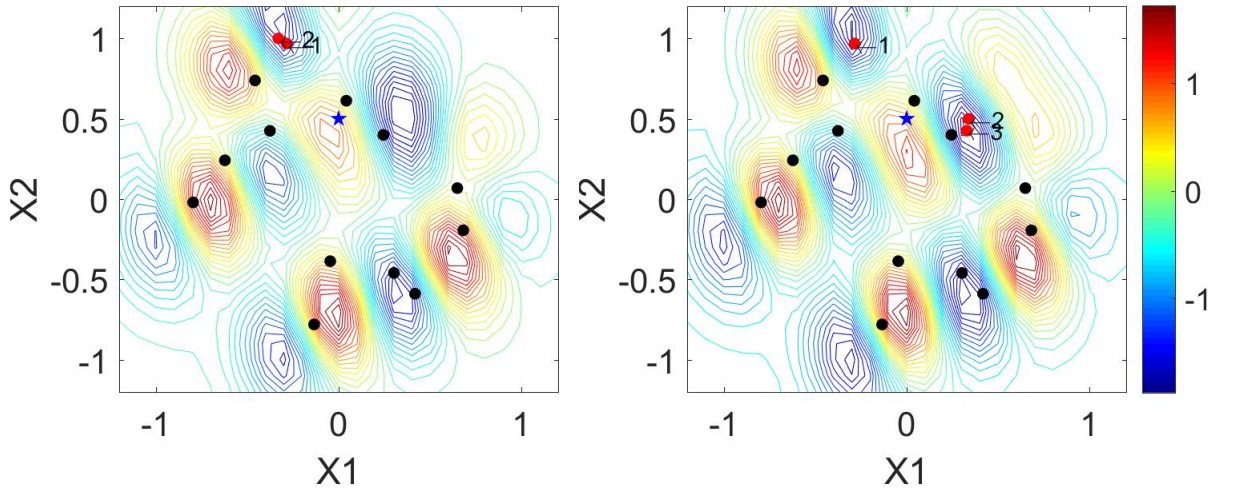
$$f(x_1, x_2) = \sin(2\pi(x_1 + x_2)) + \cos(3\pi x_1). \tag{4.42}$$

Note this function has multiple local minimums within the feasible region which is specified by $-1 \leq x_1 \leq 1$ and $-1 \leq x_2 \leq 1$, where the locations of the local minimums are given in Table 4.2. Fig. 4.6 compares the output values of the LS-SVR and GE-SVR proxy models generated with 12 input samples of (x_1, x_2) generated with LHS. For generating the LSV-proxies, we use $\sigma = 10$ and $C_0 = 1000$ and in generating the GE-SVR proxies, we use $\sigma = 10$, $C_0 = 1000$ and $C_r = 200$ for all r; see Eqs. 4.5 - 4.32. From the results of Fig. 4.6, we can observe that the true function has 9 local minimums but the LS-SVR model only identifies three of them with inaccurate locations. However, the GE-SVR proxy shows 7 of the local minimums and also provides a good approximation of the locations of these seven local minimums on the response surface. This improved accuracy of the GE-SVR model over LS-SVR should be useful in solving an optimization problem with multiple solutions and we investigation this aspect next. To do so, we run the iterative sampling algorithm nine times starting with nine initial guesses uniformly spaced on the domain $[-1, 1] \times [-1, 1]$. In this experiment, the results obtained with $\sigma_{rs} = 0$ in step 5 of the iterative sampling algorithm are compared with those obtained with $\sigma_{rs} = 0.1$



(a) LS-SVR as proxy model without perturbation scheme

(b) LS-SVR as proxy model with perturbation scheme $\sigma_{rs} = 0.1$



(c) GE-SVR as proxy model without perturbation scheme

(d) GE-SVR as proxy model with perturbation scheme $\sigma_{rs} = 0.1$

Figure 4.7: Iterative sampling process starting at the point (0, 0.5) with LS-SVR and GE-SVR as the proxy models and with and without the perturbation scheme.

Initial Guess $(x_{\text{ini},1},\, x_{\text{ini},2};\, y_t)$	LS-SVR Optimums $(x_{\text{opt},1},\, x_{\text{opt},2};\, y_{\text{proxy}};\, y_t)$	Closest Local Minimums $(x_{\text{min},1},\, x_{\text{min},2};\, y_t)$	Number of Iterative Sampling	Number of Equivalent Func. Eval.
(-0.50,-0.50; 0.00)	(-1.00, -1.00; -1.00 ; -1.00)	(-0.34, -0.91; -2.00)	2	14
(-0.50,0.00; 0.00)	(-1.00, 1.00; -1.00; -1.00)	(-1.00, 0.75; -2.00)	2	14
(-0.50,0.50; 0.00)	(1.00, -1.00; -1.00; -1.00)	(1.00, -0.25; -2.00)	2	14
(0.00,-0.50; 1.00)	(1.00, -1.00; -1.00; -1.00)	(1.00, -0.25; -2.00)	1	13
(0.00,0.00; 1.00)	(-1.00, 1.00; -1.00; -1.00)	(-1.00, 0.75; -2.00)	2	14
(0.00,0.50; 1.00)	(1.00, -1.00; -1.00; -1.00)	(1.00, -0.25; -2.00)	2	14
(0.50,-0.50; 0.00)	(0.33, -0.58; -2.00; -2.00)	(0.33, -0.58; -2.00)	5	18
(0.50,0.00; 0.00)	(-1.00, 1.00; -1.00; -1.00)	(-1.00, 0.75; -2.00)	2	14
(0.50,0.50; 0.00)	(0.34, 0.42; -2.00 ; -2.00)	(0.34, 0.42; -2.00)	6	19

Table 4.3: Nine initial guesses, estimate of optimal point for each initial guess, and the local minimum of the multi-minimum function that is closest to each optimal point generated by each application of iterative sampling with no perturbation and LS-SVR as the proxy model.

In the LS-SVR iterative sampling scheme used by Guo and Reynolds for life-cycle production [37] and by Sousa and Reynolds for history matching and prediction problems with uncertainty quantification [25], the optimum ($x^{*,n_\ell-1}$) generated from the proxy-based SQP optimization at the last iterative sampling step as the initial guess ($x^{0,n_\ell-1}$) of the proxy-based SQP optimization is used as the initial guess for the SQP optimization with the updated proxy at the current iterative sampling step. This means that, at the n_ℓth iterative sampling step, the initial guess in step 5 of the iterative sampling algorithm is $x^{0,n_\ell}=x^{*,n_\ell-1}$. However, as discussed extensively in the comment following the presentation of the detailed steps of the iterative sampling algorithm of subsection 4.3.3, we include the possibility of using a perturbation of this initial guess in the iterative sampling process. Specifically, one may use $x^{0,n_\ell}=x^{*,n_\ell-1}+\delta$ in step 5 of the iterative sampling algorithm, where δ is sampled from the normal distribution $N(0,\sigma_{rs}^2)$. For simplicity, we refer to σ_{rs} as the perturbation size, rather than the standard deviation of the perturbation. Our motivation for introducing this random perturbation was discussed in the paragraph following the presentation of the

Initial Guess $(x_{\text{ini},1},\, x_{\text{ini},2};\, y_t)$	LS-SVR Optimums $(x_{\text{opt},1},\, x_{\text{opt},2};\, y_{\text{proxy}};\, y_t)$	Closest Local Minimums $(x_{\text{min},1},\, x_{\text{min},2};\, y_t)$	Number of Iterative Sampling	Number of Equivalent Func. Eval.
(-0.50,-0.50; 0.00)	(-1.00, -1.00; -1.00; -1.00)	(-0.34, -0.91; -2.00)	2	14
(-0.50,0.00; 0.00)	(0.34, 0.42; -2.00; -2.00)	(0.34, 0.42; -2.00)	6	18
(-0.50,0.50; 0.00)	(0.33, -0.58; -2.00; -2.00)	(0.33, -0.58; -2.00)	6	18
(0.00,-0.50; 1.00)	(0.34, 0.42; -2.00; -2.00)	(0.34, 0.42; -2.00)	6	18
(0.00,0.00; 1.00)	(0.33, -0.58; -2.00; -2.00)	(0.33, -0.58; -2.00)	6	18
(0.00,0.50; 1.00)	(0.34, 0.42; -2.00; -2.00)	(0.34, 0.42; -2.00)	6	18
(0.50,-0.50; 0.00)	(-1.00, -1.00; -1.00; -1.00)	(-0.34, -0.91; -2.00)	2	14
(0.50,0.00; 0.00)	(1.00, 1.00; -1.00; -1.00)	(1, 0.75; -2.00)	11	23
(0.50,0.50; 0.00)	(1.00, -1.00; -1.00; -1.00)	(1, -0.25; -2.00)	6	18

Table 4.4: Nine initial guesses, estimate of optimal point for each initial guess, and the local minimum of the multi-minimum function that is closest to each optimal point generated by each application of iterative sampling with perturbation size of $\sigma_{rs} = 0.1$ and LS-SVR as the proxy model.

iterative sampling algorithm.

Next, we test the iterative sampling scheme on the multi-minimum function of Eq. 4.42 using the LS-SVR proxy and the GE-SVR proxy with and without the perturbation scheme, i.e., with δ fixed equal to zero in step 5 in the algorithm and with δ independently sampled from $N(0, 0.1^2)$ every time we enter step 5 of the algorithm so that δ may have a different value at each iterative sampling step. We apply the complete iterative sampling algorithm nine times starting with nine initial guesses uniformly distributed on $[-1, 1] \times [-1, 1]$. There are two objective of the exercise, namely, to compare the relative ability of LS-SVR and GE-SVR to locate several of the local minimums and to compare the effect of the size of the perturbation size, σ_{rs}, on proxy-based optimization with the iterative sampling scheme. The results of iterative samplings with the LS-SVR proxy model with no perturbation ($\delta = 0$) and with a 0.1 perturbation ($\sigma_{rs} = 0.1$), respectively, are summarized in Table. 4.3 and Table. 4.4, respectively. Similarly, the results generated with nine separate applications of the iterative sampling algorithm with the GE-SVR proxy with $\delta = 0$ and with $\sigma_{rs} = 0.1$, respectively,

Initial Guess $(x_{\text{ini},1},\ x_{\text{ini},2};\ y_t)$	GE-SVR Optimums $(x_{\text{opt},1},\ x_{\text{opt},2};\ y_{\text{proxy}};\ y_t)$	Closest Local Minimums $(x_{\text{min},1},\ x_{\text{min},2};\ y_t)$	Number of Iterative Sampling	Number of Equivalent Func. Eval.
(-0.50,-0.50; -0.59)	(-0.34, -0.91; -2.00; -2.00)	(-0.34, -0.91; -2.00)	2	28
(-0.50,0.00; 0.00)	(-0.34, 0.09; -2.00; -2.00)	(-0.34, 0.09; -2.00)	2	28
(-0.50, 0.50; 0.00)	(-0.34, 0.09; -2.00; -2.00)	(-0.34, 0.09; -2.00)	1	26
(0.00, -0.50; 1.00)	(-0.34, 0.09; -2.00; -2.00)	(-0.34, 0.09; -2.00)	1	26
(0.00, 0.00; 1.00)	(-0.34, 0.09; -2.00; -2.00)	(-0.34, 0.09; -2.00)	1	26
(0.00, 0.50; 1.00)	(-0.34, 1.00; -1.84; -1.84)	(-0.34, 1.00 ; -1.84)	2	28
(0.50 , -0.50; 0.00)	(0.33, -0.58; -2.00; -2.00)	(0.33, -0.58; -2.00)	2	28
(0.50, 0.00; 0.00)	(0.34, 0.42; -2.00; -2.00)	(0.34, 0.42; -2.00)	2	28
(0.50, 0.50; 0.00)	(0.34, 0.42; -2.00; -2.00)	(0.34, 0.42; -2.00)	1	28

Table 4.5: Nine initial guesses, estimate of optimal point for each initial guess, and the local minimum of the multi-minimum function that is closest to each optimal point generated by each application of iterative sampling with no perturbation and GE-SVR as the proxy model.

are summarized in Table 4.5 and Table. 4.6, respectively. The entries in the first column of Tables 4.3 through 4.6 are identical and represent the nine initial guesses for (x_1, x_2) together with the the value of y_t, which in every table represents the value of the true multi-minimum evaluated at the (x_1, x_2) listed with the y_t value in the table. Thus, the third entry in a three-dimensional vector of values $(\cdot,\ \ \cdot\ \ ;\ \ \cdot)$ that appear in columns 1 and 3 of the tables represents the value of the multi-minimum function evaluated at the (x_1, x_2) values given as the first two components of the vector. For the case of the four-dimensional vectors of values that appear in column 2 of Tables 4.3 through 4.6, the value of the multi-minimum function is the last entry of the vector and the value of the proxy model is the third entry where both the proxy and the multi-minimum function are evaluated at the (x_1, x_2) values given as the first two entries. In the second column, the notation $(x_{\text{opt},1},\ x_{\text{opt},2})$ and the associated values in the tables are the estimates of the optimal (x_1, x_2) obtained at convergence of the iterative sampling algorithm from each corresponding initial guess in column 1. In Column 3 of the four tables under discussion, the third column gives values of $(x_{\text{min},1}, x_{\text{min},2};\ y_t)$ where

Initial Guess $(x_{\text{ini},1},\, x_{\text{ini},2};\, y_t)$	GE-SVR Optimums $(x_{\text{opt},1},\, x_{\text{opt},2};\, y_{\text{proxy}};\, y_t)$	Closest Local Minimums $(x_{\text{min},1},\, x_{\text{min},2};\, y_t)$	Number of Iterative Sampling	Number of Equivalent Func. Eval.
(-0.50,-0.50; -0.59)	(-0.34, -0.91; -2.00; -2.00)	(-0.34, -0.91; -2.00)	2	28
(-0.50,0.00; 0.00)	(-0.34, 0.09; -2.00; -2.00)	(-0.34, 0.09; -2.00)	2	28
(-0.50, 0.50; 0.00)	(-0.34, 0.09; -2.00; -2.00)	(-0.34, 0.09; -2.00)	1	26
(0.00, -0.50; 1.00)	(-0.34, 0.09; -2.00; -2.00)	(-0.34, 0.09; -2.00)	1	26
(0.00, 0.00; 1.00)	(-0.34, 0.09; -2.00; -2.00)	(-0.34, 0.09; -2.00)	1	26
(0.00, 0.50; 1.00)	(0.34, 0.42; -2.00; -2.00)	(0.34, 0.42; -2.00)	3	30
(0.50 , -0.50; 0.00)	(0.33, -0.58; -2.00; -2.00)	(0.33, -0.58; -2.00)	2	28
(0.50, 0.00; 0.00)	(0.34, 0.42; -2.00; -2.00)	(0.34, 0.42; -2.00)	2	28
(0.50, 0.50; 0.00)	(0.34, 0.42; -2.00; -2.00)	(0.34, 0.42; -2.00)	1	28

Table 4.6: Nine initial guesses, estimate of optimal point for each initial guess, and the local minimum of the multi-minimum function that is closest to each optimal point generated by each application of iterative sampling with perturbation size of $\sigma_{rs} = 0.1$ and GE-SVR as the proxy model.

