DERIVATIVE-FREE OPTIMIZATION FOR GENERALIZED OIL FIELD DEVELOPMENT

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF
ENERGY RESOURCES ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Obiajulu Joseph Isebor
June 2013

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Louis Durlofsky, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Khalid Aziz

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Tapan Mukerji

Approved for the Stanford University Committee on Graduate Studies.

Patricia J. Gumport, Vice Provost Graduate Education

This signature page was generated electronically upon submission of this dissertation in electronic format. An original signed hard copy of the signature page is on file in University Archives.

Abstract

Given the substantial costs and potential rewards associated with oil and gas field development and management, it is essential that these operations be performed as close to optimally as possible. In this work, we consider the computational optimization of general oil field development problems. Techniques are devised to simultaneously determine the optimal number and type of new wells, the sequence in which they should be drilled, and their locations and time-varying controls. The general optimization is posed as a mixed-integer nonlinear programming (MINLP) problem and involves categorical, integer-valued, and real-valued variables. Our new formulation handles bound, linear, and nonlinear constraints. The latter are treated using filter-based techniques. Noninvasive, derivative-free and easily-parallelizable approaches are applied for the optimizations. Methods considered include Mesh Adaptive Direct Search (MADS, a local pattern search method), Particle Swarm Optimization (PSO, a heuristic global search method), a PSO-MADS hybrid, and Branch and Bound (B&B, a rigorous global search procedure that requires relaxation of the categorical variables). The filter-based treatment of nonlinear constraints is extended to PSO and to the PSO-MADS hybrid.

Example cases involving well control optimization, joint well placement and control, and generalized full-field development problems are presented. The well control optimization example highlights the positive features of the filter constraint handling treatments. For the joint well placement and control optimization examples, the PSO-MADS hybrid is shown to consistently outperform the standalone MADS and PSO procedures. The joint optimization approach is also observed to provide superior performance relative to the sequential procedure, in which the well placement and well control problems are solved separately. For the field development optimization cases

that involve the optimization of well number in addition to well locations and control, there are two examples in which B&B is applied. In these examples, the PSO-MADS hybrid is shown to provide solutions comparable to those from the more established B&B approach, but at much lower computational cost. This is significant since B&B provides a near-exhaustive search in terms of the categorical variables. A full-field development example is presented where the number, type, drilling schedule, location and control of wells are optimized. This case demonstrates the broad capabilities of the new optimization framework.

We also implement an approach for field development optimization with two conflicting objectives. For this problem, a single-objective product formulation for the biobjective optimization procedure is applied. Three biobjective field development examples, including one that highlights the applicability of biobjective techniques to optimization under geological uncertainty, are presented. Our overall conclusion from this work is that, although they can be demanding in terms of computation, the optimization methodologies presented here appear to be applicable for realistic reservoir development and management problems.

Acknowledgments

First and foremost, I would like to thank God who has been my strength throughout my academic career. It is by His grace that I have been able to finish this graduate program. All glory and honor are therefore His.

I would like to express my most sincere gratitude to my academic and research adviser, Prof. Louis J. Durlofsky, for his patience, his attention to detail, his coaching and his endless enthusiasm and support. In addition to providing excellent direction and supervision of my work, he has become a role model to me in terms of having a great work ethic and displaying professionalism in all my endeavors. I feel very privileged to have been one of his students, and I eagerly look forward to maintaining a strong relationship with him in the future. I would also like to thank my defense committee, Profs. Khalid Aziz, Tapan Mukerji and Adam Brandt, for their insightful comments on this dissertation. Special thanks go to some of my mentors, both academic and industrial, who I have had the pleasure of working with in some capacity over the years. These include Profs. Hamdi Tchelepi and Roland Horne at Stanford University, Drs. David Echeverría Ciaurri and Andrew Conn at IBM, and Drs. Michael Litvak, Jerome Onwunalu and H. Scott Lane at BP.

I am indebted to all the professors, students and staff (teaching, research and administrative) of the Department of Energy Resources Engineering for providing the most conducive and challenging learning environment I have ever been a part of. Also, I am very grateful to the industrial affiliates of the SUPRI-B (Reservoir Simulation Research) and Smart Fields Consortia at Stanford University for financial support during my studies.

I consider myself very fortunate to be surrounded by great friends and colleagues who have supported me in ways that they might not even know. Of special mention are Uche Monu, Nii Okai Addy, Tayo Oguntebi, Folake Ogunbanwo, Chinyere Nwabugwu, Richard Mills Osuamkpe, Shola Olorunnipa, Michael Krause, Mark McClure, Charles Kang, Philip Brodrick, Jack Norbeck, Matthieu Rousset, Ekin Ozdogan, Cedric Fraces, and so many others. It has really been a blessing having all these people consider me as a friend, and I would like to acknowledge them all as having played a major role in my studies and my life and helping me cope with the rigors of graduate school.

Many thanks go to my family for their love and support throughout my academic career. I would especially like to thank my mother, Mrs. Esther Isebor, for her constant encouragement without which I certainly would not be where I am today, and my father, Dr. Joseph Isebor, for always reminding me to stay grounded, humble and to never lose sight of the things in life that are the most important. I would like to acknowledge the unending support from my uncle and aunt, Mr. Eworitsemogha and Mrs. Doris Mabiaku. They are my second parents and have been selfless in their treatment of me. Mere words do no justice to the level of gratitude I feel towards them. I would like to give special thanks to my girlfriend, Dr. Kelechi Christiana-Rose Obi, for the past year of constant encouragement, love and support.

I dedicate this dissertation to my brothers, Chukwudi and Nkemdilim Isebor. I love you guys!

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Optimization problems frequently arise in engineering applications when the goal is to find a set of design and/or control variables that minimize or maximize some function that measures the cost or merit of the design or control scheme. Typically, there are also constraints in these problems, which limit the values the variables can have. An important class of optimization problems, mixed-integer nonlinear programming (MINLP) problems, are characterized by nonlinear objective and constraint functions with both discrete and continuous variables. In the absence of discrete variables, the problems are referred to as nonlinear programming (NLP) problems.

Our interest here is in the design of an optimal development and production plan for a petroleum field. This optimal plan should maximize a measure of merit such as net present value (NPV) of the project or cumulative field production. Field development optimization constitutes a challenging MINLP problem, as it may include:

- different types of variables (categorical, integer and/or continuous) to define the number of wells, well types, drilling schedule, well locations and well controls;
- different types of constraints (bound and linear constraints on the optimization variables and nonlinear constraints on simulator output) to ensure physical and economic limits are enforced;
- uncertainty in the reservoir description, leading to optimization over multiple realizations;

- multiple conflicting optimization objectives, e.g., we may seek to maximize NPV while minimizing the water injected.

In addition to these complexities, field development optimization is in general computationally demanding since the objective function evaluation requires a reservoir simulation, which is expensive for large models.

The development of computational optimization procedures for oil field operations has been an area of active research in recent years. Optimization techniques have been developed for several types of field development and operational decisions, including the determination of the optimal type and location of new wells and the optimal operation of existing wells. Traditionally, the optimization of well location has been considered separately from the optimization of well operation (the latter is often referred to as well control optimization). However, recent work has demonstrated that, as would be expected, the optimal well location depends on how the well is to be operated. Thus, the optimization of well position and well control should be considered as a joint optimization problem, rather than as two separate (sequential) optimization problems.

In addition to well locations and controls, it is also of interest in field development planning to determine the optimal number of wells to drill, the type of wells, and their drilling schedule. Physical and economic constraints as well as the presence of uncertainty in the reservoir description should also be taken into account. In general, all of these issues must be considered in order to obtain truly optimal field development plans.

1.1 Literature Review

Most of the previous work in this general area has focused on the development of optimization approaches for the separate well control and well placement problems. There has, however, been some recent work on the solution of the joint problem, and a few studies have addressed multiobjective optimization for oil field problems. In this section we present a survey of relevant work in the areas of well control optimization, well placement optimization, joint well placement and control, optimization under

uncertainty, and multiobjective optimization in the petroleum field development and production optimization literature.

1.1.1 Well Control Optimization

The well control problem involves determining the optimal values for continuous operating variables, such as well rates or bottomhole pressures (BHPs), in order to maximize an objective such as NPV or cumulative oil produced. A wide variety of computational optimization techniques have been applied. These methods can be broadly classified as gradient-based or derivative-free approaches. We now discuss these two classes of procedures.

Gradient-based Approaches

Gradient-based methods that have been applied to the well control optimization problem include the steepest descent, conjugate gradient and sequential quadratic programming (SQP) methods [90]. These methods are efficient and known to exhibit fast convergence when the gradient of the objective function with respect to the control variables is available. Different methods for computing this gradient have been used and can be classified into three categories: numerical gradients computed using finite-difference (FD) techniques [2, 82, 121, 124], ensemble-based gradients computed using multiple realizations [25, 112], and adjoint techniques based on optimal control theory [20, 66, 100, 104, 105]. The advantage of using the numerical and ensemble-based gradients is that the simulator can be treated as a 'black box' with the gradients computed from function evaluations. The adjoint method is far more efficient, but requires access to and detailed knowledge of the reservoir simulator source code.

The literature on the use of gradient-based optimization methods for oil field problems includes the following studies. Aitokhuehi [2] and Aitokhuehi and Durlofsky [3] used the conjugate gradient and Levenberg-Marquardt algorithms with numerical gradients for the real-time optimization of smart wells. They coupled these optimizations with geological model updating. Carroll [24] demonstrated the use of Newton's method, also with numerical gradients, for the optimization of production systems. Kumar [73] applied the conjugate gradient method, with numerically computed gradients, to

optimize well settings with the goal of maximizing residually trapped CO_2 in geologic carbon sequestration. Wang et al. [120, 121] used SQP with numerical gradients to solve the production optimization problem, and Litvak et al. [82] applied the method to the Prudhoe Bay E-field. Yeten et al. [124] used a conjugate gradient method for the optimization of smart well controls. Wang et al. [118] compared the steepest ascent method with FD gradients, simultaneous perturbation stochastic approximation (SPSA [110]) gradients, and ensemble-based gradients for a production optimization problem. They showed that the steepest ascent method with FD gradients was the most efficient of the three methods, if a small perturbation size was used. Chaudhri et al. [25] demonstrated the use of an improved ensemble-based gradient approach for production optimization using the conjugate gradient method. The use of numerical gradients decreases the efficiency of these gradient-based optimization methods because the number of function evaluations (simulations) typically scales with the dimension of the parameter space, i.e., with the number of optimization parameters. These methods do, however, parallelize naturally since the various simulations can be performed simultaneously on separate processors.

The use of adjoint procedures, which derive from optimal control theory, reduces computational requirements by efficiently providing the gradient of the objective function with respect to the controls. These gradients are obtained by solving an adjoint problem that uses the Jacobian matrices and other information computed during the forward simulation. Sarma et al. [104, 105] and Jansen and Brouwer [20, 67] have demonstrated the effectiveness of adjoint procedures for production optimization and closed-loop reservoir management. Earlier investigators who applied optimal control theory for enhanced oil recovery include Ramirez and co-workers [83, 87, 100], Asheim [6] and Virnovsky [116]. Adjoint-based procedures have also been widely applied for history matching; see, e.g., Li et al. [80].

The preceding authors did not include consideration of nonlinear constraints. Sarma et al. [104, 107] described a feasible direction algorithm for handling nonlinear path constraints. The nonlinear inequality constraints in production optimization are path constraints because they need to be satisfied at every time step during the reservoir simulation. Sarma distinguished between two types of algorithms for dealing with

nonlinear path constraints: algorithms that solve the path constraints implicitly together with the dynamic system (presupposes full access to the forward model or simulator), and algorithms that calculate the path constraints explicitly after the dynamic system has been solved. The algorithm in [104] and [107] is a constraint handling method of the first type. It entails a feasible direction algorithm that uses lumped constraints and a feasible line search. Some iteration to achieve feasible controls at each time step of the forward simulation is also required. Sarma et al. [104, 107] presented promising results for certain types of nonlinearly constrained problems, but the method does not yet handle all types of relevant nonlinear constraints. The nonlinear constraint handling techniques presented in this work are of the second type, which treat the simulator as a black box. This enables us to handle any type of nonlinear constraint that can be put in a functional form.

Derivative-free Approaches

Gradient-free methods do not require the calculation of objective function gradients, as they use just the objective function values obtained from function evaluations (reservoir simulation plus economic model evaluation for calculating NPV). Derivative-free methods can be further classified into deterministic (e.g., polytope or Nelder-Mead simplex method [74, 89]) and stochastic methods, including genetic algorithms (GAs) and tabu search. Carroll [24] used the polytope method for production systems optimization and showed that it is an effective alternative to gradient-based methods. Cullick et al. [31] used a global optimizer based on tabu search to optimize production strategies while accounting for risk. Harding et al. [59] used a GA with problem-specific crossover operators to select optimal well rates for the maximization of NPV. Almeida et al. [4] used the GA for production optimization with smart wells. They also accounted for technical uncertainties, such as the risk of valve failure. It should be noted that none of the works discussed above addressed nonlinear constraint handling.

There are many other derivative-free methods that have been applied in other contexts. Of particular note are the so-called direct search methods, which are stencil or frame-based optimization techniques. Kolda et al. [71] and Conn et al. [30] presented excellent reviews of these methods, including their history and convergence theory.

Direct search algorithms have been in existence for many years. Hooke and Jeeves [60] introduced their direct search method in 1961. Audet and Dennis [7] presented a more general version of pattern search methods (called Generalized Pattern Search) and Marsden et al. applied these methods to optimization problems in aeroacoustics [85] and medicine [84]. For the well control optimization problem, Isebor [63] and Echeverría et al. [40, 41] applied both gradient-based and derivative-free methods with different nonlinear constraint handling techniques. They provided comparisons between the different methods and showed that the use of pattern search methods with filter constraint handling is an effective approach for well control optimization problems.

1.1.2 Well Placement Optimization

The well placement optimization problem involves maximizing an objective function by varying well types (e.g., injector or producer, vertical or horizontal) and well locations. Due to the effects of reservoir heterogeneity, these problems can display very rough (discontinuous) optimization surfaces, with multiple local minima [94]. Well placement problems are therefore frequently solved using stochastic search procedures (which avoid getting trapped in some local optima), though some investigators have applied gradient-based approaches to this problem.

Gradient-based Approaches

Gradient-based methods applied to the well placement problem include approaches that calculate gradients using SPSA (Bangerth et al. [15]), ensemble-based techniques (Leeuwenburgh et al. [77]), and adjoint methods (Sarma and Chen [106], Zandvliet et al. [127] and Zhang et al. [128]). In the SPSA algorithm [110], starting from a current iterate in some iteration, the required gradient is approximated by first generating a random direction from the current iterate. This random direction is used to generate two new points at which the objective function is evaluated. Using these function evaluations, a direction of decrease (for minimization problems) can be determined. Unlike finite-difference methods, which provide more accurate gradients, the SPSA gradient computation is independent of the number of variables in the optimization

problem as only two function evaluations are required (as opposed to $O(n)$ evaluations, where n is the number of optimization variables, for FD). Bangerth et al. [15] compared gradient-based optimization methods, with gradients computed using FD and SPSA, with some derivative-free optimizers, including a genetic algorithm, fast simulated annealing and the Nelder-Mead simplex algorithm. They found that the SPSA algorithm outperformed the other methods. The SPSA algorithm does entail some challenges, however, including the choice of the step size for calculating new solutions. Leeuwenburgh et al. [77] optimized well placement using gradients calculated from ensemble-methods, as was done for well control problems in [25, 112].

Several authors have applied adjoint-based gradients for well placement optimization [106, 127, 128]. The main advantage over FD or SPSA methods is that the adjoint-based gradient methods accurately compute improving directions for all wells in only one forward (reservoir) and one backward (adjoint) simulation. In order to use these techniques, the optimization problem with discrete well placement variables is replaced with an equivalent problem with continuous variables from which a gradient can be calculated. For problems with two-dimensional reservoir models, Zandvliet et al. [127] used an approach that places a 'pseudowell' in the eight grid blocks that surround the well whose location is to be optimized. These pseudowells produce or inject at a very low rate and thus have a negligible influence on the overall flow. Using an adjoint method, Zandvliet et al. [127] calculated the gradients of NPV with respect to the rates of the pseudowells over the lifespan of the reservoir. Subsequently, these gradients were used to approximate an improving direction (one of eight possible directions, based on the surrounding grid blocks) in which to move the wells to achieve an increase in NPV.

Sarma and Chen [106] presented a technique similar to that of Zandvliet et al. [127]. In their method, potential well locations are surrounded by pseudowells whose geometric well indices are weighted by spatial integrals of a continuous (Gaussian) function, which itself is a function of the areal locations of the original wells. Due to the continuous functional relationship, adjoints and standard gradient-based optimization algorithms can be applied to optimize the well locations. The main advantage of this approach compared to that of Zandvliet et al. [127] is that the actual gradient with

respect to the well location variables is obtained (as opposed to gradients with respect to pseudowell rates), and thus any arbitrary search direction is possible. By contrast, the method of Zandvliet et al. [127] is limited to the eight directions associated with neighboring grid blocks.

Derivative-free Approaches

As noted above, gradient-based techniques have been applied to the well placement problem. However, due to the effects of reservoir heterogeneity, these problems can display very rough optimization surfaces, with multiple local optima. For this reason they are more commonly solved using derivative-free stochastic search procedures. The most popular of these are evolutionary strategies such as GAs, which have been used to optimize the locations of vertical wells [13, 15, 56, 96, 114] and nonconventional wells [5, 92, 103, 125]. Another evolutionary strategy that has received some attention is the covariance matrix adaptation evolution strategy (CMA-ES) [19], which has been shown to give superior performance when compared with GAs for this problem. Particle Swarm Optimization (PSO), which mimics the social interaction patterns of animal groups such as flocks of birds or swarms of fish, has also been used for well location optimization. Onwunalu and Durlofsky [94] presented applications of PSO to the optimization of vertical and nonconventional well placement. In their comparisons, PSO was found to outperform GA for these problems. Another stochastic search family of note is simulated annealing (SA). These methods have been applied for optimal well scheduling and placement by Beckner and Song [16].

In the preceding studies, the authors sought to optimize the locations of individual wells. For large-scale problems with tens to hundreds of wells, this approach becomes increasingly expensive using stochastic search methods. Therefore, for optimizing well configuration in large-scale problems, well pattern optimization procedures have been devised [81, 93, 95, 97, 98]. Pan and Horne [98] and Ozdogan et al. [97] approached large-scale well placement by optimizing well spacing with fixed well patterns. Onwunalu and Durlofsky [93, 95] performed large-scale well placement by optimizing the type and orientation of well patterns, and Litvak and Angert [81] applied these ideas to actual (giant) oil fields. In this thesis, we do not consider the optimization of well patterns as all of our example cases have no more than ten wells, but the methods

noted above could be incorporated into our general optimization framework.

As previously noted, because the optimization surfaces for well placement problems are rough, these problems are commonly approached using stochastic search procedures such as GAs. However, the use of varying amounts of local optimization hybridized with stochastic search procedures has been applied in order to accelerate the convergence of the stochastic search methods. For example, Yeten et al. [125] applied GA hybridized with a local hill climber algorithm to optimize the type and location of nonconventional wells. They demonstrated that the hybrid approach provides advantages relative to the use of only GA. Other authors have hybridized GA with the local polytope algorithm (see, e.g., Badru and Kabir [13], Guyaguler et al. [57], and Ozdogan and Horne [96]). In Chapter 2 we present our hybrid approach which combines the global stochastic PSO method with the local Mesh Adaptive Direct Search (MADS) method. Our hybrid procedure is similar to the PSwarm technique of Vaz and Vicente [115].

1.1.3 Joint Well Placement and Well Control

When well placement optimization is performed, various treatments can be used for the well controls, including fixed rates or bottomhole pressures, or the so-called 'reactive' control approach. In the latter case, production wells are closed (shut in) when the well water cut exceeds a specified limit [127]. Recent work has demonstrated that, as would be expected, the optimal well locations depend on how the wells are to be operated [17, 78, 127]. Thus, the optimization of well position and well control should be considered as a joint optimization problem, rather than as two separate problems, because a purely sequential approach will in general be suboptimal.

There have been some studies on joint well placement and control, which we now discuss. Wang et al. [117] and Forouzanfar et al. [51] applied gradient-based methods for optimizing well controls. They included an injection or production well in every grid block of the reservoir model and then eliminated wells that did not meet certain criteria, with the intent of determining the number of wells, their locations and controls. Their use of gradient methods might result in convergence to local optima, especially for a sizeable problem with many grid blocks. Their approach precludes the use of

derivative-free methods because the inclusion of a well in every grid block will lead to far too many variables for derivative-free techniques. It is also important to note that these studies did not explicitly define the well location variables but approached the joint well placement and control problem heuristically, by solving only a control problem.

Other investigators considered combined well placement and control using formulations that explicitly define variables for the different parts of the problem. Li and Jafarpour [78] applied a sequential iterative procedure for the combined problem in which they alternated between the well placement and well control optimization problems. They used a coordinate descent random search method for the well placement part and a gradient-based procedure for the well control optimization. Bellout et al. [17] considered the two optimization problems in a nested fashion, where the objective function for well placement entails optimization of the control problem. They used derivative-free direct search methods for the well placement optimization and an adjoint-based gradient procedure for the well control problem. Both of these studies demonstrated that improved results can be achieved by solving the joint optimization problem compared to a sequential optimization procedure (with an assumed well control strategy during well placement optimization).

Humphries et al. [62] also compared the use of sequential and joint optimization techniques for well placement and control. In contrast to the studies noted above, they did not find the joint approach to consistently provide superior results relative to their sequential procedure. This may be due to the fact that they included some heuristics for well control in their well placement optimizations, or to some other algorithmic treatments. Humphries et al. [62] applied a modified version of the PSwarm hybrid procedure [115] for their optimizations, which is similar to the hybrid PSO-MADS technique developed in this work. Cameron and Durlofsky [21] implemented a Hooke-Jeeves direct search for the joint optimization of well placement and control for geological carbon storage problems. Comparisons with a sequential procedure were not performed in that work.

As discussed earlier, the generalized field development problem (which involves determining the optimal number of wells, their type and drilling sequence, in addition

to well location and control) is a MINLP problem with categorical, integer and continuous variables. There have been some previous studies that have approached parts of the field development problem by including different types of variables. Gunnerud and Foss [55] approached oil production optimization using mixed-integer linear programming (MILP). Kosmidis et al. [72] and Litvak et al. [82] applied mixed-integer nonlinear programming (MINLP) for well scheduling and for optimizing a production system, respectively. In [55, 72, 82], simplified models of the reservoir and facilities were used in place of complex simulation models. Echeverría et al. [39] used MINLP formulations both for selecting wells from a given set and simultaneously optimizing well controls subject to general constraints, and for optimizing well type and locations. The MINLP methods they considered include an open-source software package for general gradient-based optimization (see Bonami et al. [18]) that is supported by convergence theory and is compatible with simulation-based optimization problems. These methods, however, unlike the derivative-free techniques applied in this work, require gradient information. As noted earlier, in many cases gradients are not available, and to construct them using an adjoint procedure requires access to (and detailed knowledge of) simulator source code.

A variety of constraints (bound, linear and nonlinear) enter the field development optimization problem. Even though previous investigators incorporated some amount of constraint treatment, none of the studies discussed above handled the full set of constraints that can appear in these problems. For the generalized field development and control problem, in addition to satisfying bound constraints on BHP ranges (which arise when BHPs are the continuous well control variables), we also need to satisfy practical constraints including well-pair distance constraints, well rate and facility constraints, water cut constraints, and constraints on the number of wells. The most common methods for handling these types of constraints are to simply discard infeasible points from consideration [81] or to use penalty methods [40, 57, 63]. In our work, we introduce a general treatment based on filter methods. In our earlier work, this approach was shown to provide comparable or better performance than penalty methods for the well control problem [40, 63].

1.1.4 Optimization Under Geological Uncertainty

The reservoir models used for field development planning have several sources of uncertainty, including geological uncertainties in property distributions (e.g., permeability), fluid property uncertainties, etc. It is of interest to incorporate these uncertainties in the optimization process in a manner that provides solutions that are robust given the uncertainties. This entails optimization over multiple realizations (which characterize the uncertainty) where the goal is to optimize a robust performance measure such as the expected value of NPV.

Yeten et al. [123] optimized well controls, incorporating the risk of inflow control valve failure, over five geological realizations. Aitokhuehi and Durlofsky [3] and Sarma [104] also optimized well controls over a set of realizations in their optimization and model updating procedures. Several authors have performed well placement optimization over multiple realizations with the goal of maximizing expected NPV. For example, Guyaguler and Horne [56] and Yeten et al. [125] used hybrid GAs, Onwunalu and Durlofsky [94] applied PSO, and Nwankwor et al. [91] used a hybrid differential evolution and PSO method. Cameron and Durlofsky [22] addressed joint well placement and control optimization, but for geologic carbon storage problems. They used the PSO algorithm to minimize the expected value of injected CO_2 mobility over a set of realizations by optimizing the placement and injection rates of horizontal CO_2 injection wells. In our work, we mostly consider optimization over a single realization, but the techniques developed here can be applied to optimizing over multiple realizations in order to incorporate geological uncertainty.

Optimization over multiple realizations is computationally expensive. However, the larger the number of realizations, the better we expect the characterization of the uncertainty to be. Therefore it is important to have techniques that reduce the computational expense in these problems by properly selecting a set of representative realizations, or to have methods that reduce the number of simulations required for a set of realizations. Artus et al. [5] used statistical proxies based on cluster analysis to reduce the number of models that required full simulation. They applied GA and their proxy-based technique to optimize the placement of monobore and dual-lateral wells and demonstrated that by simulating 10 to 20% of the scenarios, they obtained

results close to those achieved by simulating all cases. Wang et al. [119] applied a Retrospective Optimization (RO) framework that solves a sequence of optimization subproblems using an increasing number of realizations. They showed that, by using the RO procedure with cluster sampling, the computational expense of well placement under uncertainty was reduced by about an order of magnitude compared with optimizing using the full set of realizations at all iterations. Li et al. [79] performed joint well location and control optimization under uncertainty. They reduced the number of computations required by using a randomly selected subset of realizations at each iteration to approximate the expected NPV over all realizations. The procedures discussed here for reducing the computational demands in optimizations with multiple realizations are, again, compatible with the optimization framework developed in this work.

1.1.5 Multiobjective Optimization

Typically in engineering problems, a goal is to resolve conflicting objectives and obtain a design that represents an acceptable balance of all objectives. In practical field development problems, decisions may depend on multiple conflicting criteria. For example, in a waterflood we might seek to maximize oil recovery while minimizing water production, which are conflicting objectives. There is a clear trade-off in such a case, and the field of multiobjective optimization enables the solution of problems of this nature. In general there is no single solution to a multiobjective optimization problem, but rather a set of optimal solutions that define a Pareto front, which represents the optimal trade-off between the different objectives [101].

There are two broad categories of multiobjective optimization methods, depending on the goal of the decision maker. The first set of methods requires prior articulation of designer preferences. These approaches then solve the problem to find a single point on the Pareto front after some aggregation of the objectives, depending on the specified preferences. These methods require assigning weights or goals to each objective function to obtain a single aggregate objective. They have the advantage of using existing single objective optimization algorithms to approach problems with multiple objectives. These approaches include min-max formulations, weighted sums,

nonlinear approaches [111], goal programming [69] and the lexicographic method where objectives are ranked in order of importance and optimized in sequence with other objectives set as constraints [111].

The second set of methods require a posterior articulation of designer preferences. These techniques solve the multiobjective problem to generate an approximation to the Pareto front, using single or multiple optimization runs. These methods include weighted sums [34, 70], where the weights of the objectives are varied such that multiple points in the Pareto front can be generated; the widely used population-based or evolutionary methods that attempt to sample the entire front (using GA [37], PSO [29], SA [14], etc.); normal boundary intersection [35], which attempts to generate equally distributed points along the Pareto front using a sequence of optimizations; and recent ideas on the use of direct search methods to solve a sequence of single objective optimization problems. These approaches generate progressively improving Pareto approximations [10]. There are also recently devised direct search methods that attempt to sample the entire front in a singe optimization run [32]. Later in this dissertation, we present results comparing the weighted sums approach [34] and the approach for multiobjective optimization through a series of single-objective formulations [10], for field development and production problems involving two conflicting objectives.

Applications of multiobjective optimization in the petroleum engineering literature are most common in the area of history matching, where different algorithms have been used to minimize different measures of misfit. The algorithms for multi-criteria history matching include multiobjective versions of Differential Evolution (DE) [26, 58], PSO [26, 88] and GAs [48, 109]. The applications of multiobjective optimization approaches to field development optimization include the following. Gross [54] used a decision analysis framework to determine the optimal number of producer wells and gas plateau production rates in order to maximize the recovery factor and the duration of plateau rate production. In another case he optimized the number of producers and injectors, their plateau rates and minimum producer BHPs in order to maximize discounted NPV, estimated ultimate recovery (EUR) and breakthrough times. Awotunde and Sibaweihi [11] used the weighted sum approach to perform

multiobjective optimization, with the covariance matrix adaptation evolution strategy (CMA-ES) and differential evolution (DE) algorithms as the single objective optimizers. They applied their approach to optimize the location of vertical wells with the objectives of maximizing NPV and voidage replacement ratio. None of the preceding studies considered multiobjective optimization with general constraints, as will be considered in Chapter 6 of this dissertation.

1.2 Scope of Work

Our intent in this work is to develop and apply new procedures for the generalized field development optimization problem. The underlying optimization techniques considered are noninvasive (with respect to the simulator) derivative-free methods that naturally parallelize. Noninvasive (also known as black-box) methods are required when one does not have access to simulator source code. One of the motivations for using derivative-free methods is that the different types of variables (particularly the categorical variables) in the problem are not well suited for gradient-based optimization methods. In addition, the objective function surfaces to be optimized can be rough (with discontinuities) and in general display multiple local optima. This suggests the use of global search techniques, which are derivative-free in nature. Echeverría et al. [41] illustrated the applicability of noninvasive derivative-free optimization methods to well control, well placement and history matching problems.

Our solution of the generalized field development problem entails the use of Mesh Adaptive Direct Search (MADS), which is a local optimization technique with established convergence theory, and Particle Swarm Optimization (PSO), a stochastic global search procedure. A hybrid PSO-MADS procedure that combines the positive features of these two methods will be developed and tested. The methodology presented in this work also includes a treatment for general (nonlinear) constraints in the optimization. These are handled using a filter approach, as described in [40], in which the constrained problem is essentially viewed as a biobjective optimization where the two objectives are the maximization of NPV (for example) and the minimization of the aggregate constraint violation. We also introduce problem formulations that include drill/do not drill binary variables, or drill an injector/drill a producer/do not

drill ternary variables, that are associated with each well. This allows the procedure to add or eliminate wells and thus optimize the number of wells, subject to a specified maximum, along with the well type (producer or injector). Finally, we consider biobjective optimization subject to general constraints.

The key research objectives for this work are:

- Develop a general framework for addressing MINLP problems with categorical, integer and continuous variables. Different methods, including MADS, PSO and a PSO-MADS hybrid, will be implemented and compared with a traditional MINLP method, Branch and Bound (B&B), for use as the optimization engine in our framework. General constraint handling approaches will be incorporated in the framework.

- Highlight the capabilities of the novel PSO and PSO-MADS filter-based constraint handling procedures for well control optimization problems.

- Evaluate the performance of the optimization framework for the joint solution of the well placement and control optimization problem. The optimization problem at this stage is NLP rather than MINLP, as we do not include any categorical variables. The proposed joint approach will be compared to more common sequential treatments in order to investigate the importance of coupling between the well placement and well control problems.

- Address the full petroleum field development problem, stated as a true MINLP problem, where in addition to well location and control, the number of wells, the well types and their drilling schedule are also optimized. Compare the performance of the various algorithms, including B&B, for this problem.

- Develop and test an approach for performing field development optimization where we have two conflicting objectives together with general constraints. Evaluate the biobjective optimization implementation for a variety of cases, including those with geological uncertainty.

1.3 Dissertation Outline

This research entails the development and implementation of optimization methods for MINLP problems and application to generalized petroleum field development optimization. In Chapter 2 we pose the general optimization problem and describe the underlying derivative-free optimization methods used in our solution framework: Mesh Adaptive Direct Search (MADS) [9, 76], Particle Swarm Optimization (PSO) [38] and the PSO-MADS hybrid method, which is a variant and extension of the PSwarm algorithm introduced by Vaz and Vicente [115]. We also describe filter-based techniques for handling nonlinear constraints for each of these different methods. The description of a traditional MINLP solution approach, Branch and Bound (B&B), is also presented. This technique is used to provide reference solutions when we compare results from MADS, PSO, and PSO-MADS for field development problems.

In Chapter 3, we consider a generally constrained well control problem with only continuous variables to demonstrate the performance of the MADS, PSO and PSO-MADS algorithms with their respective filter-based constraint treatments. The results from this example illustrate the advantages of the filter constraint treatment and show that it provides quantitative information on the sensitivity of the optimal solution to the specified constraints. In addition to an optimized control scheme, the constraint sensitivity information can be very useful to decision makers for sizing facilities, determining the most restrictive constraints, etc.

In Chapter 4 we present the application of our framework to the joint well placement and well control problem. The optimization problem at this stage includes integer well location and continuous well control variables. We also present results using the more common sequential approach for well placement and well control, where the two problems are addressed separately. Optimization results for example cases with increasing complexity are shown. We consider a case with a two-dimensional reservoir model and only bound constraints, a case with the same model but that additionally includes nonlinear constraints, and finally a nonlinearly constrained case in which we also determine the optimal number of wells. Our results demonstrate that the PSO-MADS hybrid approach is consistently superior to standalone PSO or MADS for optimization of the joint problem, and that the joint approach has clear

advantages over the sequential procedure. The last example provides a motivation for extending our formulation to handle categorical variables in order to determine the number of wells, a topic that is fully studied in Chapter 5.

Chapter 5 presents the application of the MADS, PSO, PSO-MADS hybrid, and B&B approaches to the full field development and production problem, where we optimize the number of wells, well type and drilling schedule, in addition to the well location and controls. We describe our treatments for binary and ternary categorical variables within the MINLP framework. We then present two approaches for the categorical variable relaxations that are necessary for the B&B approach. The first two example cases involve constrained field development and control optimization of channelized fields with varying degrees of heterogeneity. These cases utilize the binary categorical MINLP formulation. The results from these cases again display the superiority of the PSO-MADS hybrid over its component methods. They also demonstrate that PSO-MADS, with our extensions for handling categorical variables, provides solutions that are comparable to the B&B approaches, but with much less computational effort. The third case involves a three-dimensional reservoir model and includes the optimization of both vertical and horizontal well locations. The fourth case utilizes the ternary MINLP formulation. In this example, we demonstrate the applicability of our framework to the solution of the full field development and production problem.

In Chapter 6, we present formulations for performing biobjective optimization and apply them to field development problems with two conflicting objectives and general constraints. The approaches considered include the linear weighting method (LWM) and the approach for biobjective optimization through a series of single-objective optimizations, as formulated by Audet et al. [10]. Audet et al. utilized MADS in their formulations, while we apply PSO-MADS in this work (for a framework we term BiPSOMADS). We will demonstrate that we achieve superior results compared to the BiMADS of Audet et al. [10] and the linear weighted approach, by which we mean that BiPSOMADS provides better approximations of the Pareto front.

Chapter 7 concludes this dissertation. In addition to summary and conclusions, we also provide recommendations for future work. These recommendations include improving the efficiency of the optimization framework, improving the treatments for optimization under uncertainty, and creating a multiobjective framework that can handle more than two objectives.

Our newly developed PSO with filters and PSO-MADS hybrid (described in Chapter 2), and some of the joint well placement and control optimization results presented in Chapter 4, have been published in [64]. The generalized field development results presented in Chapter 5 have been published in [65].

Chapter 2

Derivative-free Optimization Approaches

In this chapter, we present the general optimization problem and then provide descriptions of the derivative-free optimization methods used to solve it. These include Mesh Adaptive Direct Search (MADS, a local pattern search method) [9, 76], Particle Swarm Optimization (PSO, a global search method) [38], and our new PSO-MADS hybrid method, a variant of the PSwarm algorithm introduced by Vaz and Vicente [115], that combines the positive features of the two preceding methods. We also describe filter techniques for handling nonlinear constraints for each of these different methods. The description of a traditional MINLP solution technique, Branch and Bound (B&B), is also presented. This technique is used to provide reference solutions for MINLP example cases in Chapter 5.

2.1 Problem Statement

We now present the optimization problem for generalized field development and production, which can be posed as a mixed-integer nonlinear programming (MINLP)

problem. This MINLP problem (P) can be stated as follows:

$$(P)\begin{cases} \min\limits_{\mathbf{u}\in U,\ \mathbf{v}\in V,\ \mathbf{z}\in\mathbb{Z}^{n_c}} & f(\mathbf{p},\mathbf{u},\mathbf{v},\mathbf{z}), \\ \text{subject to} & \mathbf{g}(\mathbf{p},\mathbf{u},\mathbf{v},\mathbf{z}) = \mathbf{0}, \\ & \mathbf{c}(\mathbf{p},\mathbf{u},\mathbf{v},\mathbf{z}) \leq \mathbf{0}, \end{cases} \tag{2.1}$$

where f is the objective function to be optimized (e.g., to maximize net present value, NPV, $f = -\text{NPV}$), $\mathbf{c} \in \mathbb{R}^m$ is the vector of m nonlinear constraint functions (e.g., well water cuts, facilities constraints such as field-level injection limits, etc.), and the system of equations $\mathbf{g} = \mathbf{0}$ refers to the reservoir simulation equations that must be solved to evaluate f and $\mathbf{c}$. This constraint ensures that the governing flow equations are satisfied. It is enforced by using a reservoir simulator in the objective function evaluation. In this work, we use Stanford's General Purpose Research Simulator (GPRS) for all our simulations. Refer to [23, 68, 129] for details about GPRS, and to [12, 53, 42] for general discussions of reservoir simulation.

The vector $\mathbf{p}$ in (P) represents the dynamic state variables of the model (pressure p and saturation S at all grid blocks in an oil-water problem). The bounded sets $V = \{\mathbf{v} \in \mathbb{Z}^{n_1};\ \mathbf{v}_l \leq \mathbf{v} \leq \mathbf{v}_u\}$ and $U = \{\mathbf{u} \in \mathbb{R}^{n_2};\ \mathbf{u}_l \leq \mathbf{u} \leq \mathbf{u}_u\}$ define the allowable values for the discrete well placement variables $\mathbf{v}$ (the set V could define possible drilling locations within the reservoir) and continuous well control variables $\mathbf{u}$ (the set U could define allowable BHP ranges). There are n_1 optimization variables associated with well locations and n_2 optimization variables associated with well controls.

Even though the actual well locations are real-valued, the well placement variables are modeled as integers because most simulators, including GPRS, require well locations to be defined in terms of discrete grid blocks. In all but one of our examples, the wells are assumed to be vertical, with locations stated in terms of discrete areal (x, y) coordinates. Thus, we have $n_1 = 2(N_I + N_P)$, where N_I and N_P are the number of injection and production wells, respectively. In more general cases involving deviated or multilateral wells, additional variables are required to describe well locations/trajectories. The well controls are represented by piecewise constant functions

in time with N_t intervals. Thus, $n_2 = N_t\,(N_I + N_P)$. The vector $\mathbf{z} \in \mathbb{Z}^{n_c}$ represents the vector of n_c categorical variables that make (P) a true MINLP problem. In Chapter 5, we present formulations that include binary and ternary categorical variables.

Reservoir models are characterized by many uncertain parameters, perhaps most notably the geological property distributions. In order to incorporate geological uncertainty in the optimization process, the objective function f is not based on the response of one model but on the responses of a set of models that characterize the uncertainty. In this case the objective function could be the expected value of NPV over all models considered, i.e., $f = -\mathrm{E}[NPV]$. This implies that in light of the geological uncertainty, we wish to optimize the expected return from the reservoir development project. Here we focus mainly on optimizing with a single geological model, though we do include an example case of optimization over multiple models. It is important to note that the procedures described in this chapter can be applied for optimization over an arbitrary set of geological models, where we minimize, for example, $f = -\mathrm{E}[NPV]$, but the number of simulations required will increase substantially.

We focus on derivative-free optimization techniques. These are an important class of optimization methods that are applicable to problems where gradients are either not available, difficult/expensive to obtain, or cannot be properly defined, as is the case in the generalized field development problem. Echeverría et al. [41] have illustrated the applicability of such methods to problems in oil field development and operation.