$(x_{\text{min},1},\, x_{\text{min},2})$ is the local minimum of the true multi-minimum function which is closest to the corresponding $(x_{\text{opt},1},\, x_{\text{opt},2})$ optimal point given in column 2 and y_t denotes the value of the multi-minimum function evaluated at $(x_{\text{min},1},\, x_{\text{min},2})$. The values in column 4 of the tables give the number of iterative sampling steps required to obtain convergence for each of the initial guesses and the entries of the last column give the number of equivalent function evaluations. For the GE-SVR results, each gradient evaluation is counted as one equivalent function evaluation. Therefore, generating the one training set for the GE-SVR proxy which consists of function evaluation and gradient evaluation is equivalent to generating two training sets for the LS-SVR.

From the results of Table 4.3, we observe that most estimated optimums generated using the LS-SVR proxies in iterative sampling with no perturbation ($\delta = 0$ in step 5 of the iterative sampling algorithm) are trapped at one of the corners of the domain, i.e. at $(x_{\text{opt},1}, x_{\text{opt},2})$ equal to (-1, -1), (-1, 1), (1, -1) or (1, 1). Although the response surface of the multi-minimum function shown in Fig. 4.6a does have a minimum close to each of the

	GE-SVR, σ_{rs} =0.1 perturbation	GE-SVR no perturbation	LS-SVR, σ_{rs} =0.1 perturbation	LS-SVR no perturbation
Ave. Iter. Samp. Steps	1.66	1.56	5.67	2.67
Ave. Equiv. Func. Eval.	27.33	27.12	17.67	14.67
No. true min. found	4	5	2	2

Table 4.7: Summary the computational efficiency of the iterative sampling algorithm on the multi-minimum function with GE-SVR/LS-SVR as the proxy model with/without the perturbation scheme.

corners, none of these minimum are exactly at a corner of the domain, and no minimum close to one of the corners gives a value of the multi-minimum function that is close to its global minimum which is equal to -2. As shown in Table 4.3, when no perturbation of the initial guess in step 5 of the iterative sampling algorithm is used, the nine distinct applications of iterative sampling algorithm locate only two actual local minimum in the interior of the domain with a minimum function value equal to -2. From the results of Tables 4.4, we observe that using the perturbation scheme in step 5 of the iterative sampling algorithm with $\sigma_{rs} = 0.1$ and LS-SVR as the proxy model, we are still able to locate only two of the interior local minimums, even though the algorithm gets trapped at a corner of the domain less frequently than for the case where no perturbation is applied; see Table 4.3. However, when perturbations are used, it generally takes a larger number of iterative sampling steps to reach convergence than are required when no perturbations are done.

Now we turn to the iterative sampling results based on the utilization of GE-SVR proxies. From Table 4.5, we observe that GE-SVR-based iterative sampling without any perturbation ($\delta = 0$ in step 5 of the algorithm) correctly locates 5 of the interior local minimums out of the 9 that exist by starting from the 9 different evenly distributed initial guesses. The locations of nine different interior local minimums of the multi-minimum function and corresponding function values are summarized in Table.4.2. Moreover, convergence is achieved with at most two iterative sampling steps. By comparing the results of Tables 4.5 and 4.6, we can observe that the perturbation scheme locates 4 of the same interior local minimums

that were found using no perturbation of the initial guess in step 5 of the iterative sampling algorithm. However, by comparing the 6th rows of both Tables and Fig. 4.7 which pertain to the initial guess $(x_1, x_2) = (0.00, 0.50)$, we can see that the no perturbation scheme can find the local minimum located at $(x_1, x_2) = (-0.34, 1.00)$ which corresponds to ($x_{\text{opt},1}$,$x_{\text{opt},2}$; y_{proxy}; y_t) = (-0.34, 1.00; -1.84; -1.84), whereas when a perturbation is used, this fifth local minimum is never located, but instead, for the same initial guess, converges to a local optimum of $(x_{\text{opt},1}, x_{\text{opt},2}) = (0.34, 0.41)$ where the global optimum (-2) of the multi-minimum function is attained at the cost of one additional iterative sampling step. Whether using perturbations in step 5 of the iterative sampling scheme with GE-SVR proxies is actually better suited than using no perturbations to find global optima, whereas no perturbations are better suited to find local optimal is unknown. Resolving this question would require an exhaustive study using many optimization test functions and is beyond the scope of this work. However, for the multi-minimum function considered here, if the objective is to find as many local minimum as possible when using GE-SVR proxies, it appears to be best to use no perturbations. Most importantly, when using GE-SVR proxies in the iterative samples algorithm, more local minimum are found that when using LS-SVR proxies.

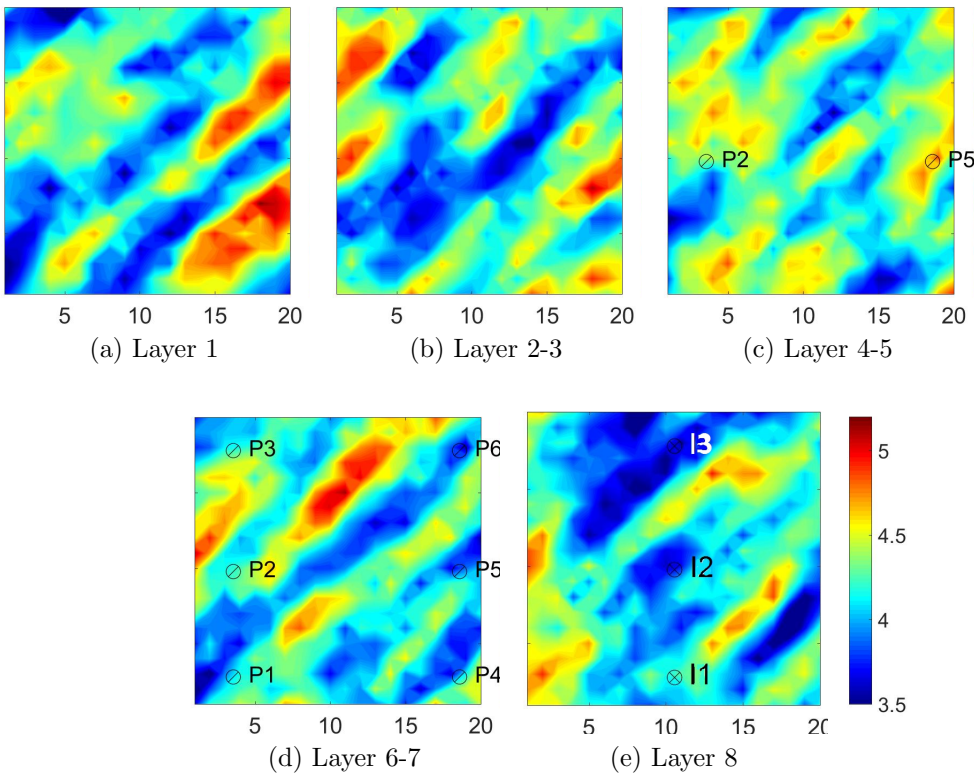
Table 4.7 summarizes the computational efficiency of the iterative sampling algorithm on the multi-minimum function with GE-SVR/LS-SVR as the proxy model with/without the perturbation scheme. The results of Table 4.7 indicate that incorporating GE-SVR proxies instead of LS-SVR proxies increases the number of equivalent function runts required to reach convergence but enables the location of more local minimums than can be found when LS-SVR proxies are used.

4.3.2 Example 2

In this example, proxy-based optimization with the GE-SVR model is applied to a synthetic 3D three-phase flow reservoir model. The reservoir model has $20 \times 20 \times 8 = 3200$ grid blocks with 8 simulation model layers. The geological properties (permeability and porosity) of the 2nd, 4th and 6th simulation layers, respectively, are same as the 3rd, 5th

and 7th simulation layers, respectively. Therefore, there are only five geological layers. The horizontal log-permeability distributions of the five geological layers are shown in Fig. 4.8. A gas cap is initially present in the 1st geological layer and an aquifer is present in the 5th geological layer. The initial pressure of the reservoir is 4,400 psi. The irreducible water saturation and residual oil saturation are both equal to 0.2. The locations of the nine restricted-entry vertical wells are also shown in Fig. 4.8. Six wells are producing well and three wells are water injection wells. The three water injection wells are completed only in the bottom layer (the aquifer). Producers P2 and P5 are completed in geologic layers 3 and 4 (simulator layers, 4-7). The other four producers are completed only in the fourth geologic layer. For this problem, only one reservoir model is used, i.e., we apply "deterministic" rather than robust optimization.

The length of each control step is 180 days and the reservoir life is 22$\times$ 180=3960 days. Thus, there are 22 control steps for each well and the total number of control variables to be optimized is $9 \times 22 = 198$. For producers, the control variables are their bottom hole pressure (BHP) and the controls of injectors are their water injection rates. The only constraints considered in this problem are bound constraints as shown in Table 4.8. Note that in the statement of our general problem (see Eq. 2.3), bound constraints Eq. 2.3b are stated individually but, in actuality, they are converted to inequality constraints of the form of Eq. 2.3c by converting Eq. 2.3b to $u_i - u_i^{\text{low}} \geq 0$ and $u_i^{\text{up}} - u_i \geq 0$ for $i = 1, 2, \cdots N_u$. Then, they are treated in the same way as nonlinear state inequality constraints in the SQP-filter optimization algorithm. Also note that in Table 4.8, constraints labeled with $\geq$ type inequality refer to an lower bound whereas constraints of the form $\leq$ refer to a upper bound, e.g., the upper bound on producers is 4,400 psi, i.e., we require $p_{\text{wf}} \leq 4400$ or equivalently $4.4 - (p_{\text{wf}}/1000) \geq 0$, where 1,000 psi is the scaling factor. According to Table 4.8, there are 2 linear constraints per well, i.e., 18 linear constraints and these must be applied at each of the 22 control steps, i.e., there are 396 constraints to be applied. For the NPV computations, the oil price is set equal to \$50/STB, the water disposal cost is \$3/STB, and the water injection cost is set equal to \$2/STB. The gas price is \$0.2/Mcf and the annual discount rate is 0.05.



(a) Layer 1 (b) Layer 2-3 (c) Layer 4-5

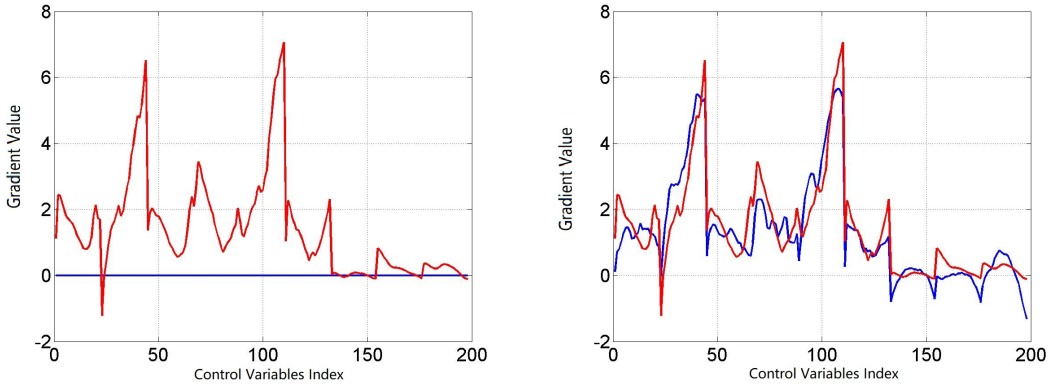
(d) Layer 6-7 (e) Layer 8

Figure 4.8: The Log permeability of each geological layer.

All producers initially operate at 3,500 psi at all control steps and the initial water injection rate for all injectors at all control steps is equal to 100 STB/day. To validate the performance of the proxy-based optimization framework and compare the performance of LS-SVR and GE-SVR proxy models, we consider production optimization with no iterative sampling scheme. However, this particular example is redone with iterative resampling in Example 3, where state constraints are also added.

Well	Cons. Type	Cons. Value	Cons. Scale	Type of inequality
P1-6	BHP (psi)	3000	1000	$\geq$
P1-6	BHP (psi)	4400	1000	$\leq$
I1,I2,I3	rate (STB/d)	0	10	$\geq$
I1,I2,I3	rate (STB/d)	2000	10	$\leq$

Table 4.8: The summary of bound constraints.



(a) $C_0 = 1,000$, $C_r = 200$ and $\sigma = 0.2$

(b) $C_0 = 1,000$, $C_r = 200$ and $\sigma = 2$

(c) $C_0 = 1,000$, $C_r = 200$ and $\sigma = 10$

(d) $C_0 = 100,000$, $C_r = 50,000$ and $\sigma = 10$

Figure 4.9: Comparison of Adjoint gradient (red) with proxy predicted gradient (blue) at the center of control space with different hyperparameters

The hyperparameters C_0 and C_r, $r = 1, 2, \cdots N_u$, in Eq. 4.5, and σ in Eq. 4.29 must be chosen appropriately to build an accurate proxy model for proxy-based optimization. To investigate the effect of hyperparameters on the predictability of the proxy model, for different values of these parameters, we compare the adjoint gradient of average life-cycle NPV based on a single realization, with the average life-cycle NPV computed from a proxy where both gradients are evaluated at the center of control space, i.e., every entry of x, the vector scaled control variables, is evaluated at 0.5. There are 198 control variables and in Fig. 4.9, we plot the derivatives of NPV with respect to each of the 198 control variables. The adjoint gradients are plotted as red lines and proxy estimated derivatives are shown as blue lines. From Figs. 4.9a, 4.9b and 4.9c, we see that the differences between the adjoint gradient and gradient of the proxy tend to decrease as the value of σ increases. From Figs. 4.9c and 4.9d, we can observe that changes in the values of C_0 and C_r do not have a significant effect on the accuracy of the derivatives calculated from the GE-SVR proxy. This is the accordance with results on LS-SVR presented by Guo and Reynolds [37] and Sousa and Reynolds [25] who found that, for LS-SVR, the value of C_0 does not have much affect on the results as long as $C_0 > 200$. Note from the results of Fig. 4.9, reasonable agreement between the adjoint derivatives and the derivatives of the GE-SVR proxy is obtained for the case where $C_0 = 1,000$, $C_r = 200$ and $\sigma = 2$, and excellent agreement is obtained between the two sets of derivative values for the case where $C_0 = 1,000$, $C_r = 200$ and $\sigma = 10$; see Fig. 4.9c. To further investigate the effect of the value of σ on the performance of the proxy model, we conduct sensitivity analysis on the effect of the value of the hyperparameter σ on the accuracy of the GE-SVR model as a proxy to replace the reservoir simulator, where the error in the proxy as an approximation is measured by the value of cross validation error. The total cross validation error is defined by the K-fold cross-validation on the training data set, where the training data set is randomly partitioned into K equally-sized subsets. Of the K subsets, a single subset is retained as the validation data for testing the model, and the remaining K-1 subsets are used as training data. The cross-validation process is repeated K times (the folds), with each of the K subsets used exactly once as the validation data

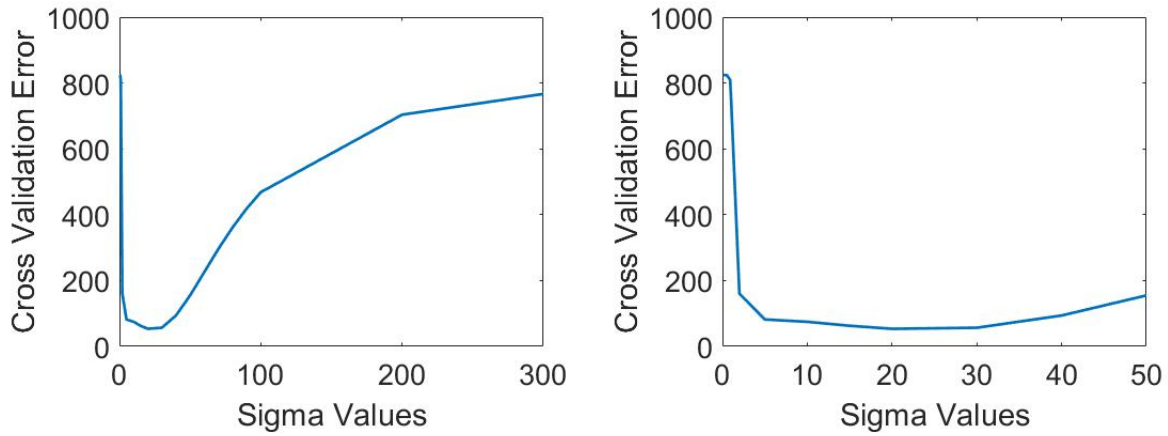
and the total error computed from the K-sets of validation errors is the objective function. For subset k as the prediction set and the complementary set as the training set, the error function for kth subset is defined as

$$E_k(\sigma) = \sum_{i=1}^{N_{s,k}} (\bar{J}_{\text{sim}}(x_k^i) - \bar{J}_{\text{proxy}}(x_k^i))^2 + ||\nabla \bar{J}_{\text{sim}}(x_k^i) - \nabla \bar{J}_{\text{proxy}}(x_k^i)||_2 \tag{4.43}$$

where $N_{s,k}$ is the number of samples in subset k; x_k^i, $i = 1, 2, \cdots N_{s,k}$ denote the scaled control vectors that are training input in the subset k. The cross-validation error as a function of the hyperparameter σ is then defined as

$$CV(\sigma) = \sum_{k=1}^{K} E_k(\sigma). \tag{4.44}$$

In this example, our initial training set for GE-SVR consists of only five input/output samples, so we perform five-fold cross validation ($K = 5$) where each subset has only one sample in it ($N_{s,k}$=1). Fig. 4.10 plots the cross validation error of GE-SVR proxy versus corresponding σ values where the values of the other hyperparameters are fixed at the following values; $C_0 = 1,000$ and $C_r = 200$. From the results of Fig. 4.10a and 4.10b, we can see that the value of the cross validation error decreases rapidly as the value of sigma increases from 0.1 to 5 and when the value of σ is larger than 30, the cross validation error starts to increase again. In Fig. 4.10b, we get a better view of the behavior of $CV(\sigma)$ by reducing the range to $0 < \sigma < 50$. From this last plot, we we can observe that the cross validation error does not change much when sigma ranges of 5 to 30. Therefore, we set $C_0 = 1,000$, $C_r = 200$ and $\sigma = 10$ as the fixed values of the hyperparameters for the GE-SVR proxy in all of the following examples. Similarly, We apply the cross validation process to find appropriate values of the hyperparameters for the LS-SVR proxy model. Because the LS-SVR proxy is generated with 25 training samples, each of the five subsets for five-fold cross validation has five samples in it ($N_{s,k}$=5). Fig. 4.11 plots the cross validation error of the LS-SVR proxy versus σ values where the value of the hyperparameter C_0 is fixed at $C_0 = 1,000$. From



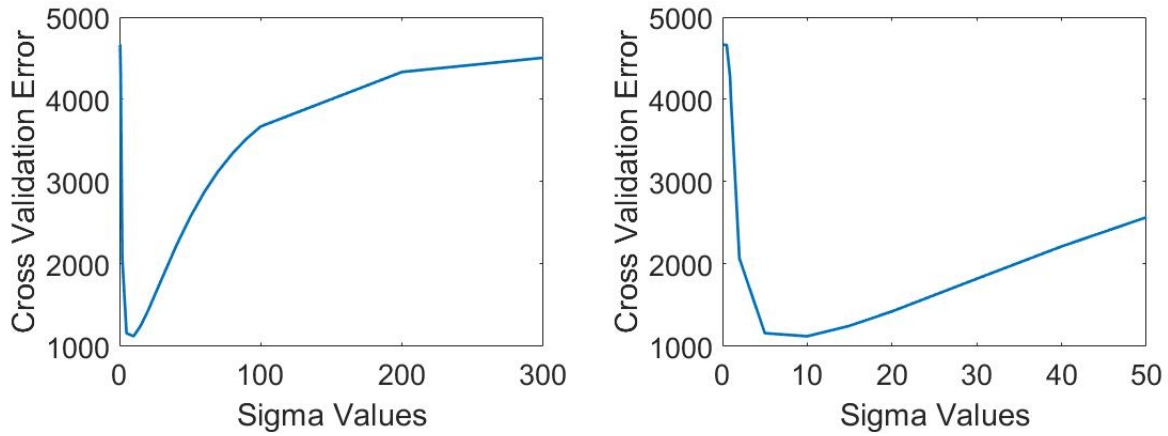
(a) Cross Validation Error (range 0.1-300) (b) Cross Validation Error (range 0.1-50)

Figure 4.10: The change of 5-fold cross validation error of GE-SVR versus different σ.

the results of Fig. 4.11a and 4.11b, we can see that the value of the cross validation error decreases rapidly as the value of sigma increases from 0.1 to 5 and when the value of σ is larger than 20, the cross validation error starts to increase again. From Fig. 4.11b, we we can observe that the cross validation error of LS-SVR proxy does not change much when sigma ranges of 5 to 30 when $C_0 = 1,000$. In the following examples, $\sigma = 10$ sad $C_0 = 1,000$ are used as the values of the hyperparameters when constructing LS-SVR proxy models.

Twenty-five training points are sampled by LHS to generate the training set for the LS-SVR proxy model so to calculate the training output corresponding to these input training samples requires 25 forward reservoir simulation runs. On the other hand, GE-SVR uses a training set consisting of 5 samples and thus requires only 5 forward runs and 5 backward runs (adjoint solutions) to compute NPV and the gradient of NPV with respect to well controls in order to obtain the training sets. Each trained proxy model replaces the reservoir simulator when applying the SQP-filter optimization algorithm. Again, the bound constraints are considered as inequality constraints in the SQP-filter framework; more details about the SQP-filter algorithm can be found in Chapter 2 of this dissertation and were also published by Liu and Reynolds [61].

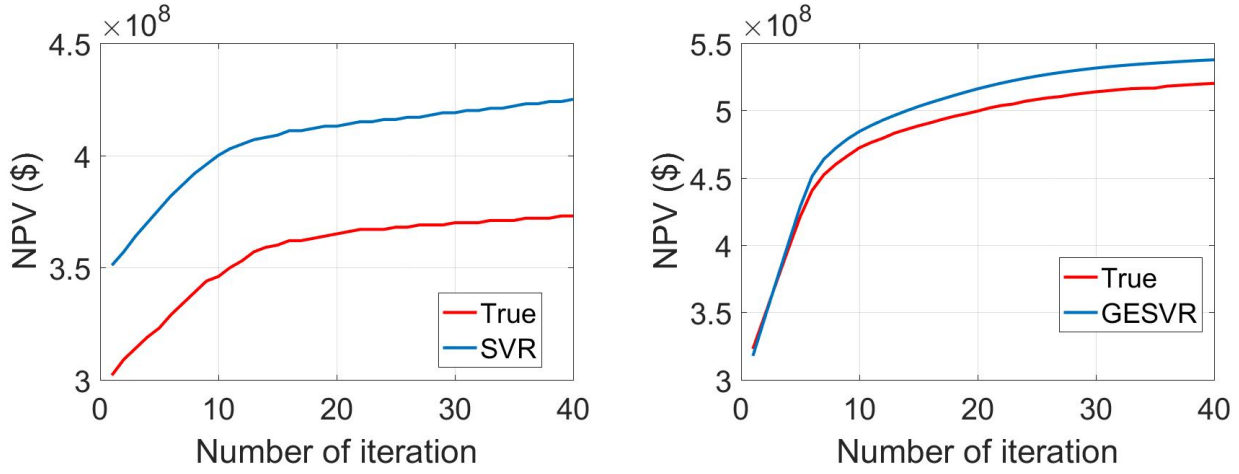
Fig 4.12 compares the life-cycle NPV results obtained at each iteration of gradient-based optimization with the corresponding life-cycle NPV values obtained by computing



(a) Cross Validation Error (range 0.1-300) (b) Cross Validation Error (range 0.1-50)

Figure 4.11: The change of 5-fold cross validation error of LS-SVR versus different σ as the proxy.

the NPV from the numerical model (reservoir simulator) using the controls obtained at each iteration of proxy-based SQP optimization. This is done to see how well the proxy tracks the true model but in practice no reservoir simulation runs during proxy-based SQP optimization. The difference between the red and blue curves implicitly gives an indication of how accurately the proxy predicts the NPV computed from the reservoir simulation during the optimization process. If the difference is large, it indicates the prediction result of the proxy model deviates from the "true" prediction from the reservoir simulator so the proxy does not give a good approximation of the true model at the corresponding control vector obtained at that iteration. By comparing the results of Fig 4.12a and 4.12b, we observe that the GE-SVR proxy provides a much better approximation of the reservoir simulator response during the optimization process than does the LS-SVR model even though the LS-SVR proxy is constructed using five times as many training samples as the GE-SVR proxy. When the number of iterations is limited to 40, the GE-SVR proxy archives a much higher NPV than does the LS-SVR model and more importantly, GE-SVR proxy-based optimization achieves an estimate of life-cycle NPV equal to 5.38×10^8 USD which is close to the one computed from the simulator (5.12×10^8 ISD) by inputting the optimal controls generated with GE-SVR proxy-based optimization into the reservoir simulating and running



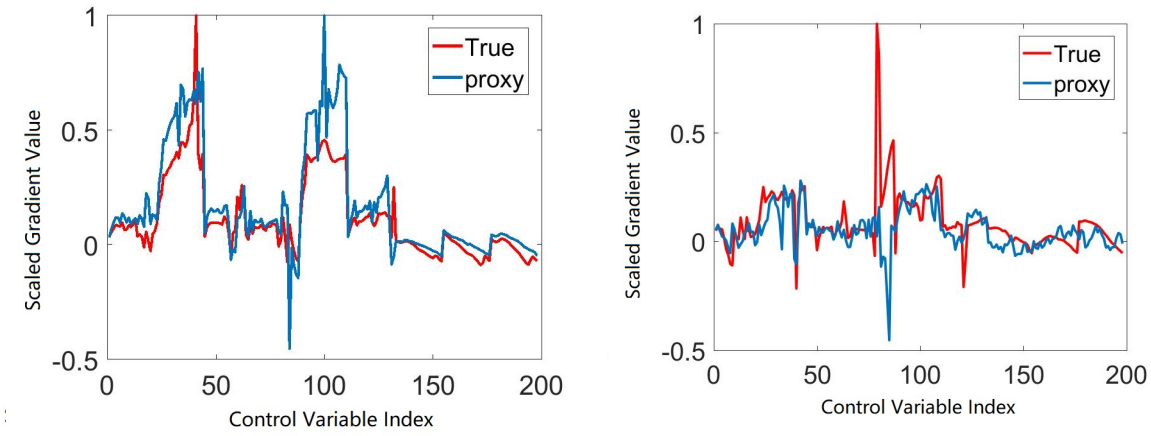
(a) LS-SVR with 25 samples

(b) GE-SVR with 5 samples

Figure 4.12: Life-cycle NPV at each iteration of the gradient-based optimization for proxy-based optimization workflow with LS-SVR and GE-SVR results compared to the value of the true simulator prediction.

it to compute life-cycle NPV. As shown later in Example 3, where we complicate the problem by adding state constraints, the agreement between the GE-SVR proxy predicted gradient of life-cycle NPV and the one predicted by the simulator can be improved by iterative resampling. The computational results are summarized in Table 4.9. With the proxy-based optimization scheme, using GE-SVR as the proxy model, we estimate the optimal NPV is equal to 5.12×10^8 USD after 40 iterations, while with the LS-SVR as the proxy model, we only achieve an NPV equal to 3.73 $\times10^8$ USD. Compared to simulator-based optimization (no proxy used) which takes 21 gradient evaluations and 50 simulation runs, the proxy-based optimization with the GE-SVR only requires 5 gradient evaluations yet using the reservoir simulator in optimization directly with no proxy model yields an NPV which is only 0.01×10^9 USD higher than the one obtained using proxy-based optimization with the GE-SVR proxy as shown in Table 4.9

In Fig. 4.13, the components of the adjoint gradient computed by the high-fidelity reservoir simulator and corresponding derivatives generated by the GE-SVR proxy model are compared at the controls obtained at the first and last iterations of GE-SVR proxy-based optimization. For the purpose of illustration, all the derivatives shown are scaled by the



(a) Comparison of GE-SVR (blue) and adjoint gradients (red) at the 1st iteration

(b) Comparison of GE-SVR (blue) and adjoint gradients (red) at the 40th iteration

Figure 4.13: Comparison approximated scaled gradient of life-cycle NPV by the GE-SVR model at the 1st and 40th iterations for deterministic production optimization with bound constraints.