In the next two sections, the two derivative-free optimization techniques used in this work are discussed. We then present a hybrid approach based on the two methods. In order to streamline the descriptions in these sections, we express the general optimization problem presented in (2.1) in the following simplified form:

$$\begin{aligned} &\min_{\mathbf{x}\in\Omega} \; f\,(\mathbf{x})\,, \quad \text{subject to} \quad \mathbf{c}\,(\mathbf{x}) \leq \mathbf{0}, \\ &\text{with} \quad \Omega = \{\mathbf{x} \in \mathbb{R}^n \;:\; \mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u\}. \end{aligned} \tag{2.2}$$

Here we do not write the explicit dependence of f and $\mathbf{c}$ on the dynamic states $\mathbf{p}$

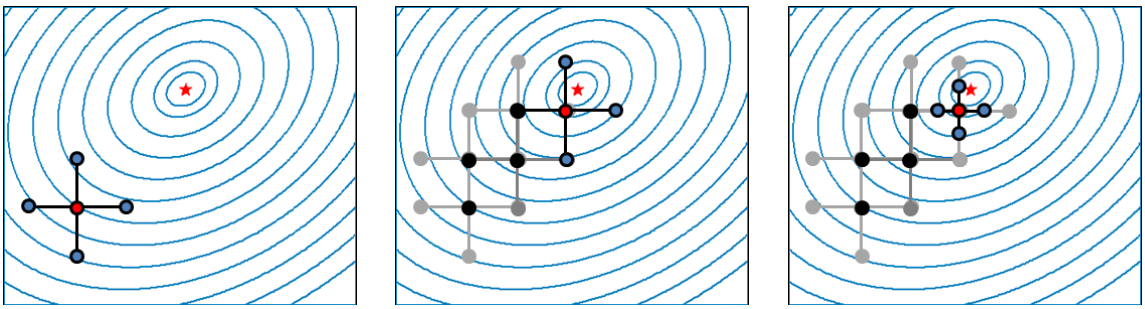
because, given a set of locations and controls, the dynamic states are determined by performing a reservoir simulation (meaning $\mathbf{g} = \mathbf{0}$) for the function evaluation. Assuming that the categorical variables can be treated as simple integer variables, we note that the vector $\mathbf{x}$ now contains both discrete and continuous variables, i.e., we use $\mathbf{x}$ here in place of $(\mathbf{u}, \mathbf{v}, \mathbf{z})$ in (2.1) to simplify the presentation.

The dimension of $\mathbf{x}$ varies with the number of wells and control periods and, for the examples considered, is less than 100. The derivative-free methods presented here may not yet be suitable for field development problems with many hundreds or thousands of optimization variables because the computational expense, which scales with the number of variables, would be excessive for a current (typical) computer cluster. However, as cluster sizes increase in coming years, the use of these methods for problems with more optimization variables will become feasible. We now describe the Mesh Adaptive Direct Search and Particle Swarm Optimization methods.

2.2 Local Derivative-free Method: Mesh Adaptive Direct Search

2.2.1 MADS Method Description

Pattern search algorithms are a family of optimization methods based on polling, which is the local exploration of the objective function surface around the current iterate. Polling is illustrated in Figure 2.1 and proceeds as follows. At any iteration k, a polling stencil is centered at the current best solution $\mathbf{x}_k$ (the poll center), as depicted in Figure 2.1(a). The stencil comprises a set of directions, of which at least one is a descent direction, and a poll stencil size, Δ_k^p. The objective function is evaluated at all of the stencil end points. If one or more of these trial poll points leads to an improvement in the objective function over that of the current poll center, the center of the stencil is moved to the point with the most improvement for the next iteration $k + 1$. See Figure 2.1(b) for a sequence of polling iterations with improvements. If no stencil poll point yields improvement, the stencil size is reduced, as illustrated in Figure 2.1(c), and polling continues with the smaller stencil size.



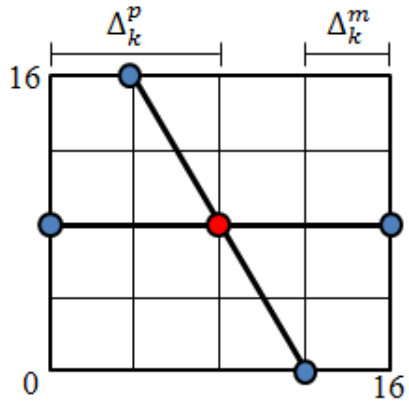
(a) Initial polling stencil (b) Successive polling iterations (c) Contraction of stencil

Figure 2.1: Illustration of a polling sequence in $\mathbb{R}^2$. The red star designates a local optimum, the blue curves are contours of the objective function, the red circles are poll centers around which the polling stencil for the current iteration is defined (with polling stencil size Δ_k^p), the blue circles are poll points to be evaluated at each iteration, and the black circles indicate the sequence of previously evaluated poll centers.

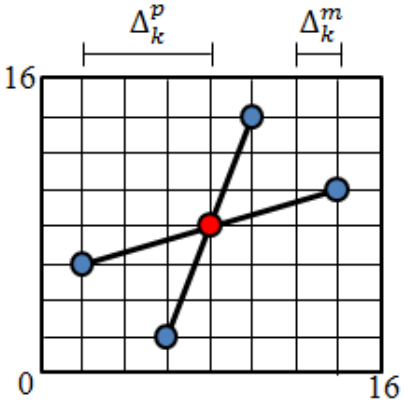
If the stencil orientation is fixed (e.g., as shown in Figure 2.1) at every iteration, the resulting method is essentially Generalized Pattern Search (GPS) [7]. However, if the stencil orientation varies from iteration to iteration, in a manner such that polling is done in an asymptotically dense set of directions, we have the Mesh Adaptive Direct Search (MADS) algorithm [9]. A key difference between GPS and MADS is that in MADS we have an underlying mesh with mesh size Δ_k^m, on which the poll points must lie and $\Delta_k^m \leq \Delta_k^p$, whereas in GPS we have a single stencil size with $\Delta_k^m = \Delta_k^p$. At unsuccessful iterations, by allowing Δ_k^m to decrease faster than Δ_k^p, the MADS algorithm is able to access more possible polling directions.

The possible progression of the MADS algorithm (showing the MADS stencil and mesh size changes) is illustrated in Figure 2.2 for a problem with two optimization variables. Figure 2.2(a) shows a MADS polling stencil at some iteration k with $\Delta_k^p = 8$ units and $\Delta_k^m = 4$ units. If polling at iteration k is unsuccessful, i.e., none of the identified poll points is better than the poll center, then both the mesh and poll sizes are reduced to produce the polling stencil for use in iteration $k+1$, as shown in Figure 2.2(b). If the polling at iteration $k+1$ is also unsuccessful, we could then have the polling stencil at iteration $k+2$ shown in Figure 2.2(c). This stencil again displays reduced mesh and poll sizes. This progression of stencils with iteration shows the variability of the MADS stencil and the fact that the underlying mesh has a mesh

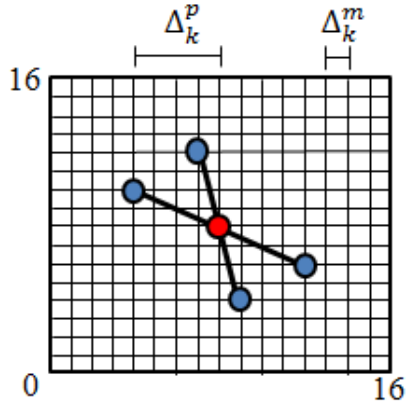
size Δ_k^m that decreases faster than Δ_k^p (note that the mesh size is halved and the poll size is reduced by a factor of $\sqrt{2}$ in this progression). This procedure enables a richer set of polling directions than are used in GPS, as shown in Figure 2.2(d). In fact, in the limit, MADS can access an infinite number of directions. Consistent with this, Audet and Dennis [9] presented results indicating that MADS yields better solutions than GPS for constrained problems. Thus, we apply MADS in this work.



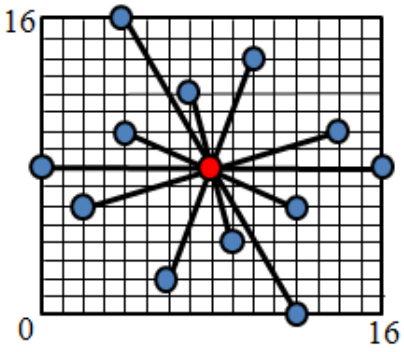
(a) MADS stencil at iteration k ($\Delta_k^p = 8$ units, $\Delta_k^m = 4$ units)



(b) MADS stencil at iteration $k+1$ ($\Delta_k^p = 5.66$ units, $\Delta_k^m = 2$ units)



(c) MADS stencil at iteration $k+2$ ($\Delta_k^p = 4$ units, $\Delta_k^m = 1$ units)



(d) Union of stencils

Figure 2.2: Illustration of MADS stencil changes with iteration. The red circles indicate poll centers around which the current poll stencil is defined (with polling stencil size Δ_k^p). The blue circles indicate poll points, which lie on a mesh defined by MADS mesh size Δ_k^m.

In MADS (and also GPS), at iteration k of the optimization process, each trial point lies on a mesh defined as

$$M_k = \{\mathbf{x} + \Delta_k^m D\boldsymbol{\zeta} \ : \ \mathbf{x} \in V_k, \ \boldsymbol{\zeta} \in \mathbb{N}^{n_D}\} \subset \mathbb{R}^n, \tag{2.3}$$

where $V_k \subset \mathbb{R}^n$, or the cache, is the set of all points evaluated before the current iteration, $\Delta_k^m \in \mathbb{R}_+$ is the mesh size parameter, and D is a fixed matrix in $\mathbb{R}^{n \times n_D}$ composed of n_D columns representing directions. The matrix D must satisfy some conditions but typically corresponds to $[A \ B]$ with $A = -B = I_n$, with I_n the identity matrix in dimension n, and $n_D = 2n$.

Pattern search algorithms such as MADS often include provision for an optional search step, in addition to the poll step. The search step enables great flexibility as it allows the use of any method to generate a finite number of search points in each iteration (these points could be generated anywhere in the same mesh as the polling points in an attempt to 'globalize' the optimization process). See Figure 2.3 for an illustration of possible search, $S_k = \{\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3\}$, and poll points, $P_k = \{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4\}$, generated at some iteration k, for an optimization with two variables.

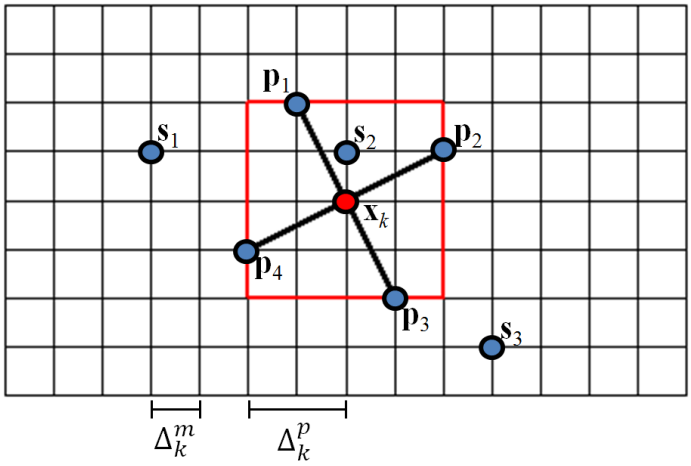


Figure 2.3: Example of MADS directions in the case $n = 2$, at iteration k of the algorithm, where underlying mesh M_k has mesh size Δ_k^m and the polling stencil size is Δ_k^p.

As the algorithm progresses, the independent evolution of the mesh and poll size parameters is designed such that the set of MADS poll directions becomes dense in the space of optimization variables, meaning that potentially every direction can be explored [9]. For discrete (integer) variables, the mesh is modified such that the coordinates corresponding to these variables are constrained to have discrete points. Note that the MADS polling process parallelizes naturally since, at every iteration, the objective function evaluations at the poll points can be accomplished in a distributed fashion. The basic MADS algorithm is summarized in Figure 2.4 (see [9] for further details).

Initialization: Let $\mathbf{x}_0$ be the initial guess such that $f(\mathbf{x}_0)$ is finite and let M_0 be the mesh defined over the solution domain with initial mesh size $\Delta_0^m = \Delta_0^p = \Delta_0 > 0$.

For $k = 0, 1, 2, \ldots$, perform the following:

1. **Search:** Use some finite strategy to seek an improved mesh point, i.e., $\mathbf{x}_{k+1} \in M_k$ defined by Δ_k^m, such that $f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$, for minimization.

2. **Poll:** If the search step was unsuccessful, evaluate f at points in the poll set defined by a stencil, with random poll directions, of size Δ_k^p centered at $\mathbf{x}_k$, in order to find an improved mesh point, $\mathbf{x}_{k+1}$.

3. **Update:** If in the search or poll steps an improved mesh point is found, update $\mathbf{x}_k$ to $\mathbf{x}_{k+1}$ and set $\Delta_{k+1}^p = \Delta_k^p/\phi$, and $\Delta_{k+1}^m = \Delta_k^m$ if $\Delta_{k+1}^p > \Delta_0$ else $\Delta_{k+1}^m = \Delta_k^m/\theta$. Otherwise, set $\mathbf{x}_{k+1} = \mathbf{x}_k$, $\Delta_{k+1}^p = \phi\Delta_k^p$, and $\Delta_{k+1}^m = \Delta_k^m$ if $\Delta_{k+1}^p > \Delta_0$ else $\Delta_{k+1}^m = \theta\Delta_k^m$, with contraction parameters $0 < \theta < \phi < 1$.

Figure 2.4: MADS algorithm, with a search-poll paradigm.

Later we will describe the use of PSO for the search step, which leads to significant global exploration of the solution space. The MADS poll step ensures theoretical local convergence and is based on poll directions that vary with iteration. In each iteration, after the search and poll steps, a final update step is performed. To accomplish this update, the algorithm must first determine if the iteration is a success or a failure. In the unconstrained case (i.e., nonlinear constraints are absent), a successful iteration occurs when the objective function is improved. For problems with nonlinear constraints, a filter method [50], as described below, is used for this assessment.

In our MADS implementation, we use contraction parameters $\theta = 0.5$ and $\phi = 1/\sqrt{2} = 0.707$. The poll size is always increased by a factor of $\sqrt{2}$ after successful iterations and decreased by a factor of 0.707 after unsuccessful iterations. The mesh size on the other hand is doubled, up to the initial mesh size Δ_0, at successful iterations, and halved at unsuccessful iterations. This ensures that $\Delta_k^p \geq \Delta_k^m$ for all k.

Different stopping criteria for terminating MADS can be considered. In our implementation, the optimization is terminated if the mesh or poll size parameters, Δ_k^m or Δ_k^p, which are reduced at every unsuccessful iteration, decrease beyond specified thresholds, or if a specified maximum number of iterations is reached. In pattern search methods, it can be seen that the convergence of the mesh or stencil sizes to zero implies the convergence of the gradient of the cost function to zero [71]. In all of the examples presented in this work, the mesh size criterion terminates the MADS optimization process.

2.2.2 Constraint Treatment in MADS

The bound constraints on the variables in (2.2) are enforced by performing the following coordinate-wise projection of trial points onto Ω:

$$\text{proj}_{\Omega}\left(x_i\right) = \begin{cases} x_{l_i} & \text{if } \; x_i < x_{l_i}, \\ x_{u_i} & \text{if } \; x_i > x_{u_i}, \\ x_i & \text{otherwise.} \end{cases} \tag{2.4}$$

For nonlinear constraints, the filter method is used. Filter methods [40, 50] are step-acceptance mechanisms that seek to avoid the robustness issues that may exist with penalty function methods and other more traditional constraint handling approaches. The use of filters can be seen as an add-on to an optimization algorithm. Instead of combining the objective function and constraint violation into a single function, as is done when using penalty functions, the problem in (2.2) is viewed as a biobjective optimization in which we aim at minimizing both the objective function $f\left(\mathbf{x}\right)$ and an

aggregate constraint violation function, defined as:

$$h(\mathbf{x}) = \left[\sum_{j=1}^{m} \left(\max\left(\bar{c}_j(\mathbf{x}), 0\right)\right)^2\right]^{1/2}, \tag{2.5}$$

where the elements of $\mathbf{c}(\mathbf{x})$ in (2.2) have been normalized. Specifically, constraints of the form $c_j(\mathbf{x}) \leq c_{\max}$ become:

$$\bar{c}_j(\mathbf{x}) = \frac{c_j(\mathbf{x})}{c_{\max}} - 1 \leq 0, \tag{2.6}$$

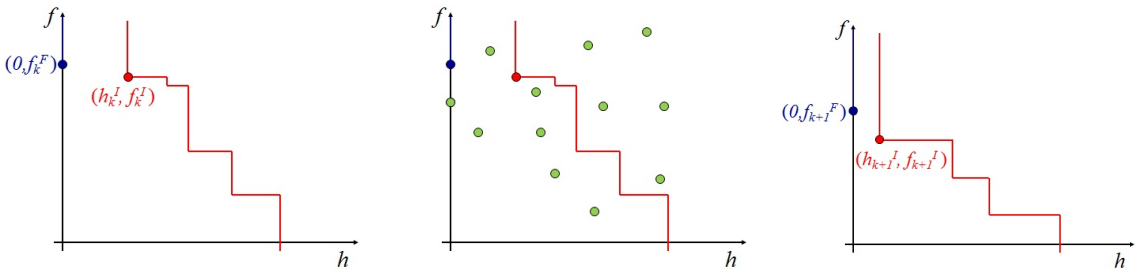
and those of the form $c_j(\mathbf{x}) \geq c_{\min}$ become:

$$\bar{c}_j(\mathbf{x}) = 1 - \frac{c_j(\mathbf{x})}{c_{\min}} \leq 0, \tag{2.7}$$

where $c_{\max}$ and $c_{\min}$ represent the constraint limits.

The second objective of minimizing $h(\mathbf{x})$ is preferred over optimizing $f(\mathbf{x})$ because the solution determined by the optimization algorithm must be feasible. Using terminology from multiobjective optimization, a point $\mathbf{x}_a$ is said to dominate another point $\mathbf{x}_b$ (written as $\mathbf{x}_a \prec \mathbf{x}_b$) if and only if $f(\mathbf{x}_a) \leq f(\mathbf{x}_b)$ and $h(\mathbf{x}_a) \leq h(\mathbf{x}_b)$, with at least one of these being a strict inequality. A filter is defined as a list of pairs $(h(\mathbf{x}_f), f(\mathbf{x}_f))$ such that no pair dominates another pair. An iterate $\mathbf{x}_k$ is considered to be acceptable, or 'unfiltered,' if $(h(\mathbf{x}_k), f(\mathbf{x}_k))$ is not dominated by any pair in the filter. Refer to [50] and [90] for more detailed discussions of the filter method, and to [40] for its application to generally constrained production optimization problems with continuous variables. We now describe the use of the filter method with MADS (this discussion follows that given by Audet and Dennis [8] for GPS).

In adapting the filter method for MADS, a filter at iteration k is defined as the set of infeasible points that are not dominated by any other points evaluated in the optimization process up to iteration k. The evaluated feasible points are considered separately and are not strictly part of the definition of a filter. At iteration k, two types of solutions are defined, as illustrated in Figure 2.5(a): the best feasible solution $\left(0, f_k^F\right)$ and the closest-to-feasible or least infeasible solution in the filter, $\left(h_k^I, f_k^I\right)$.



(a) Filter at the start of iteration k

(b) Search or poll points evaluated during iteration k

(c) Updated filter at iteration $k+1$

Figure 2.5: Illustration of the progression of a filter from iteration k to $k+1$.

During polling in the MADS algorithm, either one of these solutions can be used as the poll center, with preference given to the best feasible solution. We will refer to the best feasible solution as the primary poll center, and to the least infeasible point as the secondary poll center. Even if there is a best feasible solution, it can still be useful to poll around the least infeasible point in the filter. This enables the algorithm to explore a different, and possibly more promising, part of the parameter space.

In our implementation, a MADS iteration that generates an unfiltered point is considered a successful iteration. If no feasible point has been found up to the current iteration, the polling is performed around the least infeasible point in the filter. If feasible points have been found, and if an iteration in which we poll around the best feasible point (primary poll center) is unsuccessful, in the next iteration the polling stencil size is reduced and the secondary poll center is considered. Polling around the secondary poll center continues until an unsuccessful iteration, after which the stencil size is reduced and we return to polling around the primary poll center. We use the least infeasible point as the secondary poll center rather than another filter point (with better objective function value) because the minimization of h is preferred over the optimization of f to ensure feasible solutions.

The filter is updated at successful iterations as there are new unfiltered points that dominate some of the current filter points, as shown in the sequence of illustrations in Figure 2.5. At unsuccessful iterations, the mesh and poll size parameters, Δ_k^m and Δ_k^p, are decreased and the filter remains the same since there are no new unfiltered

points. It is interesting to note that, at the end of the optimization (and at no extra cost), the points of the final filter give a quantitative indication of the sensitivity of the objective function to the constraints. This will be illustrated in Chapter 3.

Filter methods, when compared to techniques that simply discard infeasible points, have the advantage of using infeasible points to enrich the search for an optimal solution. Filter methods have been combined with gradient-based methods [50] as well as with derivative-free algorithms [8, 40]. For more details on the MADS algorithm and the filter constraint treatments used in this work, please refer to [8] and [76].

2.3 Global Derivative-free Method: Particle Swarm Optimization

2.3.1 PSO Method Description

Particle Swarm Optimization (PSO) algorithms are a family of global stochastic search procedures introduced by Eberhart and Kennedy [38]. They are population-based methods and entail a 'swarm' (population) of 'particles' (potential solutions) that move through the solution space with certain 'velocities.' The PSO method has been applied in many application areas, including well placement optimization [94]. The behavior of PSO algorithms is dependent on a few parameters. Fernández Martínez et al. [43, 44, 45, 46, 47] analyzed the stability properties of various PSO algorithms, and their work provides guidelines for choosing parameters that result in particular behaviors (e.g., more explorative versus exploitative).

The PSO algorithm involves a swarm of S particles (S is the population or swarm size). Each particle has a location in the search space and a velocity. The new position of particle j in iteration $k+1$, denoted here as $\mathbf{x}_{k+1}^j$, is determined as follows:

$$\mathbf{x}_{k+1}^j = \mathbf{x}_k^j + \mathbf{v}_{k+1}^j \Delta t \quad \forall\, j \in \{1, \ldots, S\}, \tag{2.8}$$

where $\mathbf{v}_{k+1}^j$ is the velocity of particle j in iteration $k+1$, and Δt is a 'time' increment. Consistent with standard PSO implementations [27], we set $\Delta t = 1$. The velocity

vector associated with each particle j is given by

$$\mathbf{v}_{k+1}^{j} = \underbrace{\omega \mathbf{v}_{k}^{j}}_{\textit{inertial}\text{ term}} + \underbrace{c_1 D_{k+1}^{1}\left(\mathbf{y}_{k}^{j} - \mathbf{x}_{k}^{j}\right)}_{\textit{cognitive}\text{ term}} + \underbrace{c_2 D_{k+1}^{2}\left(\hat{\mathbf{y}}_{k}^{j} - \mathbf{x}_{k}^{j}\right)}_{\textit{social}\text{ term}}, \tag{2.9}$$

where ω, c_1 and c_2 are called the inertial, cognitive and social parameters, respectively. The matrices D_{k+1}^{1} and D_{k+1}^{2} are diagonal, with elements randomly drawn from a uniform distribution with range $[0, 1]$. The inertial term tends to move the particle in the direction it was previously moving, with the idea of continuing in a promising search direction. The cognitive term causes particle j to be attracted to its own previous best position, $\mathbf{y}_{k}^{j}$. The social term causes each particle j to also be attracted to the best position, $\hat{\mathbf{y}}_{k}^{j}$, found through iteration k by any particle in its 'neighborhood' (the definition of 'best' for cases with and without constraint violation will be provided below). In our work, we use the PSO parameters recommended by Clerc [27] ($\omega = 0.729, c_1 = c_2 = 1.494$), which were shown to perform well for a suite of test problems.

The concept of 'neighborhood' is used within PSO to specify the set of particles that particle j 'sees'; i.e., the particles to which it has information links. In a global neighborhood topology, each particle 'sees' all other particles. In this case, there is a single (global best) $\hat{\mathbf{y}}_k$, given by

$$\hat{\mathbf{y}}_{k} = \underset{\boldsymbol{\zeta} \in \{\mathbf{y}_{k}^{1}, \ldots, \mathbf{y}_{k}^{S}\}}{\arg\min} f\left(\boldsymbol{\zeta}\right). \tag{2.10}$$

In this work we use a random neighborhood topology [28], where particle j is linked to a probabilistically determined subset of the swarm. The linkages are altered (randomly) after iterations where there is no improvement in the best solution. This approach was found to be robust and to provide satisfactory performance for well placement problems by Onwunalu [93]. This reference should be consulted for further discussion of PSO neighborhood topologies.

Different stopping criteria can be used in the PSO algorithm. In our implementation, the optimization process is terminated after a given number of iterations is reached or when the norm of the velocities for all particles is smaller than a specified threshold. If all of the velocities are sufficiently small, this usually indicates that the diversity

between the particles has been lost, meaning the swarm has collapsed.

As is the case with MADS, the PSO algorithm is easily parallelizable since, at each iteration, the evaluation of all particles in the swarm can be performed concurrently. We note that all variables in our PSO implementation are treated as continuous. We round to the nearest integer to provide discrete variables when necessary. It is worth observing that even though the PSO algorithm attempts to search globally and does have a stochastic component (which enables it to avoid poor local optima), we cannot realistically expect to include enough particles to 'cover' a high-dimensional search space. Thus, PSO should be viewed as an algorithm capable of providing some amount of global exploration in the optimization search space.

Fernández Martínez and García Gonzalo [45] showed that our choice of PSO parameters ($\omega = 0.729, c_1 = c_2 = 1.494$) lies in the region of second-order stability in the PSO parameter space. This implies that the PSO particles have stable trajectories in the optimization space and that the swarm will eventually collapse. Fernández Martínez and García Gonzalo [44] presented a variety of PSO families, each with different stability regions and exploratory and convergence characteristics. In many of these PSO families, different parameters are used for different particles in the swarm. In one such algorithm [47], PSO parameters (ω, c_1, c_2) are selected from 'clouds' within stability regions, which avoids the need to perform parameter tuning. This cloud algorithm was applied by Suman [113] for the joint inversion of time-lapse seismic and production data for reservoir model updating.

2.3.2 Constraint Treatment in PSO

To satisfy bound constraints, the coordinates of a PSO-generated point that are outside the bounds are projected using (2.4), and the corresponding coordinates of the velocity vector for that particle are set to zero. For treating general (nonlinear) constraints, global stochastic search methods typically employ techniques that either discard infeasible solutions (and thus only consider feasible solutions), as in [61], or they apply penalty function approaches, as in [36] and [99]. In our work, we use filters for each PSO particle, as we now describe.

For problems without nonlinear constraints, the previous best position for PSO particle j and the neighborhood best solution, $\mathbf{y}_k^j$ and $\hat{\mathbf{y}}_k^j$ in (2.9), are determined based only on the objective function value. If the problem has nonlinear constraints, a filter is constructed from the history of each particle, as illustrated in Figure 2.6. When particle j is evaluated at a new position, its filter is updated if the new position is 'unfiltered.' The modification to the original PSO method is mainly in the manner in which we define the previous best position for each particle and the neighborhood best, for use in (2.9). If particle j has been feasible in previous iterations, $\mathbf{y}_k^j$ is the feasible point with the best objective function value. If particle j has not occupied any feasible position, then $\mathbf{y}_k^j$ is taken to be the least infeasible point in the filter for particle j, as indicated by the red circles in Figure 2.6. The neighborhood best position is defined analogously, as the best feasible $\mathbf{y}_k^j$ (in terms of objective function value) or, if there are no feasible previous positions, as the least infeasible point from all filters in the neighborhood. In Figure 2.6, the neighborhood best for the three particles would be the least infeasible point in the filter for Particle 2, since h for this point is the minimum observed.

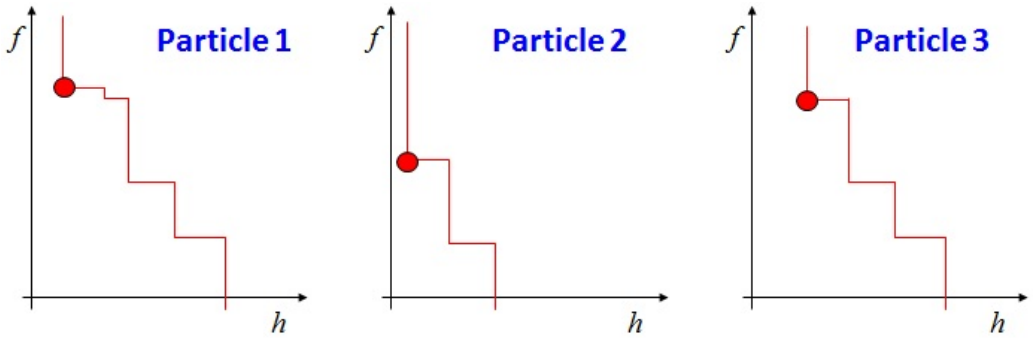


Figure 2.6: Illustration of filters for three particles in a PSO swarm at iteration k, with the least infeasible point in each filter highlighted.

2.4 Hybrid PSO-MADS Procedure

Pattern search techniques such as MADS are local methods that are designed to achieve convergence (from arbitrary starting points) to points that satisfy local optimality conditions. Although the use of a large initial stencil size enables some

amount of global search, the MADS method is not expected to provide the same degree of global exploration as a population-based stochastic search procedure such as PSO with a reasonable swarm size. Therefore, in this work we exploit the global search nature of PSO, and the rigorous convergence to stationary points provided by MADS, by creating a PSO-MADS hybrid. PSO is incorporated into the algorithm as the search step of the MADS procedure.

The hybrid method used in our work is essentially an extension of the PSwarm algorithm [115], which was developed for bound constrained problems. Our hybrid implementation is different from PSwarm in that we use MADS instead of coordinate search during the polling process, and we treat nonlinear constraints using the filter-based approaches described previously. Also, we use a random neighborhood topology in the PSO stage with particle links updated after unsuccessful iterations. Figure 2.7 illustrates the overall workflow of the PSO-MADS hybrid algorithm, and Figure 2.8 presents a detailed description.

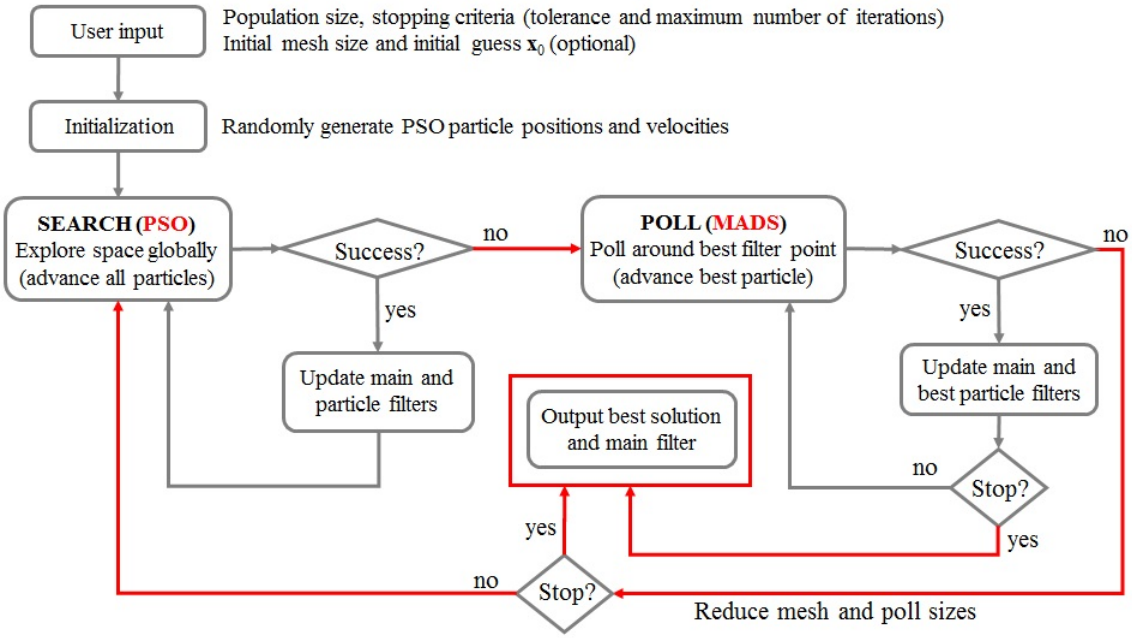


Figure 2.7: Flowchart of PSO-MADS hybrid implementation (grey arrows indicate optimization within the PSO or MADS components and red arrows indicate termination of the algorithm or coupling between the different components of the hybrid).

Initialization: Choose swarm size S, stopping tolerances $\Delta_{\text{tol}} > 0$ and $v_{\text{tol}} > 0$, and initial mesh and poll sizes $\Delta_0^m = \Delta_0^p > 0$.
Initialize number of PSO searches, $l = 0$, specify a maximum number of PSO search iterations, $l_{\max}$, and a maximum number of total iterations in the hybrid algorithm, $k_{\max}$. Randomly generate initial PSO particle information links (random neighborhood topology), initial swarm positions $\mathbf{x}_0^j$ and small velocities $\mathbf{v}_0^j$ with $||\mathbf{v}_0^j|| > v_{\text{tol}}\ \forall\ j \in \{1, \ldots, S\}$. Evaluate $f\left(\mathbf{x}_0^j\right)$ and $h\left(\mathbf{x}_0^j\right)$, and set $\mathbf{y}_0^j = \mathbf{x}_0^j\ \forall\ j \in \{1, \ldots, S\}$ with $\hat{\mathbf{y}}_0$ being the best feasible $\mathbf{y}_0^j$, or least infeasible $\mathbf{y}_0^j$ if there are none feasible.
Initialize particle filters $\mathcal{F}^j$ using each initial particle position, $\mathbf{x}_0^j$, and a main filter $\mathcal{F}^{\text{main}}$ using $\mathbf{x}_0^j\ \forall\ j \in \{1, \ldots, S\}$.
For $k = 0, 1, 2, \ldots$,$k_{\max}$, do:

1. **Search:** Skip if previous poll step was successful, $\max_{j \in \{1,\ldots,S\}} ||\mathbf{v}_k^j|| < v_{\text{tol}}$, or $l \geq l_{\max}$.
   $l \leftarrow l+1$ (increment number of PSO searches). Randomly generate new PSO particle information links if previous step is unsuccessful.
   Compute $\mathbf{v}_{k+1}^j$ and $\mathbf{x}_{k+1}^j$ using equations (2.8) and (2.9). Evaluate $f\left(\mathbf{x}_{k+1}^j\right)$ and $h\left(\mathbf{x}_{k+1}^j\right)\ \forall\ j \in \{1, \ldots, S\}$. Set $\hat{\mathbf{y}}_{k+1} = \hat{\mathbf{y}}_k$.
   For $j = 1, 2, \ldots, S$, do:
   - If $\mathbf{x}_{k+1}^j$ is unfiltered w.r.t. $\mathcal{F}^j$, update $\mathcal{F}^j$, and if w.r.t. $\mathcal{F}^{\text{main}}$, update $\mathcal{F}^{\text{main}}$, then
     - set $\mathbf{y}_{k+1}^j$ to the best feasible point for particle j or least infeasible point in $\mathcal{F}^j$ if no feasible points for particle j.
     - if $\mathbf{y}_{k+1}^j \prec \hat{\mathbf{y}}_{k+1}^j$ then set $\hat{\mathbf{y}}_{k+1}^j = \mathbf{y}_{k+1}^j$ (update neighborhood best; only deem search step and iteration successful if global best, $\hat{\mathbf{y}}$, gets updated).
   - Otherwise set $\mathbf{y}_{k+1}^j = \mathbf{y}_k^j$ (deem search step unsuccessful).
2. **Poll:** Skip if successful search step performed. Choose $\hat{\mathbf{y}}_k$ as the poll center and evaluate f and h at all poll points.
   - If there exists a MADS poll direction $\mathbf{d}_k$ such that $\hat{\mathbf{y}}_k + \Delta_k^p \mathbf{d}_k$ is unfiltered w.r.t $\mathcal{F}^{\text{main}}$, then
     - if $\hat{\mathbf{y}}_k + \Delta_k^p \mathbf{d}_k \prec \hat{\mathbf{y}}_k$, then set $\hat{\mathbf{y}}_{k+1} = \hat{\mathbf{y}}_k + \Delta_k^p \mathbf{d}_k$ (update the global best particle position).
     - update filters $\mathcal{F}^{\text{main}}$ and $\mathcal{F}^{\hat{\jmath}}$, where $\hat{\jmath}$ is the index of the global best PSO particle.
     - set $\Delta_{k+1}^m \geq \Delta_k^m$ and $\Delta_{k+1}^p \geq \Delta_k^p$ (optionally expand mesh and poll sizes in the manner shown in Figure 2.4; deem poll step and iteration successful).
   - Otherwise, if all poll points $\hat{\mathbf{y}}_k + \Delta_k^p \mathbf{d}_k\ \forall\ \mathbf{d}_k$ are filtered, then
     - set $\hat{\mathbf{y}}_{k+1} = \hat{\mathbf{y}}_k$.
     - set $\mathbf{x}_{k+1} = \mathbf{x}_k$, $\Delta_{k+1}^m = \theta \Delta_k^m$ and $\Delta_{k+1}^p = \phi \Delta_k^p$ with contraction parameters $0 < \theta < \phi < 1$ (contract mesh and poll sizes; deem poll step and iteration unsuccessful).
3. If $\Delta_{k+1}^m < \Delta_{\text{tol}}$, then stop. Otherwise, $k \leftarrow k+1$ and go to Step 1.

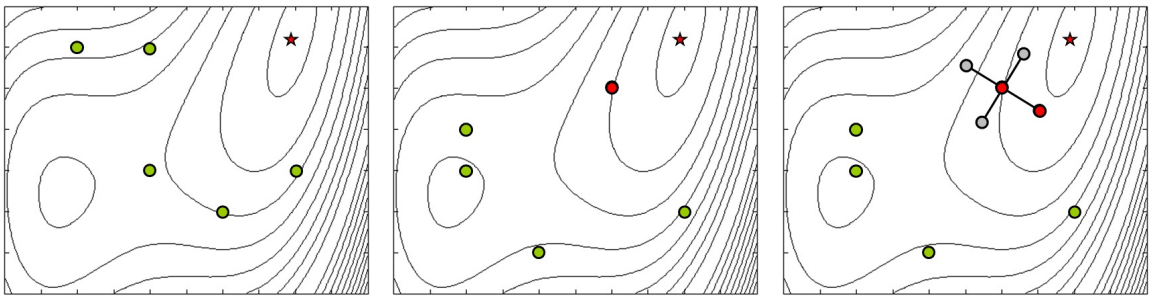
Figure 2.8: Details of PSO-MADS hybrid algorithm.

The hybrid search method begins with an initial swarm of particles, including one or more user-defined initial guesses if provided, and it then applies one iteration of PSO (using (2.8) and (2.9)). Consecutive iterations where the search step is successful are equivalent to consecutive iterations of the standalone PSO algorithm. In the hybrid implementation, for a PSO swarm with S particles, there are S particle filters built from the history of each particle, and a main filter (denoted $\mathcal{F}^{\text{main}}$ in Figure 2.8) constructed from all points evaluated in the optimization process. Note that the least infeasible point from all particle filters is the least infeasible point in the main filter.

A PSO search step is designated as successful if the global best position improves, implying generation of a new global best position that dominates the previous global best in terms of objective function f or constraint violation h. This definition of success is different than that used for MADS, where an iteration is deemed successful if any unfiltered points are found. This stricter PSO criterion is applied to avoid performing many PSO iterations during which the filter of the best particle remains unchanged (i.e., the best particle does not improve), even though the filter of a clearly suboptimal particle continues to change. This treatment acts to accelerate the convergence of the overall hybrid algorithm. Note that, as indicated in Figure 2.8, in the search step PSO uses a random neighborhood topology with neighborhood best positions $\hat{\mathbf{y}}^j$ (which appears in the social term of the PSO velocity equation). The definition of a successful PSO iteration is, however, based on improvement of the global best position, $\hat{\mathbf{y}}$, since subsequent MADS polling will be performed around this point.

When the search step is not successful (i.e., does not provide improvement), the MADS poll step is performed, centered on the global best position from the swarm (least infeasible point or best feasible point computed for any of the particles). The polling continues as long as consecutive poll steps are successful (i.e., better feasible points or unfiltered points in $\mathcal{F}^{\text{main}}$ are generated). Thus the position of the best PSO particle changes, due to MADS polling, while the positions of the other particles in the swarm remain the same. During polling, we consider $\mathcal{F}^{\text{main}}$ rather than the filter associated with the best particle because $\mathcal{F}^{\text{main}}$ contains more complete information on the progress of the overall search.

Consecutive iterations where the poll step is successful are equivalent to consecutive iterations of the standalone MADS algorithm (polling only, without a search step). At these successful MADS iterations, $\mathcal{F}^{\text{main}}$ and the particle filter corresponding to the best PSO particle, around which polling is being performed, are updated with the new unfiltered poll points. When polling is unsuccessful, the mesh and poll sizes are reduced and the hybrid algorithm returns to the PSO search with the modified position (and particle filter) for the best PSO particle (around which polling was performed), and all other PSO particles in the swarm are unchanged from the most recent PSO iteration. The PSO stage of the hybrid algorithm terminates when the particle velocities are smaller than a prescribed tolerance v_{tol} (this normally indicates that the swarm has collapsed), or when a maximum number of PSO iterations $l_{\max}$ is reached. The hybrid algorithm terminates when the mesh size decreases below a given tolerance Δ_{tol} or when a maximum number of hybrid iterations $k_{\max}$ is reached.



(a) PSO iteration k (b) PSO iteration $k+1$ (c) MADS iteration

Figure 2.9: Illustration of PSO-MADS iterations for an optimization problem with two variables.

Figure 2.9 illustrates the PSO-MADS procedure explained above for a search in two dimensions. In this figure, the contours represent the objective function we seek to minimize (note that in addition to the global optimum indicated by a red star, there is also another local optimum with higher cost function value). Figure 2.9(a) shows a five-particle PSO swarm (green points) after iteration k. The particles are advanced, using PSO, to iteration $k+1$. If at iteration $k+2$, no improvement is obtained with PSO, the best particle (red point in Figure 2.9(b)) is identified and used as the current iterate for MADS polling iterations (Figure 2.9(c)). It is evident that the

MADS iteration provides improvement in this case, so that in the subsequent PSO search, the position of the best PSO particle will have changed.

This and the two preceding sections present derivative-free optimization methods together with filter treatments for handling general constraints. The filter method for use with MADS was introduced by Audet and Dennis Jr. [8, 9]. Their group implemented a C++ code called NOMAD [1, 76] from which we run MADS with filter constraint treatment. Starting from the NOMAD base code, we built the PSO-MADS search-poll framework. This framework is used to run standalone MADS and PSO, as well as the PSO-MADS hybrid. The filter-type constraint handling in PSO, as described in Section 2.3.2, is novel and enables consistent switching between the PSO and MADS stages of the hybrid.

2.5 Branch and Bound

As discussed earlier, besides optimizing ordered integer and continuous variables, true MINLP problems include variables of a categorical nature. The general optimization problem presented in (2.2) is modified for this case as follows:

$$\begin{aligned} \min_{\mathbf{z}\in\Lambda,\ \mathbf{x}\in\Omega} \ & f(\mathbf{x},\mathbf{z}), \quad \text{subject to} \quad \mathbf{c}(\mathbf{x},\mathbf{z}) \leq \mathbf{0}, \\ \text{with} \ \ & \Omega = \{\mathbf{x}\in\mathbb{R}^n \ : \ \mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u\}, \\ \text{and} \ \ & \Lambda = \{\mathbf{z}\in\mathbb{Z}^{n_c} \ : \ \mathbf{z}_l \leq \mathbf{z} \leq \mathbf{z}_u\}. \end{aligned} \tag{2.11}$$

Here we separate the vector of n_c categorical variables $\mathbf{z}$ from the other variables $\mathbf{x}$. In a field development problem, the categorical variables could for example indicate whether to drill a well or not.

Branch and Bound (B&B) is a general solution strategy proposed by Land and Doig [75] for optimization problems with discrete-valued variables. The method is based on the observation that the enumeration of all candidate integer solutions has a tree structure. For example, in a problem with five categorical binary variables, the complete enumeration of all variable combinations entails a tree with 2^5 'leaf nodes.' Since the full enumeration tree will be too large to evaluate for real problems with

even a moderate number of variables, the B&B approach avoids evaluating the entire tree, and proceeds by discarding large subsets of poor solutions. This is accomplished by using estimated upper and lower bounds of the cost function. Specifically, the algorithm determines which nodes are the most promising by estimating a bound on the best value of the objective function that can be obtained by 'growing' the subtree from the node. The *branching* occurs when a node is selected for further growth and the next generation of 'children nodes' is created from that parent node. The *bounding* refers to determining a bound for the best value that can be attained by growing from a given node.

If the bound associated with a particular node is deemed poor (by techniques we will describe shortly), the node is not expanded further and is *pruned* (i.e., none of the nodes in its subtree is evaluated). At the end of the B&B process, only a very small fraction of the full enumeration tree is actually evaluated. The remainder of the tree is not considered to be attractive in terms of the cost function. It is important to note that the success of the B&B approach is contingent upon effectively approximating upper and lower bounds for the actual solution to the MINLP problem. These accurate bounds are required to avoid pruning nodes whose subtrees contain the true MINLP solution.

For the maximization of a cost function with integer-valued variables, B&B proceeds as follows. First, upper and lower global bounds for the cost function are estimated for the complete optimization space. The upper bound can be obtained by maximizing the cost function with all integer variables relaxed to real-valued variables. The lower bound can be taken as the cost function evaluated at a point in the (discrete) optimization space. If the upper and lower bounds coincide, then the point used for determining the lower bound is the optimal solution, since no other point (even in the relaxed space) can be expected to improve the bound. In this case, the search is terminated and optimal solution reported. Otherwise, the optimization space is divided in two, and for each of these two regions we proceed as for the complete space (branching the search space if needed, and terminating the search in the corresponding region if the global bounds coincide).

The process can be accelerated when the upper bound for a region is lower than

the lower bound for another region. In that case, we can eliminate (prune) the first region from the search, since the best cost function that can be expected in that region (according to the bound provided) is worse than the cost function already determined for the other region. Because the B&B algorithm relies strongly on the quality of the global upper bounds computed (for maximization problems), the solution of each relaxed problem should be performed with a method with global exploration features. In this work, we use the hybrid PSO-MADS for that purpose.

We now illustrate B&B on a simple problem with eight binary variables. The optimized continuous variables in the problem are designated by $\mathbf{u}^*$. This example illustrates the B&B process for the unconstrained version of the first MINLP problem studied in the Chapter 5. The tree explored in this B&B process is illustrated in Figure 2.10. The first node (labeled with the index zero) represents the solution of the relaxed problem for the complete optimization space, which yields an upper bound for the cost function of 25.98. From this node, we should in principle explore the 2^8 elements in the search space representing all the different combinations of the binary variables. It should be noted that some of the components of this relaxed solution are real-valued. Rather than determining a lower bound (since we are unlikely, at this preliminary search stage, to find an optimal solution), we proceed with the branching.

The branching is achieved by setting one of the binary variables first to zero and then to one (this yields two new search regions, each with seven binary variables). The variable selected for the branching is the one that is closest to 0.5 in the optimized relaxed solution (fourth component, z_4, of the solution at node 0 in Figure 2.10). This selection is motivated by the fact that this variable is the most 'undecided,' in the sense that it is the variable furthest from an integer value.

We then perform optimizations with z_4 set to 0 (node 1) and with z_4 set to 1 (node 2). This gives the two upper bounds shown in Figure 2.10. There are now two subregions that need to be explored. Given a number of regions at any iteration, we select the region with highest upper bound, since that may be the most promising in terms of cost function. In the example in Figure 2.10, this region is the subtree (with 2^7 elements) expanding from node 2. We proceed as for node 0, and we first define

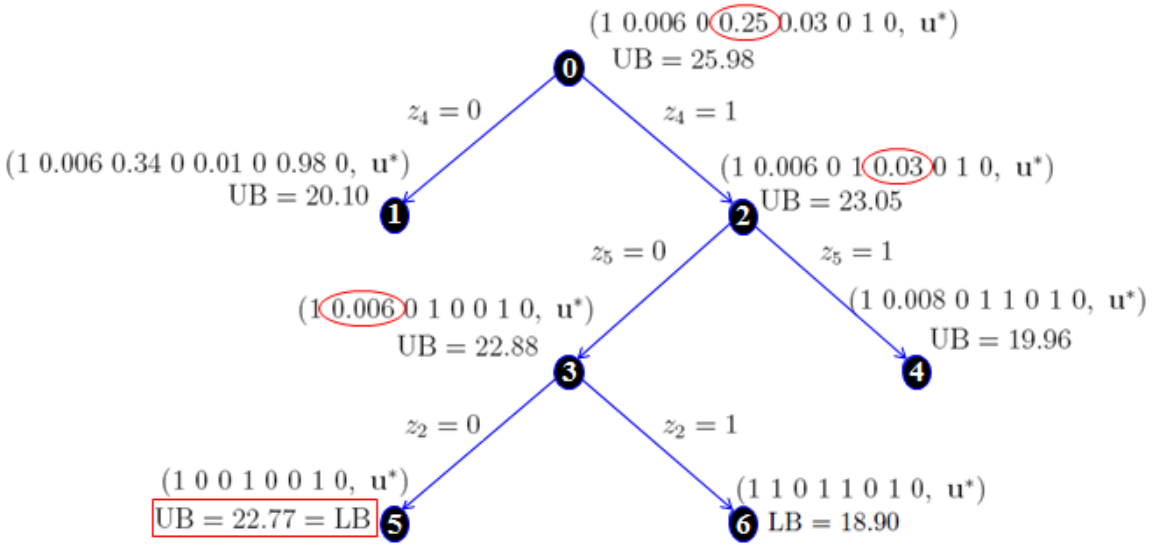


Figure 2.10: Illustration of B&B for a maximization problem with eight binary categorical and eight continuous variables (only values of binary variables shown).

optimization subproblems for nodes 3 and 4 by setting the fifth variable z_5 to 0 and 1. After branching from node 3 (with respect to the second integer variable z_2), we solve and obtain solutions for the subproblems in nodes 5 and 6.

At this point the complete search can be terminated based on the following observations. First, nodes 5 and 6 yield upper bounds that are obtained with integer-valued variables (in spite of having performed the optimization in a relaxed space of five real-valued variables). Therefore, we can find lower bounds for these two nodes that coincide with the upper bounds (by just selecting the solutions shown in the figure). Since the upper/lower bound for node 5 is larger than the upper/lower bound for node 6, we can eliminate node 6 from the search. Additionally, since the lower bound for node 5 is larger than the upper bound for nodes 1 and 4, we can also prune the subtrees expanding from these two nodes. Consequently, and consistent with the global upper bound, the integer-valued solution found at node 5 is the global maximum for all of the 2^8 possible configurations. We note that the computational cost for the overall process in this case is equivalent to the solution of seven (relaxed) optimizations.

B&B can be easily extended to true MINLP problems. Whenever an upper bound is required, both relaxed and (actual) real-valued variables are jointly optimized. Lower bounds for the integer search can be computed by selecting a group of integer-valued variables, and optimizing for the remaining real-valued variables. Although B&B is relatively straightforward conceptually, its implementation entails the use of some heuristics (based on several decades of B&B usage) that may accelerate the optimization considerably. In the results shown later, we use our own B&B implementation. It is possible that improved results could be obtained if we used a highly tuned B&B implementation.

It is evident that the computation required by B&B can increase very quickly as the number of integer variables increases. This is because the number of potential nodes that must be evaluated will be larger, and because the complexity of each relaxed optimization problem will be higher. In the examples presented in Chapter 5, we observe performance deterioration with B&B for a problem with only ten binary variables. More heuristic methods such as PSO, by contrast, can be very useful in practical applications where it is not essential to explore rigorously the complete discrete space, but rather to simply perform some amount of global search.

Chapter 3

Well Control Optimization

In this chapter we consider a generally constrained well control example with only continuous (well control) variables. This example illustrates the performance of the MADS, PSO and PSO-MADS approaches applied to a continuous problem and highlights the effectiveness of the new PSO and PSO-MADS filter treatments. We also illustrate the constraint sensitivity information that can be obtained from the trial points evaluated during the course of the optimization run.

3.1 Problem Setup

The MADS, PSO and PSO-MADS methods presented in Chapter 2 will now be applied to solve a constrained well control problem. The reservoir is modeled using a two-dimensional model with 40×40 grid blocks. The channelized permeability field is depicted in Figure 3.1 (note that k designates permeability, which is assumed to be isotropic). This and many of the geological models used in this work were generated using the Stanford Geostatistical Modeling Software (SGeMS) [102]. The sand-shale facies model was generated with the object-based simulation program (TetrisTiGen) in SGeMS, while the different facies were populated with permeability values using sequential Gaussian simulation, also in SGeMS.

For the well control optimization problem, the well locations are fixed at the positions indicated in Figure 3.1. The well bottomhole pressures (BHPs) will be optimized in

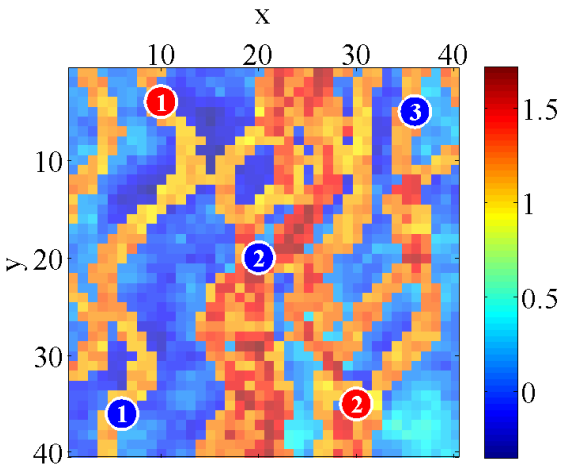


Figure 3.1: Permeability field ($\log_{10} k$, with k in md) used for well control example, showing locations of injection and production wells (in blue and red respectively).

order to maximize undiscounted NPV. For this case, the MINLP optimization problem presented in (2.1) reduces to the following NLP:

$$\min_{\mathbf{u}\in U} f(\mathbf{p},\mathbf{u}) = -\text{NPV}(\mathbf{p},\mathbf{u}) = -p_o Q_o(\mathbf{p},\mathbf{u}) + c_{pw} Q_{pw}(\mathbf{p},\mathbf{u}) + c_{iw} Q_{iw}(\mathbf{p},\mathbf{u}),$$
$$\text{subject to} \quad \mathbf{g}(\mathbf{p},\mathbf{u}) = \mathbf{0}, \quad \mathbf{c}(\mathbf{p},\mathbf{u}) \leq \mathbf{0}, \tag{3.1}$$

where p_o, c_{pw} and c_{iw} are the price of oil, cost of handling produced water and cost of water injection, respectively (all in \$/STB), and Q_o, Q_{pw} and Q_{iw} are the total produced oil, produced water and injected water (all in STB) obtained from the reservoir simulator. The simulation model involves two-phase oil-water flow. The oil and water relative permeability curves are shown in Figure 3.2. Other simulation and optimization parameters are summarized in Table 3.1.

The reservoir is simulated for 10 years (3650 days), with BHPs updated every 730 days, implying five control periods. As there are three injectors and two producers in the model, the total number of optimization variables for this problem is 25 (5 wells $\times$ 5 control periods). The bound constraints (injection and production BHP

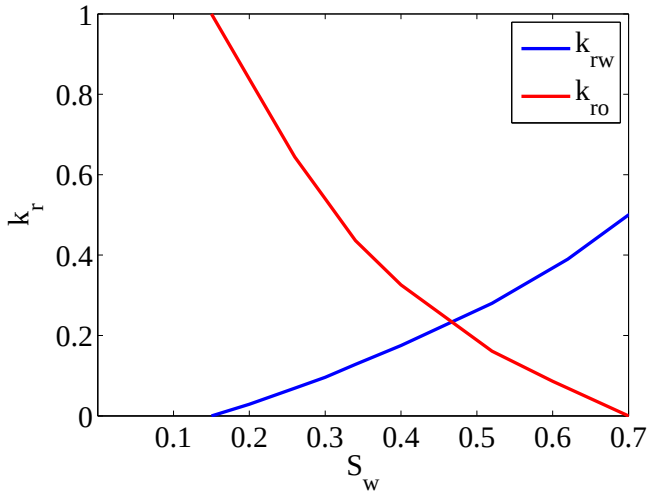


Figure 3.2: Relative permeability curves for the oil and water phases.

Table 3.1: Simulation and optimization parameters for well control example

Grid cell dimensions	60 ft × 60 ft × 10 ft
Initial pressure p_i, at datum	5080 psi at 8500 ft
μ_o and μ_w at p_i	1.9 and 0.325 cp
ρ_o and ρ_w	54.0 and 58.0 lbm/ft^3
B_o and B_w at p_i	1.00 RB/STB
p_o, c_{pw} and c_{iw}	\$80, \$10 and \$5/STB
Drilling cost	\$5 million per well
Injection BHP range	6000 - 10000 psi
Production BHP range	500 - 4500 psi
Maximum water injection rate	600 STB/day
Minimum oil production rate	200 STB/day
Maximum fluid production rate	600 STB/day
Maximum well water cut	0.7

ranges) and nonlinear constraints (rate and water cut constraints) are given in Table 3.1. The latter constraints are nonlinear because our control variables are BHPs, so the values for rates and water cuts are only obtained after the reservoir simulation, which is a nonlinear function evaluation procedure, is performed.

For this case, the MADS, PSO and PSO-MADS methods are applied to determine the BHPs that maximize NPV while satisfying all specified constraints. The parameters

used for the different algorithms are as follows. In standalone MADS, a $2n$-polling stencil is used, implying a maximum of $2n$ function evaluations per MADS iteration ($2n = 50$ for this case). It should be noted that the evaluation of some of the poll points can be avoided if these points have already been visited in previous iterations and the objective and constraint values are saved in cache memory. The initial MADS mesh and poll sizes are set to 1000 psi (recall that all optimization variables are BHPs). In standalone PSO, a swarm size of 50 is used, implying a maximum of 50 function evaluations per PSO iteration (fewer if some positions have been previously evaluated and saved). The same parameters used for the standalone PSO and MADS iterations are considered in the PSO-MADS hybrid method.

The function evaluations in all three methods are parallelized using a computing cluster. For these runs, we typically had about 50 processors available, meaning the iterations were fully parallelized. To convert from total simulations (proportional to total computational time) to equivalent simulations (proportional to actual elapsed time), the number of total simulations should be divided by 50. Note that the overhead in the parallelization process, which leads to a true speedup that is slightly less than 50, is not accounted for in this conversion.

3.2 Optimization Results

Considering the stochastic nature of these algorithms, each of the three methods is run five times starting from different initial guesses. The NPVs for the five initial guesses, together with their mean and standard deviation, designated <NPV> and σ, are shown in Table 3.2. The amount of normalized constraint violation, $h(\mathbf{u})$, associated with each initial guess is also presented.

The first initial guess has all of the wells operating at the average BHP within the specified range, i.e., all injectors are set to 8000 psi and producers to 2500 psi. The remaining four initial guesses are randomly generated from a uniform distribution within the BHP ranges. From Table 3.2 we see that all of the initial guesses are infeasible since they all have nonzero constraint violation values.

Table 3.2: NPVs and constraint violations from the five initial guesses used in the optimizations (best NPV is italicized)

Run #	Initial guess NPV($\mathbf{u}$) [\$ MM]	Constraint violation $h(\mathbf{u})$
1	21.59	0.563
2	20.95	0.616
3	*26.51*	0.915
4	23.17	0.948
5	22.67	0.912
<NPV>	22.98	0.791
σ	2.16	0.185

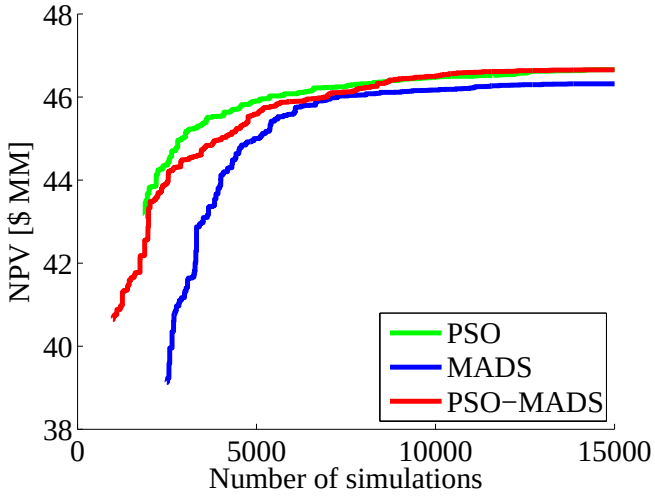


Figure 3.3: Evolution of mean NPV for the five runs.

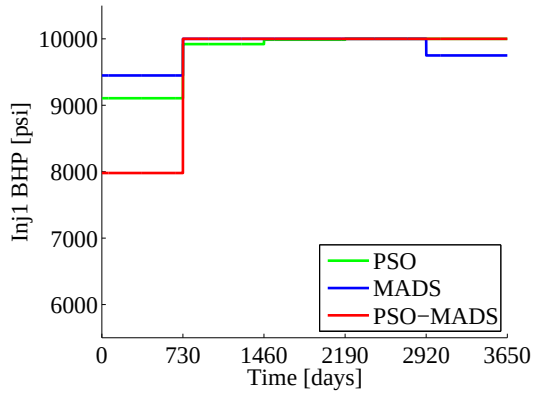
Figure 3.3 displays the evolution of mean NPV for the feasible solutions for the PSO, MADS and PSO-MADS methods. Table 3.3 summarizes the final NPVs from all runs. The curves in Figure 3.3 are not plotted until about 1000 simulations have been performed because the initial guesses and the earlier simulations lead to infeasible solutions. The results in Table 3.3 for the five runs for each method do not show much distinction between the performance of the different methods. All three methods perform well and increase the average NPV of the initial guesses by about 100%, while satisfying the specified rate and water cut constraints. The 15 runs yield final NPVs that are within 2% of the best NPV obtained by the second

Table 3.3: Final NPVs from five runs for the three different methods (best values are italicized)

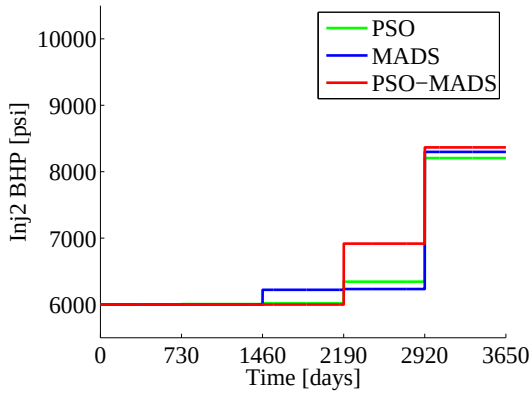
Run #	PSO NPV [$ MM]	MADS NPV [$ MM]	PSO-MADS NPV [$ MM]
1	46.64	46.04	46.50
2	*46.84*	46.10	46.63
3	46.50	46.67	46.72
4	46.79	*46.71*	*46.73*
5	46.51	46.12	46.69
<NPV>	46.66	46.33	46.65
σ	0.16	0.33	0.09

PSO run. These results display the effectiveness of the new PSO with filter and the hybrid PSO-MADS with filter. The fact that these algorithms give results that are comparable to those from the more established MADS with filter method suggests that our implementation is reasonable. In later chapters, where we increase the complexity of the optimization problem by including integer and categorical variables, we will see that the PSO-MADS hybrid provides the best results, which motivates the search-poll hybridization strategy.

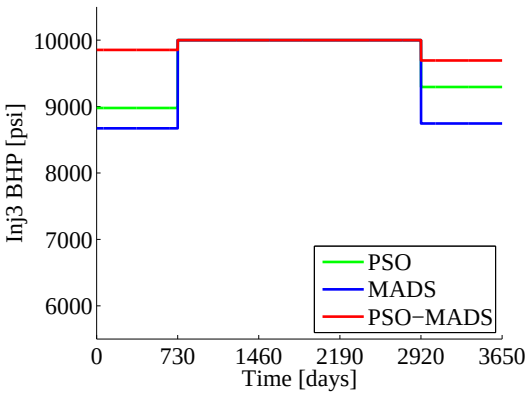
Figure 3.4 shows the best optimized BHP solutions for each of the different optimization methods (solutions are from PSO Run 2, MADS Run 4 and PSO-MADS Run 4). The results show that there are similarities between the different solutions, e.g., all three methods keep Injector 2 at the lower bound until 1460 days and have Producer 1 at its lower bound for the entire ten years of simulation. Note, however, that even though all three optimized NPVs are within 0.3% of the best solution of $46.84 million, the optimal controls do show differences. This observation is significant because it supports the hypothesis that, even for cases with general constraints, instead of a single optimal solution, we may have a 'ridge' of optimal (or near-optimal) solutions for the well control problem. Jansen et al. [67] had similar observations in their waterflooding examples, which did not include nonlinear constraints. They observed that significantly different waterflooding schemes resulted in nearly the same NPV, and attributed this to the use of more optimization variables than are necessary.



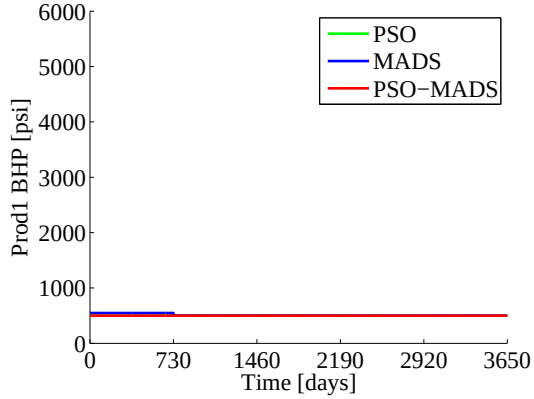
(a) BHP profile for Injector 1



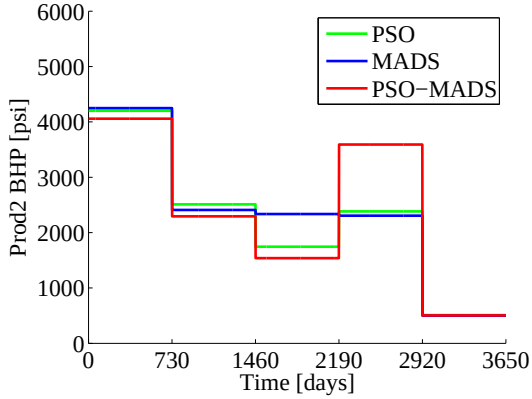
(b) BHP profile for Injector 2



(c) BHP profile for Injector 3

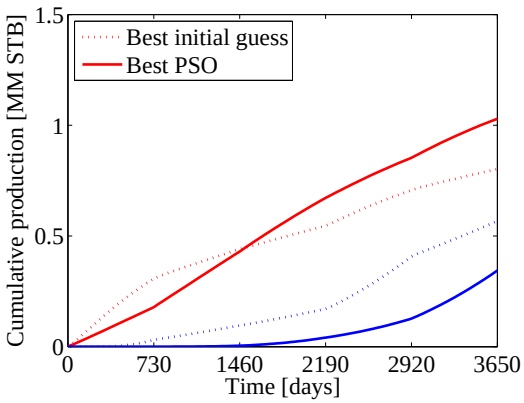


(d) BHP profile for Producer 1

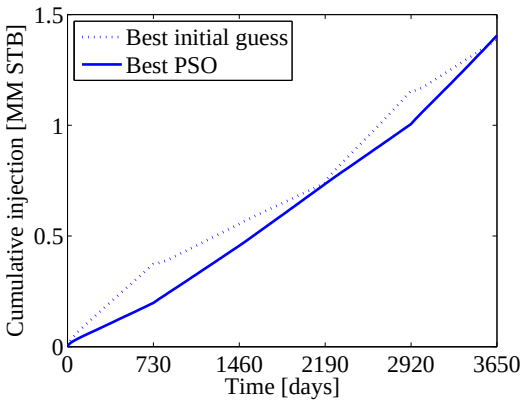


(e) BHP profile for Producer 2

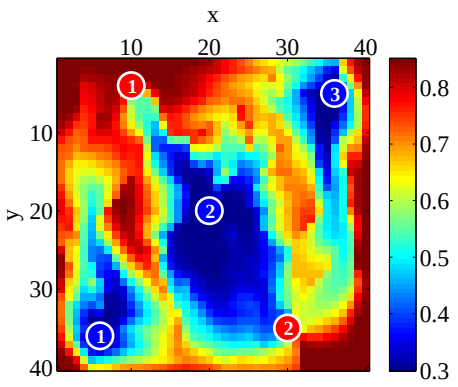
Figure 3.4: Comparison of the best optimized BHP controls for the three injection and two production wells, from the three different methods.



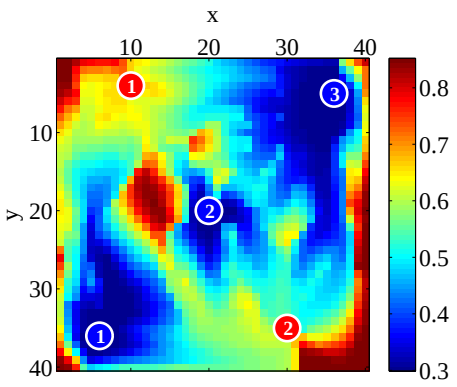
(a) Cumulative oil (red) and water (blue) production



(b) Cumulative water injection



(c) Final oil saturation, initial guess Run 3



(d) Final oil saturation, PSO Run 2

Figure 3.5: Comparison of the cumulative production and injection profiles, and final oil saturation maps (red indicates oil and blue water), for the best initial guess (Run 3) and best optimized solution (PSO Run 2).

We now examine the optimization results further to see the effect of the optimized controls on the reservoir performance. Figure 3.5 shows comparisons of the cumulative fluid production and injection profiles, as well as the final oil saturation maps, for the best initial guess (Run 3) and the best solution achieved (PSO Run 2). From Figure 3.5(b), we see that the optimized control scheme injects just as much water over the course of ten years as the best initial control scheme. However, the optimized control scheme leads to a 28% increase in cumulative oil produced and a 41% reduction in

cumulative water produced. Figures 3.5(c) and 3.5(d) illustrate the improvement in sweep efficiency of the optimal control scheme over the best initial guess.

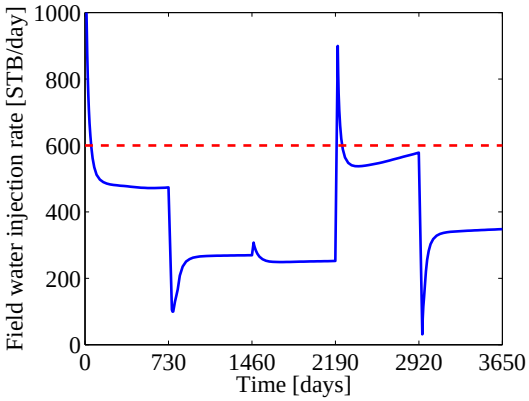
Recall that for this well control problem we have four different types of nonlinear constraints (last four rows of Table 3.1). Figure 3.6 shows the field rate and water cut results for the best initial guess, together with the constraint limits. From the plots in this figure we observe that the initial guess control scheme leads to violation of the maximum field injection, minimum oil production, and maximum water cut in Producer 2 constraints.

Figure 3.7 presents analogous information for the optimized control scheme. Note that the rates show spikes at the start of the simulation ($t \leq 1$ day), which are a result of the instantaneous application of the BHP controls. We only include data for $t > 1$ day in our calculation of $h(\mathbf{u})$, so this behavior is not considered a constraint violation. From the plots in Figure 3.7, we see that all specified constraints are thus satisfied in the optimal solution. This demonstrates the ability of the filter approach to satisfy the nonlinear constraints during optimization. Note that the field injection and production rates in Figures 3.7(a) and 3.7(b), as well as the water cut for Producer 2 in Figure 3.7(e), reach their constraint limits. This indicates that these constraints are active at the constrained optimal solution.

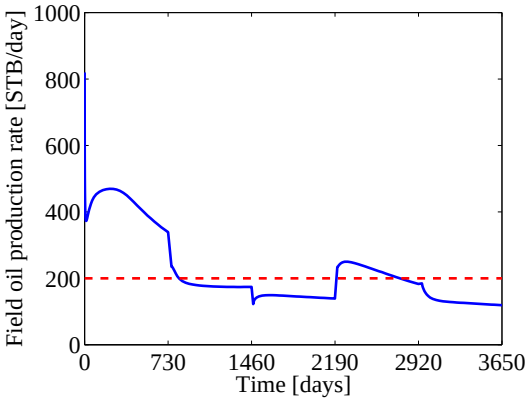
3.3 Constraint Sensitivity Analysis

One of the advantages of the filter constraint handling approach is that we can extract additional information about the optimization process and the optimal solution that is not available from other constraint handling techniques. For example, the final filter provides the sensitivity of the objective function to the constraint violations. It depicts how much the objective function can be improved by allowing a certain level of constraint violation. In petroleum field development, this may be relevant to the sizing of facilities, as these sizes determine the maximum flow rate constraints.

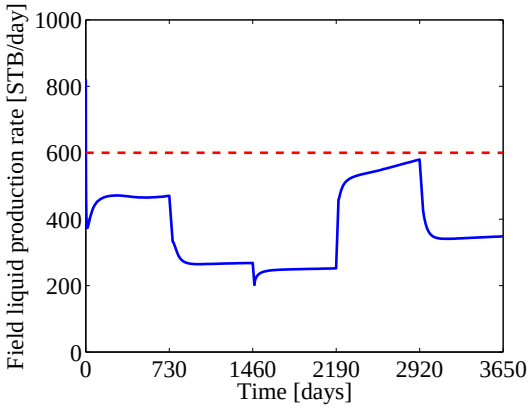
The final filters from the best optimization run for each method (PSO Run 2, MADS Run 4 and PSO-MADS Run 4) are shown in Figure 3.8. Recall that $h(\mathbf{u})$ entails aggregation of the individual normalized constraint violations, $\bar{c}_j$ in (2.5). Consistent



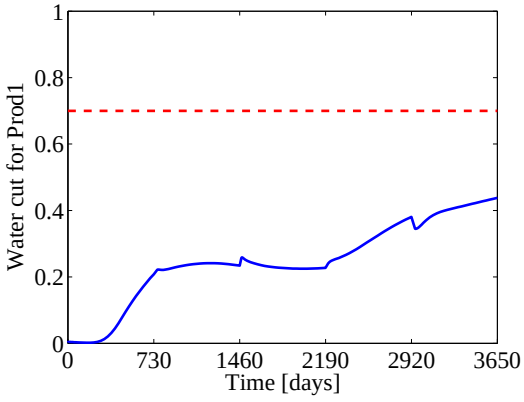
(a) Field water injection rate



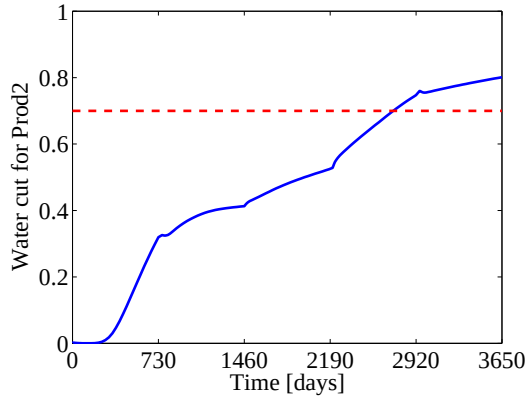
(b) Field oil production rate



(c) Field fluid production rate

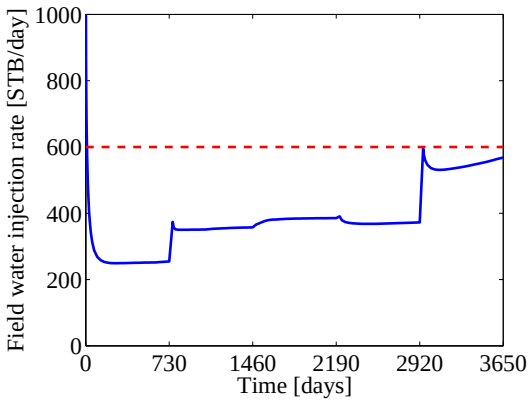


(d) Water cut for Producer 1

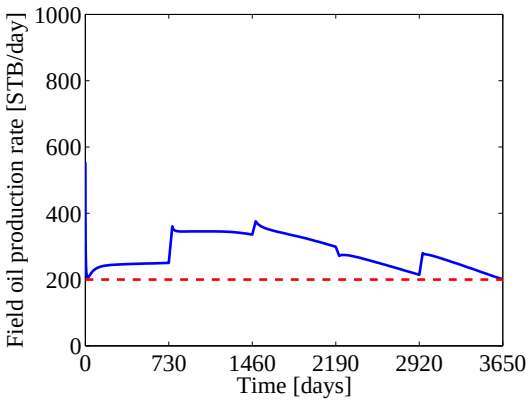


(e) Water cut for Producer 2

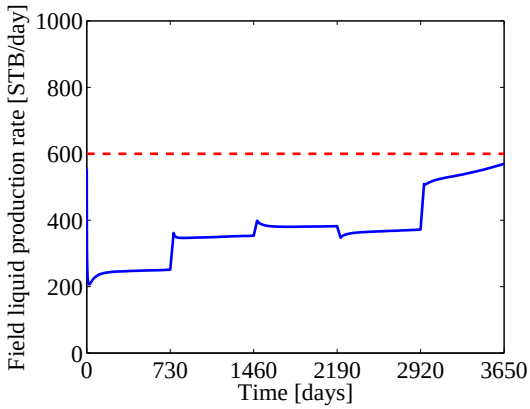
Figure 3.6: Field rates and well water cuts for the best initial guess (Run 3). Constraint limits shown as red dashed lines.



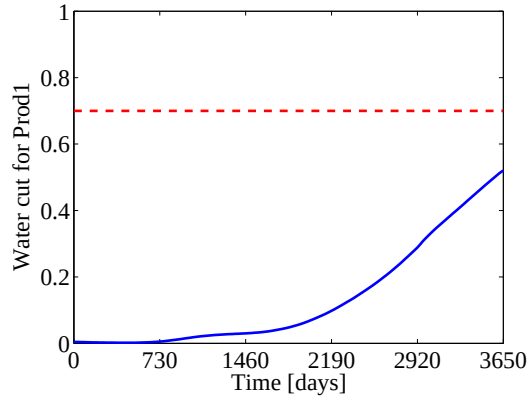
(a) Field water injection rate



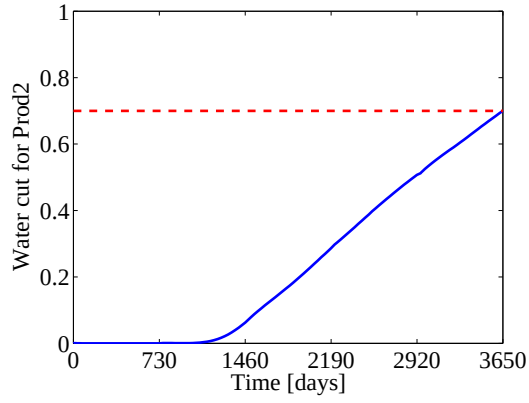
(b) Field oil production rate



(c) Field fluid production rate

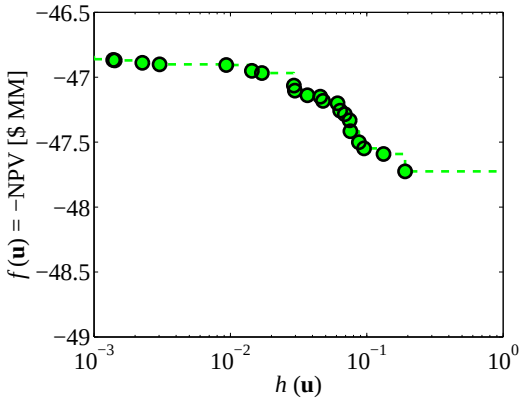


(d) Water cut for Producer 1

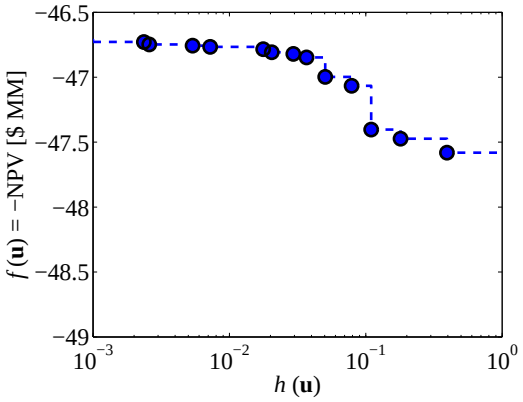


(e) Water cut for Producer 2

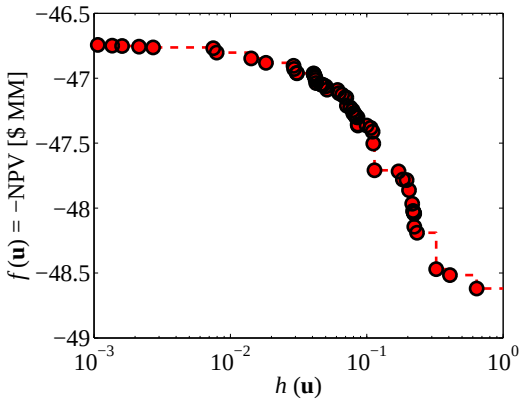
Figure 3.7: Field rates and well water cuts for the best optimized solution (PSO Run 2). Constraint limits shown as red dashed lines.