	Optimal NPV (USD) Evaluated by Proxy	Optimal NPV (USD) Evaluated by Simulator	Number of Simulations
Simulator	-	5.13×10^8	50 runs +21 grad. evaluation
LS-SVR	4.27×10^8	3.73×10^8	25 runs
GE-SVR	5.38×10^8	5.12×10^8	5 runs +5 grad. evaluations

Table 4.9: Comparison of bound constrained optimization using reservoir simulator-based optimization with LS-SVR and GE-SVR proxy-based optimization without iterative sampling scheme.

infinity norm of the absolute values of the adjoint gradient so all scaled derivatives have values between -1 and 1. Recall there are 198 control variables and in Fig. 4.13, we plot the derivatives of life-cycle NPV with respect to each of the 198 control variables. Note there is a fairly good agreement between the derivatives of the GE-SVR proxy model and the derivatives computed from the adjoint solution provided by our reservoir simulator. However, even at the last iteration of the optimization, for some derivatives, the derivative values generated from the proxy and simulator are in significant disagreement. Although not shown here, iterative sampling improves the agreement between the derivatives generated

from the two models as we will see in the next example.

4.3.3 Example 3

In reality, field production facilities have a limited capacity and thus impose limitations on phase rates which result in nonlinear state constraints. In this example, we solve the life-cycle production optimization problem with nonlinear state constraints with our proxy-based optimization framework. We first run a deterministic production optimization with state constraints in which only one geological realization is considered and compare the results with simulator-based life-cycle production optimization. Then we run a robust production optimization problem with state constraints to illustrate the capability of proxy-based optimization with the GE-SVR model as the proxy. All settings are exactly same as in example 2 except that the maximum field gas production rate is set as as 25,000 Mcf/day, the maximum field liquid production rate is set as 6,000 STB/day and the maximum field water production rate is set as 4,000 STB/day. In order to save computation, all bound constraints are enforced by the truncation scheme in this example. Recall that the scaled control vector x is confined within the $[0, 1]$ scale by Eq. 4.1. For the scaled control vector x, if any entry of x is smaller than 0 during the iteration of the optimization, we set this the value of this entry as 0. If any entry of x is larger than 1 during the iteration of the optimization, we set this the value of this entry as 1. All nonlinear state constraints are summarized in Table 4.10.

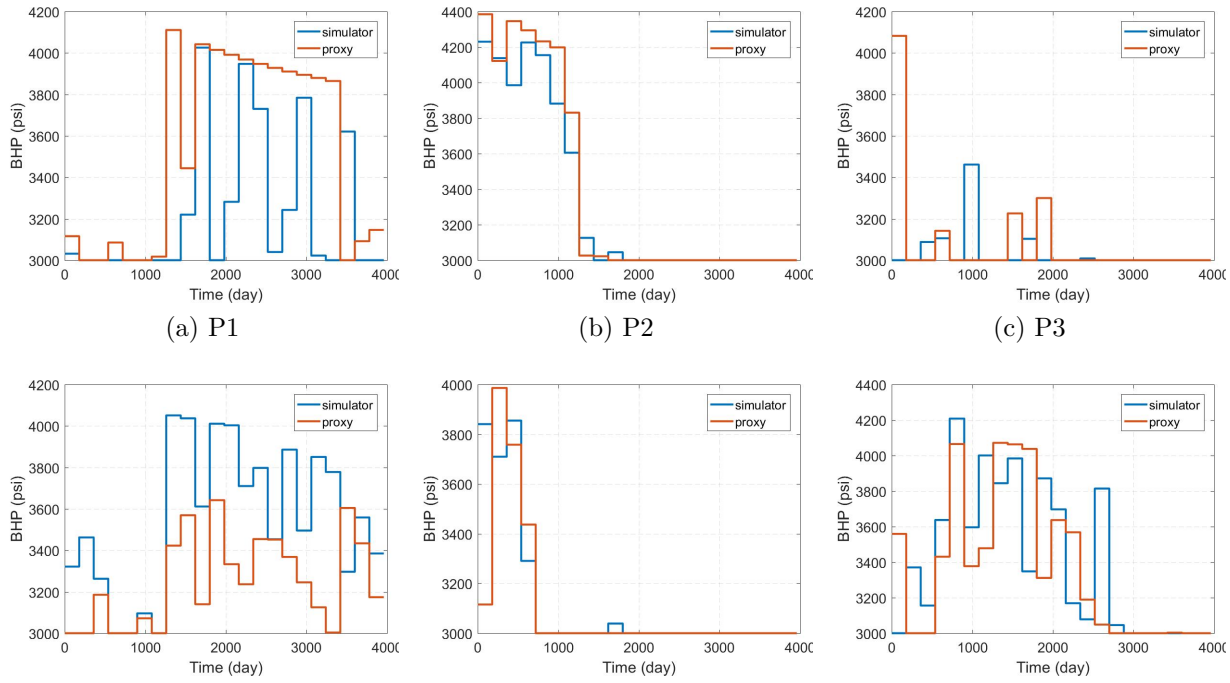
To compare the performance of the proxy-based optimization framework with the simulator-based optimization framework, we apply two different optimization frameworks in this example. In the first framework, we apply the SQP-filter to solve the constrained optimization problem using the high-fidelity reservoir simulator (no proxy generated) in which the gradient information is computed by adjoint method. In the second framework, we apply the SQP-filter to solve the problem with a trained GE-SVR model. In this deterministic optimization case, we try to further reduce the number of initial training samples from 5 to 4 to see if the GE-SVR proxy could solve the proxy-based production optimization problem

Well	Cons. Type	Cons. Value	Cons. Scale	Type of inequality
P1-6	rate	0.001	10	$\geq$
I1,I2,I3	BHP (psi)	4,600	1,000	$\leq$
FWR	rate (STB/d)	4,000	4,000	$\leq$
FLR	rate (STB/d)	6,000	6,000	$\leq$
FGR	rate (Mcf/d)	25,000	25,000	$\leq$

Table 4.10: The summary of inequality nonlinear state constraints for the proxy-based production optimization problem with life-cycle NPV as the objective function .

with state constraints with small sets of training samples when iterative sampling workflow is applied; see Fig 4.1.

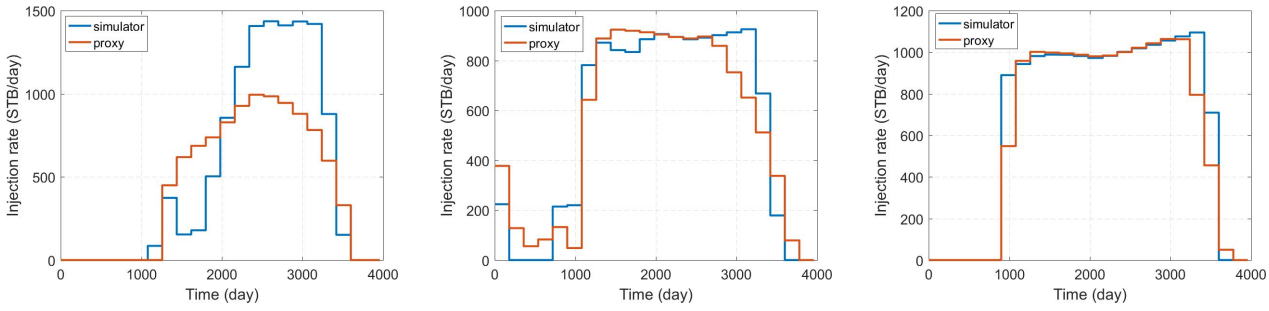
Fig. 4.14 compares the optimal well controls generated by reservoir simulator based optimization (blue lines) with those generated with proxy-based optimization (red lines). Although the estimated optimal controls generated from SQP optimization using the simulator with adjoint capability and those generated by optimization of the GE-SVR proxy model tend to be in reasonable qualitative agreement, there definitely exist quantitative differences; see Fig 4.14 and 4.15. Such quantitative differences are not surprising as the well control problem generally has multiple solutions [53, 61]. Even through the two sets of controls have clear differences, the optimal NPV values estimated using the two different models are very close, 4.37×10^8 USD for simulator-based optimization versus 4.39×10^8 USD for GE-SVR proxy-based optimization with iterative sampling. Fig. 4.16 shows the oil rates of the producers computed from the proxy-based and simulator-based optimal controls of Figs. 4.14 and 4.15. Except for the oil rate at well P1 between 1800 and 3,600 days, the well rates obtained from the two methods are in reasonable agreement. From the results of Fig. 4.14, we observe that at both wells P2 and P5, the production rates are restricted by maintaining relatively high BHPs at the beginning of the lifecycle. Recall that, unlike the other wells, producers P2 and P5 are completed in geologic layer 3, which is relatively close to the gas cap. Thus, the high BHP's in wells P2 and P5 at the early control steps restrict



(a) P1 (b) P2 (c) P3

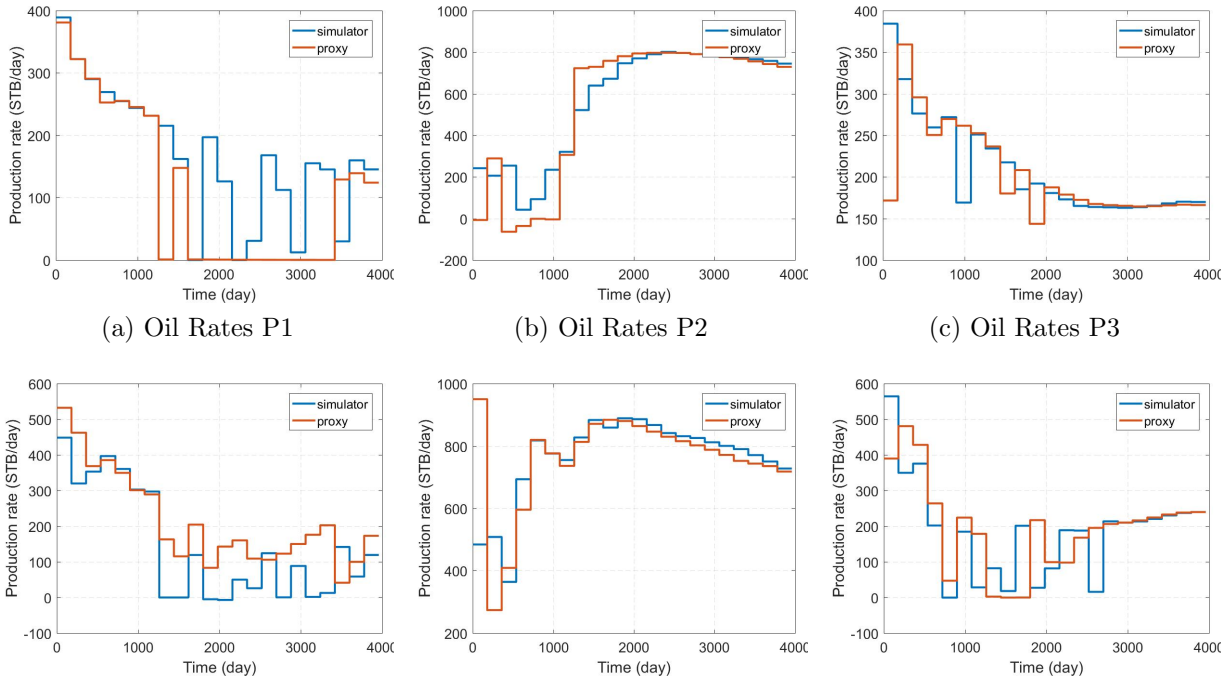
(d) P4 (e) P5 (f) P6

Figure 4.14: Comparison of the optimal BHPs for producers generated by simulator-based life-cycle production optimization (blue lines) and proxy-based optimization (red lines) for a deterministic case with state constraints.



(a) Water Rates I7 (b) Water Rates I8 (c) Water Rates I9

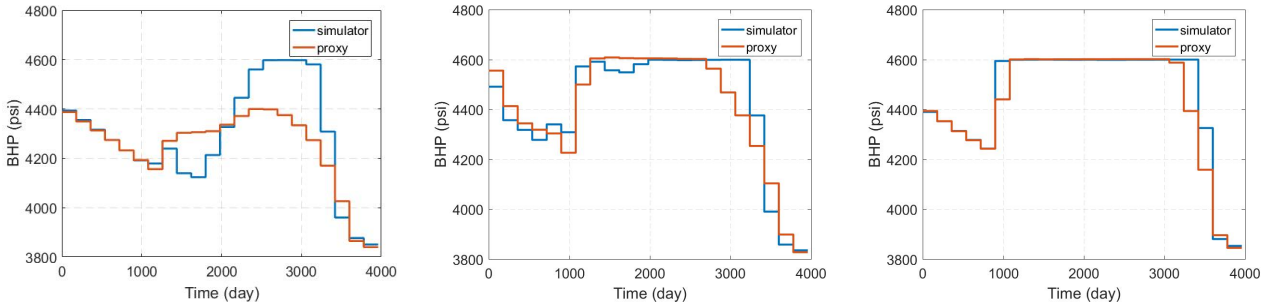
Figure 4.15: Comparison of the optimal water injection rates generated by simulator-based optimization (blue lines) and proxy-based optimization (red lines) for deterministic life-cycle production optimization with state constraints.