(a) Best PSO (Run 2) final filter



(b) Best MADS (Run 4) final filter



(c) Best PSO-MADS (Run 4) final filter

Figure 3.8: Final filters from best PSO, MADS and PSO-MADS runs, excluding points dominated by best feasible solution.

with the values in Table 3.3, the best feasible solution for PSO shown in Figure 3.8(a) is \$46.84 million (intersection of green dashed line with y-axis of plot). The green points in the plot constitute actual control schemes that are infeasible and that define the filter. The filters for all three methods clearly illustrate the tradeoff between the objective function, $f = -\text{NPV}$ in (3.1), and the aggregate constraint violation, h in (2.5). For instance, it is evident from Figure 3.8(a) that the objective function value can be improved by 1.5%, to \$47.55 million, by allowing an aggregate constraint violation of 10%, and by 1.9%, to \$47.73 million, by allowing a constraint violation of 20%. Interestingly, PSO-MADS has found a point for which a 20% constraint

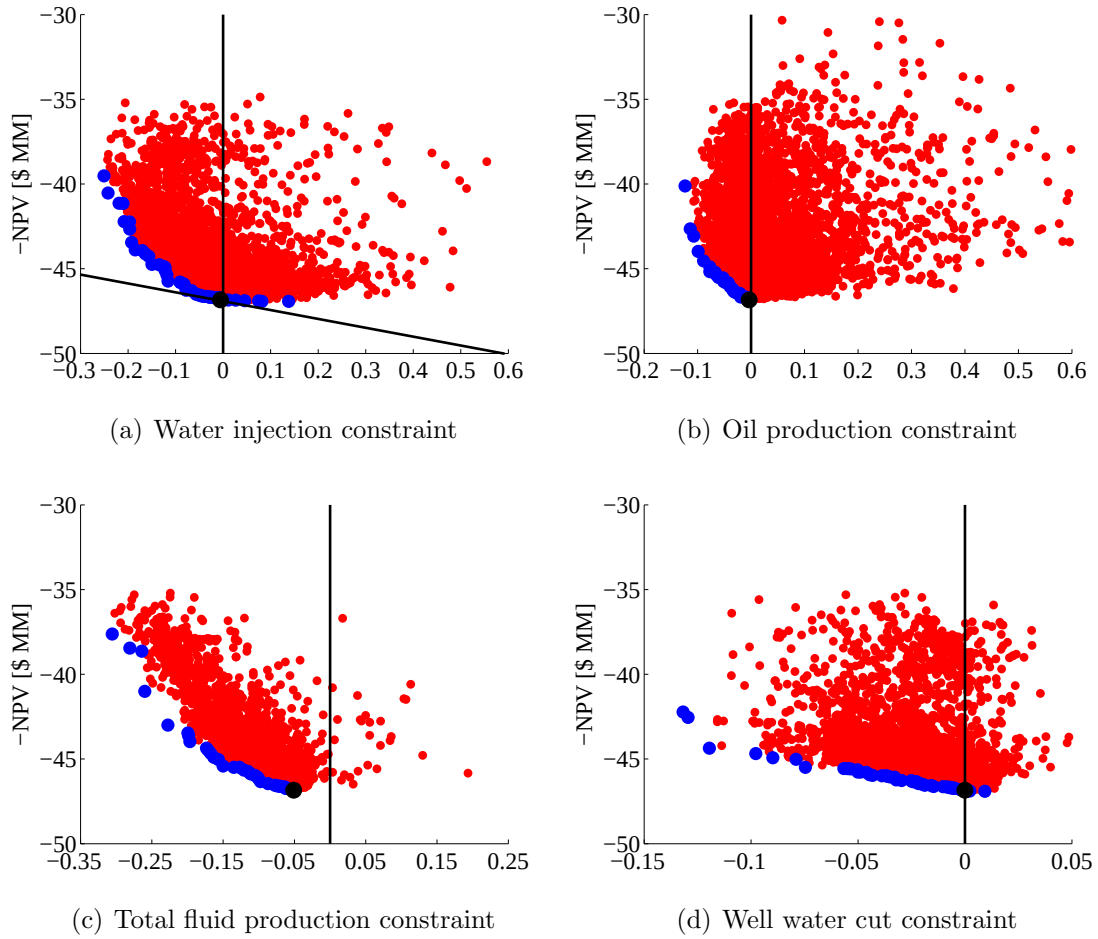
violation leads to an NPV of \$48 million (see Figure 3.8(c)).

The sensitivity of the optimal objective function value to the individual constraints is also of interest. In order to obtain this information, we must post-process the results from the optimization procedure. This post-processing, which introduces negligible computational cost, is as follows. During the optimization, all evaluated search or poll points are saved. For each evaluated point, $\mathbf{x}^i$, the objective function and normalized constraint values are also saved, as follows:

$$\mathbf{x}^i \rightarrow \left[f\left(\mathbf{x}^i\right) \;\; \bar{c}_1\left(\mathbf{x}^i\right) \;\; \bar{c}_2\left(\mathbf{x}^i\right) \;\; \ldots \;\; \bar{c}_m\left(\mathbf{x}^i\right) \right], \tag{3.2}$$

where m is the number of nonlinear constraints. The point $\mathbf{x}^i$ is feasible if $\bar{c}_j\left(\mathbf{x}^i\right) \leq 0 \;\forall\; j \in \{1, 2, \ldots, m\}$ and infeasible if any $\bar{c}_j\left(\mathbf{x}^i\right) > 0$. During post-processing, for each constraint c_j we plot points in objective function versus constraint space that are feasible in terms of every other constraint c_k such that $k \neq j$. That is, when creating the sensitivity plot for constraint j, we plot points $\left(f\left(\mathbf{x}^i\right), \bar{c}_j\left(\mathbf{x}^i\right)\right)$ for all saved $\mathbf{x}^i$ such that $\bar{c}_k\left(\mathbf{x}^i\right) \leq 0 \;\forall\; k \neq j$.

We thus obtain, for PSO Run 2, the four constraint sensitivity plots presented in Figure 3.9. For each plot in Figure 3.9, the red dots are the points for the particular constraint when all other constraints are feasible. The blue points show the set of nondominated points that dominate all other points in the space. These blue points quantify the explicit tradeoff of the objective function value and a particular constraint. The black dots identify the optimal solution (in this case the solution that yields an NPV of \$46.84 million). The vertical black lines designate the feasibility boundary for each constraint (where $\bar{c}_j = 0$). The points to the left of these lines ($\bar{c}_j \leq 0$) are feasible for all constraints, and the points to the right of the lines ($\bar{c}_j > 0$) are feasible for all other constraints except constraint j. Since the points to the left of the lines indicate feasibility for all constraints, all plots have the same number of points to the left of the constraint boundaries, i.e., all the feasible points generated during the optimization process. Tightening a specific constraint would imply moving the feasibility boundary (black line) to the left, while relaxing the constraint would imply moving the feasibility boundary to the right.



(a) Water injection constraint

(b) Oil production constraint

(c) Total fluid production constraint

(d) Well water cut constraint

Figure 3.9: Plots displaying information about objective function sensitivity to the different specified constraints (set of nondominated points shown in blue and all other evaluated points in red).

Significant information can be gleaned from the sensitivity plots in Figure 3.9. First, we can easily identify the active constraints at the optimal solution as the constraints with plots where the optimal solution (black dot) is aligned with the constraint boundary (black line). Here, we see that the water injection, oil production and well water cut constraints are the active constraints, consistent with Figure 3.7. Second, we can obtain information about the ease with which the optimizer was able to satisfy each constraint. The plots with more points to the left of the constraint boundary indicate that particular constraint was easier to satisfy. This implies that the total

fluid production rate constraint, Figure 3.9(c), was relatively easy to satisfy, while the oil production rate constraint, Figure 3.9(b), was more difficult to satisfy.

Third, and perhaps most importantly, we can see the sensitivity of the optimal objective function to each constraint. With reference to the primal-dual formulation of constrained optimization problems [90], the slope of the curve approximated by the set of nondominating points at the optimal solution corresponds to the Lagrange multiplier, λ, for that constraint. The Lagrange multiplier represents the quantitative sensitivity of the optimal objective function value to the specified constraint limit ($c_{\max}$ or $c_{\min}$ in (2.6) or (2.7)). Specifically, the Lagrange multiplier for the water injection constraint is the slope of the tangent line depicted in Figure 3.9(a). This Lagrange multiplier, which is only nonzero for active constraints and can be obtained from some gradient-based optimizers, is important because it shows how the optimal objective function value will change with changes to the constraint limit (e.g., increasing or decreasing the 600 STB/day maximum water injection constraint limit).

The plots in Figure 3.9 provide more information than is available from the Lagrange multiplier, however, since they give an explicit description of the objective function sensitivity to the constraint over a range, rather than just at the optimal solution. In addition, constraints in which the set of nondominating points and the optimal solution are to the left of the feasibility boundary (Figure 3.9(c)) can be tightened without changing the optimal solution. In our case, for example, the maximum total fluid production constraint can be tightened by 5%, from the specified 600 STB/day to 570 STB/day, without changing the optimal solution. This knowledge might conceivably enable some facilities to be resized.

Plots that have some points in the nondominating set (blue dots) to the right of the feasibility boundary indicate where we might achieve improvements in the objective function value by relaxing the constraint. We see here that the 1.5% increase in NPV can be achieved by relaxing the maximum water injection constraint (Figure 3.9(a)) by 10% from 600 STB/day to 660 STB/day. Figure 3.9(b) shows that we cannot improve the objective function by relaxing the oil production constraint, but we can quantify the decrease in NPV, e.g., we would reduce the NPV to \$45 million if we tightened this minimum constraint by 10%, requiring oil production of 220 STB/day

instead of the specified 200 STB/day.

From this analysis we see that, in addition to an optimal control scheme, we can obtain some potentially useful information from our optimizations regarding constraint sensitivities. This information could affect development decisions. We emphasize that these sensitivities are obtained at essentially no cost since the constraint plots are generated from points that are evaluated and saved during the optimization process.

Chapter 4

Joint Well Placement and Control

In Chapter 3, we presented a well control example with only continuous variables that represented the BHP control scheme for a reservoir with wells that have already been drilled. In this chapter, we present results using our optimization framework, which includes the MADS, PSO and PSO-MADS methods, for the joint well placement and well control problem.

In order to motivate our use of derivative-free optimization methods with global search capabilities, consider the following well placement optimization problem. We have four production wells fixed at the corners of a reservoir as shown in Figure 4.1, and we seek to optimize the location of the single injection well. This well location problem will be solved for each of the four different reservoir models shown in Figure 4.2.

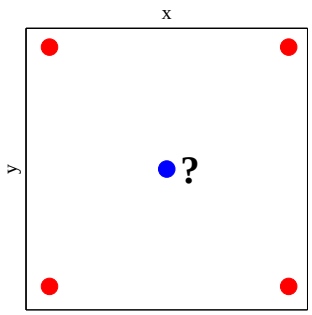
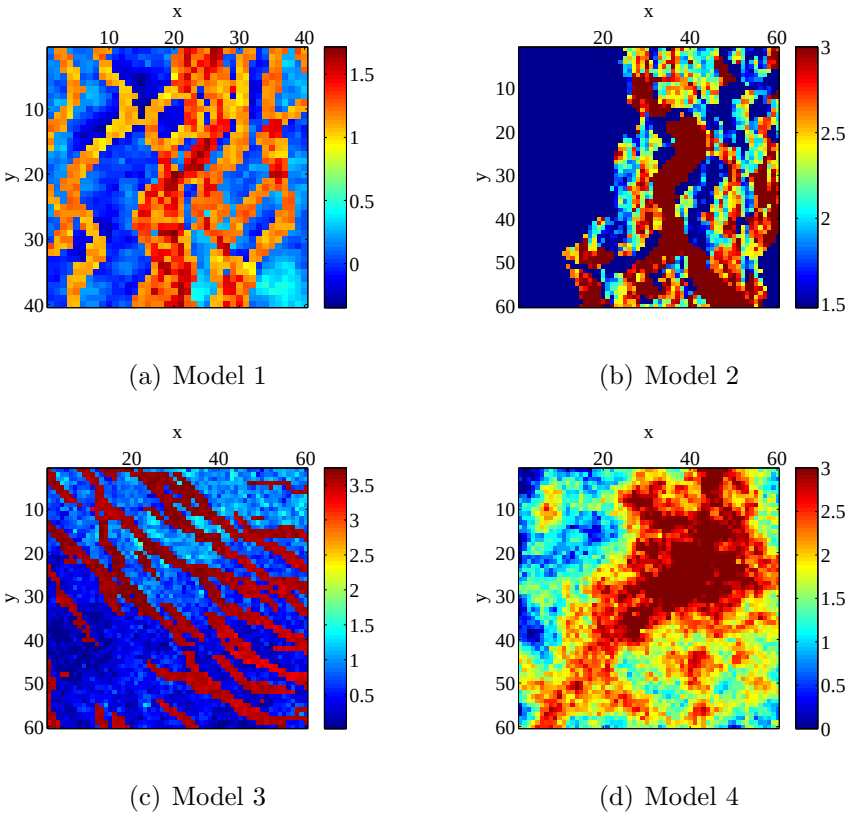


Figure 4.1: Problem schematic – fixed production wells (red) and injection well (blue) with location to be determined.



(a) Model 1

(b) Model 2

(c) Model 3

(d) Model 4

Figure 4.2: Permeability fields ($\log_{10} k$, with k in md) for four different reservoir models.

The BHPs of the four existing production wells and the injection well to be placed are all fixed. The location of the injection well is fully described by two variables (x and y grid-block indices), and these are the only variables in the optimization problem (i.e., the search space is two-dimensional). Note that if the grid blocks were numbered from 1 to $N_x \times N_y$, where N_x and N_y represent the number of grid blocks in the x and y directions, we would have only one optimization variable, though the size of the search space in that single dimension would of course be larger. Thus for Models 2 – 4, there are only 3596 possible injection well locations (the models are 60×60 and there are four production wells), so we can construct the entire NPV surface with 3596 simulations (note Model 1 is 40×40 and requires 1596 simulations). These are the objective function surfaces that the optimization algorithm would need to explore.

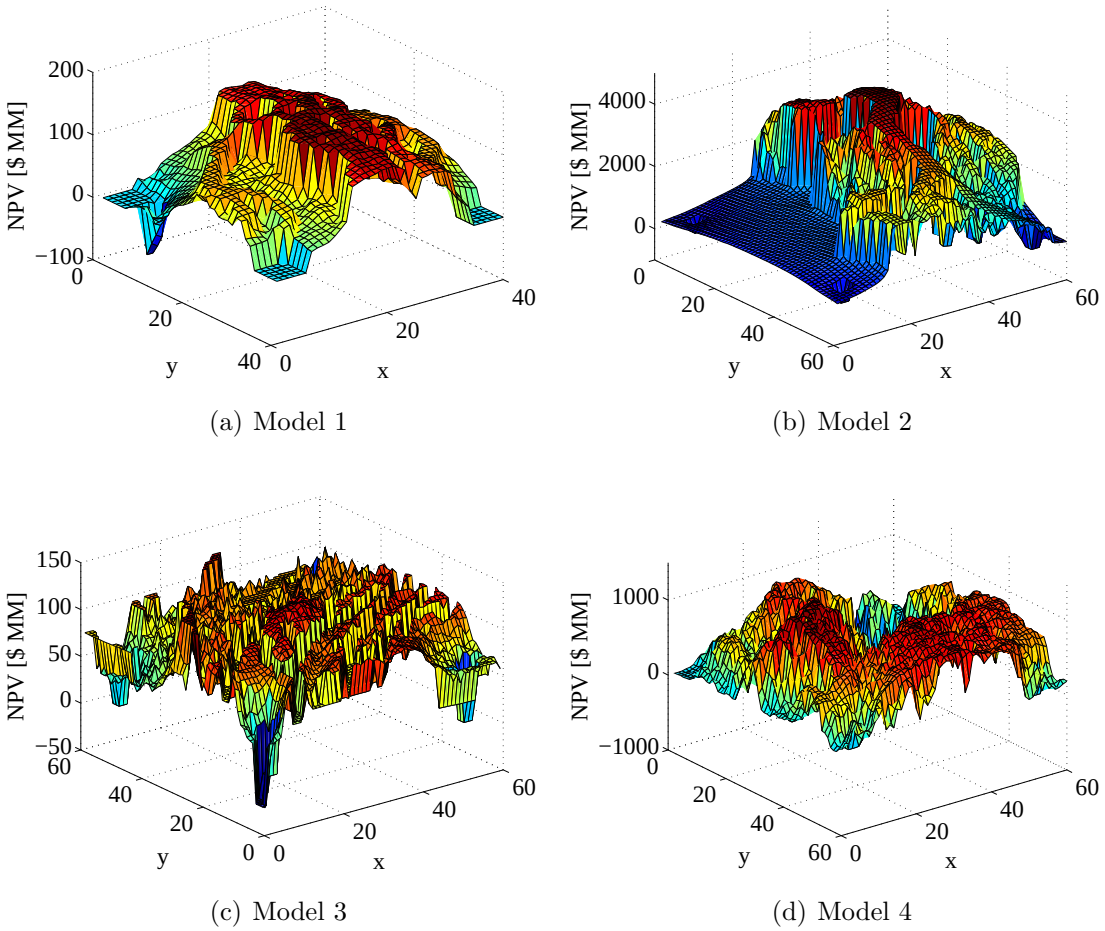


Figure 4.3: Resulting NPV surfaces for the injection well location problem using the four different reservoir models depicted in Figure 4.2.

The NPV surfaces for the four models are shown in Figure 4.3. From these plots, we see that even for a simple well placement problem with only two variables, we obtain very complex objective function surfaces, which the optimization algorithm will have to navigate. These surfaces are discontinuous due to the heterogeneity in the geological models, and they display multiple local optima. The objective function surfaces may change significantly when we consider multiple wells and the joint optimization of well location and control. We still expect complex and potentially discontinuous objective function surfaces, however. This motivates the use of derivative-free optimizers with global search capabilities that still provide local convergence. Thus our

selection of the MADS (for local convergence) and PSO (for global search character) methods, and the creation of the PSO-MADS hybrid, seem to be appropriate for this problem.

4.1 Optimization Problem and Solution Approaches

The optimization problem considered here entails the determination of the optimal locations of some number of injection and production wells together with their optimal well settings (time-varying BHPs). The MINLP optimization presented in (2.1) thus reduces to the NLP problem given by:

$$\min_{\mathbf{u}\in U,\ \mathbf{v}\in V} f(\mathbf{p},\mathbf{u},\mathbf{v}) = -\mathrm{NPV}(\mathbf{p},\mathbf{u},\mathbf{v}) = \underbrace{\sum_{j=1}^{N_I+N_P} C^j}_{\text{drilling costs}} + \sum_{j=1}^{N_I}\sum_{k=1}^{N_t} \Delta t_k \underbrace{c_{iw} q_{j,k}^{iw}(\mathbf{p},\mathbf{u},\mathbf{v})}_{\text{water injection cost}} -$$
$$\sum_{j=N_I+1}^{N_I+N_P}\sum_{k=1}^{N_t} \Delta t_k \left(\underbrace{p_o q_{j,k}^{o}(\mathbf{p},\mathbf{u},\mathbf{v})}_{\text{oil revenue}} - \underbrace{c_{pw} q_{j,k}^{pw}(\mathbf{p},\mathbf{u},\mathbf{v})}_{\text{water disposal cost}} \right),$$
$$\text{subject to } \ \mathbf{g}(\mathbf{p},\mathbf{u},\mathbf{v}) = \mathbf{0},\ \mathbf{c}(\mathbf{p},\mathbf{u},\mathbf{v}) \le \mathbf{0}, \tag{4.1}$$

where C^j is the cost to drill well j, N_t is the number of time steps in the reservoir simulation, Δt_k represents the time step size at time step k, c_{iw} and c_{pw} are the costs per barrel of injected and produced water, and p_o is the sale price of the produced oil. The terms $q_{iw}^{j,k}, q_{pw}^{j,k}$ and $q_o^{j,k}$ represent the rates of injected and produced water, and produced oil, from well j in time step k. These rates are functions of the optimization variables and are obtained from the reservoir simulator. Note that, in the examples in this chapter, we seek to optimize undiscounted NPV.

4.1.1 Sequential Solution Approach

The well placement and control problems have traditionally been addressed in a decoupled manner, with the well placement part solved first and the well control

optimization solved second, in contrast to the joint optimization in (4.1). Using a sequential procedure, the well placement variables are optimized with an assumed control strategy by solving

$$\mathbf{v}_S^* = \arg\min_{\mathbf{v}\in V} \; f\left(\mathbf{u}_0, \mathbf{v}\right), \quad \text{subject to} \;\; \mathbf{c}\left(\mathbf{u}_0, \mathbf{v}\right) \leq \mathbf{0}, \tag{4.2}$$

where $\mathbf{u}_0 \in U$ defines the assumed control strategy. Possible control strategies include using constant BHPs (typically set at the bounds, implying wells injecting or producing at maximum rates) or the potentially more effective 'reactive' control strategy, which is used in this work when we perform sequential optimizations. Under the reactive control strategy considered here, injection wells always operate at their maximum BHP bounds. Production wells operate at their minimum BHP bounds until a prescribed limit is reached. This limit can be defined in terms of a maximum allowable fraction of water in the produced fluid, or (analogously) when the water production cost ($c_{pw} q_{pw}^{j,k}$) exceeds oil production revenue ($p_o q_o^{j,k}$) for the well. Although it can give reasonable results in some cases, reactive control represents a heuristic treatment that will, in general, be suboptimal. After the solution of (4.2), the well locations are fixed at $\mathbf{v}_S^*$ and the following well control optimization problem is solved:

$$\mathbf{u}_S^* = \arg\min_{\mathbf{u}\in U} \; f\left(\mathbf{u}, \mathbf{v}_S^*\right), \quad \text{subject to} \;\; \mathbf{c}\left(\mathbf{u}, \mathbf{v}_S^*\right) \leq \mathbf{0}. \tag{4.3}$$

As discussed in the Chapter 1, a number of different optimization techniques – both gradient-based and derivative-free – have been applied for the problems defined by (4.2) and (4.3).

4.1.2 Joint Solution Approach

The sequential approach defined above has the advantage of solving two smaller problems (well placement, of dimension n_1, and well control, of dimension n_2) instead of one large problem (of dimension $n_1 + n_2$). As shown by Bellout et al. [17] and Zandvliet et al. [127], the controls applied during the well placement optimization affect the optimized well locations. Thus, any approach that does not optimize the location and control variables simultaneously can be expected to be suboptimal. This motivates the joint optimization of the problem defined in (4.1).

As noted in Chapter 1, joint optimization approaches have been developed previously. Bellout et al. [17] presented an approach based on a nested bi-level optimization. In that method, well placement is the master problem, and in order to evaluate the objective function associated with a particular configuration of wells, the well controls are optimized to a certain degree. This approach allows the use of different optimization methods for the two problems, and in the Bellout et al. [17] implementation, the well placement optimization was accomplished using derivative-free direct search procedures, while the well control subproblem was addressed with an efficient adjoint-based gradient technique. Li and Jafarpour [78] used an iterative procedure in which they alternated between optimizing well placement and well control. Again, different optimization approaches were used for the two problems. The results presented in both studies demonstrated the advantages of joint optimization compared to sequential procedures.

Our approach can be seen as an alternative to these earlier treatments, but we also introduce several important extensions. We solve the joint well placement and control problem with a single optimization method, in contrast to the earlier procedures that addressed the joint problem but used different treatments for the two subproblems. We also include general (nonlinear) constraints in our framework (which do not appear to have been considered previously for the joint problem), and present results using our PSO-MADS hybrid optimization procedure.

4.2 Example Cases

The methods described in Chapter 2 will now be applied to solve the optimization problem defined by (4.1) for a synthetic field subject to waterflooding. The geological model is represented on a two-dimensional 60×60 grid. The permeability field is shown in Figure 4.4, together with an initial guess for the locations of the five wells (two injection and three production wells) used in the examples. Figure 4.5 shows the relative permeability curves used in the simulations, and Table 4.1 presents other key simulation and optimization parameters.

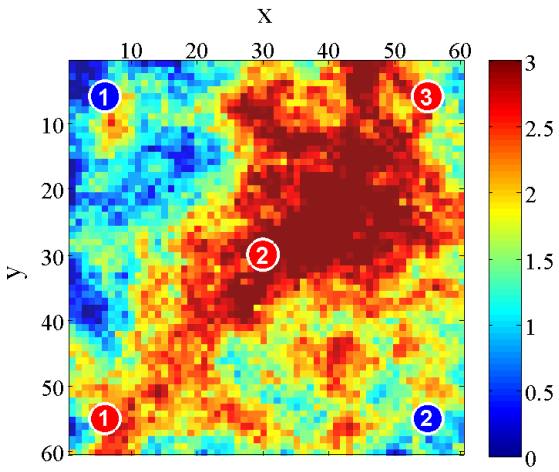


Figure 4.4: Geological model ($\log_{10}$ of isotropic permeability field, with permeability expressed in md) used for all Chapter 4 examples, showing initial-guess injection (in blue) and production (in red) well locations.

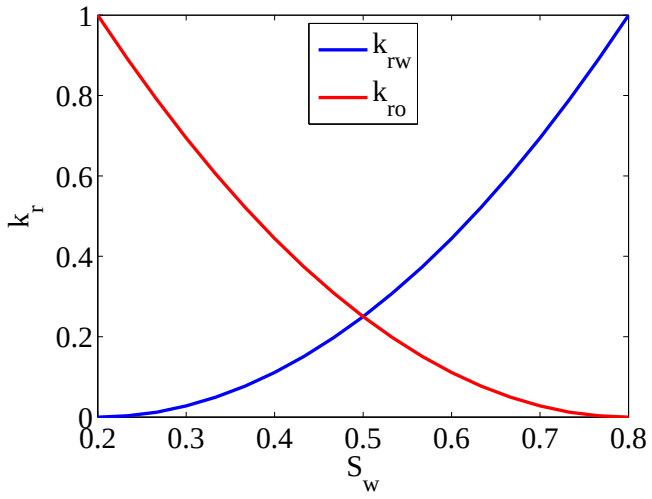


Figure 4.5: Relative permeability curves for the oil and water phases.

The permeability field shown in Figure 4.4 is the same as that used by Bellout et al. [17] in their example cases, though other aspects of the problem are different. The production time frame is 2,920 days, with the well BHPs updated every 584 days, for a total of five control intervals. The BHP is held constant over each control interval.

Table 4.1: Simulation and optimization parameters for joint well placement and control examples

Grid cell dimensions	130 ft × 130 ft × 20 ft
Initial pressure p_i, at datum	4012 psi at 8620 ft
μ_o and μ_w at p_i	0.5 and 0.3 cp
ρ_o and ρ_w	53.1 and 62.4 lbm/ft^3
B_o and B_w at p_i	1.00 RB/STB
p_o, c_{pw} and c_{iw}	\$80, \$10 and \$5/STB
Drilling cost	\$20 million per well
Injection BHP range	4100 - 6000 psi
Production BHP range	1000 - 3500 psi
Maximum water injection rate	9000 STB/day
Minimum oil production rate	4000 STB/day
Maximum fluid production rate	9000 STB/day
Maximum well water cut	0.7
Minimum well-to-well distance	1300 ft

Results for three cases will be presented. The first case involves only bound constraints, the second case additionally incorporates the nonlinear constraints listed in Table 4.1 (last five rows), and the third case includes the nonlinear constraints plus binary optimization variables that allow us to also determine the optimum number of wells. The total number of optimization variables n for Cases 1 and 2 is 35 (two areal location variables and five control variables for each of the five wells), while in the third case there are 40 (an additional binary variable for each well).

4.2.1 Case 1: Bound Constraints Only

For this case, the nonlinear field rate and well water cut constraints are not included in the optimization. Thus, the problem involves only the bound constraints on well BHPs. The MADS, PSO and PSO-MADS methods are applied to this problem. For each MADS iteration, a maximum of $2n = 70$ function evaluations is performed in the polling process (fewer if previously evaluated points have been saved). The initial MADS mesh sizes correspond to 20% of the variable ranges. In the standalone PSO iterations, a swarm size of 50 particles is used, which means that a maximum of 50 function evaluations is performed in each PSO iteration. The same parameters used

for the standalone PSO and MADS iterations are applied for the hybrid PSO-MADS method. The function evaluations in all three methods are parallelized using a computing cluster with sufficient cores to perform all function evaluations concurrently at each iteration.

Considering the stochastic nature of these algorithms and the fact that the optimization surface is expected to contain multiple optima, each of the three methods is run five times starting from five different initial guesses. The NPVs for the five initial guesses, together with their mean and standard deviation, designated <NPV> and σ, are shown in Table 4.2. The first of these runs has as the initial guess the well locations indicated in Figure 4.4, with the injection well BHPs at their upper bounds (6000 psi) and production well BHPs at their lower bounds (1000 psi). The remaining four initial guesses are randomly generated from a uniform distribution within the bounds of the problem.

Table 4.2: NPVs from the five initial guesses used in the optimizations (best value is italicized)

Run #	Initial guess NPV($\mathbf{u}, \mathbf{v}$) [$ MM]
1	*1015*
2	554
3	283
4	-663
5	580
<NPV>	354
σ	626

The optimization results for the three derivative-free methods are summarized in Figure 4.6 and Table 4.3. Figure 4.6 shows the NPV evolution of the best solution versus the number of simulations, averaged over the five runs, for the three methods. From this figure, we see that the PSO-MADS hybrid (red curve) outperforms its component methods. Table 4.3 presents the final optimized NPVs for all of the runs, together with the mean and standard deviation of the optimized NPVs over the five runs. The best NPV from each method is italicized in the table.

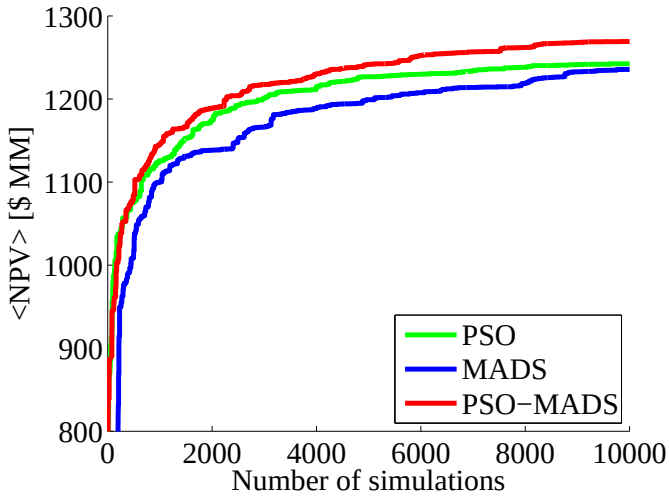


Figure 4.6: Evolution of mean NPV for the five runs (Case 1).

Table 4.3: Final NPVs from five runs for the three different methods (Case 1, best values are italicized)

Run #	PSO NPV [$ MM]	MADS NPV [$ MM]	PSO-MADS NPV [$ MM]
1	*1283*	*1297*	*1304*
2	1257	1083	1247
3	1263	1262	1289
4	1239	1247	1262
5	1169	1289	1244
<NPV>	1242	1236	1269
σ	44	88	26

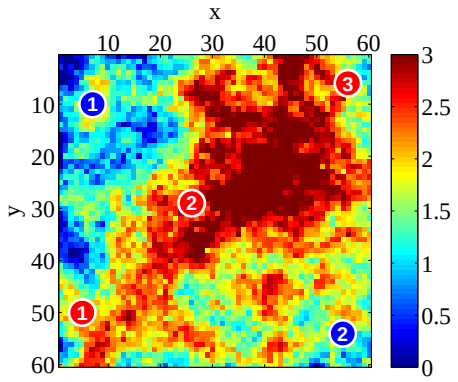
Figure 4.6 and Table 4.3 highlight some of the characteristics of the three methods. As a result of its global search nature, PSO is able to avoid poor local optima and provide fairly robust solutions (evident from the relatively small σ in Table 4.3). However, in contrast to MADS, the PSO algorithm is not guaranteed to converge to a local minimum, although we do expect eventual 'convergence' to the so-called center of attraction [45]. MADS results, on the other hand, can depend strongly on the initial guess, and because of its local search nature, MADS may converge to

poor local optima, as is the case for the second run. Since the PSO-MADS hybrid combines some of the advantages of the PSO and MADS algorithms, it displays strong convergence to better quality solutions than standalone MADS and PSO, along with better robustness features than PSO, which is evident from the smaller σ for PSO-MADS. We note additionally that, even though the total number of simulations required for these optimizations is around 10,000 (see Figure 4.6), the number of equivalent simulations using 50 processors, which gives an indication of elapsed time, is ideally only around 200. This represents a manageable level of computation.

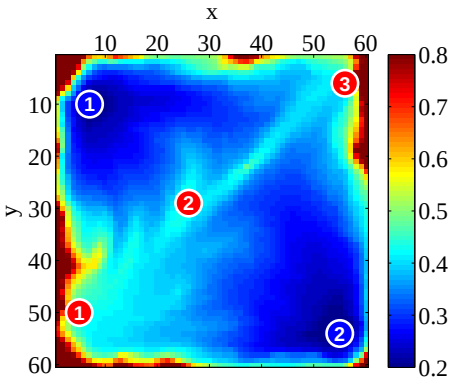
Figure 4.7 shows the optimal well locations with the final oil saturation map, together with the injection and production well BHP controls, and resulting injection and production well rates for the best PSO-MADS solution (Run 1 with NPV of \$1304 million). Comparing Figure 4.7(a) to Figure 4.4 we see that the optimized well locations are close to the initial guess locations, implying that the initial guess was quite reasonable (note that the best NPVs for all three methods correspond to the runs with this initial guess). The slight shifts in well locations, together with the modifications to the well controls shown in Figures 4.7(c) and 4.7(d), account for the observed 17% improvement in NPV from the initial guess.

The results presented in Figure 4.6 and Table 4.3 are obtained by solving the combined well placement and control problem in the joint fashion proposed in (4.1). In order to compare our joint approach to a sequential procedure, we also solve the problem sequentially. In this case we first address the well placement problem (4.2), using reactive controls (wells operate at their BHP limits, with production wells closed when water cut exceeds an economic limit of 0.9) and PSO-MADS with the same parameters as used for the joint solution. Then, with the optimized locations from (4.2), we solve the well control problem (4.3), again using PSO-MADS. We run the sequential approach five times with the same initial guesses as were used for the joint optimizations.

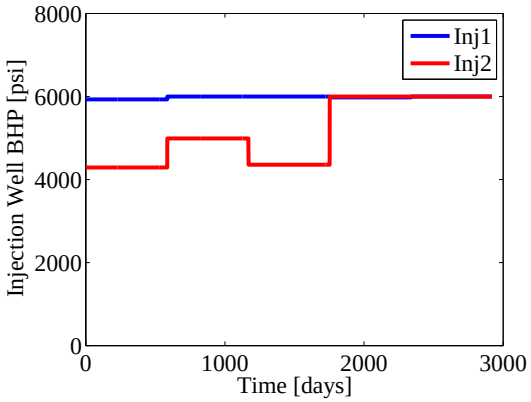
Figure 4.8 displays the average performance from the sequential and joint approaches, while Table 4.4 presents the final optimized NPVs for all five runs, together with the mean and standard deviation of the optimized NPVs. Since the sequential method involves solving two smaller optimization problems (the well placement problem has



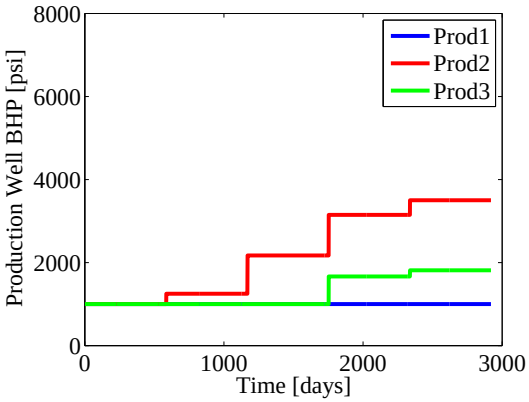
(a) Optimized well locations



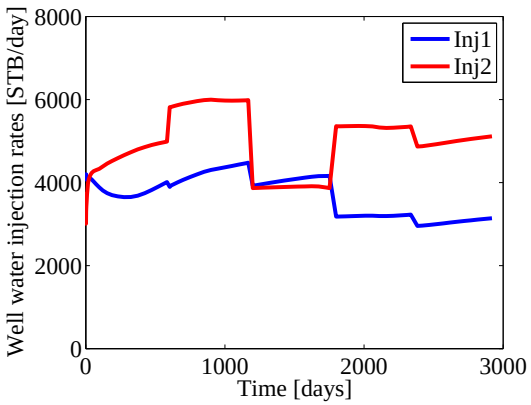
(b) Final oil saturation map



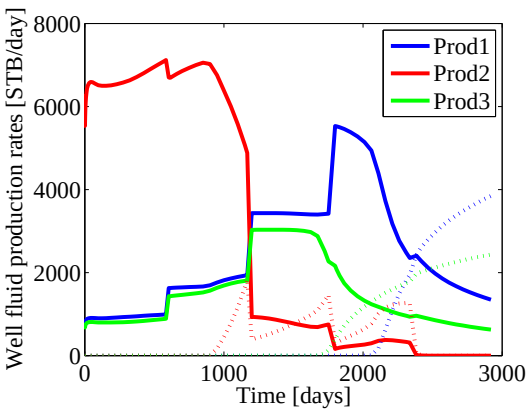
(c) Optimized injection BHPs



(d) Optimized production BHPs



(e) Resulting injection rates



(f) Resulting oil (solid) and water (dotted) production rates

Figure 4.7: Best PSO-MADS solution, showing well locations, final oil saturation, BHP versus time profiles, and resulting injection and production rates (Case 1).

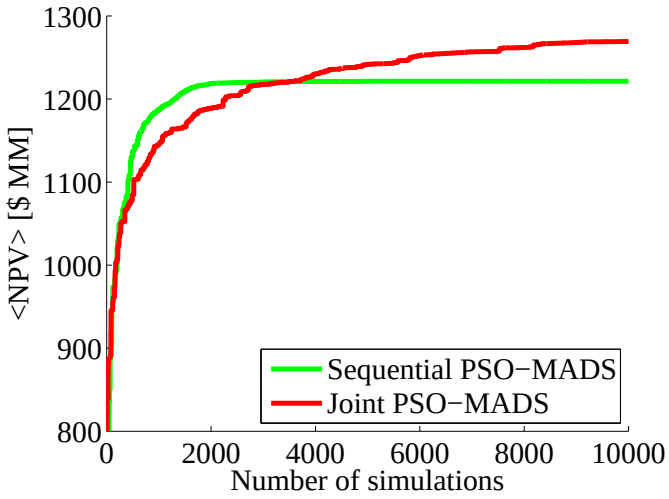


Figure 4.8: Evolution of mean NPV from five PSO-MADS runs for sequential and joint procedures (Case 1).

Table 4.4: Final NPVs from five runs for sequential and joint procedures (Case 1, best values are italicized)

Run #	PSO-MADS seq. NPV [$ MM]	PSO-MADS joint NPV [$ MM]
1	*1290*	*1304*
2	1235	1247
3	1132	1289
4	1205	1262
5	1245	1244
<NPV>	1221	1269
σ	59	26

10 variables and the control problem has 25 variables, compared to 35 variables for the joint approach), it exhibits faster overall convergence (see Figure 4.8). Despite the faster convergence, the solutions from the sequential procedure display lower NPVs on average than those from the joint approach. This is because, in contrast to the joint procedure, the sequential approach does not completely capture the coupling between the well placement and well control problems. In addition, the sequential approach appears to be less robust, as is evident from the larger σ in Table 4.4. Our observation that the joint approach provides better solutions than the sequential

procedure is consistent with the results of Bellout et al. [17] and Li and Jafarpour [78], and we expect this to be the case in other problems as well.

Although our joint optimization procedure differs in several respects from that applied by Bellout et al. [17], it is nonetheless instructive to compare our results with theirs. In the second example in their paper, a problem very similar to that considered above was addressed (they used the permeability field shown in Figure 4.4, though they specified smaller grid block sizes, higher costs for produced and injected water, and they used more control steps). We modified our problem specification to enable solution of the same problem using our joint PSO-MADS hybrid optimization procedure. Bellout et al. [17] ran their optimization using different direct search procedures for well placement (the best results, in terms of average objective function value, were achieved using the Hooke-Jeeves direct search method) and an adjoint-based gradient technique for well control. Their results for nine optimization runs (using different initial guesses) show a mean NPV of \$363 million and a standard deviation of \$27 million over the nine runs. For the same case we achieved a mean NPV of \$361 million and a standard deviation of \$19 million (again over nine runs). Thus, the two procedures appear to be quite comparable in terms of the quality of results. Both of these mean NPVs are well above the mean NPV of about \$300 million reported for sequential optimization in this example [17].

Bellout et al. [17] required about 4,000 simulations for each of their optimization runs, whereas we use around 30,000 simulations for our runs (note that this problem contains 10 control steps, resulting in 60 optimization variables, which is more than the 35 optimization variables in Case 1). Their approach requires fewer runs because they apply an efficient gradient-based procedure for the well control optimization, with gradients computed using the adjoint technique of Sarma et al. [108]. Although it is very efficient, the adjoint approach is simulator invasive and does not easily parallelize. This means that part of their nested approach can be parallelized (the well placement part, solved with direct search procedures) and part of it cannot (the well control part, solved with a gradient-based technique). Our method, by contrast, is fully parallelizable and would effectively lead, if 50 computing cores were available on a cluster, to about 600 equivalent simulations instead of 30,000. Therefore, there are

clearly relative advantages and disadvantages between the approach used by Bellout et al. [17] and that applied here, and the method of choice will depend on the number of cores available, on access to simulation source code, and on the ability to implement and maintain an adjoint-based gradient computation code.

4.2.2 Case 2: Nonlinear Constraints

Case 1 above dealt with only bound constraints. We now treat a case that also includes nonlinear constraints, which render the problem more difficult. The nonlinear constraints considered are the well distance, field rate and well water cut constraints listed in Table 4.1. The rate and water cut constraints are nonlinear in nature because the relationship between individual well BHPs (the control variables) and the field rates and well water cuts involve reservoir simulation (i.e., nonlinear function evaluations). We use the same PSO, MADS and PSO-MADS parameters as in Case 1. The three algorithms, together with the filter constraint handling techniques implemented for each method (as described in Chapter 2), are applied to solve the joint well placement and control problem. We again run each method five times using the same initial guesses as in Case 1 (recall that one initial guess involves the well locations shown in Figure 4.4).

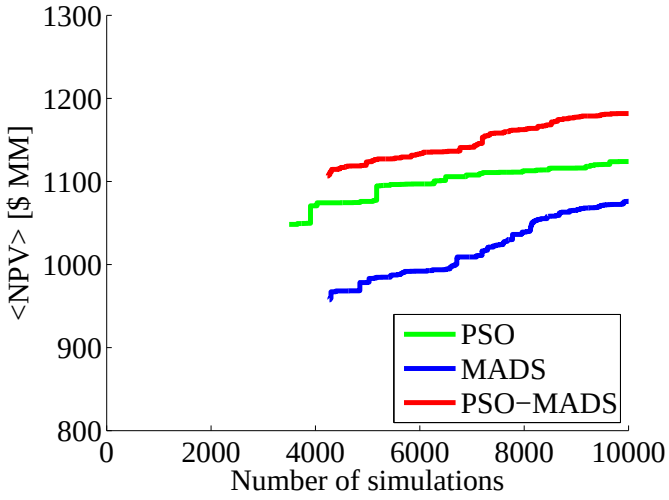


Figure 4.9: Evolution of mean NPV for the five runs (Case 2).

Table 4.5: Final NPVs from five runs for the three different methods (Case 2, best values are italicized)

Run #	PSO NPV [$ MM]	MADS NPV [$ MM]	PSO-MADS NPV [$ MM]
1	*1225*	1143	1206
2	1060	1032	*1228*
3	1063	972	1154
4	1138	1076	1200
5	1134	*1157*	1120
<NPV>	1124	1076	1182
σ	68	77	44

Figure 4.9 displays the evolution of mean NPV for the feasible solutions for the three methods tested, and the results for all runs are summarized in Table 4.5. The curves in Figure 4.9 do not appear until several thousand simulations have been performed because the initial guesses and the earlier simulations lead to infeasible solutions (i.e., solutions that violate the nonlinear constraints). From the results in Figure 4.9 and Table 4.5, it is again apparent that the PSO-MADS procedure outperforms its component methods. Note that the optimized NPVs achieved in this case are lower, for all runs, than those for Case 1 (shown in Table 4.3). This is consistent with the fact that this case involves a more constrained problem.

It is important to note that, even though the early solutions using all three methods are infeasible, through use of the filter method we improve the objective function value and reduce the constraint violation simultaneously. Hence, by the time a feasible solution is found, its NPV is already relatively high. Specifically, for this case the mean NPV for the five initial guesses (all of which are infeasible) is $354 million, while for all three algorithms, the mean NPV when feasible solutions appear (after about 4000 simulations) is in the range of $950–$1100 million. This observation highlights the effectiveness of the filter constraint handling techniques applied in this work.

The final filter from the best PSO-MADS joint optimization run is shown in Figure 4.10. Consistent with Table 4.5, the best feasible solution has an NPV of $1228

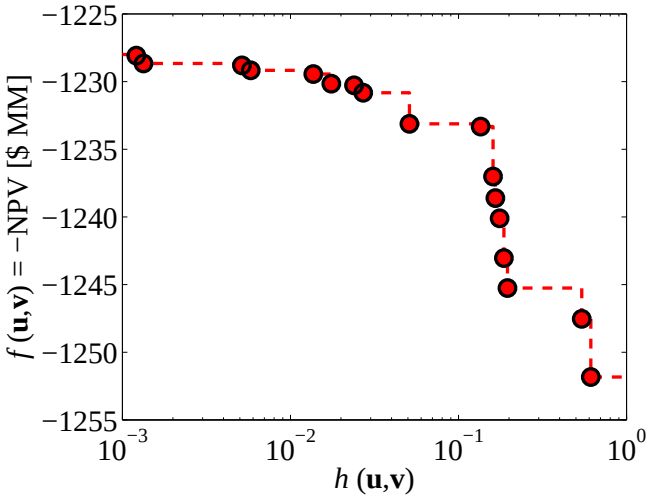


Figure 4.10: Final filter, excluding points dominated by the best feasible solution, from the best PSO-MADS joint optimization run (Case 2).

million. The points in the plot constitute the infeasible points that define the filter. This filter illustrates the tradeoff between the objective function, $f(\mathbf{u}, \mathbf{v})$, and the aggregate constraint violation, $h(\mathbf{u}, \mathbf{v})$. It quantifies how much the constraints need to be relaxed in order to improve the objective function by a certain amount. In this example, for most of the filter points with small $h(\mathbf{u}, \mathbf{v})$ values, only the minimum field oil production is violated. The filter in Figure 4.10 shows that if we are able to accommodate a constraint violation of about 20% (i.e., we allow the minimum oil production rate to be relaxed from 4000 STB/day to 3200 STB/day), there is a field development scenario where the NPV increases to $1245 million while satisfying the relaxed constraints.

As in Case 1, we also performed sequential (rather than joint) optimizations for Case 2, using the same optimization parameters. The PSO-MADS procedure was applied for both the well placement and well control stages. Figure 4.11 and Table 4.6 provide comparisons of the joint and sequential approaches. Four of the five sequential runs yield feasible solutions, with a mean NPV of $1105 million and a standard deviation of $116 million. This is inferior in terms of average performance and robustness to the results achieved using the joint approach (<NPV> of $1182 million, σ of $44

million). In addition, the best of the five solutions for the joint approach has an NPV that is 3.5% higher than that from the best solution for the sequential approach.

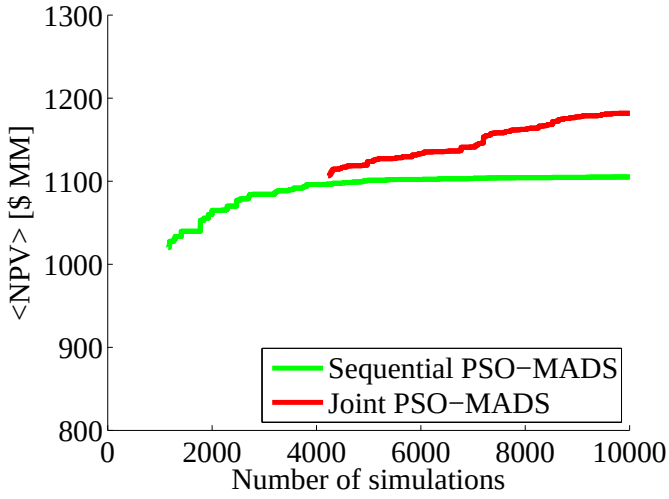
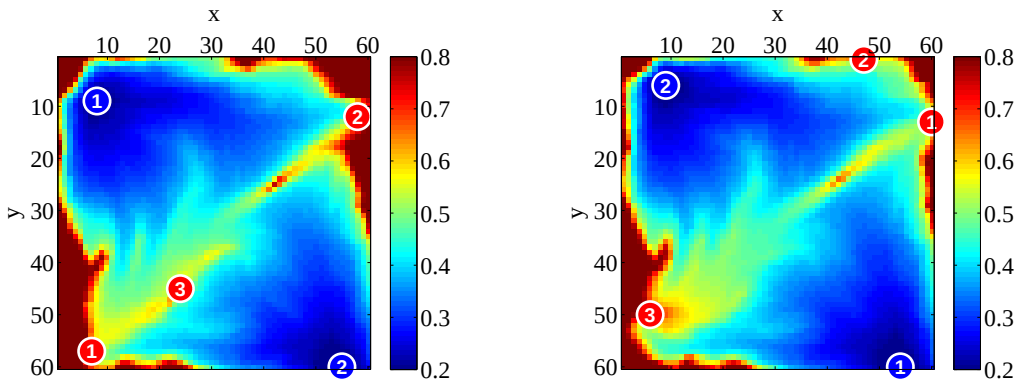


Figure 4.11: Evolution of mean NPV from five PSO-MADS runs for sequential and joint procedures (Case 2).

Table 4.6: Final NPVs from five runs for sequential and joint procedures (Case 2, best values are italicized)

Run #	PSO-MADS seq. NPV [$ MM]	PSO-MADS joint NPV [$ MM]
1	*1187*	1206
2	–	*1228*
3	1107	1154
4	941	1200
5	1186	1120
<NPV>	1105	1182
σ	116	44

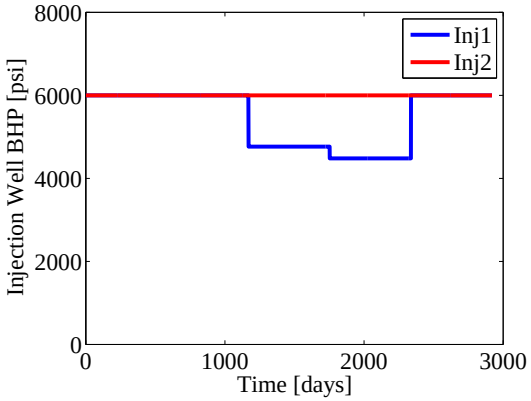


(a) Oil saturation from sequential approach (b) Oil saturation from joint approach

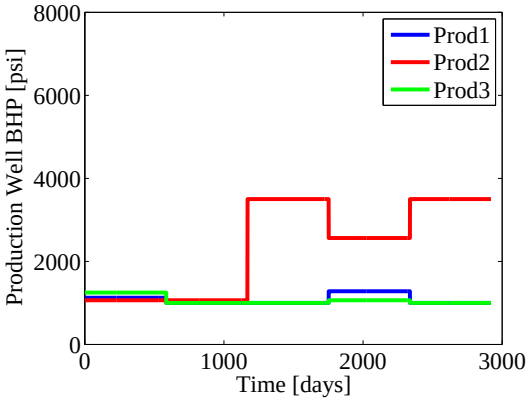
Figure 4.12: Final oil saturation maps (red indicates oil and blue water) and well configurations from the best PSO-MADS solutions for the sequential and joint approaches (Case 2).

In Figure 4.12 we present maps of the final oil saturation from the best solutions for both approaches. The well configurations are different between the two solutions, and it is evident that the solution for the joint approach provides slightly better overall sweep (compare, e.g., the saturation fields in the upper right corners in both figures).

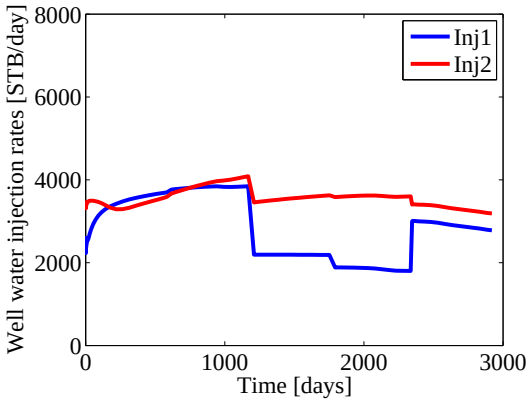
The well BHP controls and the resulting individual well rates for the best PSO-MADS solution for the joint approach are presented in Figure 4.13. The well BHPs display some similarities with the BHPs for the case without nonlinear constraints (Figures 4.7(c) and 4.7(d)), though there are clear differences in the detailed profiles. Figure 4.13(c) shows the well rates for the two injection wells in Figure 4.12(b), and Figure 4.13(d) displays the oil production (solid lines) and water production (dashed lines) rates for the three production wells. From these rate plots, we see that the two injection wells operate at similar rates, whereas the production well rates vary significantly.



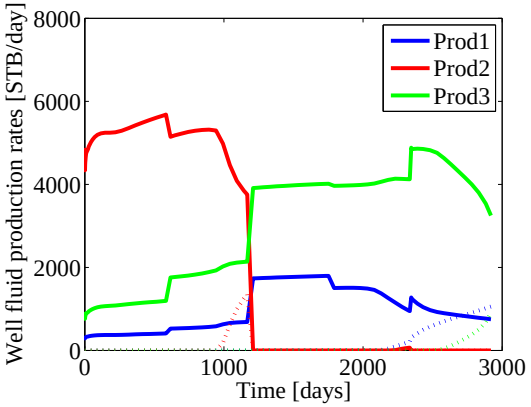
(a) Optimized injection BHPs



(b) Optimized production BHPs



(c) Resulting injection rates



(d) Resulting oil (solid) and water (dotted) production rates

Figure 4.13: Well BHPs and resulting injection and production rates for the best PSO-MADS solution for the joint approach (Case 2).

4.2.3 Case 3: Determination of Optimal Number of Wells

It is evident from the results in Figure 4.13 that some wells operate at higher rates than others. This leads us to question whether all five wells are required, and suggests that we optimize the number of wells along with well locations and controls. We thus consider a more general problem, as defined in (2.1) in Chapter 2, which includes binary variables (drill/do not drill) to enable the determination of the optimal number of wells. Here we treat the binary variables as regular integers (specifically, we round continuous values to either 0 or 1 in PSO and use only $\{0, 1\}$ mesh points in MADS).

Case 3, which we now consider, is identical to Case 2 in terms of problem specification and bound and nonlinear constraints, but it also includes binary categorical variables. We now have a total of 40 optimization variables: 10 integer well location variables, 25 continuous well control variables, and 5 binary categorical variables. The three joint optimization algorithms (PSO, MADS and PSO-MADS) are again each run five times from different initial guesses (one user-supplied and four randomly generated, which are different from those used in Cases 1 and 2 as we now have binary variables).

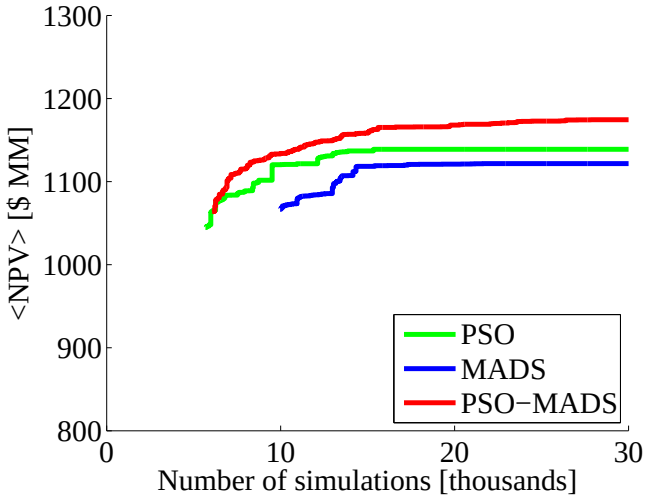


Figure 4.14: Evolution of mean NPV for the five runs (Case 3).

The results from these runs are presented in Figure 4.14 and Table 4.7. In this table, in addition to the final NPVs, we also show the optimized number of wells for each run. We again see that the PSO-MADS hybrid outperforms its component methods in terms of mean NPV and best solution, while providing similar robustness as PSO. In addition, it is interesting to note that the best results are obtained when only four wells are drilled instead of five, leading us to believe that for this case the optimal number of wells is indeed four (two injectors and two producers). Of the five runs for each method, the MADS algorithm converges to a solution with four wells once, the PSO algorithm twice, and the hybrid PSO-MADS method four times. This again highlights the superiority of PSO-MADS relative to its component methods. The inclusion of the binary variables leads to a more difficult optimization problem, as

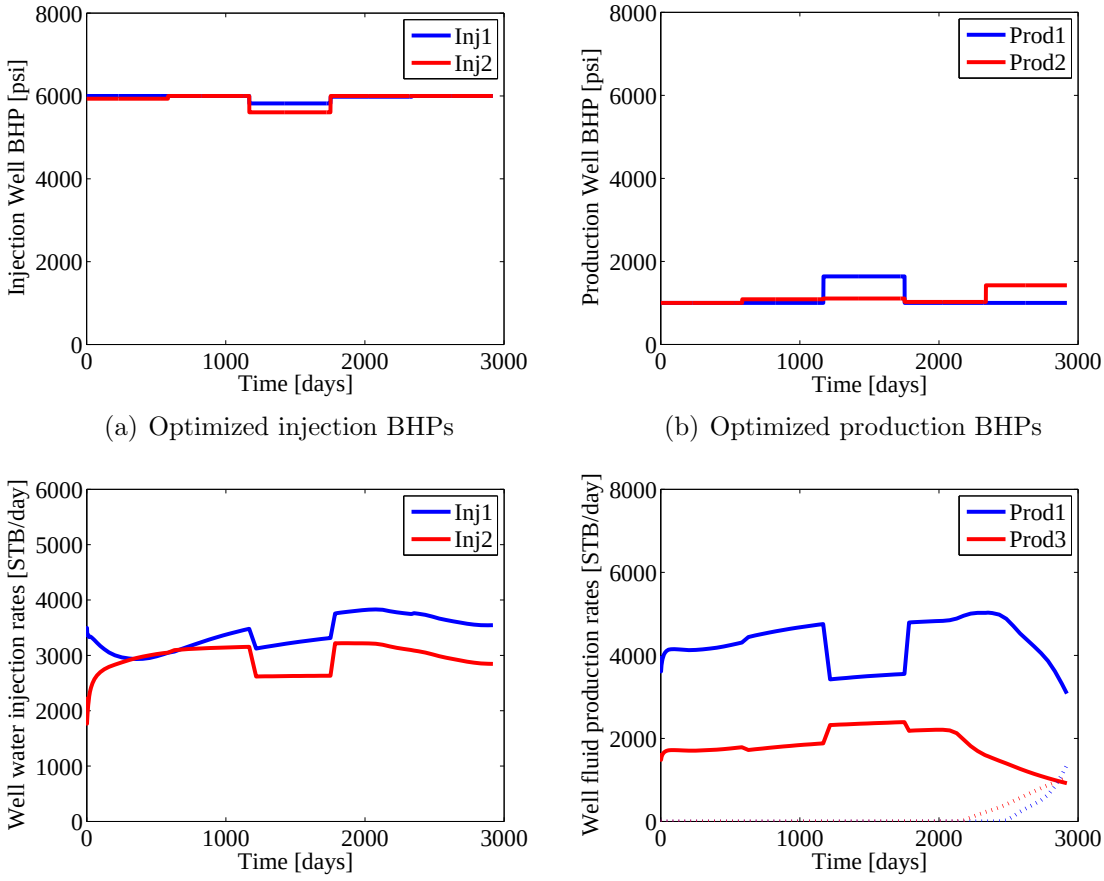
Table 4.7: Final NPVs from five runs for the three different methods (Case 3, best values are italicized). Number of wells for each run shown in parentheses

Run #	PSO NPV [$ MM] (# of wells)	MADS NPV [$ MM] (# of wells)	PSO-MADS NPV [$ MM] (# of wells)
1	1138 (5)	*1223* (4)	*1247* (4)
2	*1197* (4)	1127 (5)	1162 (4)
3	1151 (4)	1073 (5)	1158 (4)
4	1111 (5)	1194 (5)	1143 (5)
5	1096 (5)	991 (5)	1161 (4)
<NPV>	1139	1122	1174
σ	39	94	41

is evident from the fact that about 2–3 times more simulations are required than in Case 2 (20,000–30,000 simulations versus 10,000, as can be seen from Figures 4.9 and 4.14). This is perhaps not surprising since we are adding significant complexity to an already challenging problem.

The well BHP profiles and resulting injection and production rates for the best PSO-MADS solution (Run 1) are shown in Figure 4.15. Comparing the well controls and rates in Figure 4.15 to those in Figure 4.13 (five-well case), we see that the four-well solution has some attractive features; i.e., the wells operate close to their BHP bounds and the resulting well rates are less variable. In addition, water breakthrough is significantly delayed.

The optimized well configurations, with final oil saturation maps, for the worst, median and best PSO-MADS solutions, together with the initial guess configurations used for these runs, are presented in Figure 4.16. From these figures, we can see that even with an initial guess that leads to poor sweep and NPV, the PSO-MADS procedure is able to optimize the number of wells and their associated locations and controls to provide reasonable results. Comparing Figure 4.16(f) to Figure 4.12(b), we see that by including the binary variables in the optimization, we obtain a four-well solution that is comparable to the five-well solution in Case 2, except that one



(a) Optimized injection BHPs

(b) Optimized production BHPs

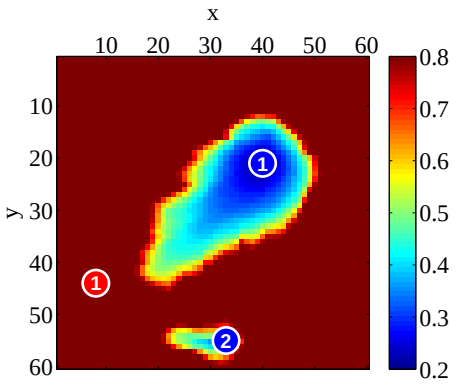
(c) Resulting injection rates

(d) Resulting oil (solid) and water (dotted) production rates

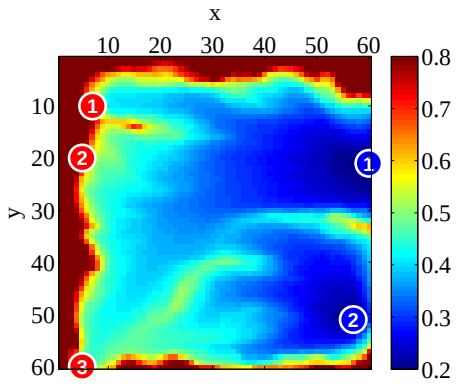
Figure 4.15: Well BHPs and resulting injection and production rates for the best PSO-MADS solution (Case 3).

of the production wells in the upper right portion of the model has been eliminated. Nonetheless, the sweep in the two cases appears quite comparable.

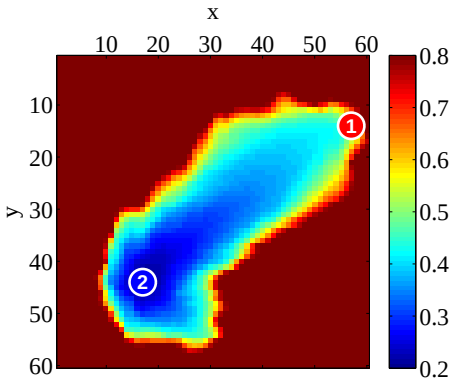
The NPV from the best PSO-MADS solution improves from \$1228 million for Case 2 (five wells) to \$1247 million for Case 3 (four wells). This increase in NPV of \$19 million corresponds closely to the cost of drilling one well (\$20 million). For Case 2, the best PSO-MADS solution yields a cumulative oil production of 17.9 million STB and cumulative water injection of 18.7 million STB; for Case 3, the corresponding values are 17.8 million STB of oil production and 18.5 million STB of water injection.



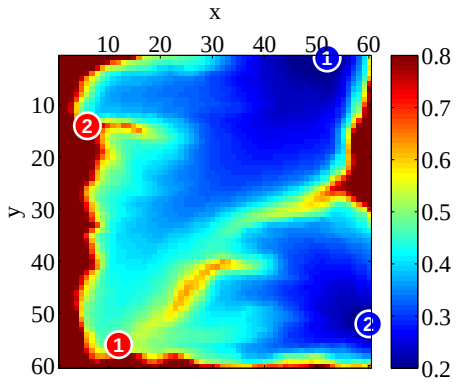
(a) Initial guess, Run 4 ($245.2 MM)



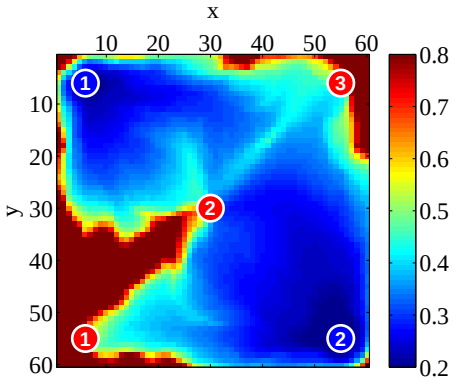
(b) Worst PSO-MADS, Run 4 ($1143 MM)



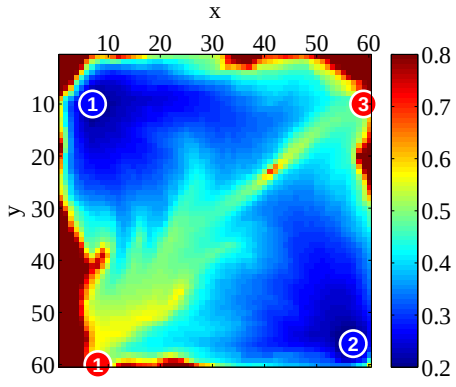
(c) Initial guess, Run 5 ($510.0 MM)



(d) Median PSO-MADS, Run 5 ($1161 MM)



(e) Initial guess, Run 1 ($1015 MM)



(f) Best PSO-MADS, Run 1 ($1247 MM)

Figure 4.16: Well configurations, NPVs and final oil saturation maps (red indicates oil, blue indicates water) from the initial guesses and optimized solutions for the worst, median and best PSO-MADS runs (Case 3).

Thus, the fluid injection and production volumes for the two cases are quite similar, though the solution in Case 3 accomplishes this oil recovery with one less well. This example illustrates the benefits that can be achieved by optimizing the number of wells along with the location and control variables. This represents a first assessment of the applicability of our optimization framework to the full MINLP field development and production problem where we determine the optimal number of wells, in addition to their locations and controls. This problem, together with more traditional MINLP solution approaches, will be further studied in Chapter 5.

4.3 Summary

In this chapter we presented applications of the MADS, PSO and PSO-MADS noninvasive derivative-free methods to the joint optimization of well locations and controls (BHPs). We presented three example cases of increasing complexity involving multiple vertical wells, several well control periods, and different types of constraints. All of the optimizations were performed in a distributed computing environment. For all cases, the PSO-MADS hybrid method was shown to outperform the standalone MADS and PSO approaches, both in terms of average NPV and standard deviation of NPV over multiple runs, demonstrating that the hybrid algorithm does indeed improve upon its component methods. For the first two examples, we included comparisons against a sequential method (where we first optimize well locations and then well controls), and the joint approach was shown to provide superior results. The last example was a nonlinearly constrained problem that additionally dealt with the determination of the optimum number of wells. In this case, the optimization provided an improvement in NPV by eliminating one of the production wells that was included in the previous examples (where the number of wells was fixed).

Our treatment for optimizing the number of wells in this chapter represents an initial approach to the general MINLP problem. In the following chapter, this problem is studied in more detail. Specifically, in Chapter 5 we generalize our optimization framework and also explore the use of an existing MINLP approach, Branch and Bound (B&B) [18], to solve the MINLP problem given by (2.1).

Chapter 5

Optimization Results for Generalized Field Development

In Chapter 4, we applied the MADS, PSO and PSO-MADS derivative-free methods introduced in Chapter 2 to solve combined well placement and control problems. In this chapter, we approach the full field development problem, where in addition to optimizing well locations and controls, we seek to determine the optimal number of wells, their type (injector or producer), and the drilling schedule.

5.1 Problem Formulation

Recall from Chapter 2 that the generalized field development problem can be posed as the following MINLP problem (repeated here for convenience):

$$(P)\begin{cases} \min\limits_{\mathbf{u}\in U,\ \mathbf{v}\in V,\ \mathbf{z}\in\mathbb{Z}^{n_c}} & f\left(\mathbf{p},\mathbf{u},\mathbf{v},\mathbf{z}\right), \\ \text{subject to} & \mathbf{g}\left(\mathbf{p},\mathbf{u},\mathbf{v},\mathbf{z}\right)=\mathbf{0}, \\ & \mathbf{c}\left(\mathbf{p},\mathbf{u},\mathbf{v},\mathbf{z}\right)\leq\mathbf{0}, \end{cases} \tag{5.1}$$

where $f = -\text{NPV}$ is the objective function. The optimization variables are comprised of categorical variables $\mathbf{z}$ that determine the number, type and drilling schedule of wells, integer variables $\mathbf{v}$ that prescribe the well locations, and continuous variables $\mathbf{u}$ that define the well controls.

5.1.1 Binary Categorical MINLP Formulation

We first consider a formulation where in addition to the well placement and control decisions, we also determine the optimal number of wells to drill, given a maximum number of possible wells. This is accomplished by including binary categorical variables for each well. These 'drill/do not drill' binary categorical variables, together with the discrete well location and continuous well control variables, form the basis for a mixed-integer nonlinear optimization problem.

We consider a waterflooding field development case, with a maximum number of wells K_{max} some of which are specified to be injectors and some producers. The goal is to determine the $K \leq K_{max}$ wells to drill, along with their locations and controls, such that we maximize NPV. This problem can be stated as:

$$\left(\hat{P}\right) \begin{cases} \min\limits_{\mathbf{u}\in U,\ \mathbf{v}\in V,\ \mathbf{z}} & -\text{NPV}\left(\mathbf{p},\mathbf{u},\mathbf{v},\mathbf{z}\right) = \sum\limits_{j=1}^{N_I}\left[z_j C_j + \sum\limits_{k=1}^{N_t} \dfrac{\Delta t_k c_{iw} q_{j,k}^{iw}\left(\mathbf{u},\mathbf{v}\right)}{\left(1+b\right)^{t_k/\tau}}\right] - \\ & \sum\limits_{j=N_I+1}^{N_I+N_P}\left[-z_j C_j + \sum\limits_{k=1}^{N_t} \dfrac{\Delta t_k \left(p_o q_{j,k}^{o}\left(\mathbf{u},\mathbf{v}\right) - c_{pw} q_{j,k}^{pw}\left(\mathbf{u},\mathbf{v}\right)\right)}{\left(1+b\right)^{t_k/\tau}}\right], \\ \text{subject to} & \mathbf{g}\left(\mathbf{p},\mathbf{u},\mathbf{v},\mathbf{z}\right) = \mathbf{0}, \\ & \mathbf{c}\left(\mathbf{p},\mathbf{u},\mathbf{v},\mathbf{z}\right) \leq \mathbf{0}, \\ & \sum\limits_{j=1}^{N_I+N_P} z_j \leq K_{max}, \\ & z_j \in \{0,1\} \quad \forall j = 1,\ldots,N_I+N_P, \end{cases} \tag{5.2}$$

where C_j is the cost of drilling well j, N_t is the number of time steps in the reservoir simulation, t_k and Δt_k represent the time and time step size at time step k, b is the fractional discount rate, τ provides the appropriate normalization for t_k (e.g., $\tau = 365$ days), c_{iw} and c_{pw} are the costs of injected and produced water, and p_o is the sale price of produced oil. The variables $q_{j,k}^{iw}, q_{j,k}^{pw}$ and $q_{j,k}^{o}$ are the rates of injected and produced water and produced oil from well j in time step k. The binary variable z_j prescribes whether or not a well is drilled (the well is drilled when $z_j = 1$ and not drilled when $z_j = 0$). We do not incur the well cost and there are no injected or produced fluids for well j if $z_j = 0$.

5.1.2 Binary Categorical Variable Relaxation Approaches

Some of the optimization methods (specifically PSO, MADS and PSO-MADS) discussed in Chapter 2 can address the optimization problem ($\hat{P}$) directly and do not require categorical variable relaxations. Many other MINLP methods, however, such as Branch and Bound (also discussed in Chapter 2), typically do require the relaxation of the categorical variables.

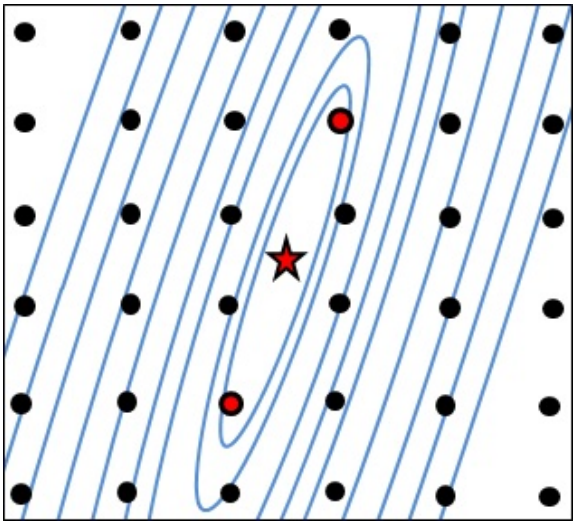


Figure 5.1: Two-dimensional illustration of a discrete optimization surface and its underlying continuous relaxation.

The main idea behind variable relaxation is illustrated in Figure 5.1. The dots in the figure show feasible solutions for the integer programming (IP) problem, with the red dots representing the integer-optimal solutions. The blue contours in the figure denote the underlying continuous nonlinear programming (NLP) problem with optimal solution depicted by the red star. Relaxing the integer problem entails solving the underlying continuous NLP. Figure 5.1 shows a situation where a single continuous global optimum is obtained from an integer problem with two local integer optima. It is important to notice (consistent with the example in Figure 5.1) that simply solving the continuous NLP and rounding the final solution to the nearest integer point does not in general yield integer optimal solutions. Thus, relaxations must be applied within a framework that judiciously combines the use of NLP solvers with a

systematic search for integer solutions. Such a framework is provided by the Branch and Bound (B&B) method.

We now describe two binary variable relaxation approaches for the MINLP problem stated in ($\hat{P}$). The first approach is based on the linear constraints that correspond to the well controls $\mathbf{u}$. A relaxed optimization problem ($\hat{P}^r$) is obtained by eliminating the restriction that the elements of $\mathbf{z}$ should be binary (i.e., $z_j \in \{0,1\} \quad \forall\ j = 1,\ldots,N_I+N_P$), and replacing the continuous well control bound constraints, $\mathbf{u}_l \leq \mathbf{u} \leq \mathbf{u}_u$, with the following linear constraints:

$$\begin{aligned} u_{lj} \leq \quad u_j \quad &\leq u_{lj} + z_j\,(u_{uj} - u_{lj}) \quad \forall\ j \in I, \\ u_{uj} + z_j\,(u_{lj} - u_{uj}) \leq \quad u_j \quad &\leq u_{uj} \qquad\qquad\qquad\quad \forall\ j \in P, \\ 0 \leq \quad z_j \quad &\leq 1 \qquad\qquad\qquad\qquad\quad \forall\ j \in I \cup P, \end{aligned} \tag{5.3}$$

where I and P refer to the sets of injection and production wells, respectively. With this relaxation, for an injection well j, as z_j approaches zero, the well controls u_j approach the lower BHP bound, effectively shutting off the well (provided the lower BHP bound is close to the well-block pressure). On the other hand, as z_j approaches one, the well controls can vary fully between the lower and upper bounds. The bounds for production wells are modified analogously. It should be noted that the drilling costs in the objective function in ($\hat{P}$) are scaled consistent with z_j being a real-valued variable.

The second binary relaxation approach considered in this work entails modification of the well index. For multiphase flow between well-block i and well j (which is perforated in block i), the well rate q_j^p for phase p is now given by:

$$q_j^p = [z_j\, WI_{ij}]\, \lambda_i^p\, (p_i - p_j)\,, \tag{5.4}$$

where WI_{ij} is the well index for well j in block i, λ_i^p is the mobility of phase p in block i, p_i is the grid-block pressure, and p_j is the BHP for well j (note that capillary pressure effects are neglected). It is evident that, as z_j varies from 0 to 1, the 'effective' well index (and hence q_j^p) varies in accordance. This second relaxation is simpler than the first from an optimization perspective since there are no additional constraints.

5.1.3 Ternary Categorical MINLP Formulation

Given a maximum number of injection and production wells, in addition to optimizing the well locations and controls, the binary MINLP formulation above optimizes the number of wells by selecting a subset of wells to be drilled. In order to also optimize the well type and drilling schedule, we extend this formulation and create a ternary categorical MINLP formulation. The categorical variable now associated with each well can take values of $\{-1, 0, 1\}$. These three values correspond to drilling an injection well (-1), not drilling (0), and drilling a production well (1). Note that we have introduced a logical ordering in this categorical variable, which will be useful in the optimization algorithm; i.e., not drilling can be interpreted as an action between injecting and producing fluid from the reservoir. As in the binary MINLP formulation above, we still need to specify the maximum number of wells to be drilled, K_{max}.

Additionally, the order in which the K_{max} categorical variables appear in the decision vector $\mathbf{z}$ determines the drilling sequence. Each of the K_{max} wells has a drilling time t_j associated with it. This time is specified by the user, and may be related to the minimum amount of time needed to drill a well. Therefore, the well corresponding to variable z_1 is drilled first, at time t_1, the well corresponding to z_2 is drilled at time t_2, and so forth. If $z_j = 0$, no action is taken at time t_j. The optimization problem can now be represented as:

$$\left(\tilde{P}\right) \begin{cases} \min\limits_{\mathbf{u}\in U,\ \mathbf{v}\in V,\ \mathbf{z}} & -\mathrm{NPV}\left(\mathbf{p},\mathbf{u},\mathbf{v},\mathbf{z}\right) = \sum\limits_{j\in I}\left[\dfrac{|z_j|C_j}{(1+b)^{t_j/\tau}} + \sum\limits_{k=1}^{N_t}\dfrac{\Delta t_k c^{iw} q^{iw}_{j,k}\left(\mathbf{u},\mathbf{v}\right)}{(1+b)^{t_k/\tau}}\right] - \\ & \sum\limits_{j\in P}\left[-\dfrac{|z_j|C_j}{(1+b)^{t_j/\tau}} + \sum\limits_{k=1}^{N_t}\dfrac{\Delta t_k\left(p^o q^o_{j,k}\left(\mathbf{u},\mathbf{v}\right) - c^{pw} q^{pw}_{j,k}\left(\mathbf{u},\mathbf{v}\right)\right)}{(1+b)^{t_k/\tau}}\right], \\ \text{subject to} & \mathbf{g}\left(\mathbf{p},\mathbf{u},\mathbf{v},\mathbf{z}\right) = \mathbf{0}, \\ & \mathbf{c}\left(\mathbf{p},\mathbf{u},\mathbf{v},\mathbf{z}\right) \le \mathbf{0}, \\ & \sum\limits_{j=1}^{K_{max}} |z_j| \le K_{max}, \\ & z_j \in \{-1,0,1\} \quad \forall j = 1,\ldots,K_{max}, \end{cases} \tag{5.5}$$

where I is the set of injection wells (wells for which $z_j = -1$) and P is the set of

production wells (wells for which $z_j = 1$). With this formulation, the optimization algorithm determines the optimal number of wells, drilling sequence, well type (injector or producer), well locations, and well controls (for a maximum of K_{max} wells). In this work we do not consider relaxation schemes for this ternary categorical MINLP formulation, and therefore we only optimize it using methods that do not require relaxation. This is in part motivated by the fact that our implementation of B&B, although robust in terms of performing a global search, is computationally expensive. This is the case even for the binary MINLP formulation, which is somewhat simpler than the ternary categorical problem.