(a) Oil Rates P1 (b) Oil Rates P2 (c) Oil Rates P3

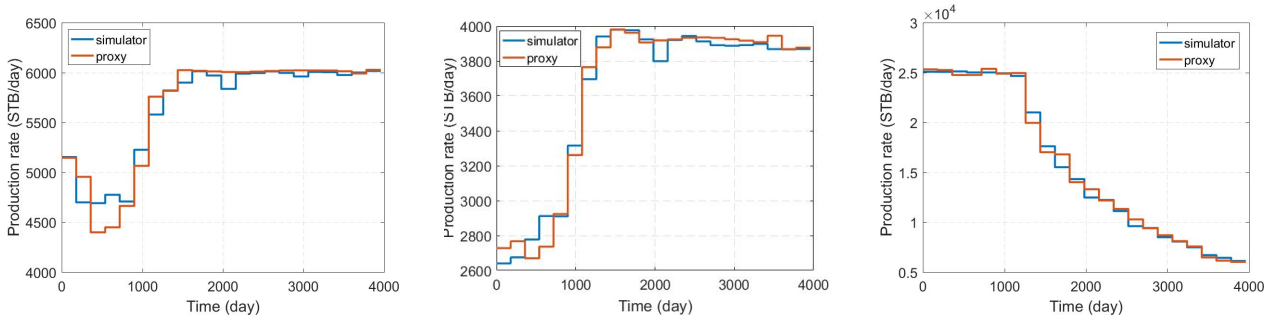
(d) Oil Rates P4 (e) Oil Rates P5 (f) Oil Rates P6

Figure 4.16: Comparison of the oil production rates of each producer under the optimal controls generated by simulation-based life-cycle production optimization (blue lines) and proxy-based optimization (red lines) for the deterministic production optimization with state constraints.



(a) BHP I1 (b) BHP I2 (c) BHP I3

Figure 4.17: Comparison of the BHP at each injector under the optimal controls generated by simulation-based optimization (blue lines) the proxy-based optimization (red lines) for the deterministic production optimization with state constraints.



(a) FLR (b) FWR (c) FGR

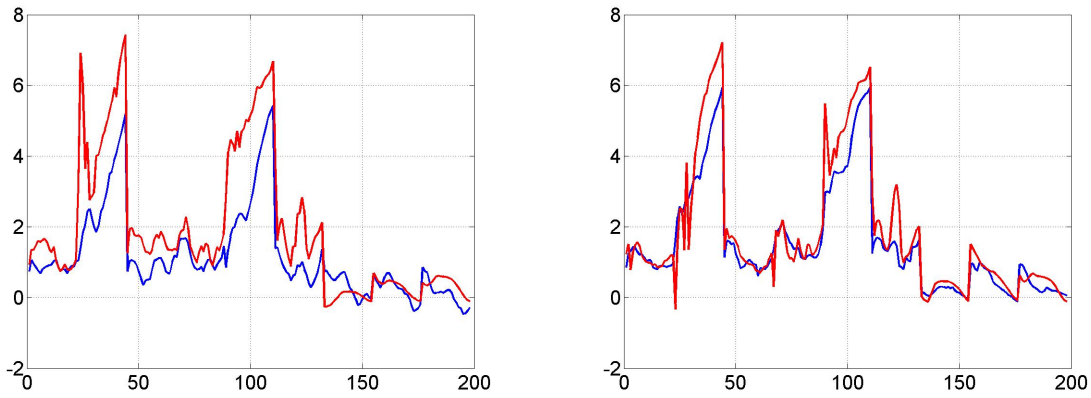
Figure 4.18: Comparison of the FLR, FWR and FGR under the optimal controls generated by simulation-based optimization (blue lines) the proxy-based optimization (red lines) for the deterministic production optimization with state constraints.

gas production in order to satisfy the constraint on the FGR. Note that water injection wells are controlled by the water injection rates, thus the upper BHP bounds at the injectors are nonlinear state constraints for this constrained optimization problem. Fig. 4.17 shows that all the BHP state constraints of injectors are satisfied.

Fig. 4.18 shows the field liquid production rate (FLR), field water production rate and field gas production rate (FGR) at each control step obtained for both simulator-based optimization and GE-SVR proxy-based optimization where for each method, the estimates of the optimal wells controls are used to compute production rates. From the results of Fig. 4.18a, we observe that the FLR is close to its upper bound (6,000 STB/day) from day 1500 to the end of the production period. Moreover, the FWR is also close to its upper bound

(4,000 STB/day) from from day 1500 to the end of production period. From Fig. 4.18c, we see that the FGR is close to its upper bound of 25,000 Mscf/d from early-times to 1200 days. The large gas production rate results largely from the gas cap that occupies the top layer. The high value of BHP controls of producers P2 and P5 (Fig. 4.14) result from the need to satisfy the FGR constraint. From the beginning of the production to 1200 days, the FGR in the active constraint. From 1500 days to the end of the production period, both the FLR and FWR become active constraints. Note that even though the FLR and FWR are fairly far below their upper bounds for times prior to 1,000 days, the water injection rate controls are all zero during this time period. If these water injection rates were not kept at zero and BHP controls were unchanged, the oil rate, and hence the rate of the production of dissolved gas would increase resulting in a violation of the FGR. In essence, at early times, the gas cap and aquifer provide the energy needed to produce oil.

To investigate the ability of GE-SVR to accurately predict the output from the reservoir simulator near an optimal point generated during iterative sampling, we plot the comparison of the adjoint gradients of life-cycle NPV with the proxy-predicted gradients at different iterative sampling steps in Fig. 4.19 where both simulator and proxy derivatives are evaluated at the optimal well controls generated by the proxy-based deterministic optimization with the GE-SVR proxy, except that in Fig. 4.19a,derivatives are evaluated at the initial guess used in the iterative sampling workflow. The derivatives, which are the components of the adjoint gradients are shown by red lines and the corresponding derivatives computed from the GE-SVR proxy are shown as blue lines. From Fig. 4.19a, we can observe that the initial GE-SVR proxy gives derivatives in reasonably good agreement with those obtained with the adjoint even though this initial GE-SVR proxy is trained with only five training samples which are generated by sampling the scaled control variables from LHS and running the reservoir simulator with the corresponding values of unscaled well controls to compute corresponding outputs. As the iterative sampling scheme goes on, the results of Figs. 4.19b, 4.19c and 4.19d show that the differences between gradients of life-cycle NPV computed from the proxy and the reservoir simulator decrease gradually. At the end of



(a) Before iterative sampling

(b) At 1st iterative sampling step

(c) At 3rd iterative sampling step

(d) At 5th iterative sampling step

Figure 4.19: Comparison of adjoint gradient of life-cycle NPV with respect to well controls (red) with proxy predicted gradient of life-cycle NPV (blue) at different iterative sampling steps

iterative sampling, the components of the adjoint gradient of the objective life-cycle NPV calculated from the reservoir simulator are almost identical to the corresponding values of derivatives computed from the GE-SVR proxy which means the proxy model is quite accurate in the region around the optimum obtained at the end of iterative sampling.

Table 4.11 summarizes the performance of the iterative sampling steps of proxy-based optimization for the deterministic constrained life-cycle production optimization example under consideration. The state constraints are handled by the min-max scheme defined in Section 2.2.5. The first column denotes the index of the iterative sampling step. The second column denotes the angles between the adjoint gradients ($\nabla \bar{J}_{\text{sim}}$) and the GE-SVR proxy predicted gradients ($\nabla \bar{J}_{\text{proxy}}$) which are computed using the estimate of the optimal well controls obtained at the end of each proxy-based SQP-filter optimization. The formula for calculating the angles between the two gradients is given in Eq. 2.29. The third column lists the relative differences in magnitude between $\nabla \bar{J}_{\text{sim}}$ and $\nabla \bar{J}_{\text{proxy}}$ as $\frac{||\nabla \bar{J}_{\text{sim}} - \nabla \bar{J}_{\text{proxy}}||_2}{||\nabla \bar{J}_{\text{sim}}||_2}$. The fourth and fifth columns of Table 4.11, respectively, represent the corresponding negative life-cycle NPV at the optimums obtained at each iterative sampling step computed from the reservoir simulator and GE-SVR proxy, respectively. The sixth column lists the average constraint violations estimated by the reservoir simulator at the optimums of each iterative sampling step. From the table, we can observe that both angles and the relative differences between the $\nabla \bar{J}_{\text{sim}}$ and $\nabla \bar{J}_{\text{proxy}}$ decrease as the iterative sampling process goes on and this result can also be observed in Fig. 4.19. Therefore, we could conclude that the iterative sampling process can improve the accuracy of the GE-SVR proxy around the optimum. By observing the values in the fourth column, we can see the changes in the value of $\bar{J}_{\text{sim}}$ decrease as iterative sampling process goes on. By comparing the values at columns 4 and 5, we can see the relative differences between the $\bar{J}_{\text{sim}}$ and $\bar{J}_{\text{proxy}}$ decrease as the iterative sampling process goes on and the convergence criterion on

$$\frac{|\bar{J}_{\text{sim}} - \bar{J}_{\text{proxy}}|}{|\bar{J}_{\text{sim}}|} \leq \epsilon_{\text{p}} \tag{4.45}$$

is satisfied after only two iterative sampling steps which means this criterion is not the dominant criterion to determine the convergence of the iterative sampling algorithm. The values of the average constraint violations $\bar{\theta}_{\text{sim}}(u^{*,n_\ell})$ is also decreasing as the number iterative sampling steps increases and the results indicate that $\bar{\theta}_{\text{sim}}(u^{*,n_\ell})$ is the dominant criterion to control the convergence of iterative sampling process in this example problem. In this problem, we can observe that at the optimal controls generated from the iterative sampling scheme, many state constraints such as FLR, FWR and FGR in Fig. 4.18 are very close to corresponding constraints. Therefore, enforcing the nonlinear state constraints is a challenging process and the criterion on the average constraint violation is the dominant one for this deterministic production optimization problem but we cannot make an general conclusion from this one particular example. Azad et al. in the TUPREP report 2020 [3] introduces an additional convergence criterion on the change in the objective function computed from the simulator output between two successive iterative sampling steps. To be specify, they use the convergence criterion defined by

$$\frac{|\bar{J}_{\text{sim}}(x^{*,n_\ell}) - \bar{J}_{\text{sim}}(x^{*,n_\ell})|}{|\bar{J}_{\text{sim}}(x^{*,n_\ell})|} \leq \epsilon_{\text{sim}}, \tag{4.46}$$

where $J^{*,n_\ell}_{\text{sim}}$ denotes the average negative NPV computed using the simulator output generated using the optimal controls generated by proxy-based optimization with the n_ℓth GE-SVR proxies, i.e., at the end of the n_ℓth iterative sampling step and $J^{*,n_\ell-1}_{\text{sim}}$ is defined similarly. Azad et al.[3] suggest that by setting $\epsilon_{\text{sim}} = 0.01$, a proxy-based optimization algorithm can achieve a higher life-cycle NPV for optimization a huff-n-puff process involving CO_2 injection but no nonlinear state constraints are considered in their work. From the values in the fourth column of Table. 4.11, we see that the criterion of Eq. 4.46 is satisfied after only two iterative sampling steps. Thus, in this example adding a convergence criterion based on Eq. 4.46 does not affect the optimal well controls that are obtained. Nevertheless, the criterion of Eq. 4.46 is a reasonable one to enforce and the option of also using this convergence criterion in addition to the convergence criteria in Eq. 4.40 and 4.40 is embedded

in our in-house software.

Iterative sampling step	Angle between $\nabla\bar{J}_{\text{sim}}$ & $\nabla\bar{J}_{\text{proxy}}$	$\frac{\|\nabla\bar{J}_{\text{sim}}-\nabla\bar{J}_{\text{proxy}}\|_2}{\|\nabla\bar{J}_{\text{sim}}\|_2}$	$\bar{J}_{\text{sim}}(u^{*,n_\ell})$ USD	$\bar{J}_{\text{proxy}}(x^{*,n_\ell})$ USD	$\bar{\theta}_{\text{sim}}(u^{*,n_\ell})$
0	24.77	0.43	-4.65×10^8	-4.39×10^8	0.59
1	10.95	0.24	-4.31×10^8	-4.27×10^8	0.23
2	8.65	0.17	-4.35×10^8	-4.36×10^8	0.22
3	5.67	0.11	-4.39×10^8	-4.38×10^8	0.10
4	5.62	0.10	-4.39×10^8	-4.40×10^8	0.05
5	4.37	0.08	-4.39×10^8	-4.39×10^8	0.01

Table 4.11: Summary of iterative sampling steps of proxy-based optimization for deterministic constrained production optimization

Some results of this deterministic constrained optimization case are summarized in Table 4.12. From the results, we observe that both simulator-based and proxy-based optimization produce similar values of the optimal NPV with all constraints satisfied. In terms of the computational cost, simulator-based optimization requires 44 gradient evaluations and 50 extra forward reservoir simulations (runs where adjoint solutions are not computed). Note, in this example, the reservoir simulator needa to estimate the gradient of both objective function (life-cycle NPV) and nonlinear state constraints (e.g., FLR, FWR and FGR) in the adjoint solution since we need to train GE-SVR proxy models for the objective function and each of the state constraints and their derivatives. The adjoint solution consists of a forward reservoir simulation run, which is a regular simulation run, and a backward simulation run to calculate the gradient. Therefore, in Table 4.12, each "run" is equivalent to regular one forward simulation run and each "grad. eval." (gradient evaluation) is equivalent to one forward simulation run and one backward simulation run. The proxy-based optimization framework only requires 4 gradient evaluations of the initial training samples and 5 more gradient computations due to the iterative sampling steps. The com-

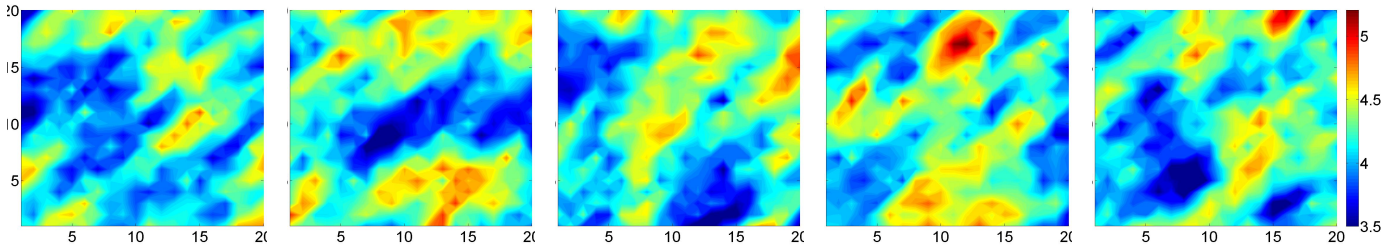
putation cost of evaluating the GE-SVR proxy model is not included in Table 4.12 since its cost is negligible compared to the computational cost of a simulation run. Compared with simulator-based optimization, proxy-based optimization framework with GE-SVR as the proxy model saves significant computational time in terms of the number of gradient evaluations and forward simulations. Assuming that the computational cost of an adjoint calculation (backward simulation) is equivalent to the cost of a simulation run, the results indicate that the simulator-based optimization is on the order of one magnitude more costly than GE-SVR proxy-based optimization on this deterministic production optimization with state constraints.