5.1.4 Categorical Variable Treatments

Recall from Chapter 2 that the integer variables are handled by using an underlying integer mesh for the MADS polling stencil and by rounding the PSO particles to the nearest integer mesh point in PSO. The integer polling stencil works for integer variables such as well locations where there is an inherent order in the variable values and a well-defined neighborhood structure (e.g., 1 is closer to 2 than it is to 3). This neighborhood structure is needed to guarantee convergence of the algorithms to an optimal solution that is better than all other neighboring solutions. With categorical variables, the integers do not necessarily have an implicit ordering or neighborhood structure. With categorical variables that have more than two categories, it may be difficult to define an appropriate neighborhood structure. Thus, even though we can still treat the variables as simple integers during the MADS polling process, this approach would introduce a heuristic component into the optimization.

A variant of this approach, which was implemented in this work, is to replace the usual MADS integer polling stencil with a stencil that evaluates all of the categories for each categorical variable. This is similar to a one-factor-at-a-time experimental design approach [33]. Figure 5.2 illustrates this approach for a two-dimensional optimization problem with categorical variables z_1 (with six categories) and z_2 (with eight categories). In the figure, the poll center is depicted by the red dot. Using the standard approach, the MADS poll stencil at the current iteration might consider the yellow points, with the point with the black dashed circle being the next poll center.

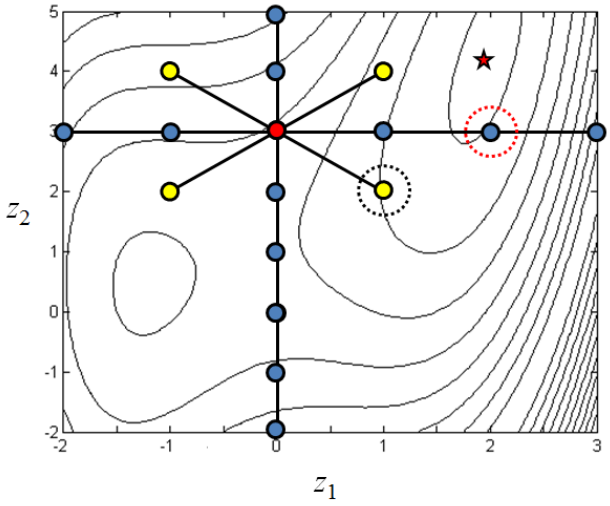


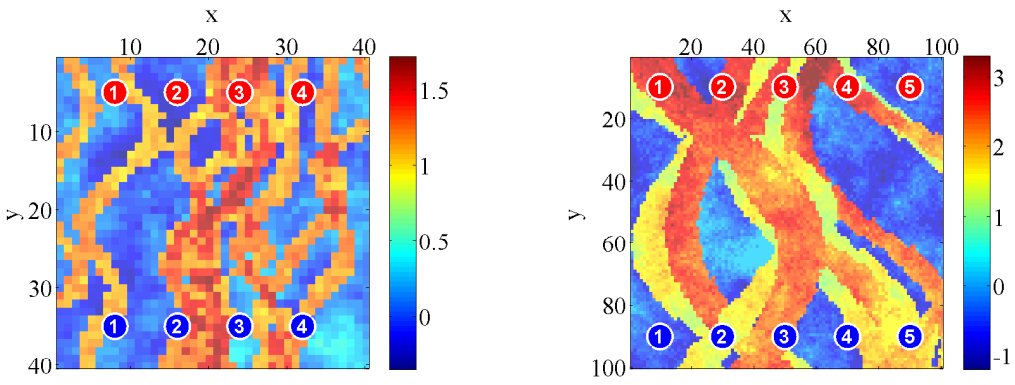
Figure 5.2: Illustration of extended stencil for treatment of categorical variables during polling in MADS and PSO-MADS.

With the proposed treatment, we replace the integer polling stencil with a stencil that evaluates all of the categories coordinate-wise. This gives the stencil depicted by the blue points. Note that this stencil does not suppose any neighborhood structure since it simply evaluates all categories. The new poll center in this case will be the blue point with the red dashed circle around it. We will show in Section 5.2.4 that for problems with categorical variables, with more than two categories, the proposed approach on average outperforms the original integer mesh polling procedure. The new approach will, however, be expensive for categorical variables with many categories.

5.2 Example Cases

The optimization methods and approaches described in Chapter 2 and Section 5.1 will now be applied to four example cases. In the first two cases, we utilize the binary categorical formulation to optimize the number of wells and their locations and controls. We assess the performance of standalone PSO, standalone MADS, the PSO-MADS hybrid, and B&B with two different relaxation procedures. These first two cases involve two-dimensional reservoir models. In the third case, we optimize

the development of a three-dimensional reservoir model with vertical injection and horizontal production wells. Due to the computational expense of running three-dimensional simulations, only the PSO-MADS hybrid is applied. In the fourth case, we additionally optimize the well type and drilling schedule using the ternary categorical formulation. PSO, MADS and PSO-MADS are considered for this case.

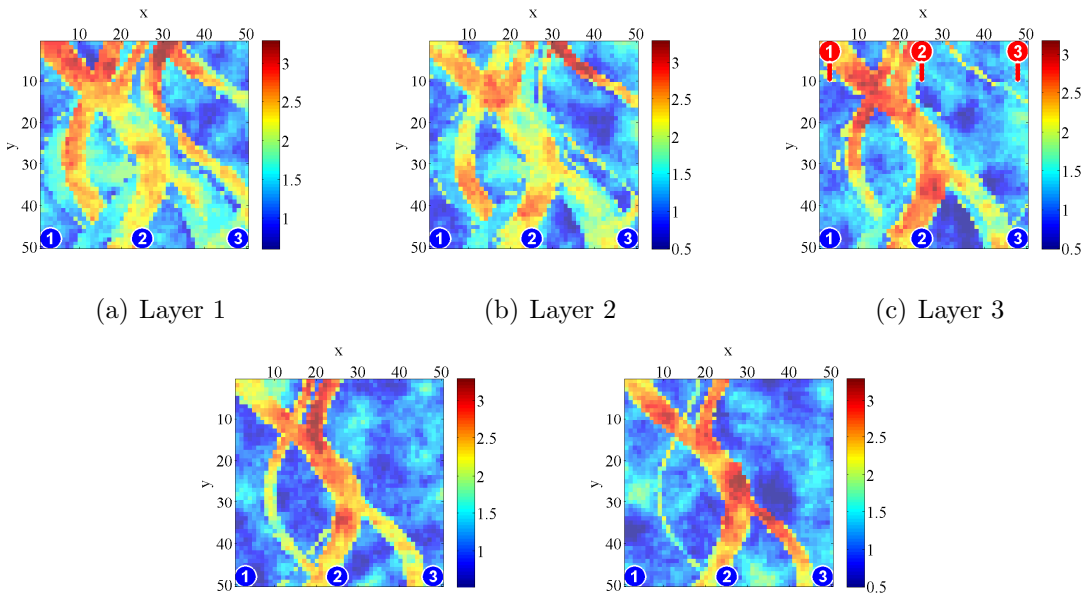


(a) Geological model used in Cases 1 and 4

(b) Geological model used in Case 2

Figure 5.3: Two-dimensional geological models used in Cases 1, 2 and 4 ($\log_{10}$ of permeability, with permeability in md, is shown). Some injection (blue) and production (red) wells are also shown.

The geological models used in the examples are depicted in Figures 5.3 and 5.4. All models correspond to fluvial depositional systems and were generated using SGeMS [102]. The model in Figure 5.3(a), used in Cases 1 and 4, was also applied in [39, 40]. It contains 40×40 grid blocks and displays about two orders of magnitude variation in permeability (a moderate degree of heterogeneity). The second model, used in Case 2, is larger (100×100 grid blocks) and represents a complex channel-levee-shale fluvial system. This model displays about four orders of magnitude variation in permeability, and is thus referred to as highly heterogeneous. The three-dimensional model ($50 \times 50 \times 5$) in Figure 5.4, used in Case 3, also represents a channel-levee-shale fluvial system. For all of the models, areal permeability is isotropic, fluid relative permeability curves are as shown in Figure 3.2, and porosity is constant at 20%. Table 5.1 presents other simulation parameters common to all of the cases considered.



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

(d) Layer 4 (e) Layer 5

Figure 5.4: Three-dimensional geological model used in Case 3 ($\log_{10}$ of permeability, with permeability in md, is shown). Some vertical injection (blue) and horizontal production (red) wells are shown.

Table 5.1: Simulation parameters common to all four example cases

Initial pressure p_i, at datum	5080 psi at 8100 ft
Rock compressibility c_R	0.5 $\times 10^{-5}$ 1/psi
Oil and water viscosity, μ_o and μ_w at p_i	1.20 and 0.325 cp
Oil and water density, ρ_o and ρ_w	54.0 and 58.0 lbm/ft^3
Formation volume factors, B_o and B_w at p_i	1.00 RB/STB

5.2.1 Case 1: Development of a Moderately Heterogeneous Channelized Reservoir

In this first case, we seek to optimize the field development and control plan for the channelized reservoir depicted in Figure 5.3(a). The production time frame is 3000 days. We consider a single control period (this implies that the well BHP value is kept constant throughout the complete production time frame). Additional simulation and optimization parameters particular to this case are provided in Table 5.2.

Table 5.2: Simulation and optimization parameters (Case 1)

Grid size	40×40
Grid cell dimensions	50 ft $\times$ 50 ft $\times$ 15 ft
p_o, c_{pw} and c_{iw}	\$80, \$36 and \$18/STB
Well drilling cost	\$4 million per well
Fractional discount rate b	0
Injection BHP range	6000 - 9000 psi
Production BHP range	2500 - 4500 psi
Max. water injection rate	270 STB/day
Min. oil production rate	200 STB/day
Max. fluid production rate	320 STB/day
Max. well water cut	0.5

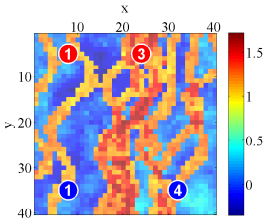
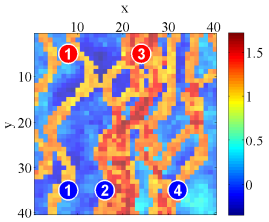
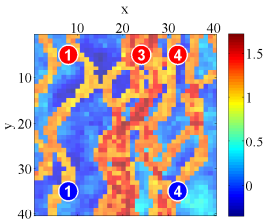
The BHP ranges define the bound optimization constraints for the continuous well controls (BHPs). The field level injection and production constraints, along with the well water cut constraints, constitute the nonlinear optimization constraints, which can only be evaluated after a reservoir simulation has been performed. Note that the simulation parameters and constraint values were chosen to make the resulting MINLP field development and control problem challenging to solve over the time frame considered (3000 days). In practice, we might (for example) operate for longer periods and use higher water cut constraints.

In this first example we initially do not consider the optimization of well placement. Rather, we are given eight possible injection and production wells in the line drive configuration depicted in Figure 5.3(a), and the goal of the optimization is to determine which of the eight wells to drill and how to control them. This optimization entails solution of a MINLP problem with eight categorical binary variables (drill/do not drill) and eight continuous control variables (well BHPs), for a total of 16 decision variables. The problem is thus posed using a reduced version of the binary categorical MINLP formulation presented earlier, with the well placement variables eliminated.

Since we have eight binary variables, there are $2^8 = 256$ possible combinations of these variables, meaning 256 possible well configurations. From a brute-force perspective, this problem can be addressed by solving the 256 well control optimization problems

that correspond to each of these configurations. We performed this exhaustive evaluation (using PSO-MADS for each of the 256 optimization runs) in order to better assess the results obtained with our MINLP algorithms. Of the 256 configurations assessed, only 16 yielded feasible solutions (i.e., optimal controls that satisfy all of the constraints in Table 5.2).

Table 5.3: Best three well configurations and their associated NPVs (Case 1, wells at fixed locations) obtained after exhaustive search

Ranking	Well Configuration	NPV [$ MM]
1		19.22
2		15.58
3		15.45

The best three (in terms of NPV) of the 16 feasible well configurations are shown in Table 5.3. The best configuration has a categorical variable solution $\mathbf{z} = [1\ 0\ 1\ 0]$ for the production wells and $\mathbf{z} = [1\ 0\ 0\ 1]$ for the injection wells. This corresponds to the well configuration shown in the first row of Table 5.3. From our exhaustive search results, we see that the optimal MINLP solution for this problem is a four-well solution with two producers and two injectors, yielding an optimal NPV of $19.22 million. Assuming that we are indeed finding the global optimum (or a solution 'close' to this

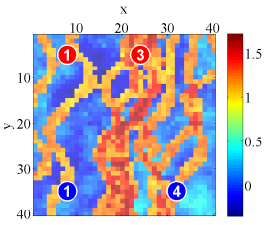
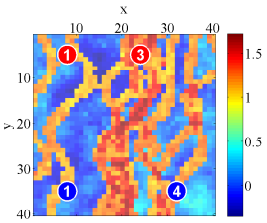
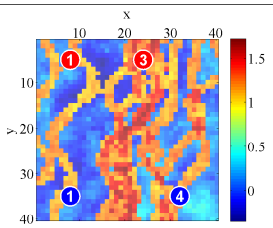
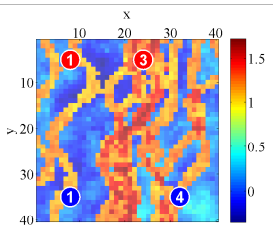
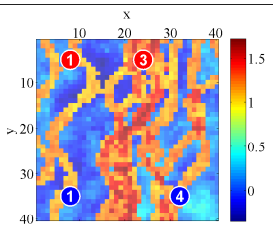
optimum) for the well control optimization problem corresponding to each configuration of the eight binary variables, we would expect that the best configuration shown in Table 5.3 represents the integer component of the global optimum of the MINLP problem under consideration.

With the results of the exhaustive search as a reference, we now assess the performance of the optimization techniques presented earlier. The standalone PSO and the PSO stage in the PSO-MADS hybrid are run with a swarm of 30 particles (thus each PSO iteration requires a maximum of 30 reservoir simulations; fewer if previously evaluated points have been saved). The standalone MADS and the MADS stage of PSO-MADS use polling stencils with 32 points (meaning each MADS iteration entails a maximum of 32 reservoir simulations), which is twice the total number of optimization variables. The PSO, MADS and PSO-MADS algorithms are run five times each, starting from different initial guesses. The initial well configuration for the first run is as shown in Figure 5.3(a) (the eight binary categorical variables are all equal to one, meaning that all of the wells are drilled), while the other four runs used randomly generated initial combinations. The B&B runs are performed with the initial guess as shown in Figure 5.3(a), with the linear constraint (LC) relaxation and the well index (WI) scaling relaxation, and using PSO-MADS for solution of the relaxed problems at each B&B node. The B&B problems are each solved only once due to the high computational cost.

The optimization results for the different methods are summarized in Table 5.4. In the last column, for PSO, MADS and PSO-MADS, the upper NPV indicates the best optimum found in the five runs, and the lower value is the average over the five runs. In terms of <NPV>, we see that PSO-MADS consistently outperforms the standalone procedures. Both B&B runs found the four-well configuration shown in Table 5.3 and the (presumed) global optimum. All five PSO-MADS runs also obtained the four-well configuration, but the standalone MADS and PSO runs each found this configuration in only two of the five runs performed.

The second column in Table 5.4 indicates the number of iterations per run (recall that PSO, MADS and PSO-MADS were run five times each, and the B&B runs were performed once). In the third column of the table, we see the average number of

Table 5.4: Optimization results from different methods (Case 1, wells at fixed locations)

Method	Iterations per run	Simulations (equiv. sims) per run	Best well configuration	NPV_{max} <NPV> [$ MM]
PSO	100	2423 (100)		19.21 <16.63>
MADS	68	1869 (68)		17.92 <16.05>
PSO-MADS	114	2316 (114)		19.22 <19.16>
B&B, LC relaxation	1059	23301 (1059)		19.22
B&B, WI relaxation	1329	31746 (1329)		19.22

simulations per run, together with the equivalent number of simulations (which accounts for parallelization) per run in parentheses. Note that, since the swarm size in PSO is comparable to the stencil size in MADS (in both methods around 30), the computational cost associated with each iteration of these algorithms (and also of PSO-MADS) is comparable and equivalent to around 30 simulations. Since B&B utilizes PSO-MADS as its basic optimization method (NLP solver), every B&B iteration also requires a similar computational cost. The elapsed time is essentially proportional to the number of equivalent simulations. Since we fully distribute computations on 32 cores for this case, the number of equivalent simulations is equal to the number of iterations. The actual elapsed time would presumably be slightly greater due to communication time and any inefficiencies related to the parallel implementation.

The global search capability of PSO may explain why the PSO-MADS method outperforms MADS on average. Even though the best MADS run does find the optimal well configuration, it apparently gets trapped in a local optimum, and does not converge to the control values that yield an NPV of \$19.22 million. It is worth reiterating that both B&B runs identify the best well configuration, as might be expected given the thorough search they perform. However, the computational cost of B&B is significantly higher than that of PSO-MADS, even for this relatively simple problem.

We next consider a variation of the previous example. In this case we also include the well location variables in the optimization. The problem now contains a total of 32 optimization variables. The size of the swarm for PSO, and the PSO stage of PSO-MADS, is increased to 50 particles. The maximum number of poll points in MADS is 64 ($2n$). We again perform five runs from different initial guesses for the PSO, MADS and PSO-MADS, and one run each for B&B with LC and WI relaxations.

The progression of the optimizations (for PSO, MADS and PSO-MADS) is shown in Figure 5.5. Prior to the appearance of the curves in the figure, solutions are infeasible. The optimization results for this case are summarized in Table 5.5. The fourth column in this table shows, for each of the different methods, the well configuration from the run with the highest final NPV. The well locations in all cases clearly differ from those in the previous case. It is apparent that, by allowing the wells to move, the highest NPV (obtained using B&B with WI relaxation) increases by 68%, from about \$19

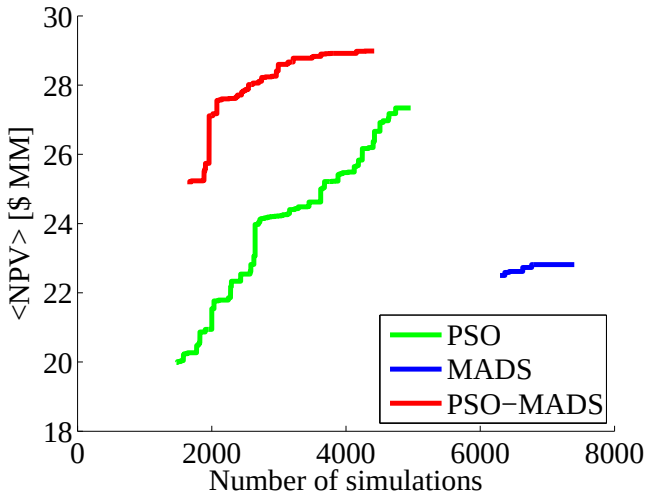
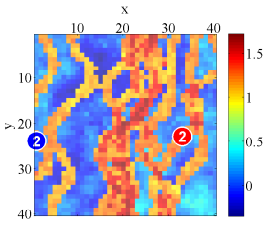
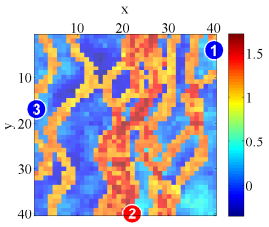
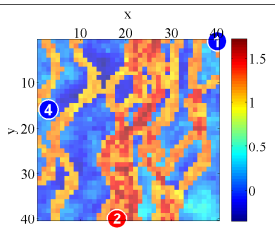
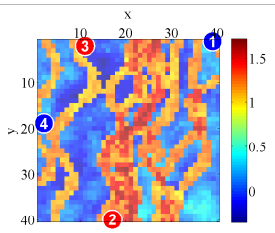
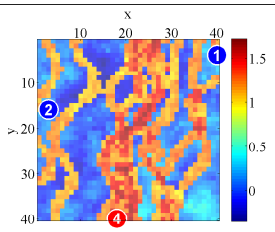


Figure 5.5: Evolution of mean NPV for the five PSO, MADS and PSO-MADS runs (Case 1).

million (Table 5.4) to $32 million. Additionally, the best configuration is now a three-well solution (two injectors and one producer) instead of a four-well solution. We again see that the PSO-MADS algorithm outperforms its component methods, and that it is competitive in terms of solution quality with the B&B approaches (within 0.6% of the best B&B solution), while requiring considerably less computational effort.

It is of interest to note that the two B&B runs (which use different binary variable relaxations) perform very differently in this case. While B&B with WI relaxation finds a configuration similar to that from PSO-MADS, the B&B solution obtained with LC relaxation contains an additional production well. Furthermore, B&B with LC relaxation terminates the search after using only about 1/8 the number of iterations used by B&B with WI relaxation. This difference may be due to the fact that the WI relaxation introduces a nonlinear relationship between the binary variables z_j and the cost function and constraints, while in the LC relaxation, the dependence is only in the constraints and it is linear. The additional nonlinearity associated with the WI relaxation may complicate the search.

Table 5.5: Optimization results from different methods (Case 1, well locations included in the optimization)

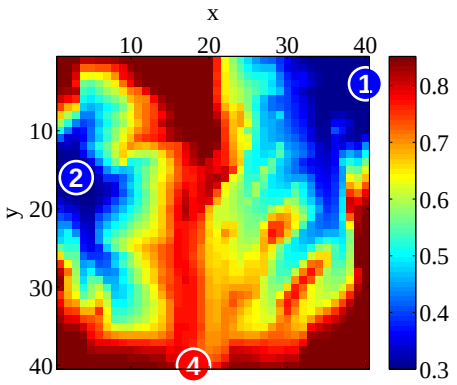
Method	Iterations per run	Simulations (equiv. sims) per run	Best well configuration	NPV_{max} <NPV> [$ MM]
PSO	100	4963 (100)		30.89 <27.34>
MADS	122	7401 (122)		25.43 <22.90>
PSO-MADS	104	4419 (104)		31.85 <28.99>
B&B, LC relaxation	1001	42357 (1001)		30.21
B&B, WI relaxation	8450	324526 (8450)		32.03

The new linear constraints that arise in the LC relaxations are handled by means of the filter method. Because the filter approach was devised for general constraints, this treatment may cause the optimization algorithm to perform suboptimally. The inclusion of linear constraints in MADS using the approach described in [71] will likely enhance the performance of the B&B method. Although the careful treatment of these linear constraints may improve B&B performance, we still expect it to require significantly more function evaluations than the other methods.

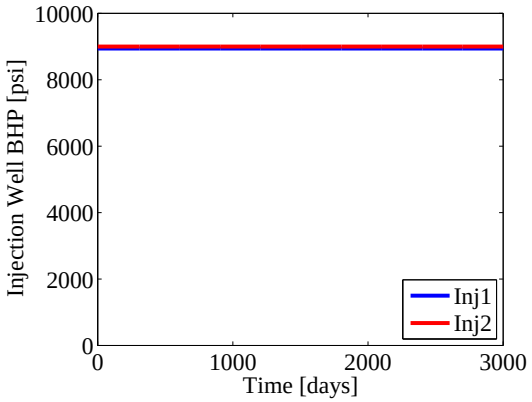
Figure 5.6 displays simulation results from the best solution found by any of the methods (B&B with WI relaxation). Figure 5.6(a) demonstrates the high degree of reservoir sweep achieved with the optimal well configuration and BHP controls. Figures 5.6(b) and 5.6(c) show the optimal well BHP profiles. The optimization procedure places the wells far enough apart that the optimal BHP controls, which do not vary in time since this example involves only a single control period, are at the bounds. These well BHPs result in the optimal injection and production rates shown in Figures 5.6(d) and 5.6(e). It is evident that the field-level constraints for water injection and oil production (depicted by the dashed red lines in Figures 5.6(d) and 5.6(e)) are indeed satisfied at all times.

5.2.2 Case 2: Development of a Highly Heterogenous Channelized Reservoir

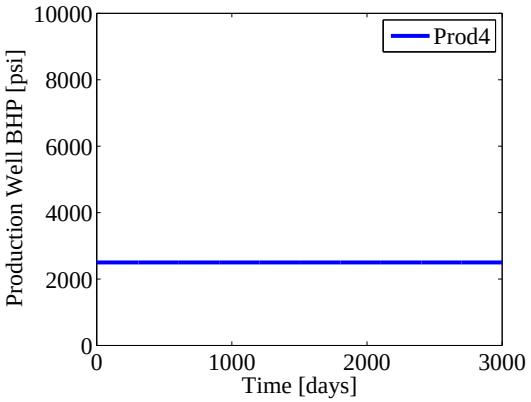
In the second example, we consider the reservoir model shown in Figure 5.3(b). As in Case 1, the production time frame is again 3000 days, but we consider three control periods of 1000 days each (implying that we set the well BHPs at the start of the run, at 1000 days and at 2000 days). The field development in this case can include a maximum of five injectors and five producers. We seek to optimize the number of wells (subject to these maxima) along with their corresponding locations and controls. The ten possible wells imply ten binary variables, 20 areal (x,y) well location variables and, since we include three control periods per well, 30 well control variables. Therefore, the total number of optimization variables is equal to 60. In addition to the simulation parameters presented in Table 5.1, we also have the simulation and optimization parameters, including bound and nonlinear constraints, shown in Table 5.6.



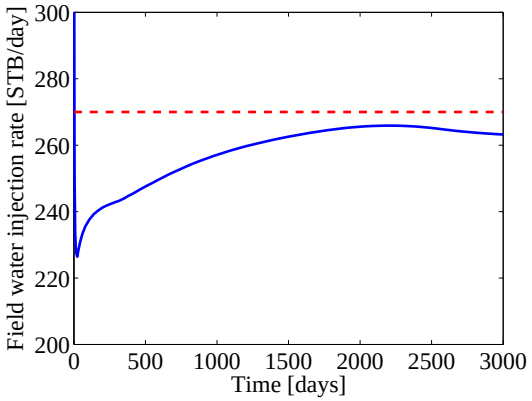
(a) Final oil saturation map



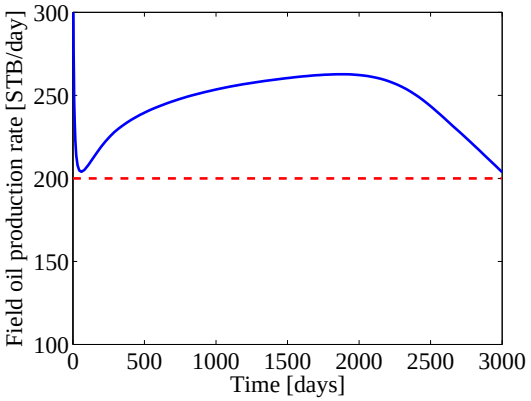
(b) Optimized injector BHPs



(c) Optimized producer BHP



(d) Field water injection rate



(e) Field oil production rate

Figure 5.6: Simulation results from the solution obtained with B&B with WI relaxation (Case 1, with well locations optimized), showing the final oil saturation map, well BHPs, and field rates with constraint limits (red dashed lines).

Table 5.6: Simulation and optimization parameters (Case 2)

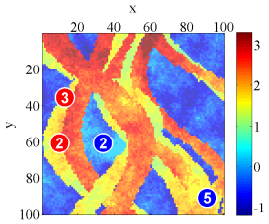
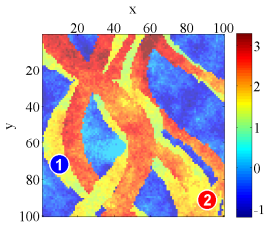
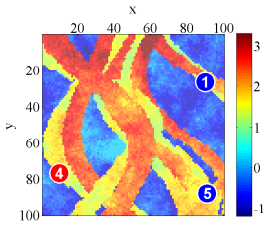
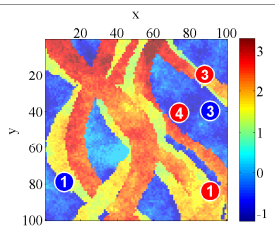
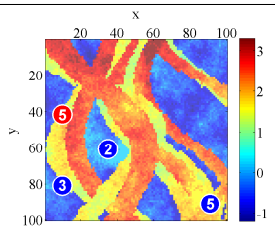
Grid size	100 × 100
Grid cell dimensions	50 ft × 50 ft × 15 ft
p_o, c_{pw} and c_{iw}	\$80, \$10 and \$5/STB
Well drilling cost	\$5 million per well
Fractional discount rate b	0
Injection BHP range	6000 - 9000 psi
Production BHP range	1500 - 4500 psi
Max. water injection rate	1200 STB/day
Min. oil production rate	500 STB/day
Max. fluid production rate	1200 STB/day
Max. well water cut	0.8

The PSO and PSO-MADS algorithms use the same PSO parameters as in Case 1 except for the swarm size, which is now set to 80 particles. The MADS polling stencil size in this example is equal to 120, since there are 60 optimization variables. All of the runs were parallelized using 40 computer cores. Therefore, one iteration of MADS is equivalent in elapsed time to about three simulation runs, and one iteration of PSO to about two simulation runs. The PSO, MADS and PSO-MADS methods are run five times each, starting from different initial guesses. The configuration shown in Figure 5.3(b) represents one initial guess; the others are randomly generated.

The optimization results for this case are summarized in Table 5.7, while the progress of the PSO, MADS and PSO-MADS algorithms for this case are shown in Figure 5.7. The fourth column of the table displays the well configuration for the run with the highest objective function value. We see again that the PSO-MADS hybrid outperforms its component methods and yields solutions, in terms of objective function, comparable to those obtained with B&B. Consistent with our previous finding, PSO-MADS requires considerably less computational effort than the more rigorous B&B approaches.

As was observed in the results for the second test in Case 1, B&B with LC relaxation terminates the search after many fewer iterations than B&B with WI relaxation. We also see that the best PSO-MADS configuration and the optimal B&B configurations are very different from one another but still have about the same NPV. This is

Table 5.7: Optimization results from different methods (Case 2)

Method	Iterations per run	Simulations (equiv. sims) per run	Best well configuration	NPV_{max} <NPV> [$ MM]
PSO	200	15580 (400)		156.6 <149.4>
MADS	231	26089 (693)		164.4 <150.8>
PSO-MADS	157	12046 (381)		174.3 <161.4>
B&B, LC relaxation	4496	372254 (11958)		172.2
B&B, WI relaxation	25238	2353822 (72649)		171.7

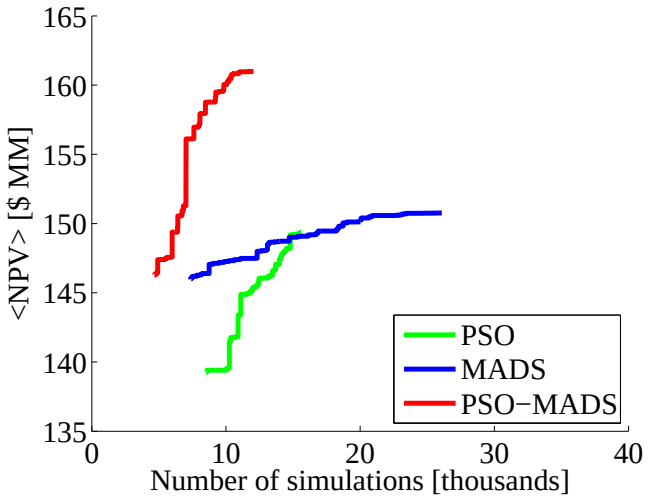
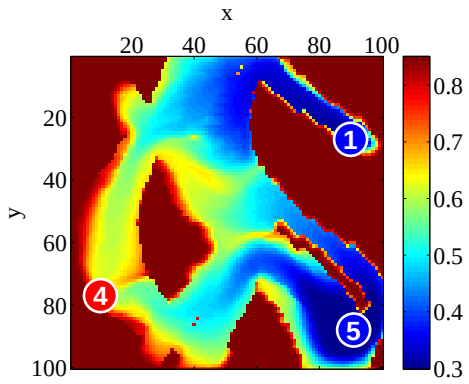


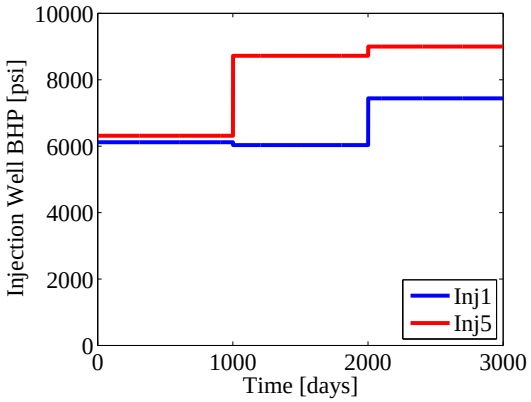
Figure 5.7: Evolution of mean NPV for the five PSO, MADS and PSO-MADS runs (Case 2).

consistent with our expectation that the objective function surface is very complex, with multiple local optima.

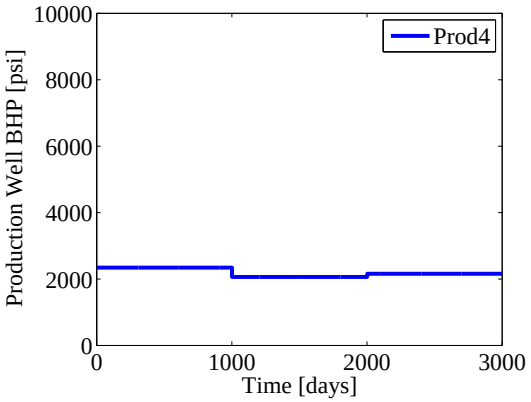
Figure 5.8 displays simulation results for the best solution found using PSO-MADS. It is important to note that this case is more challenging than Case 1 because permeability variation is much greater here, which makes it more difficult to achieve complete reservoir sweep. Nonetheless, the optimization procedure places three wells (two injectors and one producer) in the locations shown in Figure 5.8(a), with optimized BHP controls as shown in Figures 5.8(b) and 5.8(c), such that the oil in the channel and levee sands is produced. The resulting field water injection and oil production rates, together with their constraint limits, are shown in Figures 5.8(d) and 5.8(e), where it can be observed that the constraints are satisfied at all times. From the rate plots, the effects of the changes in well BHPs at 1000 and 2000 days are evident.



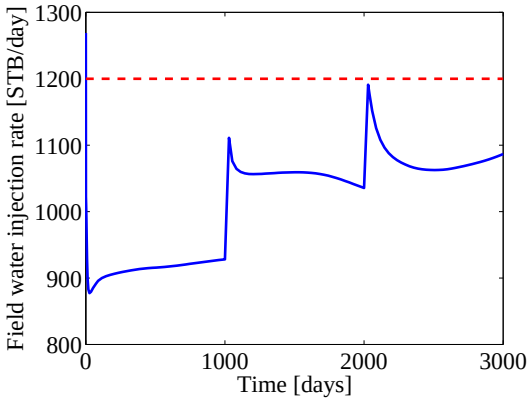
(a) Final oil saturation map



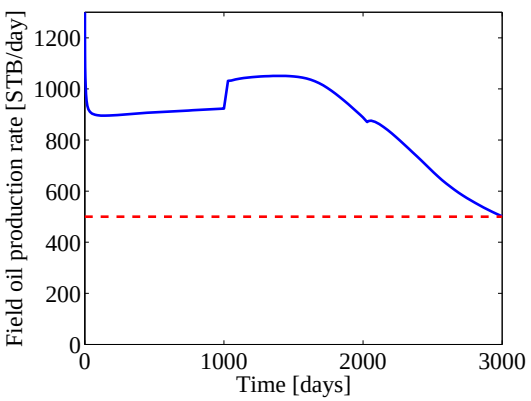
(b) Optimized injector BHPs



(c) Optimized producer BHP



(d) Field water injection rate



(e) Field oil production rate

Figure 5.8: Simulation results from the best solution obtained with PSO-MADS (Case 2), showing the final oil saturation map, well BHPs, and field rates with constraint limits (red dashed lines).

5.2.3 Case 3: Development of a 3D Channelized Reservoir

This example involves the development of the three-dimensional reservoir model depicted in Figure 5.4. The production time frame for this case is 1500 days. There are three control periods of 500 days each. The options for field development include drilling a maximum of three vertical injection wells and three horizontal production wells. The horizontal wells are constrained to be drilled in the y-direction. The binary categorical variable formulation is utilized to decide which wells to drill. In addition to the areal location of each vertical injection well, the completion interval for the well is also optimized. Thus for each vertical well, we have four location variables: x and y areal variables and the completion interval defined by the first and last layers in which the well is open to flow (l_1 and l_2). For each of the horizontal production wells, we also have four location variables (x areal variable, the layer l in which the well is drilled, and y_1 and y_2 defining the start and end of the well). The optimization variables associated with the vertical injection and horizontal production wells are illustrated in Figure 5.9. With the three BHP control periods for each well, there are a total of 48 optimization variables in this example (one binary categorical, four location and three control variables for each of the six possible wells).

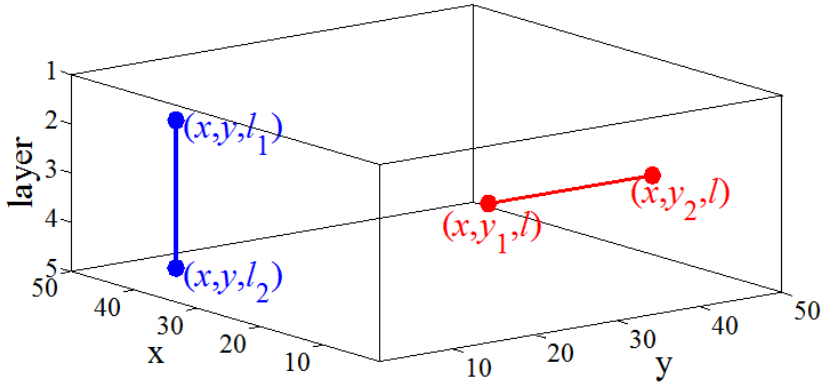


Figure 5.9: Well location variables associated with vertical injection (blue) and horizontal production (red) wells (Case 3).

Table 5.8: Simulation and optimization parameters (Case 3)

Grid size	$50 \times 50 \times 5$
Grid cell dimensions	100 ft $\times$ 100 ft $\times$ 15 ft
k_v/k_h ratio	0.1
p_o, c_{pw} and c_{iw}	\$80, \$10 and \$5/STB
Well drilling cost	\$80 million per well
Vertical completion costs	\$10000/ft
Horizontal completion costs	\$12000/ft
Fractional discount rate b	0
Injection BHP range	6000 - 9000 psi
Production BHP range	1500 - 4500 psi
Max. water injection rate	30000 STB/day
Min. oil production rate	10000 STB/day
Max. fluid production rate	30000 STB/day
Max. well water cut	0.75

In addition to the simulation parameters presented in Table 5.1, this case also includes the simulation and optimization parameters shown in Table 5.8. Note that there are now additional per-foot drilling, completion and perforation costs for vertical and horizontal wells, in addition to the drilling cost associated with reaching the top of the reservoir. For this example case, we perform the field development optimization using only the PSO-MADS hybrid. The swarm size for the PSO stage is 50, and the number of MADS poll points per iteration is 96. As in the Case 1 examples, these function evaluations are fully parallelized. The PSO-MADS optimizer is run five times, starting from different initial guesses. In the first initial guess, the well configuration is as shown in Figure 5.4, where we have three fully penetrating injection wells and three short horizontal producers drilled in the middle layer. The well controls for this initial guess are set to 3000 psi for the producers and 7500 psi for the injectors. The other four initial guesses were randomly generated within the bounds of the problem.