Optimization Type	Optimal NPV, USD (estimated by simulator)	Optimal NPV, USD (estimated by proxy)	Number of Simulations
Simulator-based	4.37×10^8	-	44 grad. eval.+ 50 runs
Proxy-based	4.39×10^8	4.39×10^8	9 grad. eval.

Table 4.12: Comparison of the performance of simulator-based and proxy-based optimizations on the deterministic production optimization problem with state constraints. (Example 3 part I)

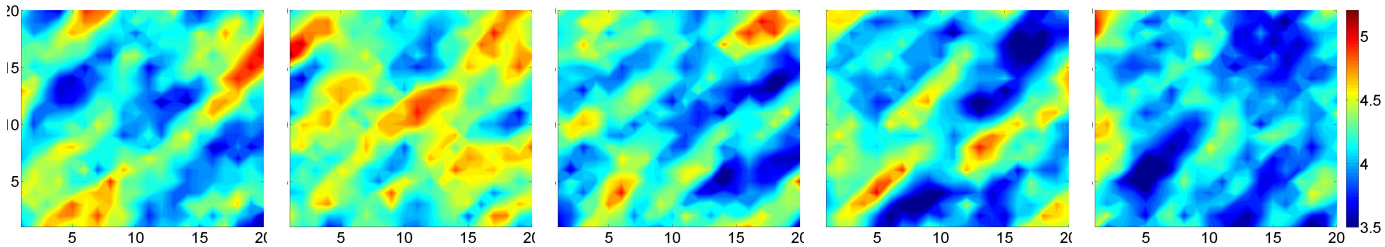
In the second part of this example, the proxy-based robust optimization scheme with the GE-SVR proxy model is applied to solve a constrained robust production optimization problem with nonlinear state constraints. The settings of this robust optimization case are exactly same as for the previous deterministic case except there are 10 distinct geological realizations of the reservoir model used to compute the average life cycle NPV in the robust case. Figs. 4.20 and 4.21 illustrate the log permeability of each layer of the reservoir of the realization 1 and realization 2, respectively. By comparing Figs. 4.20 and 4.21, we can see the each realization are distinct to each other.

Five different control vectors are sampled with LHS and the reservoir simulator is run for each reservoir model with each sample of the vector of controls to obtain corresponding output in order to form the initial training set to build the initial proxy model for the iterative



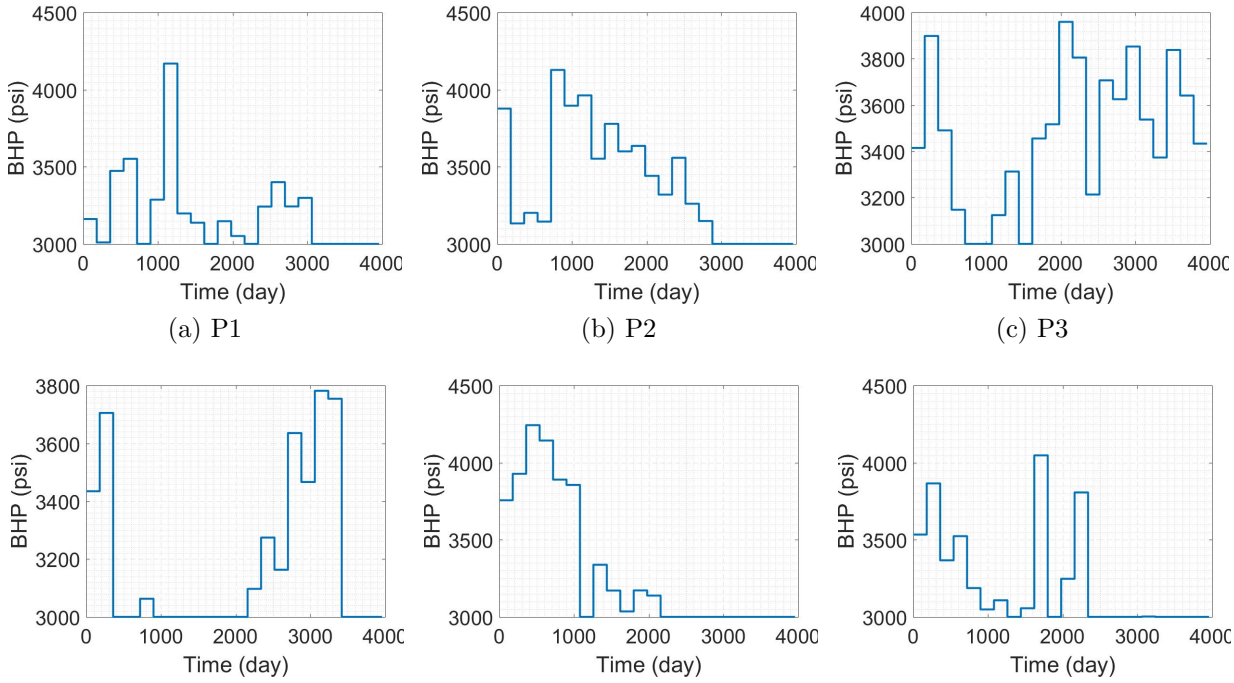
(a) Layer 1 (b) Layer 2 (c) Layer 3 (d) Layer 4 (e) Layer 5

Figure 4.20: The Log permeability of each geological layer for realization 1 .



(a) Layer 1 (b) Layer 2 (c) Layer 3 (d) Layer 4 (e) Layer 5

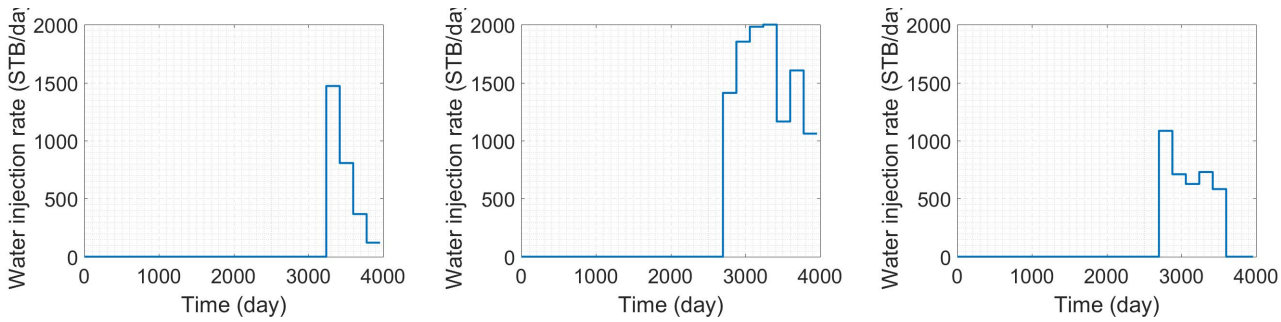
Figure 4.21: The Log permeability of each geological layer for realization 2 .



(a) P1 (b) P2 (c) P3

(d) P4 (e) P5 (f) P6

Figure 4.22: The optimal BHP at producers generated from proxy-based constrained robust production optimization.



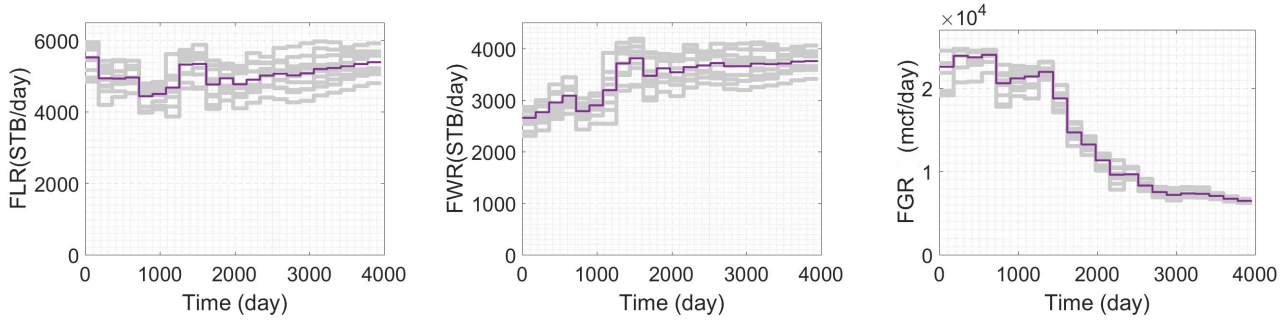
(a) I7 (b) I8 (c) I9

Figure 4.23: The optimal water injection rates for injectors generated form the proxy-based robust production optimization.

sampling optimization procedure. Thus, since there are 10 geological realizations, 50 forward and backward simulation runs are required to generate life-cycle NPV and state constraints values and corresponding gradients. To solve the proxy-based robust production optimization problem with state constraints, we need to train the GE-SVR proxies for the objective function and each state constraints for each reservoir model. Therefore, the backward run of the reservoir simulator is required to calculate the adjoint solutions on the objective function and each of the state constraints for each reservoir model. Note this means we are effectively training $N_{ic}+1$ GE-SVR proxy models for each reservoir model, where N_{ic} is the number of state constraints for each reservoir model. Training of the GE-SVR model basically solves the linear system proposed in Eq. 4.20. As stated in section 4.2.1, the left hand side the matrix in the linear system given by Eq. 4.20 only depends on input data, i,e, the x^k, $k = 1, 2, \ldots N_s$ and does not depend on the training output function $y(x)$. By replacing the right hand side on the training output function, we can train the objective function and all state constraints for each reservoir model once with only one inversion of the matrix involved in the linear system. The min-max scheme (see section 2.2.5) is implemented to reduced the number of state constraints to be enforced in the proxy-based SQP-filter process from $N_{ic} \times N_e$ to N_{ic}, where N_{ic} denotes the number of state constraints for each reservoir model and N_e denotes the number of reservoir models. To be specific, with the mix-max scheme used to handle constraints, for each state constraint, the proxy model needs only

to be trained on the realization of the reservoir model which has the maximum violation of that state constraint among all reservoir models. Our results indicate that the proxy-based optimization utilizes 8 iterative sampling steps to achieve convergence and thus 70 more forward and backward reservoir simulation runs are used to reach the final optimal solution. Due to the limitation in computational resources, we did not solve this problem by the simulator-based optimization scheme however, based on the literature [53] solving the constrained robust production optimization problem with gradient-based algorithm is a computationally intense process which would require on the order of 1000 or more forward and backward runs. Therefore, by training a GE-SVR model and using the iterative sampling scheme, the proxy-based scheme can significantly reduce the computational cost.

Figs. 4.22 and 4.23, respectively, show the optimal BHP's of producers and injection rates for water injection wells. . There are some similarities between the optimal controls generated from deterministic optimization (see Fig. 4.14) and the optimal controls generated from the robust proxy-based optimization case shown in Fig. 4.22. For example, to restrict the gas production from the gas cap, producer P5 operates at and high pressure for the first 2 control steps and producer P2 operates a high pressure during controls steps, 1 and 5-8 in Fig 4.22, while the optimal controls generated by the deterministic optimization restrict BHP at P2 from the first to seventh control steps and restrict BHP at P5 for the first 4 control steps. From the results, we see that all producers except P3 operate at the minimum BHP (3,000 psi) near the end of the life cycle of the reservoir development whereas P3 severely restricts production at late times by increasing the BHP. Moreover, all injectors tend to inject water only at the end of production period similar to the deterministic case. Fig. 4.24 illustrates the field liquid production rates (FLR), field water production rates (FWR) and field gas production rates (FGR) of each realization at the optimal well controls. The field rates for different realizations are denoted as grey lines and the average field rates over all realizations are represented as purple lines. In Fig. 4.24a, we observe that for one or more realizations, the FLR is close to its upper boundary over the entire production life-cycle. Similarly, for one or more realizations, the FWR in Fig. 4.24b is near its upper bound from



(a) FLR (b) FWR (c) FGR

Figure 4.24: The FLR, FWR and FGR under the optimal controls generated by the proxy-based optimization for the robust production optimization with state constraints.

1400-3960 days. Moreover, we see that the maximum value of FGR over all realizations is around its upper limit from the beginning of production to 1600 days. Therefore, we can conclude that the proxy-based optimization schemes can handle the nonlinear state constraints for every realization at every control step when the min-max method for constraints is used in the SQP-filter algorithm.

Fig. 4.25 illustrates the cumulative distribution function (CDF) for NPV at the initial well controls (blue line) and at the optimal well controls (red line). The mean value of NPV of all realizations operating under the initial set of well controls is 2.988 $\times 10^8$ USD and the mean value of that NPV operating under the optimal controls is 4.069 $\times 10^8$ USD, i.e., optimization achieves a 36% increase in the mean value of the NPV.

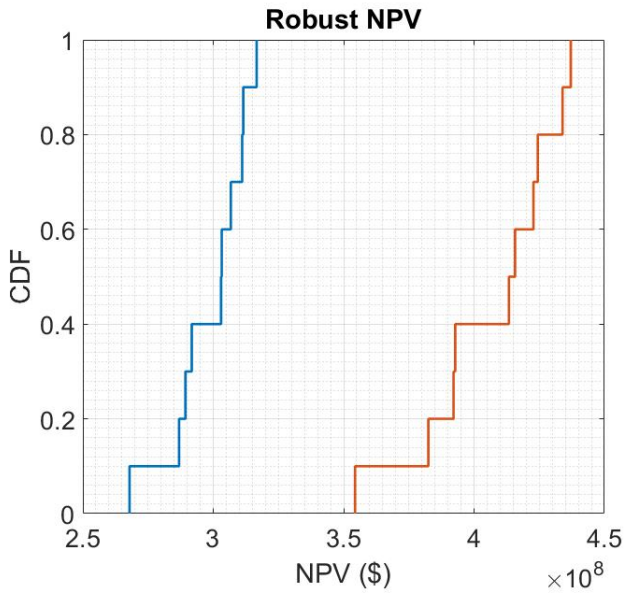


Figure 4.25: Comparison of CDF of the robust life-cycle NPV generated by operating under the initial well controls (blue) and optimal well control (red).

CHAPTER 5

DISCUSSION AND CONCLUSIONS

In this work, we aim to solve the robust production optimization problem with nonlinear state constraints. We first proposed a novel SQP-filter framework based on the stochastic gradient (StoSAG) for robust optimization with nonlinear state constraints where nonlinear state constraints are enforced with the filter method. Two schemes are proposed to improve the accuracy of the derivatives of phase flow rates as approximations of the corresponding derivatives computed from an adjoint method. As state constraints on field rates and the NPV functional are simply linear combinations of phase rates at wells, the basic idea is to improve the quality of the stochastic gradients of the state constraints and NPV with respect to the vector of controls (pressures for production wells, water rates for injection wells) by improving the accuracy of the derivatives of the phase rates with respect to pressure controls. The first improvement applies truncation wherein we set stochastic derivatives obtained from the standard StoSAG gradient that physically should be zero equal to zero. The second method applies a temporal damping of the derivatives computed from the normal StoSAG algorithm. The damping heuristic is motivated by an analytical derivation based on single-phase flow. By combing both modifications, we show that the resulting StoSAG derivative of rates provide a better approximation of the derivative generated with the adjoint method than does standard StoSAG. Conceptually, every realization of the model should satisfy the constraints but this is very computationally expensive as in this case, constraints have to be enforced at every control step for every reservoir model and thus, the stochastic gradients of all constraints need to be computed for every reservoir model . To improve computational efficiency, we tried three procedures. In the first (expected value scheme), we apply the constraint only to the average constraint function where the average is over all

reservoir models. In the second (min-max scheme), at each iteration of SQP, we explicitly enforce each constraint on the reservoir model that gives the maximum violation of that constraint. In the third method (heuristic enforcement), we first apply StoSAG with only bound constraints, and then insert the optimal controls found into the reservoir simulator and use the internal heuristics of the simulator to modify the control to enforce the state constraints. Based on the results shown in Chapter 2 of this work, as well as theoretical discussions, the following conclusions are warranted:

- The modified StoSAG gradients of nonlinear constraints are more accurate than the standard StoSAG gradients and thus result in a vastly improved solution of the robust, life-cycle optimal well control problem under nonlinear state constraints than the solution using gradients from standard StoSAG when the SQP-filter framework is applied to solve the constrained optimization problem.

- Obtaining accurate gradients of the state constraints for linearization of these constraints in SQP is critical for good performance of the algorithm because the linearized constraints define the feasibility region for the QP subproblem. If the feasibility region is not defined with sufficient accuracy, the subproblem may not have a feasible solution.

- To obtain sufficiently reliable approximate gradients of nonlinear state constraints, both truncation, and temporal damping should be used whereas a useful gradient of the average NPV can be generated by simply performing truncation.

- The computational efficiency of the filter method for enforcing constraints can be improved by setting an upper bound on infeasibility in order to prevent too large a step at any iteration of the SQP-filter optimization algorithm.

- When the expected value method is applied to enforce constraints, for some reservoir models, some state constraints are violated by 25 to 50%.

- Heuristic enforcement of constraints by the reservoir simulator results in a vastly inferior CDF for NPV than is obtained with the min-max scheme.

- Although, in robust constrained optimization, extra perturbations are introduced for computing gradients of constraints by the min-max scheme, the min-max scheme can generate optimal controls under which FLR and FWR constraints are satisfied for every realization.

In Chapter 3, we extend the SQP-filter framework with the stochastic gradient (StoSAG) for robust production optimization with nonlinear state constraints to solve the multi-objective production optimization problem. The life-cycle NPV, short-term NPV and down-side risk are the three objectives considered in this work. The down-side risk is characterized by the worst-case NPV among all realizations of reservoir models. The robust multi-objective optimization problem with state constraints is solved with a modified lexicographic method proposed in this work. The first step for both the standard and modified lexicographic methods requires the solution of the single objective robust production optimization problem with state constraints by the SQP-filter method using the improved StoSAG gradient and the min-max scheme. At the end of this optimization, we record the average short-term NPV and minimum NPV over the set of realizations at the optimal well controls u^*. Then, in contrast with the standard lexicographic method which switches the primary objective and sets as a state constraint a lower bound on the average life-cycle NPV, the modified lexicographic method maximizes the live-cycle NPV again but with state constraints set as a constant greater than 1 times the average short term NPV at u^* and/or another constant greater than one times the worst-case NPV evaluated at u^*. By changing the constants, we can try to find Pareto optimal points. The Brugge case is used to illustrate the utility of this multi-objective robust production optimization framework. Based on the computational results, we have the following conclusions:

- The modified lexicographic procedure developed in this work provides an effective method for bi- and tri-objective robust production optimization with state constraints. The main advantage that the modified lexicographic method has over the the standard lexicographic method is that it allows the generation of potential Pareto optimal points

which are uniformly spaced in the values of the second objective that one wishes to improve by bi-objective optimization.

- In the modified lexicographic method, the value of the average life-cycle NPV as the primary objective function obtained by bi- or tri- objective optimization is often higher than its value obtained by optimizing only average life-cycle NPV (single objective).

- Pareto fronts generated by the modified lexicographic method with the improved StoSAG are different than expected in the sense that, within a certain range of the bounds placed on the other objective functions, the average life-cycle NPV increases as we increase the lower bounds imposed on the short-term NPV and worst-case NPV incorporated as state constraints. Once these lower bounds become large enough, the two secondary objectives incorporated as state constraints act as if there are truly in conflict with the primary objective, the maximization of average life-cycle NPV, we start to generate points on the Pareto front.

- Despite the aforementioned atypical behavior, the points generated in an attempt to find the Pareto front provide choices that enable the operational engineer to set controls based on his or her short-term and life-cycle production goals and risk tolerance.

In Chapter 4, we propose the proxy-based optimization technique to solve life-cycle production optimization with nonlinear state constraints. Our major contribution here is the development of a machine learning proxy which is trained to match both function values and corresponding values of the derivatives of the function which are generated from a reservoir simulator with full adjoint-solution capability. In this approach, a proxy (GE-SVR) is trained to match simulator training output for the life-cycle NPV and state constraint functions and their derivatives. In the case of life-cycle NPV, a separate proxy is trained to match the simulator values of training output of NPV and associated derivatives for each of the reservoir models used to characterize geological uncertainty. Since the min-max scheme is applied to enforce the nonlinear state constraint, each facility state constraint is only imposed on the reservoir model that gives the maximum violation of that constraint. Thus, the proxy model

for each constraint function and its derivatives needs to be trained for only one reservoir model at each iteration of the SQP filter algorithm. Because both function and derivative information are used to train the proxies, the proxies are referred to as gradient-enhanced support vector regression (GE-SVR) proxies. The GE-SVR proxies then replace the reservoir simulator during robust life-cycle production optimization with the SQP-filter method where the min-max scheme is applied to enforce nonlinear state constraints. In the optimization process, iterative sampling is applied. For two simple standard optimization test cases and one synthetic reservoir model we compare the performance of proxy-based optimization with GE-SVR with the performance of LS-SVR. On the basis of the results of Chapter 4, the following conclusions are warranted.

- Compared with LS-SVR, which can treat the reservoir simulator as a back-box and only requires objective and constraint function values from the simulator to generate training samples, generating training samples for constructing the GE-SVR proxy requires a reservoir simulator which has the adjoint capability necessary to compute the gradients of the objective and state constraint functions with respect to the vector of well controls.

- Generating the GE-SVR proxies only requires the inversion of a single matrix once to train the GE-SVR proxy models for the objective function and all of the state constraints.

- The proxy-based iterative sampling scheme with GE-SVR as proxy not only can find more local optimums but also can reach lower optimum values with fewer iterative sampling steps than the case with the LS-SVR as the proxy.

- Computational results on deterministic optimization with state constraints indicate that the proxy-based optimization with iterative sampling and GE-SVR proxy models requires on the order of 5-times fewer equivalent reservoir simulation runs than are required for the simulator-based optimization with the adjoint solution.

- The iterative sampling scheme improves the accuracy of the GE-SVR proxy in the region around the optimal solution and thus leads to an improvement in the estimate of the optimal solution.

- The min-max implemented here for satisfying state constraints vastly improves computational efficiency compared to enforcing constraints on all individual realizations.

BIBLIOGRAPHY

[1] Andreas Antoniou and WuSheng Lu. *Practical optimization: algorithms and engineering applications*. Springer Science & Business Media, 2007.

[2] H. Asheim. Maximization of water sweep efficiency by controlling production and injection rates. In *Proceedings of the SPE European Petroleum Conference*, number SPE 18365, 1988.

[3] Almasov Azad et al. Production optimization of the CO2 Huff and Puff process in unconventional oil reservoirs using LS-SVR and GPR based machine learning proxies. TUPREP Research Report 36, The University of Tulsa, 2020.

[4] A Bao, E Gildin, and H Zalavadia. Development of proxy models for reservoir simulation by sparsity promoting methods and machine learning techniques. In *ECMOR XVI-16th European Conference on the Mathematics of Oil Recovery*, 2018.

[5] D. R. Brouwer, Geir Nævdal, J. D. Jansen, Erland H. Vefring, and C. P. J. W. van Kruijsdijk. Improved reservoir management through optimal control and continuous model updating. In *Proceedings of the SPE Annual Technical Conference and Exhibition, Houston, Texas, 26-29 September*, number SPE 90149, 2004.

[6] D.R. Brouwer and J.D. Jansen. Dynamic optimization of water flooding with smart wells using optimial control theory. *SPE Journal*, 9(4):391–402, 2004.

[7] Andrea Capolei, Bjarne Foss, and John Bagterp Jørgensen. Profit and risk measures in oil production optimization. *IFAC-PapersOnLine*, 48(6):214–220, 2015.

[8] Marco A Cardoso and Louis J Durlofsky. Linearized reduced-order models for subsurface flow simulation. *Journal of Computational Physics*, 229(3):681–700, 2010.

[9] A Castellini, H Gross, Y Zhou, J He, and W Chen. An iterative scheme to construct robust proxy models. In *ECMOR XII-12th European Conference on the Mathematics of Oil Recovery*, 2010.

[10] Bailian Chen. *A stochastic simplex approximate gradient for production optimization of WAG and continuous water flooding*. The University of Tulsa, 2017.

[11] Bailian Chen, Rahul-Mark Fonseca, Olwijn Leeuwenburgh, and Albert C Reynolds. Minimizing the risk in the robust life-cycle production optimization using stochastic simplex approximate gradient. *Journal of Petroleum Science and Engineering*, 153:331–344, 2017.

[12] Bailian Chen and Albert C Reynolds. Ensemble-based optimization of the water-alternating-gas-injection process. *SPE Journal*, 21(03):786–798, 2016.

[13] Chaohui Chen. *Adjoint-gradient-based production optimization with the augmented Lagrangian Method*. 2011.

[14] Chaohui Chen, Gaoming Li, and A. C. Reynolds. Constrained optimization algorithms for the production optimization step of closed loop reservoir management. TUPREP research report, The University of Tulsa, 2009.

[15] Chaohui Chen, Gaoming Li, and Albert Reynolds. Robust constrained optimization of short-and long-term net present value for closed-loop reservoir management. *SPE Journal*, 17(03):849–864, 2012.

[16] Chaohui Chen, Gaoming Li, and Albert C. Reynolds. Closed-loop reservoir management on the Brugge test case. *Computational Geosciences*, 14(4):691–703, 2010.

[17] Chaohui Chen, Gaoming Li, and Albert C. Reynolds. Robust constrained optimization of short and long-term npv for closed-loop reservoir management. *SPE Journal*, 17(3):849–864, 2012.

[18] Yan Chen and Dean S Oliver. Ensemble-based closed-loop optimization applied to Brugge field. *SPE Reservoir Evaluation & Engineering*, 13(01):56–71, 2010.

[19] Yan Chen, Dean S. Oliver, and Dongxiao Zhang. Efficient ensemble-based closed-loop production optimization. *SPE Journal*, 14(4):634–645, 2009.

[20] Yan Chen and Oliver Dean Stuart. Localization of ensemble-based control-setting updates for production optimization. *SPE Journal*, 17(01):122–136, 2012.

[21] Andrew R. Conn, Katya Scheinberg, and Luis N. Vicente. *Introduction to Derivative Free Optimization*. SIAM, Philadelphia, PA, 2009.

[22] Indraneel Das and John E Dennis. Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM journal on optimization*, 8(3):631–657, 1998.

[23] Vahid Dehdari and Dean S Oliver. Sequential quadratic programming for solving constrained production optimization–case study from Brugge field. *SPE Journal*, 17(03):874–884, 2012.

[24] Sy Do and Albert C. Reynolds. Theoretical connections between optimization algorithms based on an approximate gradient. *Computational Geosciences*, 17(6):959–973, 2013.

[25] Emilio Paulo dos Santos Sousa and Albert C Reynolds. Markov chain monte carlo uncertainty quantification with a least-squares support vector regression proxy. In *SPE Reservoir Simulation Conference*. Society of Petroleum Engineers, 2019.