Table 5.9 presents the NPVs for the initial guess and optimized solutions for the five PSO-MADS runs. The evolution of the average objective function over the five runs is shown in Figure 5.10. It should be noted that all initial guesses are infeasible, while all of the optimized solutions are feasible. Table 5.9 also shows the number of vertical

Table 5.9: Initial and final NPVs from five PSO-MADS runs (Case 3, best values are italicized). Number of injection and production wells for each run shown in parentheses

Run #	Initial NPV [$ MM] (# injectors, # producers)	Total iterations	Optimized NPV [$ MM] (# injectors, # producers)
1	*586* (3,3)	213	*1224* (2,1)
2	-408 (2,2)	215	1157 (1,1)
3	-149 (0,2)	250	1111 (2,2)
4	-58 (1,1)	235	1177 (2,1)
5	349 (2,2)	201	1217 (3,1)
<NPV>	64		1177
σ	399		46

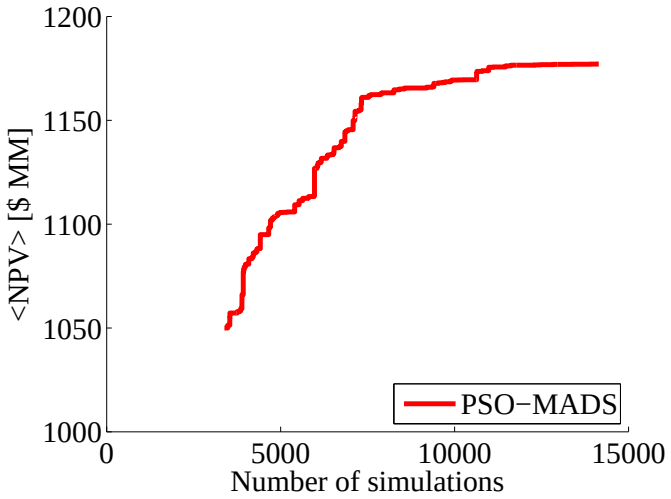
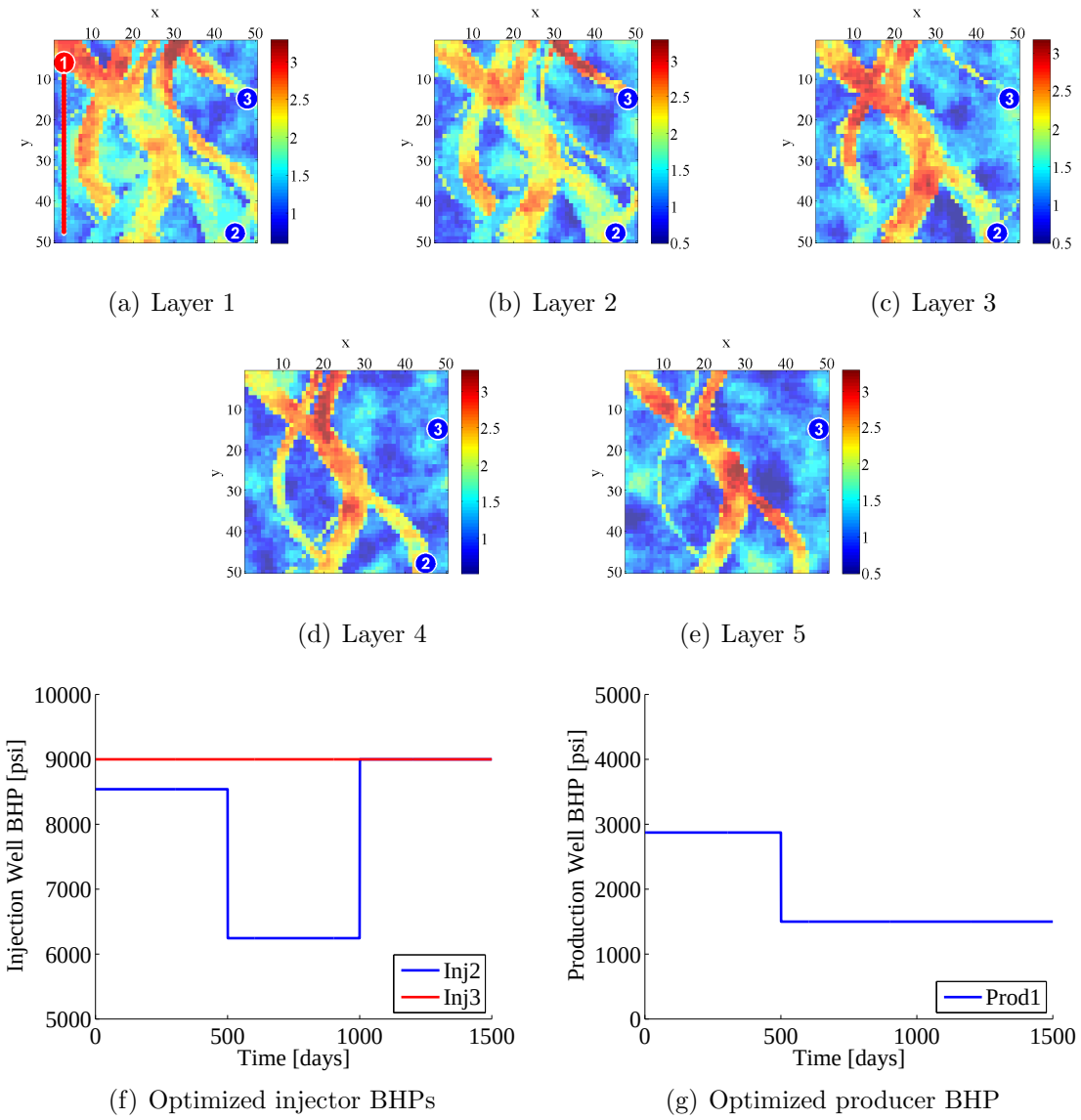


Figure 5.10: Evolution of mean NPV for the five PSO-MADS runs (Case 3).

injection and horizontal production wells, together with the number of total iterations required in each optimization run. The results in this table illustrate the effectiveness and robustness of the PSO-MADS algorithm in obtaining reasonable solutions, even starting from poor initial guesses. The three best solutions, in terms of NPV, contain one production well and two to three injection wells, suggesting that a three to four well configuration is optimal.



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

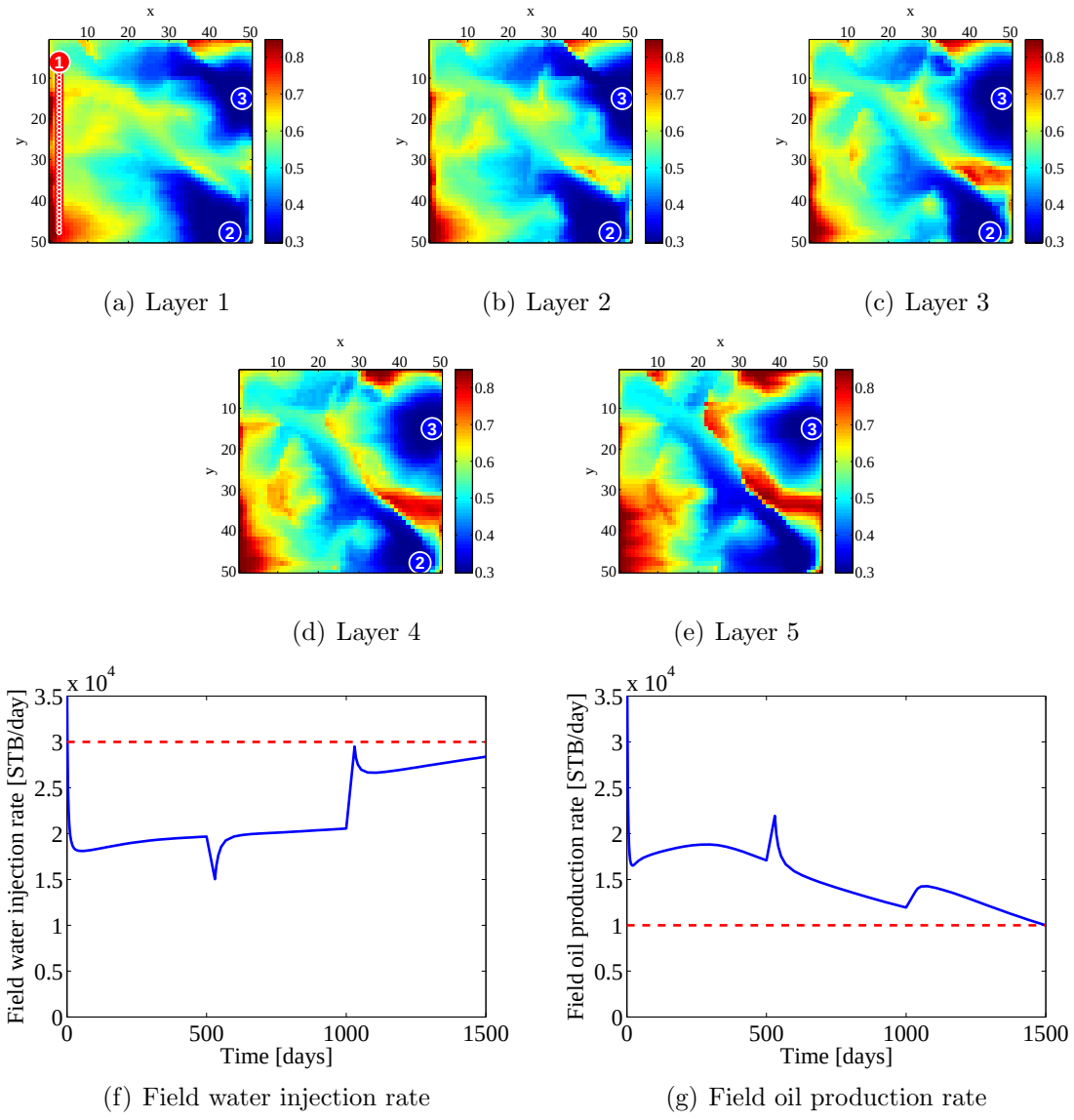
(d) Layer 4 (e) Layer 5

(f) Optimized injector BHPs (g) Optimized producer BHP

Figure 5.11: Best PSO-MADS solution, showing well locations and BHP profiles (Case 3).

Figure 5.11 presents the best PSO-MADS solution (Run 1). The locations of the two injection wells and one production well are shown Figures 5.11(a) to 5.11(e), and their BHP controls are displayed in Figures 5.11(f) and 5.11(g). Injector 2 is completed in the first four layers of the reservoir (note that it does not appear in Figure 5.11(e)), while Injector 3 penetrates all layers. Producer 1 is drilled in layer 1 along the left

side of the reservoir. The optimized well configuration thus entails waterflooding from right to left through the reservoir.



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

(d) Layer 4 (e) Layer 5

(f) Field water injection rate (g) Field oil production rate

Figure 5.12: Simulation results from best PSO-MADS solution (Case 3), showing final oil saturation maps and field rates with constraint limits shown as red dashed lines.

Figure 5.12 displays simulation output for the best PSO-MADS solution. Figures 5.12(a) to 5.12(e) show the final oil saturation maps for the five layers of the reservoir. These

maps illustrate the effective right-to-left sweep. Figures 5.12(f) and 5.12(g) present the resulting field rates, together with their constraint limits. Only two constraints are shown, but all specified constraints are satisfied by all of the optimized solutions.

5.2.4 Case 4: Full Field Development of a Moderately Heterogenous Channelized Reservoir

In this example, in addition to determining the optimal number of wells together with their locations and controls, we will also optimize the well type and drilling schedule using the ternary MINLP formulation. The ternary variable for each well specifies if the well is drilled as an injector, a producer, or not drilled at all. Each of these ternary variables has an associated predefined drilling time, t_j. As in Cases 1 and 2, the production time frame for this reservoir is 3000 days. However, here we consider nine control periods (the first eight are 210 days each and the last is 1320 days). Although the geological model is the same as in Case 1 (see Figure 5.3(a)), some of the optimization parameters have been modified, as noted in Table 5.10 (e.g., a 10% discount rate is now used in the NPV calculation).

Table 5.10: Simulation and optimization parameters (Case 4)

Grid size	40×40
Grid cell dimensions	50 ft $\times$ 50 ft $\times$ 15 ft
p_o, c_{pw} and c_{iw}	\$80, \$36 and \$18/STB
Well drilling cost	\$4 million per well
Fractional discount rate b	0.1
Injection BHP range	6000 - 9000 psi
Production BHP range	2500 - 4500 psi
Max. water injection rate	600 STB/day
Min. oil production rate	50 STB/day
Max. fluid production rate	600 STB/day
Max. well water cut	0.5

The maximum number of wells that can be drilled is equal to eight, which means we have eight ternary categorical variables. There are nine control periods per well, and the start times for the first eight periods coincide with the eight possible drilling times t_j. Thus a well can be drilled, if needed, every 210 days. Since there are two

areal location variables per well, this problem would appear to have a total of 96 optimization variables. However, before well j is drilled, its controls for times before t_j are immaterial and are fixed to shut-in values. Therefore, the actual number of optimization variables that must be determined is 68.

Because the B&B relaxations used in the first two example cases are not directly applicable to the ternary variables present in this example, and because of the high computational cost associated with the B&B algorithms, we consider the use of MADS, PSO and PSO-MADS in this case. We apply PSO-MADS with both the standard MADS stencil and the extended categorical stencil discussed in Section 5.1.4. We note, however, that if finding the best solution is of paramount importance regardless of the computational cost, then B&B should also be considered. The swarm size in PSO and PSO-MADS, and the stencil polling size in MADS and the hybrid method, are now equal to 50 and 136, respectively. We distribute function evaluations over 70 computer cores. Thus, as in Case 2, one iteration of PSO (for which elapsed time is equivalent to one simulation) is less expensive than one iteration of MADS (elapsed time equivalent to two simulations). The four methods are run five times each, starting from different initial guesses.

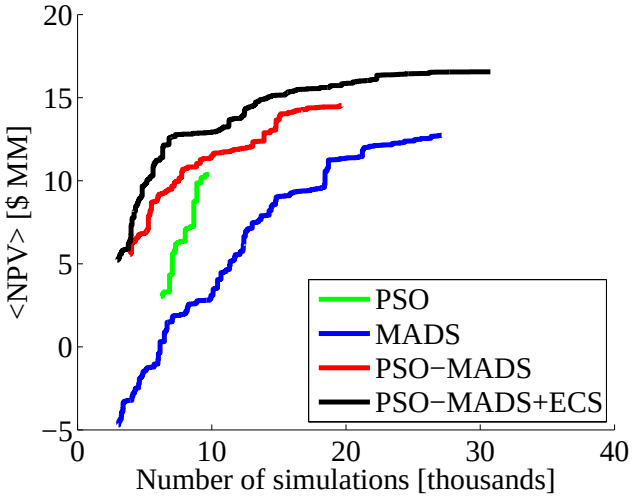
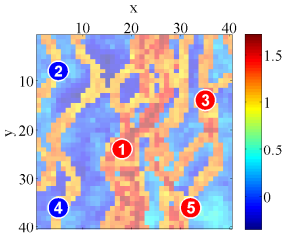
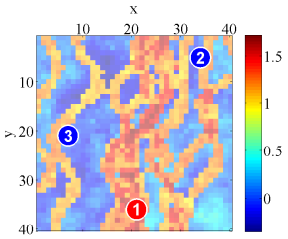
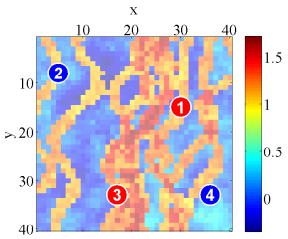
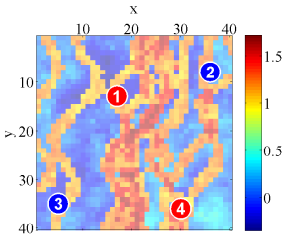


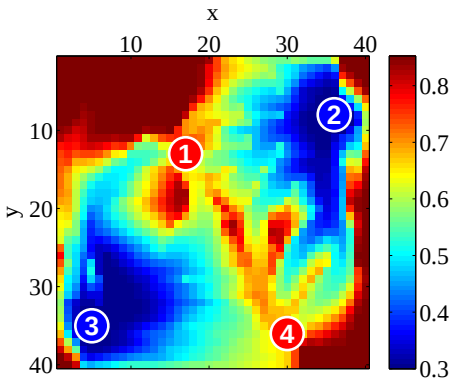
Figure 5.13: Evolution of mean NPV for the five PSO, MADS, PSO-MADS and PSO-MADS+ECS (extended categorical stencil) runs (Case 4).

Table 5.11: Optimization results from different methods (Case 4)

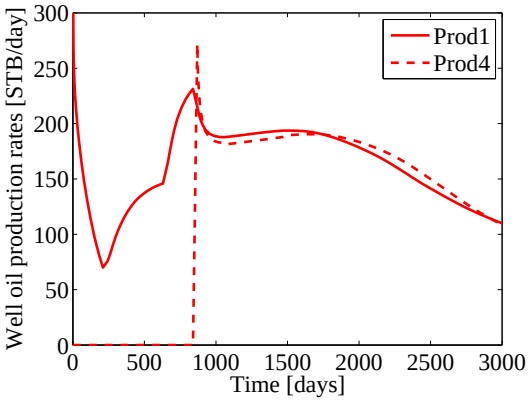
Method	Iterations per run	Simulations (equiv. sims) per run	Best well configuration and sequence	NPV_{max} <NPV> [$ MM]
PSO	200	9789 (200)		14.61 <10.63>
MADS	210	27124 (420)		14.13 <12.82>
PSO-MADS	178	19658 (296)		17.13 <14.70>
PSO-MADS with extended categorical stencil	297	30752 (534)		17.22 <16.56>

The progress of the four optimization algorithms is shown in Figure 5.13. The optimization results are summarized in Table 5.11. The fourth column in the table indicates the well configuration from the run with the highest NPV, with the wells numbered according to their place in the drilling sequence (note that in the previous cases, the well numbers just indicated well names). For example, for the best

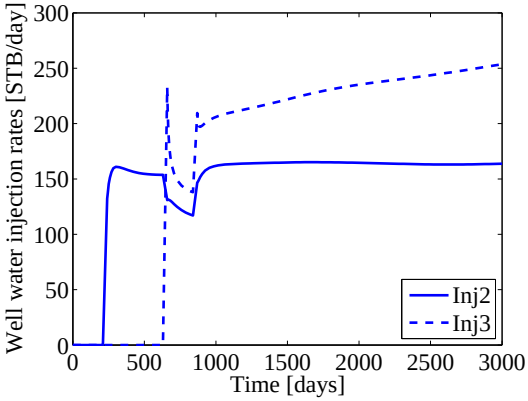
PSO-MADS solution, the producer labeled 1 is drilled first, followed by the injector labeled 2, etc. The results in Table 5.11 again show that the PSO-MADS hybrid method outperforms its component methods both in terms of average performance and the best solution obtained from the five runs. It is interesting to note that all of the solutions shown in Table 5.11 have a production well drilled first, to begin oil extraction as soon as possible, and an injection well drilled second to provide pressure support, which seems intuitively reasonable. The best method, both in terms of average performance and maximum NPV obtained, is PSO-MADS with the extended categorical stencil. This result validates the extended stencil idea for variables with multiple categories where a neighborhood structure is not evident.



(a) Final oil saturation map



(b) Well oil production rates



(c) Well water injection rates

Figure 5.14: Best solution obtained using PSO-MADS with extended categorical stencil (Case 4).

Figure 5.14 presents simulation results for the optimization solution from the best PSO-MADS with extended categorical stencil run. In Figure 5.14(b), we see oil production from Producer 1, drilled at the beginning of the production time frame, quickly decline due to decreasing reservoir pressure. Pressure support is provided at 210 days when the second well (Injector 2) is drilled (see Figure 5.14(c)). Production from Producer 1 then increases, and additional support is provided by Injector 3, which is drilled at 630 days. Producer 4 is drilled at 840 days to drain the oil in the lower right portion of the reservoir model. Oil production decreases, though relatively slowly, at later times. The reservoir sweep at the end of the run is displayed in Figure 5.14(a).

This example demonstrates that, using the ternary MINLP formulation for petroleum field development and control introduced in this work, along with our PSO-MADS hybrid implementation with extension for categorical variables, we are able to simultaneously optimize the well configuration (number of wells, well type and locations), the drilling sequence, and the well controls. Thus the methodology presented here appears to be applicable for complicated and realistic field management decisions.

5.3 Summary

In this chapter we applied the general procedures described in Chapter 2 to simultaneously determine the optimal number and type of wells, their drilling sequence, and the corresponding well locations and controls (e.g., time-varying BHPs). The field development and control problem was posed as a mixed-integer nonlinear programming (MINLP) problem with (binary/ternary) categorical variables, integer-valued ('pseudo-continuous') location variables, and real-valued control variables. We also introduced an extended MADS stencil for the categorical variables.

Four example cases were presented. Two different categorical variable relaxation treatments were considered in the B&B procedures that were applied in the first two cases. In these cases, the PSO-MADS hybrid algorithm was found to outperform its component methods (standalone PSO and MADS) in all aspects. In addition, although it contains some heuristic components, PSO-MADS provided solutions that

were comparable, in terms of NPV, to the solutions obtained with B&B, a method that performs a systematic categorical variable search. The computational cost associated with PSO-MADS was in both cases significantly less than that for B&B. In the third case, the PSO-MADS hybrid was used to optimize field development for a three-dimensional model that contained vertical injection and horizontal production wells. The optimizer obtained a solution with two injection wells and one production well, while satisfying all specified constraints. In the last case, we optimized the number of wells to be drilled, their type and drilling schedule, in addition to the well locations and controls. For this case, we applied the standalone PSO and MADS algorithms, the PSO-MADS hybrid used in the previous examples, and a new PSO-MADS algorithm with an extended categorical stencil. This new PSO-MADS hybrid was shown to provide the best performance for this general and challenging case.

Due to its solid theoretical foundations and exhaustive search capabilities, B&B may still be an attractive choice for these problems. The performance of the two B&B relaxation procedures considered in the first two cases differed significantly, especially in terms of computational cost. This emphasizes the importance of the choice of categorical variable relaxation in B&B formulations, and suggests that other treatments should be developed and assessed. In addition, the applicability of other MINLP solution approaches, such as outer approximation [18], which requires gradient information, should be considered for the generalized field development problem.

Chapter 6

Biobjective Field Development

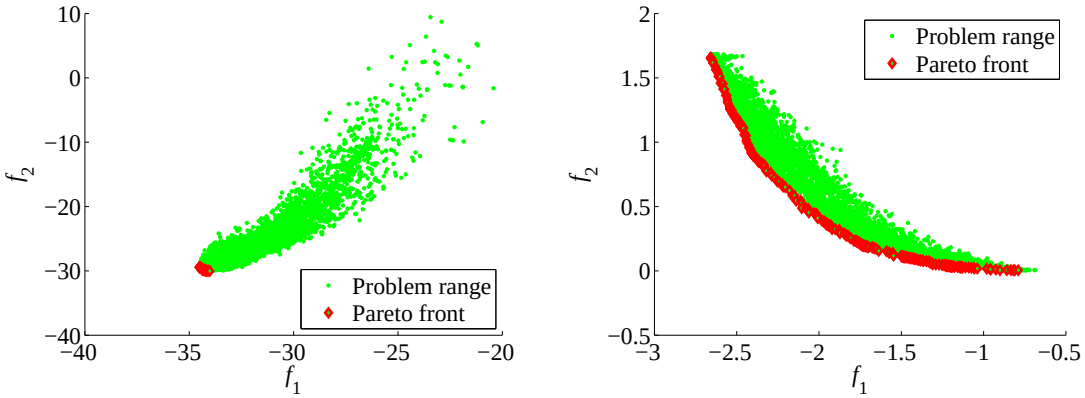
In field development and production planning, as in many other real-world engineering problems, decisions can involve multiple, possibly conflicting, criteria. Two different objectives are considered to be conflicting if designs or control schemes that improve one objective worsen the other. The goal of optimization with multiple conflicting objectives is to generate the so-called Pareto front, which is a representation of the optimal trade-off between the different objectives. This leads to multiobjective optimization, where instead of optimizing a singe objective f, as in the general MINLP problem (P) in (2.1), we optimize the vector $\mathbf{f}$ as follows:

$$\min_{\mathbf{x}\in\Omega} \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_L(\mathbf{x})], \tag{6.1}$$

where L is the number of objectives. Note that $\mathbf{x}$, used here to simplify the presentation, consists of discrete and continuous variables and replaces $(\mathbf{u}, \mathbf{v}, \mathbf{z})$ in (2.1). In this work we focus on formulations for optimizing just two objectives, i.e., we treat the biobjective optimization problem, though much of our discussion is also relevant for multiobjective optimization.

Similar to the filter method, described in Section 2.2.2, in multiobjective optimization, optimal solutions are selected based on Pareto dominance. The dominance relation [126] states that when comparing two points $\mathbf{x}_a$ and $\mathbf{x}_b$, $\mathbf{x}_a$ is said to dominate $\mathbf{x}_b$ (written as $\mathbf{x}_a \prec \mathbf{x}_b$) if and only if $f_i(\mathbf{x}_a) \leq f_i(\mathbf{x}_b)\ \forall\ i \in \{1, 2, \ldots, L\}$, with at least one of these being a strict inequality. This dominance relation provides the

definition of the Pareto optimal solutions for the multiobjective problem. The set of nondominated points $X_{\mathcal{P}}$ that dominate all of the other points in the feasibility region of the problem defines the Pareto front $Y_{\mathcal{P}}$ (in our notation X indicates points in the parameter space and Y indicates the mapping of those points to the objective-function space).



(a) Nonconflicting objectives

(b) Conflicting objectives

Figure 6.1: Illustrations of biobjective Pareto fronts (in red) from optimizations with conflicting and nonconflicting objectives. Green points indicate solutions computed during the optimization.

Figure 6.1 presents biobjective minimization results for problems with nonconflicting and conflicting objectives (these results are obtained using the procedure described later in this chapter). In these plots, the green points approximate the range of each problem (objective function values of points in the feasible region of the parameter space or domain) and the red points depict the nondominated points in objective space, which approximate the Pareto fronts for these problems. In Figure 6.1(a) we see that the objectives are not conflicting because solutions that improve f_1 generally also improve f_2. Thus we could have optimized one of the objectives and the results would be near-optimal for the other objective. As a result, we have a very small Pareto front in this case. In Figure 6.1(b), by contrast, there is conflict between the two objectives since, in general, solutions that improve f_1 worsen f_2. This results in an extensive Pareto front over the range of the problem. We next describe solution approaches for multiobjective optimization problems that can be used to obtain Pareto fronts.

6.1 Solution Approaches

As noted in Section 1.1.5, where we provided a brief review of multiobjective optimization approaches, there are two broad classes of approaches. The first type of treatment requires an aggregation of the multiple criteria into a single objective. Then a single-objective optimization algorithm is used to obtain a single Pareto point that satisfies designer preferences. Here we focus on the second class of approaches, which attempt to obtain the entire Pareto front. We present two methods in this second class, a linear weighting method and a single-objective product formulation. We then present a BiPSOMADS algorithm, which involves the single-objective product formulation and is based on the BiMADS method of Audet et al. [10], but uses PSO-MADS instead of MADS as the core optimization algorithm.

6.1.1 Linear Weighting Method

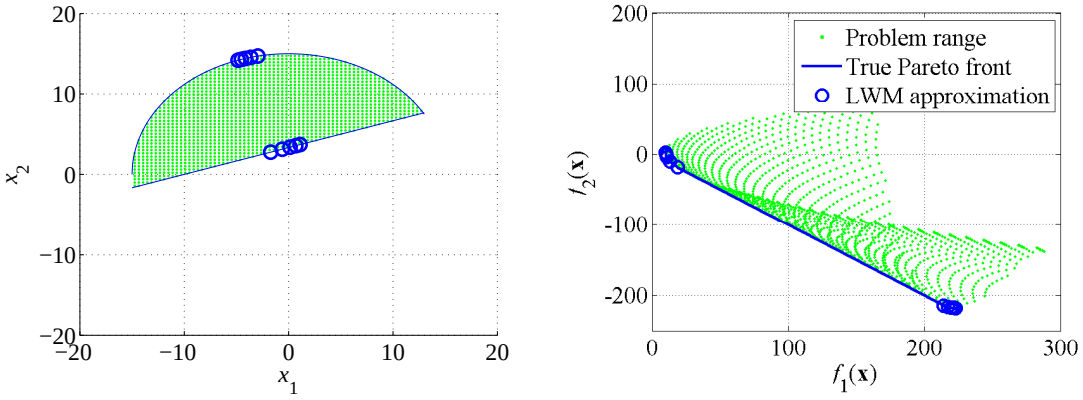
In the linear weighting method (LWM), the multiobjective optimization problem in (6.1) is converted to a single-objective problem that involves a convex combination of the objectives:

$$\min_{\mathbf{x}\in\Omega} \sum_{i=1}^{L} w_i f_i(\mathbf{x}), \tag{6.2}$$

where w_i, for all $i = 1, 2, \ldots, L$, are weights such that $\sum_{i=1}^{L} w_i = 1$. Any optimal solution of (6.2) is a Pareto optimal solution of (6.1). Therefore, LWM attempts to generate the Pareto front by solving (6.2) for different weight combinations. There are, however, two drawbacks to this approach. The first is that even using uniform distributions of the weight combinations, we often obtain clustered Pareto points rather than the desired uniform distribution of points along the Pareto front. The second concern with LWM is that it cannot produce Pareto points in the nonconvex part of the front, which means portions of the solution are not computed.

These drawbacks can be illustrated with the following biobjective example:

$$\begin{aligned} \min\ \mathbf{f}(\mathbf{x}) &= \begin{bmatrix} (x_1-2)^2+(x_2-1)^2+2 \\ 9x_1-(x_2-1)^2 \end{bmatrix} \\ \text{subject to:} & \\ x_1^2+x_2^2 &\leq 225, \\ x_1-3x_2 &\leq -10, \\ -20 \leq x_1,\ x_2 &\leq 20. \end{aligned} \tag{6.3}$$



(a) Design space (feasible domain)

(b) Objective space (range)

Figure 6.2: Results of LWM for the analytical biobjective optimization problem in (6.3), showing the true Pareto front (blue line) and LWM approximation (blue circles).

Figure 6.2 shows the results of optimizing (6.3) using LWM. Every point in the feasible design space depicted in Figure 6.2(a) has a corresponding point in the objective function space shown in Figure 6.2(b). In obtaining the LWM Pareto approximation, we optimize $F(\mathbf{x}) = \omega f_1(\mathbf{x}) + (1-\omega) f_2(\mathbf{x})$, subject to the constraints in (6.3), by setting $\omega = 0, 0.1, 0.2, \ldots, 1$. The LWM Pareto points in Figure 6.2(b) are obtained by solving the aggregated single-objective problem using the SQP optimization algorithm in MATLAB® (*fmincon* function [86]). This figure clearly shows the drawbacks of LWM. Specifically, we see the nonuniform distribution of the generated Pareto points even when ω is varied uniformly, and the lack of points in the nonconvex front section.

6.1.2 Single-objective Product Formulation

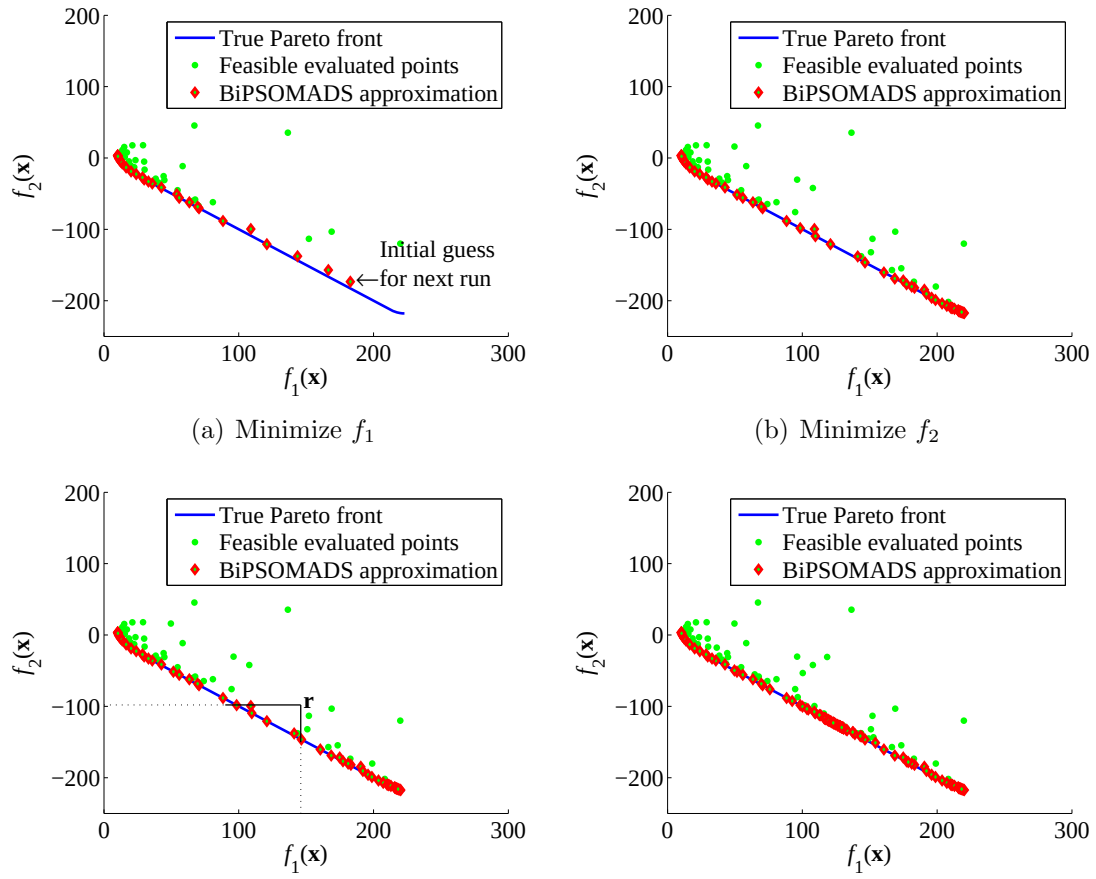
There are several alternate formulations that avoid the drawbacks associated with LWM (see [10] for a brief review of such methods). One such approach, which is the method considered here, solves multiobjective problems through a series of so-called single-objective product formulations (SOPFs). With this approach, each single-objective optimization problem requires the identification of a reference point $\mathbf{r}$ in the objective space. The idea is to place $\mathbf{r}$ in the interior of the objective space, and to then generate points that dominate $\mathbf{r}$ such that the distances between these generated solution points and $\mathbf{r}$ are maximized. These solution points can only extend as far as the boundary of the feasible region and thus can identify subsets of the Pareto front. By varying the position of $\mathbf{r}$, different subsets of the Pareto front can be generated. The method proceeds until the entire front is identified.

In SOPF, the problem in (6.1) is replaced with

$$\min_{\mathbf{x}\in\Omega} \psi_{\mathbf{r}}(\mathbf{x}) = -\prod_{i=1}^{L} \left(\max\{(r_i - f_i(\mathbf{x})), 0\}\right)^2, \tag{6.4}$$

where $\mathbf{r} \in \mathbb{R}^L$ is the reference point in objective space. By varying $\mathbf{r}$, we solve a sequence of problems that correspond to different $\psi_{\mathbf{r}}(\mathbf{x})$. The first two optimizations performed consider each objective individually.

The biobjective optimization procedure based on SOPF, described more formally in Section 6.2, is illustrated in Figure 6.3 for the simple problem introduced in (6.3). Figure 6.3(a) illustrates the first iteration of the procedure where the first objective function f_1 is minimized, using PSO-MADS. All of the feasible points that are evaluated during this optimization (in green), as well as the set of nondominated points (in red) which provide the current approximation of the Pareto front, are shown in the figure. The nondominated point with the smallest f_2 value is chosen as the initial guess for the second iteration of the biobjective optimization procedure. This second iteration entails minimizing f_2, again using PSO-MADS. The evaluated points and the Pareto approximation at this stage are shown in Figure 6.3(b).



(a) Minimize f_1

(b) Minimize f_2

(c) Reference point selection

(d) Solve SOPF in (6.5)

Figure 6.3: Illustration of BiPSOMADS minimization procedure using SOPF and a reference point $\mathbf{r}$, for the problem presented in (6.3).

After the first two iterations (minimizing f_1 and then f_2), each subsequent iteration involves identifying a reference point $\mathbf{r}$ followed by the solution of a SOPF problem (minimizing $\psi_{\mathbf{r}}$). The SOPF corresponding to (6.3) is:

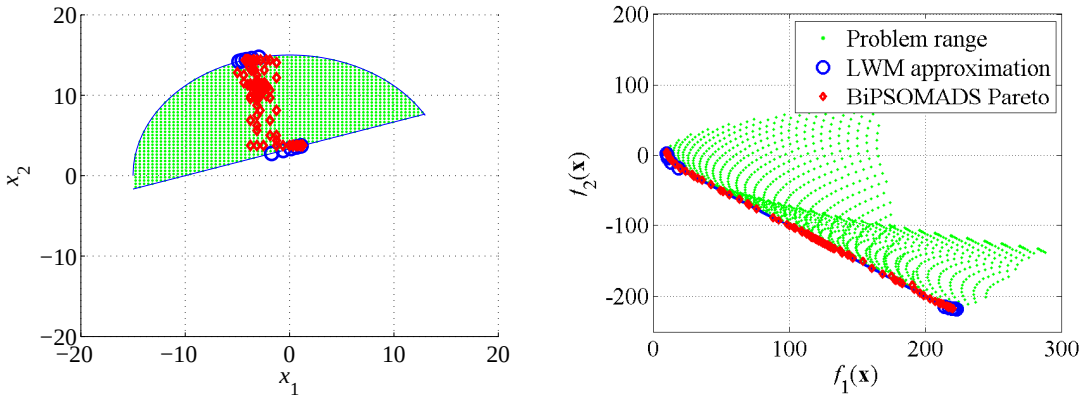
$$\begin{aligned} &\min \ -(\max\{r_1 - (x_1-2)^2 - (x_2-1) - 2, 0\})^2(\max\{r_2 - 9x_1 + (x_2-1)^2, 0\})^2, \\ &\text{subject to: } x_1^2 + x_2^2 \leq 225,\ x_1 - 3x_2 \leq -10,\ -20 \ \leq \ x_1,\ x_2 \leq 20. \end{aligned} \tag{6.5}$$

The reference point at each SOPF iteration is chosen such that the sparsest portion of the current approximation of the Pareto front is in the dominance zone of $\mathbf{r}$ (region

of objective space with solutions that dominate **r**). Within the dominance zone of **r**, (6.5) can be expressed as:

$$\begin{aligned} &\min\ -(r_1-(x_1-2)^2-(x_2-1)-2)^2(r_2-9x_1+(x_2-1)^2)^2, \\ &\text{subject to: } x_1^2+x_2^2\le 225,\ x_1-3x_2\le -10,\ -20\ \le\ x_1,\ x_2\le 20. \end{aligned} \tag{6.6}$$

This minimization seeks points that dominate **r** and are as 'far' from **r** as possible. Solving the SOPF (using PSO-MADS) with **r** selected as shown in Figure 6.3(c) yields the new Pareto points displayed in Figure 6.3(d). Thus after three PSO-MADS runs (from the three iterations of the biobjective procedure), a reasonable approximation of the true Pareto front has been generated. In order to obtain a better approximation of the front, additional PSO-MADS runs, using different values of **r**, must be performed.



(a) Design space (feasible domain)

(b) Objective space (range)

Figure 6.4: Results for SOPF (using BiPSOMADS) and LWM for the biobjective optimization problem (6.3), showing the true Pareto front (blue line), LWM approximation (blue circles) and BiPSOMADS Pareto front after three PSO-MADS runs (red diamonds).

Figure 6.4 shows the Pareto fronts computed using LWM and the BiPSOMADS algorithm. It is evident from Figure 6.4(b) that by solving a series of SOPF problems, we obtain a much better Pareto front than was obtained using LWM. We now describe BiPSOMADS in detail.

6.2 BiPSOMADS Algorithm

In this work, our focus is on problems with two objectives. Here we present a detailed description of the algorithm for solving a series of SOPFs in which the location of $\mathbf{r}$ is systematically varied to generate the Pareto front in a biobjective problem. The algorithm presented is essentially the same as BiMADS introduced by Audet et al. [10] except we use PSO-MADS instead of MADS to solve the single-objective subproblems (note that we could also use standalone PSO or MADS in our implementation). PSO-MADS handles nonlinear constraints using the filter treatment described in Chapter 2. Our presentation here follows that of Audet et al. [10].