[26] D. Eydinov, G. Gao, G. Li, and A. C. Reynolds. Simultaneous estimation of relative permeability curves and porosity/permeability fields by history matching production data. *J. Canadian Pet. Tech.*, pages 13–25, 2009.

[27] Dmitry Eydinov, Sigurd Ivan Aanonsen, Jarle Haukas, and Ivar Aavatsmark. A method for automatic history matching of a compositional reservoir simulator with multipoint flux approximation. *Computational Geosciences*, pages 209–225, 2008.

[28] Roger Fletcher and Sven Leyffer. A bundle filter method for nonsmooth nonlinear optimization. *University of Dundee, Report NA/195*, 1999.

[29] Roger Fletcher and Sven Leyffer. Nonlinear programming without a penalty function. *Mathematical programming*, 91(2):239–269, 2002.

[30] R. Fonseca, S. S. Kahrobaei, L. J. T. van Gastel, O. Leeuwenburgh, and J. D. Jansen. Quantification of the ensemble size on the quality of an ensemble gradient using principles of hypothesis testing. In *Proceedings of SPE Reservoir Simulation Symposium*, number SPE 173236, 2015.

[31] R. Fonseca, A. S. Stordal, O. Leeuwenburgh, P. Van den Hof, and J. D. Jansen. Robust ensemble-based multi-objective optimization. In *Proceedings of 14th European Conference on the Mathematics of Oil Recovery (ECMOR XIV), Catania, Italy, 8-11 September*, 2015.

[32] Rahul Fonseca, Olwijn Leeuwenburgh, Ernesto Della Rossa, Paul MJ Van den Hof, and Jan-Dirk Jansen. Ensemble-based multiobjective optimization of on/off control devices under geological uncertainty. *SPE Reservoir Evaluation & Engineering*, 18(04):554–563, 2015.

[33] Rahul Rahul-Mark Fonseca, Bailian Chen, Jan Dirk Jansen, and Albert Reynolds. A stochastic simplex approximate gradient (stosag) for optimization under uncertainty. *International Journal for Numerical Methods in Engineering*, 109(13):1756–1776, 2017.

[34] Saul Gass and Thomas Saaty. The computational algorithm for the parametric objective function. *Naval research logistics quarterly*, 2(1-2):39–45, 1955.

[35] Mitsuo Gen and Runwei Cheng. *Genetic algorithms and engineering optimization*, volume 7. John Wiley & Sons, 2000.

[36] Gene H. Golub and Urs Von Matt. Generalized cross-validation for large-scale problems. *Journal of Computational and Graphical Statisics*, 6(1):1–34, 1997.

[37] Zhenyu Guo and Albert C. Reynolds. Robust life-cycle production optimization with a support-vector-regression proxy. *SPE Journal*, 2018.

[38] Zhenyu Guo and Albert C Reynolds. Insim-ft in three-dimensions with gravity. *Journal of Computational Physics*, 380:143–169, 2019.

[39] Zhenyu Guo, Albert C Reynolds, and Hui Zhao. Waterflooding optimization with the insim-ft data-driven model. *Computational Geosciences*, pages 1–17, 2018.

[40] Jincong He and Louis J Durlofsky. Reduced-order modeling for compositional simulation by use of trajectory piecewise linearization. *SPE Journal*, 19(05):858–872, 2014.

[41] Obiajulu J Isebor and Louis J Durlofsky. Biobjective optimization for general oil field development. *Journal of Petroleum Science and Engineering*, 119:123–138, 2014.

[42] Obiajulu J. Isebor and Louis J. Durlofsky. A derivative-free methodology with local and global search for the constrained joint optimization of well locations and controls. *Computational Geosciences*, 18:463–482, 2014.

[43] Obiajulu J. Isebor, Louis J. Durlofsky, and David Echeverría Ciaurri. A derivative-free methodology with local and global search for the constrained joint optimization of well locations and controls. *Computational Geosciences*, 18(3):463–482, Aug 2014.

[44] Jan Dirk Jansen. Adjoint-based optimization of multi-phase flow through porous media. *Computers & Fluids*, 46:40–51, 2011.

[45] 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.

[46] H. Klie, W. Bangerth, M. F. Wheeler, M. Parashar, and V. Matossian. Parallel well location optimization using stochastic algorithms on the grid computational framework. In *Proceedings of 9th European Conference on the Mathematics of Oil Recovery*, pages 1–8, 2004.

[47] Drosos Kourounis, Louis J. Durlofsky, Jan Dirk Jansen, and Khalid Aziz. Adjoint formulation and constraint handling for gradient-based optimization of compositional reservoir flow. *Computational Geosciences*, 18(2):117–137, Apr 2014.

[48] J. F. B. M. Kraaijevanger, P. J. P. Egberts, J. R. Valstar, and H. W. Buurman. Optimal waterflood design using the adjoint method. In *Proceedings of the SPE Reservoir Simulation Symposium*, number SPE 105764, page 15, 2007.

[49] Jorge L Landa and Bars Güyagüler. A methodology for history matching and the assessment of uncertainties associated with flow prediction. In *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers, 2003.

[50] Gaoming Li and Albert C. Reynolds. Uncertainty quantification of reservoir performance predictions using a stochastic optimization algorithm. *Computational Geosciences*, 15(3):451–462, 2011.

[51] Ruijian Li, A. C. Reynolds, and D. S. Oliver. History matching of three-phase flow production data. *SPE Journal*, 8(4):328–340, 2003.

[52] Ruijian Li, Albert C. Reynolds, and Dean S. Oliver. Sensitivity coefficients for three-phase flow history matching. *J. Canadian Pet. Tech.*, 42(4):70–77, 2003.

[53] Xin Liu. *Multiobjective optimization of a waterflood with gradient-based algorithms*. PhD thesis, The University of Tulsa, 2016.

[54] Xin Liu and Albert C Reynolds. Gradient-based multiobjective optimization with applications to waterflooding optimization. In *ECMOR XIV-14th European Conference*

on the Mathematics of Oil Recovery, volume 2014, pages 1–21. European Association of Geoscientists & Engineers, 2014.

[55] Xin Liu and Albert C Reynolds. Augmented Lagrangian method for maximizing expectation and minimizing risk for optimal well-control problems with nonlinear constraints. *SPE Journal*, 21(05):1–830, 2016.

[56] Xin Liu and Albert C Reynolds. Gradient-based multiobjective optimization for maximizing expectation and minimizing uncertainty or risk with application to optimal well-control problem with only bound constraints. *SPE Journal*, 21(05):1–813, 2016.

[57] Yimin Liu, Wenyue Sun, and Louis J. Durlofsky. A deep-learning-based geological parameterization for history matching complex models. *Mathematical Geosciences*, 51(6):725–766, Aug 2019.

[58] Zhe Liu. Clustering-based robust optimization of smart wells in naturally fractured reservoirs. *Master Thesis*, University of Tulsa ,2016.

[59] Zhe Liu and Fahim Forouzanfar. Ensemble clustering for efficient robust optimization of naturally fractured reservoirs. *Computational Geosciences*, 22(1):283–296, 2018.

[60] Zhe Liu, Fahim Forouzanfar, and Yu Zhao. Comparison of SQP and al algorithms for deterministic constrained production optimization of hydrocarbon reservoirs. *Journal of Petroleum Science and Engineering*, 171:542–557, 2018.

[61] Zhe Liu and Albert C Reynolds. An SQP-filter algorithm with an improved stochastic gradient for robust life-cycle optimization problems with nonlinear constraints. In *SPE Reservoir Simulation Conference*. Society of Petroleum Engineers, 2019.

[62] R. J. Lorentzen, A. M. Berg, G. Nævdal, and E. H. Vefring. A new approach for dynamic optimization of waterflooding problems. In *Proceedings of the SPE Intelligent Energy Conference and Exhibition*, number SPE 99690, 2006.

[63] Ranran Lu, Fahim Forouzanfar, and Albert C Reynolds. An efficient adaptive algorithm for robust control optimization using StoSAG. *Journal of Petroleum Science and Engineering*, 159:314–330, 2017.

[64] R Timothy Marler and Jasbir S Arora. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6):369–395, 2004.

[65] Michael D McKay, Richard J Beckman, and William J Conover. Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.

[66] James Mercer. Functions of positive and negative type, and their connection the theory of integral equations. *Philosophical transactions of the royal society of London. Series A, containing papers of a mathematical or physical character*, 209(441-458):415–446, 1909.

[67] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, 2006.

[68] Diego Felipe Oliveira and Albert C. Reynolds. Hierarchical multiscale methods for life-cycle production optimization: A real field case study. *SPE Journal*, 18(5):896–907, October 2015.

[69] Dean S. Oliver and Yan Chen. Recent progress on reservoir history matching: a review. *Computational Geosciences*, 15:185–221, 2011.

[70] Dean S. Oliver, Albert C. Reynolds, and Ning Liu. *Inverse Theory for Petroleum Reservoir Characterization and History Matching*. Cambridge University Press, Cambridge, UK, 2008.

[71] Donald 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.

[72] L. Peters, R.J. Arts, G.K. Brouwer, C.R. Geel, S. Cullick, R.J. Lorentzen, Y. Chen, K.N.B. Dunlop, F.C. Vossepoel, R. Xu, P. Sarma, A.H. 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.

[73] Margarita Reyes-Sierra and CA Coello Coello. Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *International journal of computational intelligence research*, 2(3):287–308, 2006.

[74] Albert C. Reynolds, Ruijian Li, and Dean S. Oliver. Simultaneous estimation of absolute and relative permeability by automatic history matching of three-phase flow production data. *J. Canadian Pet. Tech.*, 43(3):37–46, 2004.

[75] Pallav Sarma, Khalid Aziz, and Louis J Durlofsky. Implementation of adjoint solution for optimal control of smart wells. In *proceedings of SPE Reservoir Simulation Symposium*. SPE, 2005.

[76] Pallav Sarma, Khalid Aziz, and Louis J Durlofsky. Implementation of adjoint solution for optimal control of smart wells. In *SPE reservoir simulation symposium*. Society of Petroleum Engineers, 2005.

[77] Pallav Sarma, Wen H Chen, Louis J Durlofsky, and Khalid Aziz. Production optimization with adjoint models under nonlinear control-state path inequality constraints. In *Intelligent Energy Conference and Exhibition*. Society of Petroleum Engineers, 2006.

[78] Mehrdad G Shirangi and Louis J Durlofsky. Closed-loop field development under uncertainty by use of optimization with sample validation. *SPE Journal*, 20(05):908–922, 2015.

[79] James C. Spall. Implementation of the simultaneous perturbation algorithm for stochastic optimization. *IEEE Transactions on Aerospace and Electronic Systems*, 34(3):817–823, 1998.

[80] Andreas S. Stordal, Slawomir P. Szklarz, and Olwijn Leeuwenburgh. A theoretical look at ensemble-based optimization in reservoir management. *Mathematical Geosciences*, 48(4):399–417, May 2016.

[81] Johan AK Suykens, Jos De Brabanter, Lukas Lukas, and Joos Vandewalle. Weighted least squares support vector machines: robustness and sparse approximation. *Neurocomputing*, 48(1-4):85–105, 2002.

[82] Jorn F. M. van Doren, Renato Markovinović, and Jan-Dirk Jansen. Reduced-order optimal control of water flooding using proper orthogonal decomposition. *Computational Geosciences*, 10(1):137–158, Mar 2006.

[83] Gijs Van Essen, Paul Van den Hof, and Jan-Dirk Jansen. Hierarchical long-term and short-term production optimization. *SPE Journal*, 16(01):191–199, 2011.

[84] G.M. van Essen, M.J. Zandvliet, P.M.J. Van den Hof, O.H. Bosgra, and J.D. Jansen. Robust waterflooding optimization of multiple geological scenarios. *SPE Journal*, 14(1):202–210, 2009.

[85] O. Volkov and D.V. Voskov. Advanced strategies of forward simulation for ajoint-based optimization. In *SPE Reservoir Simulation Symposium*, 2013.

[86] Chunhong Wang, Gaoming Li, and Albert C. Reynolds. Production optimization in closed-loop reservoir management. *SPE Journal*, 14(3):506–523, 2009.

[87] Zhan Wu. *Conditioning Geostatistical Models to Two-Phase Flow Production Data*. PhD thesis, The University of Tulsa, 1999.

[88] Zhan Wu, Albert C Reynolds, and Dean S Oliver. Conditioning geostatistical models to two-phase production data. *SPE Journal*, 1999.

[89] Xia Yan and Albert C. Reynolds. An optimization alorithm based on combining finite difference approximations and stochastic gradients. In *Proccedings of the SPE Reser-*

voir Simulation Symposium, The Woodlands, Texas, USA, 18-20 February, number SPE 163645, 2013.

[90] Ali A Yousef, Pablo Hugo Gentil, Jerry L Jensen, and Larry Wayne Lake. A capacitance model to infer interwell connectivity from production and injection rate fluctuations. In *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers, 2005.

[91] Lofti Zadeh. Optimality and non-scalar-valued performance criteria. *IEEE transactions on Automatic Control*, 8(1):59–60, 1963.

[92] Iskander S. Zakirov, Sigurd I. Aanonsen, Ernest S. Zakirov, and Boris M. Palatnik. Optimizating reservoir performance by automatic allocation of well rates. In *Proceedings of the 5th European Conference on the Mathematical Oil Recovery - Leoben, Austria, 3-5 September*, 1996.

[93] H. Zhao, C. Chen, S. Do, D. Oliveira, G. Li, and A. Reynolds. Maximization of a dynamic quadratic interpolation model for production optimization. *SPE Journal*, 18(6):1012–1025, 2013.

[94] Hui Zhao, Zhijiang Kang, Xiansong Zhang, Haitao Sun, Lin Cao, and Albert C Reynolds. A physics-based data-driven numerical model for reservoir history matching and prediction with a field application. *SPE Journal*, 21(06):2–175, 2016.

[95] Xiao Jian Zhou and Ting Jiang. Enhancing least square support vector regression with gradient information. *Neural Processing Letters*, 43(1):65–83, 2016.

[96] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2):173–195, 2000.