Figure 6.5 presents an outline of the BiPSOMADS algorithm. BiPSOMADS is an iterative algorithm that constructs sets of points that approximate the Pareto optimal set $X_{\mathcal{P}}$. At each iteration, the current set of nondominated points, which dominate all other points found thus far, is denoted $X_{\mathcal{L}}$. The mapping of $X_{\mathcal{L}}$ under $\mathbf{f}$ to the objective space is denoted $Y_{\mathcal{L}} \in \mathbb{R}^L$. Note that $Y_{\mathcal{L}}$ defines the current approximation of the Pareto front $Y_{\mathcal{P}}$.

In the first step of the BiPSOMADS algorithm, the following two single-objective problems are solved:

$$\min_{\mathbf{x}\in\Omega} f_1(\mathbf{x}) \text{ and } \min_{\mathbf{x}\in\Omega} f_2(\mathbf{x}). \tag{6.7}$$

We apply the PSO-MADS algorithm from a given initial guess $\mathbf{x}_0 \in \Omega$. A first list of nondominated points, $X_{\mathcal{L}}$, is obtained from the set of points evaluated during these two optimization runs. The length of this list is J. The sets $X_{\mathcal{L}}$ and $Y_{\mathcal{L}}$ are sorted in ascending order of f_1 value, which corresponds to a descending order in the f_2 value. Gaps between nondominated points are quantified by computing Euclidean distances between successive (ordered) solutions in $Y_{\mathcal{L}}$. This strategy enables the determination of the next reference point. It should be noted that the set of nondominated points can be ordered in this manner in biobjective problems, but not in more general multiobjective problems. Since this ordering is essential for the BiPSOMADS algorithm, problems with more than two objectives will require alternate solution strategies.

From Figure 6.5 it can be seen that each iteration of BiPSOMADS consists of three steps. First, the ordered list $Y_{\mathcal{L}}$ is used to identify a reference point $\mathbf{r}$ in the objective

Initialization:

- Apply the PSO-MADS algorithm with initial guess $\mathbf{x}_0$ to solve $\min_{\mathbf{x}\in\Omega} f_1(\mathbf{x})$ and $\min_{\mathbf{x}\in\Omega} f_2(\mathbf{x})$.
- Let $X_{\mathcal{L}} = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^J\}$ be an ordered list of pairwise nondominated points such that $f_1(\mathbf{x}^1) < f_1(\mathbf{x}^2) < \ldots < f_1(\mathbf{x}^J)$ and $f_2(\mathbf{x}^1) > f_2(\mathbf{x}^2) > \ldots > f_2(\mathbf{x}^J)$. Initialize the weight $w(\mathbf{x}) = 0 \ \forall \ \mathbf{x} \in \Omega$ and let $\delta > 0$.

Main Iterations: Repeat

- **Reference Point Determination:**
  - If $J > 2$, let $\hat{\jmath} \in \arg\max_{j=2,\ldots,J-1} \ \delta^j = \frac{||\mathbf{f}(\mathbf{x}^j)-\mathbf{f}(\mathbf{x}^{j-1})||^2+||\mathbf{f}(\mathbf{x}^j)-\mathbf{f}(\mathbf{x}^{j+1})||^2}{w(\mathbf{x}^j)+1}$, and define the reference point $\mathbf{r} = \left(f_1\left(\mathbf{x}^{\hat{\jmath}+1}\right), f_2\left(\mathbf{x}^{\hat{\jmath}-1}\right)\right)$.
  - If $J = 2$, let $\mathbf{x}^{\hat{\jmath}} = \mathbf{x}^2$, define $\mathbf{r} = \left(f_1\left(\mathbf{x}^2\right), f_2\left(\mathbf{x}^1\right)\right)$, and $\delta^{\hat{\jmath}} = \frac{||\mathbf{f}(\mathbf{x}^2)-\mathbf{f}(\mathbf{x}^1)||^2}{w(\mathbf{x}^2)+1}$.
  - If $J = 1$, let $\mathbf{x}^{\hat{\jmath}} = \mathbf{x}^1$, $\delta^{\hat{\jmath}} = \frac{\delta}{w(\mathbf{x}^{\hat{\jmath}})+1}$ and apply the PSO-MADS algorithm with initial guess $\mathbf{x}^{\hat{\jmath}}$ to solve $\min_{\mathbf{x}\in\Omega} f_1(\mathbf{x})$ and $\min_{\mathbf{x}\in\Omega} f_2(\mathbf{x})$. Terminate PSO-MADS when the mesh size parameter Δ^m decreases below $\Delta\left(\delta^{\hat{\jmath}}\right) = O\left(\delta^{\hat{\jmath}}\right)$ and continue to the step **Update $X_{\mathcal{L}}$.**
- **Single-objective Formulation Minimization:**
  Solve a single-objective product formulation $\min_{\mathbf{x}\in\Omega} \psi_{\mathbf{r}}(\mathbf{x})$ using the PSO-MADS algorithm from initial guess $\mathbf{x}^{\hat{\jmath}}$. Terminate PSO-MADS when the mesh size parameter Δ^m decreases below $\Delta\left(\delta^{\hat{\jmath}}\right) = O\left(\delta^{\hat{\jmath}}\right)$ or if a maximum number of iterations is reached.
- **Update $X_{\mathcal{L}}$:**
  Add to $X_{\mathcal{L}}$ all nondominated points found in the current iteration, remove dominated points from $X_{\mathcal{L}}$, and order the resulting list of points.
  Increase weights $w\left(\mathbf{x}^{\hat{\jmath}}\right) \leftarrow w\left(\mathbf{x}^{\hat{\jmath}}\right) + 1 \ \forall \ \mathbf{x} \in X_{\mathcal{L}}$.

Figure 6.5: The BiPSOMADS algorithm for biobjective optimization (after [10]).

space. If $J > 2$, the strategy considers the weighted sum δ^j of squared distances between each nondominated point $\mathbf{f}(\mathbf{x}^j) \in Y_{\mathcal{L}}$ and its predecessor $\mathbf{f}(\mathbf{x}^{j-1})$ and successor $\mathbf{f}(\mathbf{x}^{j+1}) \ \forall \ j = 2, 3, \ldots, J-1$. The point $\mathbf{f}\left(\mathbf{x}^{\hat{\jmath}}\right)$ that maximizes the measure $\delta^{\hat{\jmath}}$ is identified from the list $Y_{\mathcal{L}}$. If there are only two nondominated points ($J = 2$), $\mathbf{x}^{\hat{\jmath}}$ is set to $\mathbf{x}^2$, $\mathbf{r}$ is set to $(f_1(\mathbf{x}^2), f_2(\mathbf{x}^1))$, and $\delta^{\hat{\jmath}}$ is set to the weighted squared distance between the two nondominated points. The weights are updated in a way that reduces frequent definitions of $\mathbf{r}$ around the same point. This is accomplished

by adding 1 to $w\left(\mathbf{x}^{\hat{\jmath}}\right)$ at the end of the iteration. If $J = 1$, i.e., if a single point $\mathbf{x}^1$ dominates all other points evaluated thus far, then the algorithm again solves the two single-objective problems in (6.7).

After the determination of the reference point, the second step of the BiPSOMADS iteration entails the solution of the single-objective formulation $\min_{\mathbf{x}\in\Omega} \psi_{\mathbf{r}}(\mathbf{x})$ using the PSO-MADS hybrid algorithm. Many of the points produced during this optimization are expected to dominate $\mathbf{r}$ (recall Figure 6.3(c)). The PSO-MADS optimization terminates when the mesh size parameter Δ^m drops below a specified threshold or if a maximum number of iterations is reached. The final step of the BiPSOMADS approach involves updating the set of nondominated points $X_{\mathcal{L}}$ at the end of each optimization run.

In our work, the algorithm is terminated after a fixed number of BiPSOMADS iterations (PSO-MADS optimization runs). A termination criterion based on $\delta^{\hat{\jmath}}$ reaching a prescribed threshold could also be used. Any single-objective optimization algorithm can be used to solve the SOPF subproblems, so the overall method is quite general. We apply PSO-MADS because of its global search and local convergence character, which was shown to be very useful in the field development examples in the preceding chapters.

6.3 Example Cases

The optimization approaches presented in the preceding sections will now be applied to biobjective reservoir management and field development problems. In the first example, we determine the optimal controls for water injection wells in order to maximize both NPV and cumulative oil production (COP). In the second case, we optimize the number of wells, their type, drilling sequence, locations and controls, to again maximize both NPV and COP. This second case is essentially a biobjective version of Case 4 in Chapter 5. In the last example, we utilize our biobjective procedure to optimize field development while considering geological uncertainty. The goal in this case is to generate field development scenarios that are Pareto optimal and robust in terms of both the expected NPV and the worst-case NPV.

6.3.1 Case 1: Well Control Optimization

In this case, the objectives are to maximize both NPV and COP in a waterflooding problem. The reservoir contains three production wells and two injection wells. The reservoir model is the same as in Case 1 of Chapter 5. The simulation parameters for the model are as shown in Tables 5.1 and 5.2, except for the following changes. The oil price and water cost parameters here are p_o = \$80/STB, c_{pw} = \$20/STB and c_{iw} = \$15/STB. Water costs are set to very high values to emphasize the difference between the two objective functions considered. There are no nonlinear constraints specified for this problem. The producer BHPs are fixed at 2500 psi and the injector BHPs can vary between 4000 psi and 9000 psi. The production time frame is 3000 days. This is divided into five control periods of 600 days each. The injection and production wells are fixed in the locations shown in Figure 6.6.

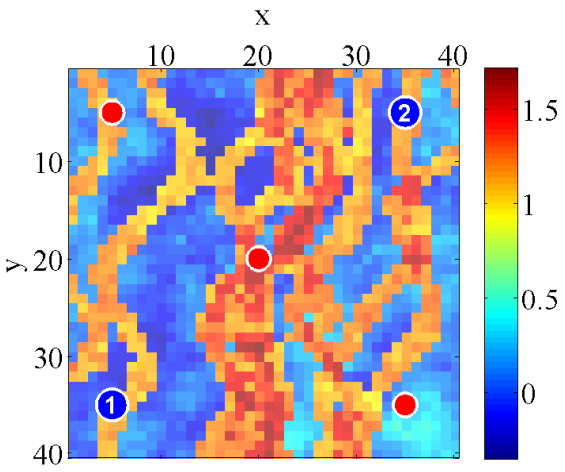
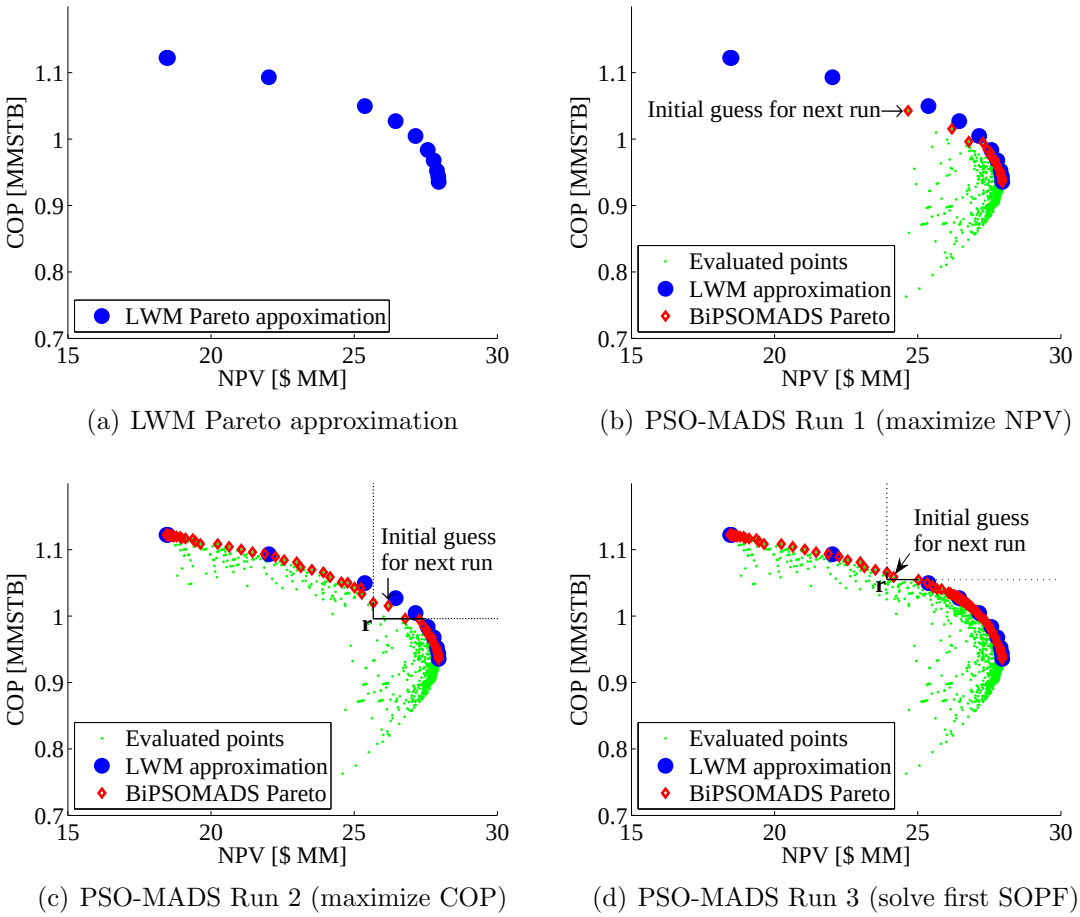


Figure 6.6: Geological model ($\log_{10}$ of isotropic permeability field, with permeability in md), showing locations of injection (in blue) and production (in red) wells.

Only the injection well BHPs are optimized, so we have just ten optimization variables (five BHP controls for each of the two injection wells). The LWM and BiPSOMADS optimization approaches are applied to this biobjective well control problem. Both approaches use PSO-MADS as the core optimization algorithm. The PSO swarm size is ten particles and the maximum number of poll points for MADS is 20 ($2n$). The function evaluations are fully parallelized.



(a) LWM Pareto approximation

(b) PSO-MADS Run 1 (maximize NPV)

(c) PSO-MADS Run 2 (maximize COP)

(d) PSO-MADS Run 3 (solve first SOPF)

Figure 6.7: LWM Pareto front approximation and Pareto approximations from the first three BiPSOMADS steps (Case 1).

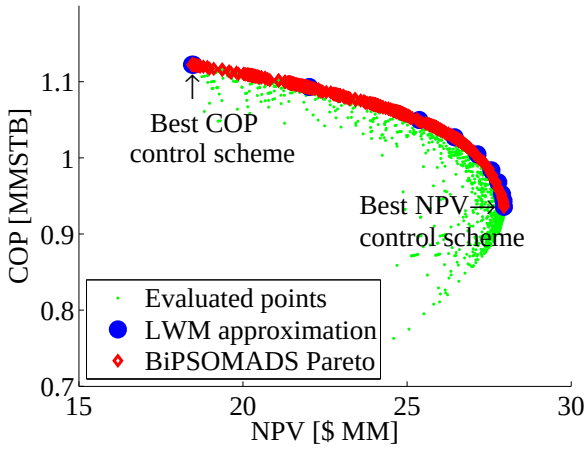
The results of LWM and the first three steps of the BiPSOMADS procedure are shown in Figure 6.7. Figure 6.7(a) presents the LWM Pareto approximation, obtained from the maximization of $F(\mathbf{u}) = \omega f_1(\mathbf{u}) + (1-\omega) f_2(\mathbf{u})$, where f_1 is NPV and f_2 is COP. These results are generated by setting $\omega = 0, 0.1, 0.2, \ldots, 1$, implying 11 PSO-MADS runs. Here the vector of continuous control variables $\mathbf{u}$ represents the injection well BHPs. From Figure 6.7(a) we can observe one of the drawbacks of LWM, which is that even for uniformly distributed ω values, the resulting Pareto front is nonuniformly distributed. This drawback in LWM, which is due to the use of a uniform distribution of ω values, can be alleviated by using the normal-boundary intersection method of

Das and Dennis [35] or by using an adaptive weighted sum approach, as was done by Kim and de Weck [70]. Neither of these approaches was considered in this work.

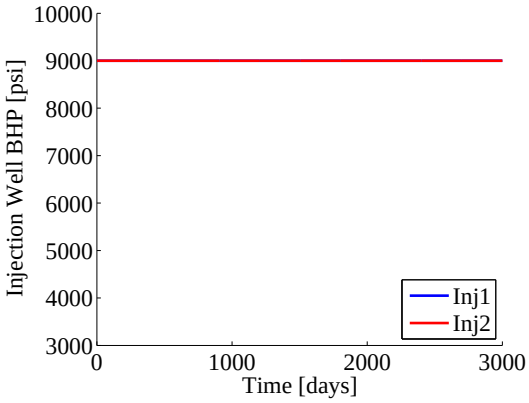
Figure 6.7(b) shows the results of the first step of the BiPSOMADS procedure, which is the PSO-MADS optimization of $f_1 = \text{NPV}$. After this optimization, the nondominated point with the best COP value is selected as the initial guess for the next step, which is the optimization of $f_2 = \text{COP}$. The results of this optimization are presented in Figure 6.7(c). After the first two optimization steps, subsequent steps involve selection of a reference point $\mathbf{r}$, determined from the sparsest part of the current Pareto approximation, and optimization of the corresponding SOPF. The results of the first SOPF optimization (third PSO-MADS run) are illustrated in Figure 6.7(d). It should be observed that the newly generated points in this figure are all in the dominance zone of $\mathbf{r}$ as illustrated in Figure 6.7(c). Figure 6.7(d) shows the selection of $\mathbf{r}$ for the next optimization run. This process of reference point selection and SOPF solution is continued in order to obtain progressively better approximations of the Pareto front.

The final results, obtained after ten PSO-MADS runs in the BiPSOMADS procedure, are illustrated in Figure 6.8. Figure 6.8(a) shows the final LWM and BiPSOMADS Pareto front approximations. BiPSOMADS clearly provides a more uniform distribution of points along the Pareto front than LWM. Also, we see that for this problem, the NPV and COP objectives are conflicting. The best NPV control scheme yields a NPV of \$27.95 million and COP of 0.9364 MMSTB (this represents a 17% decrease from the maximum achievable cumulative oil recovery). The best COP control scheme yields a COP of 1.122 MMSTB and a NPV of \$18.45 million (a 34% decrease from the maximum achievable NPV).

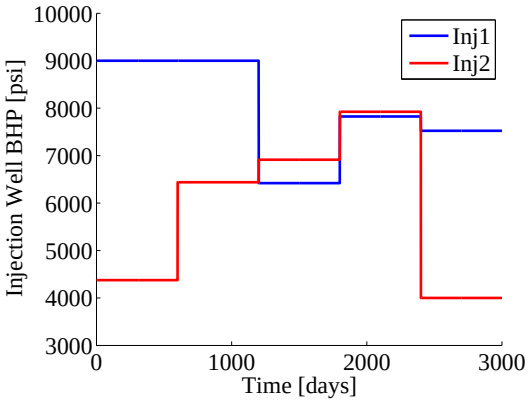
Figures 6.8(b) and 6.8(c) present the injection BHP profiles, and Figures 6.8(d) and 6.8(e) depict the resulting cumulative production and injection profiles, for the best COP and best NPV control schemes. From these plots we observe that, in order to maximize oil recovery, the best control scheme is to operate the injection wells at their maximum allowable BHPs. This acts to inject as much water as possible into the reservoir. However, due to the water handling costs, injecting too much water will negatively affect NPV. Hence the best NPV solution varies the injection BHPs to better manage water use, even though this results in the production of less oil.



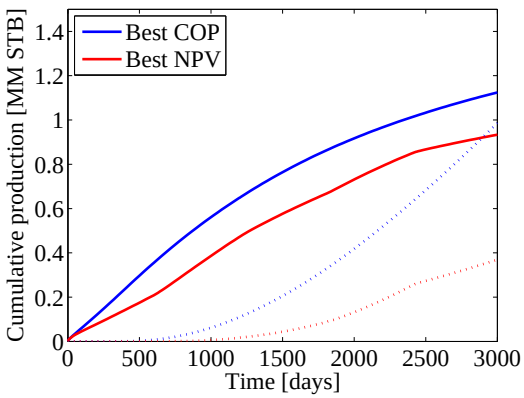
(a) LWM and BiPSOMADS Pareto front approximations, after eleven LWM and ten BiPSOMADS steps, each involving PSO-MADS runs



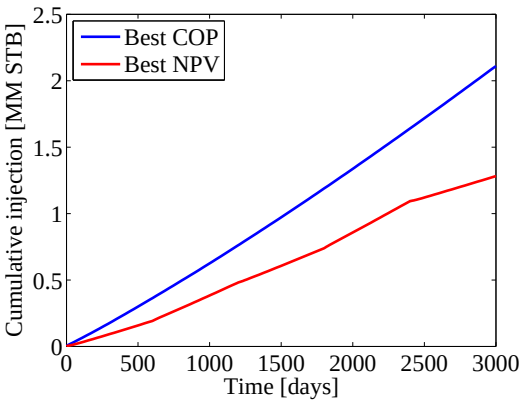
(b) Best COP



(c) Best NPV



(d) Cumulative field oil (solid) and water (dotted) production



(e) Cumulative field water injection

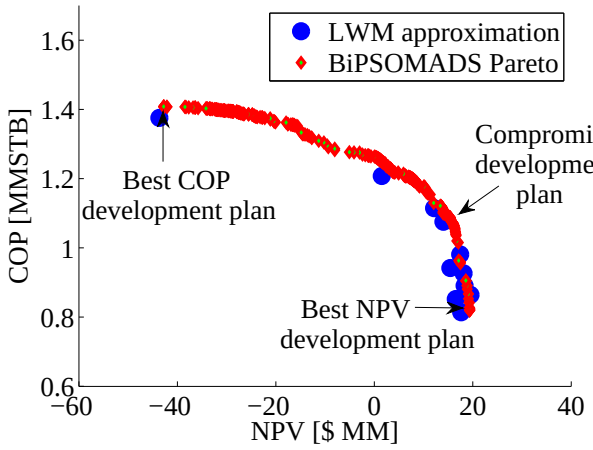
Figure 6.8: Pareto front approximations, together with the best BiPSOMADS COP and NPV control schemes for injection well BHPs. Resulting cumulative production and injection profiles from the two control schemes are also shown (Case 1).

It is important to reiterate that every point in the LWM and BiPSOMADS Pareto approximations corresponds to particular control schemes that provide specific NPV and COP. The exact choice of which control scheme to implement will depend on the preferences of the decision maker. This example demonstrates the utility of BiPSOMADS for enabling this type of decision.

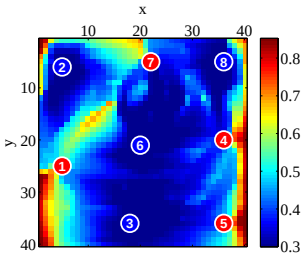
6.3.2 Case 2: Full Field Development

In this example, we consider the biobjective optimization of a field development case that involves the optimization of the number of wells, their well type and drilling sequence, and their locations and controls. This example is a biobjective extension of Case 4 in Chapter 5, where the single objective was NPV. Here we address the same problem but consider both NPV and COP as in the preceding case. The simulation and optimization parameters are as described in Section 5.2.4. Initially, we do not include the nonlinear constraints in the biobjective optimization (they are included later). This problem contains 68 optimization variables, with eight ternary categorical variables (handled using the MADS extended polling stencil discussed in Section 5.1.4). The same PSO-MADS parameters as were used in Section 5.2.4 are used here (e.g., PSO swarm size of 50).

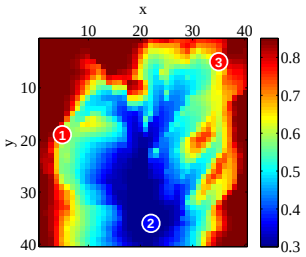
The results for the biobjective field development problem, without nonlinear constraints, are shown in Figure 6.9. The LWM Pareto approximation is again obtained from maximizing the aggregate function $F(\mathbf{u}, \mathbf{v}, \mathbf{z}) = \omega f_1(\mathbf{u}, \mathbf{v}, \mathbf{z}) + (1-\omega) f_2(\mathbf{u}, \mathbf{v}, \mathbf{z})$, with $\omega = 0, 0.1, 0.2, \ldots, 1$ (11 PSO-MADS runs). Ten PSO-MADS optimizations are again run in the BiPSOMADS procedure. The final Pareto approximation is shown in Figure 6.9(a). The best COP solution entails a development plan with eight wells (four injection and four production wells) drilled in the order and in the locations shown in Figure 6.9(b). This plan yields 1.408 MMSTB of oil and a NPV of -$42.79 million. In this case, almost all of the accessible oil in the reservoir is produced, as is evident in the nearly complete sweep in Figure 6.9(b). This scenario entails the use of a very large amount of water, as can be seen in Figures 6.9(e) and 6.9(f). As a result the NPV for this development plan is negative.



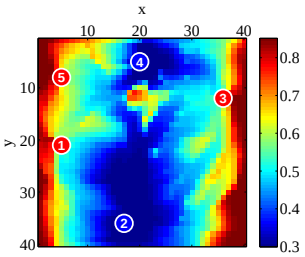
(a) LWM and BiPSOMADS Pareto front approximations, after eleven LWM and ten BiPSOMADS steps, each involving PSO-MADS runs



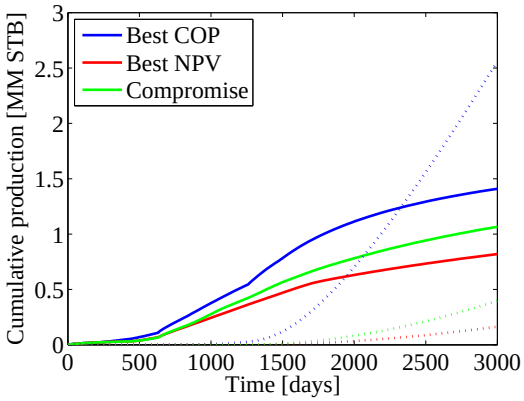
(b) Best COP



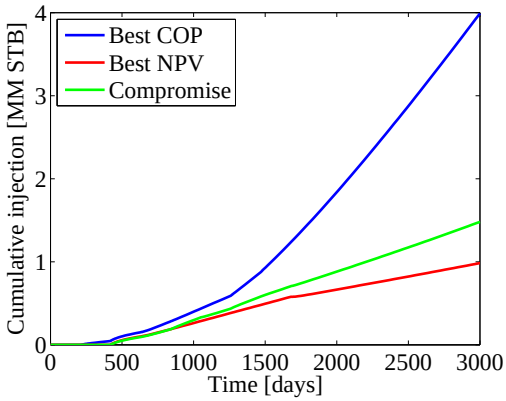
(c) Best NPV



(d) Compromise



(e) Cumulative field oil (solid) and water (dotted) production



(f) Cumulative field water injection

Figure 6.9: Pareto front approximations, together with the best COP, best NPV and compromise field development solutions, showing well configurations, drilling sequences and final oil saturations. Also shown are the resulting cumulative production and injection profiles for the three development plans (Case 2).

The best NPV solution, by contrast, entails a three-well (two producers and one injector) development plan, as depicted in Figure 6.9(c). This scenario provides \$19.39 million in NPV and 0.8189 MMSTB of recovered oil (42% less than the best achievable COP). Figure 6.9(d) shows a possible 'compromise' solution selected from near the bend in the Pareto front (see Figure 6.9(a)). This five-well development plan yields 18% more oil, while only sacrificing 8% in NPV, relative to the best NPV case. From Figure 6.9 we see that, as would be expected, the compromise solution leads to reservoir sweep and cumulative production and injection profiles that are between the best COP and best NPV cases.

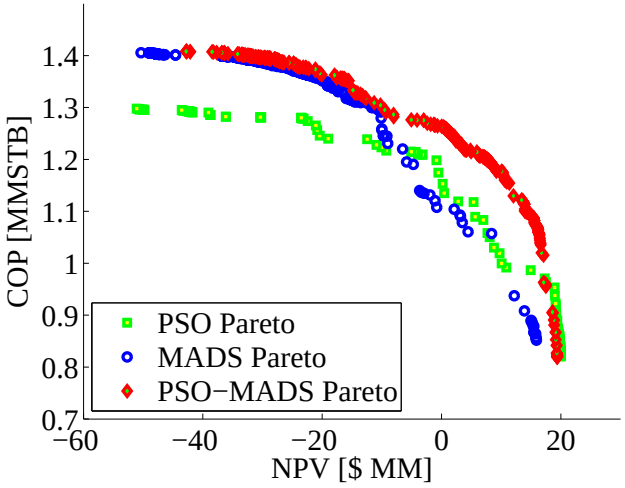


Figure 6.10: Pareto fronts from SOPF biobjective procedure using PSO, MADS and PSO-MADS as core optimization algorithms.

As noted earlier, any single-objective optimization algorithm can be used to solve the single-objective subproblems within the biobjective optimization framework. Figure 6.10 presents a comparison of the resulting Pareto approximations, after ten optimization runs in the biobjective procedure, using PSO, MADS and PSO-MADS as the core optimization algorithms. Note that this usage of MADS corresponds to the BiMADS algorithm of Audet et al. [10]. It is evident that the use of PSO-MADS in the SOPF biobjective procedure leads to the best Pareto front approximation. This observation is consistent with our earlier finding that the PSO-MADS hybrid outperforms its component methods for general field development optimization.

It should be noted that a large portion of the Pareto front in this problem corresponds to negative NPVs. This is due to high water handling costs (see Table 5.10) and to the fact that we have not included nonlinear constraints, such as a maximum field water injection rate, in the problem. Thus the best COP solution involves injecting and producing very large amounts of water. This issue can be addressed by including the nonlinear constraints in Table 5.10 in the biobjective field development problem. We accomplish this using the filter method, described in Chapter 2, for handling the constraints in each single-objective subproblem of BiPSOMADS.

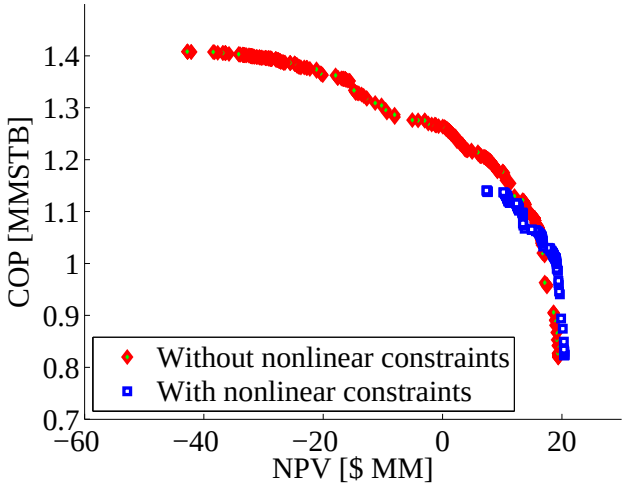


Figure 6.11: Pareto fronts from BiPSOMADS with and without nonlinear constraints included in biobjective optimization problem (Case 2).

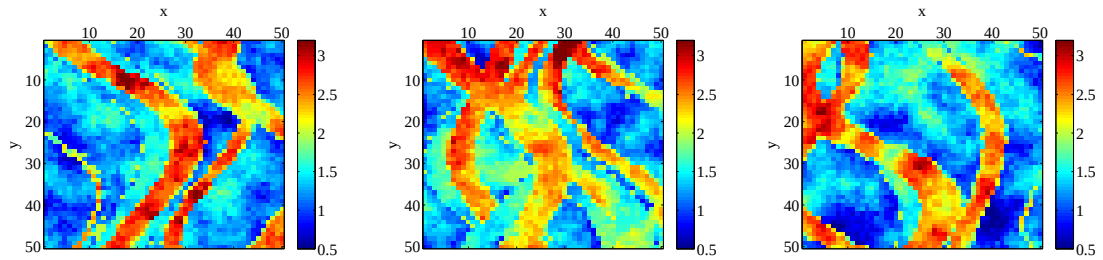
With this approach, for the nonlinearly constrained problem, we obtain the Pareto front (after ten PSO-MADS runs) shown in Figure 6.11. It is clear from the figure that the nonlinear constraints, especially the maximum field water injection constraint, prevent the injection of excessive amounts of water into the reservoir. This leads to positive NPVs at all points on the Pareto front. This result demonstrates that we can utilize BiPSOMADS with the filter constraint treatment to optimize field development cases with two objectives, subject to general constraints.

6.3.3 Case 3: Biobjective Approach for Optimization Under Uncertainty

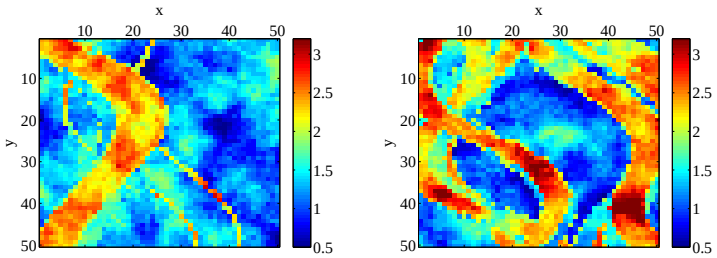
The BiPSOMADS procedure is now applied to perform field development optimization under geological uncertainty. This uncertainty is characterized by ten geological realizations, depicted in Figure 6.12, which are assumed to be equally probable. We will consider all of these realizations in the optimizations. Each realization is a 50×50 model of a fluvial reservoir, with meandering channel and levee sands predominantly oriented in the y-direction. The realizations were generated using object-based simulation for creating the channel, levee and shale facies, followed by sequential Gaussian simulation for the assignment of permeability. SGeMS [102] was applied for both stages. The reservoir is to be produced by waterflooding. The simulation and optimization parameters are the same as those used in Case 3 of Chapter 5. The relative permeability curves for the reservoir fluids are as shown in Figure 3.2. The other simulation and optimization parameters are as presented in Tables 5.1 and 5.8, except here we have two-dimensional 50×50 models, a drilling cost of \$10 million per well, and we neglect nonlinear constraints. The production time frame is 1500 days and there are three BHP control periods of 500 days each. The field development optimization problem involves the determination of the optimal number, well type, locations and controls for a maximum of six wells. There are thus a total of 36 optimization variables in this problem (6 ternary categorical variables, 12 areal location variables and 18 BHP control variables).

The goal of these optimizations is to generate optimal solutions that are 'robust' given the geological uncertainty. We consider the expected value of NPV (E[NPV]) and the worst-case NPV (min[NPV]) over the ten realizations. More specifically, we apply BiPSOMADS to maximize $f_1(\mathbf{u}, \mathbf{v}, \mathbf{z})$ and $f_2(\mathbf{u}, \mathbf{v}, \mathbf{z})$ where

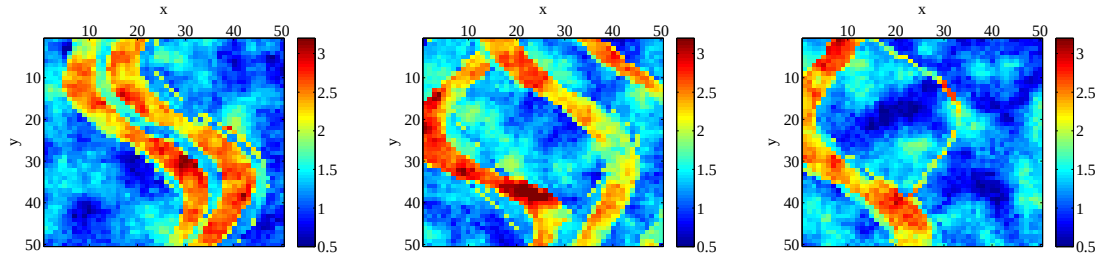
$$\begin{aligned} f_1(\mathbf{u}, \mathbf{v}, \mathbf{z}) &= \mathrm{E}[NPV] = \frac{1}{N_R} \sum_{i=1}^{N_R} \mathrm{NPV}_i(\mathbf{u}, \mathbf{v}, \mathbf{z}), \\ f_2(\mathbf{u}, \mathbf{v}, \mathbf{z}) &= \min[NPV] = \min_{i \in \{1,\ldots,N_R\}} \mathrm{NPV}_i(\mathbf{u}, \mathbf{v}, \mathbf{z}). \end{aligned} \tag{6.8}$$



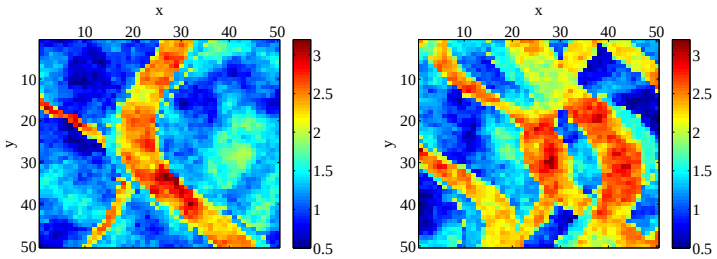
(a) Realization 1 (b) Realization 2 (c) Realization 3



(d) Realization 4 (e) Realization 5



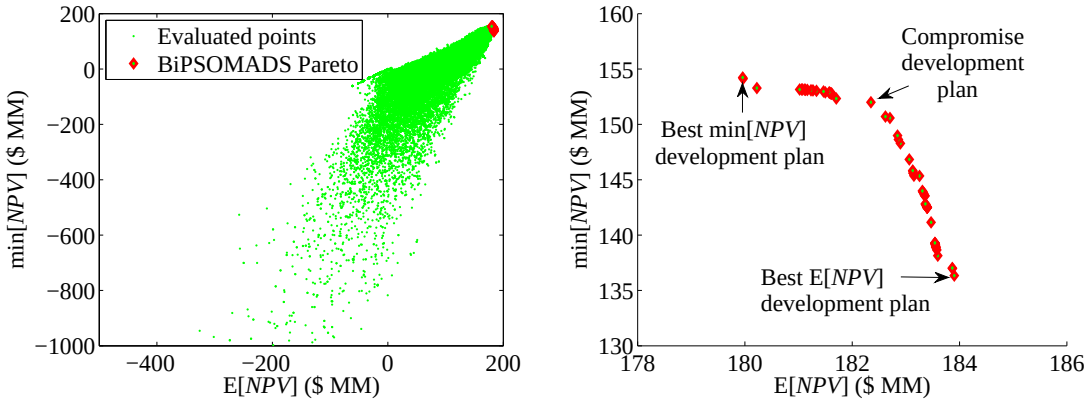
(f) Realization 6 (g) Realization 7 (h) Realization 8



(i) Realization 9 (j) Realization 10

Figure 6.12: Ten geological realizations ($\log_{10}$ of isotropic permeability field, with permeability expressed in md) of a fluvial reservoir (Case 3).

Here N_R is the number of realizations ($N_R = 10$ in this case) and $\text{NPV}_i(\mathbf{u}, \mathbf{v}, \mathbf{z})$ represents the NPV that results from applying the development plan defined by $(\mathbf{u}, \mathbf{v}, \mathbf{z})$ to realization i. The motivation for this choice of f_1 and f_2 is that, although we seek to maximize the expected NPV, we also wish to protect against the worst-case scenario. Note that, because this example includes ten realizations, the worst case corresponds to the 'P10' result, where P10 refers to the p-value (10th percentile) on the cumulative distribution function. In the PSO-MADS algorithm applied in the BiPSOMADS procedure, the PSO swarm size is 50, and we use the extended categorical stencil in MADS with a maximum of 72 poll points.



(a) All evaluated points

(b) Pareto approximation

Figure 6.13: Results of biobjective field development optimization under uncertainty. BiPSOMADS used to optimize expected NPV and worst-case NPV from ten geological realizations (Case 3).

Figure 6.13 presents the results from the BiPSOMADS optimization procedure after ten PSO-MADS runs. Figure 6.13(a) shows that the expected NPV and worst-case NPV are generally nonconflicting objectives. This seems reasonable since the field development plan that maximizes the NPV for the worst-performing realization will also usually improve performance for other realizations, which leads to improved $\text{E}[NPV]$. Despite this generally nonconflicting nature of the two objectives, Figure 6.13(b) shows that it might still be useful to approach this problem in a biobjective fashion in order to quantify the detailed trade-offs between the two objectives.

From Figure 6.13(b) we observe that optimizing only expected NPV yields E[NPV] = \$183.9 million and min[NPV] = \$136.3 million, while optimizing only the worst-case NPV yields E[NPV] = \$180.0 million and min[NPV] = \$154.2 million. Thus there is a slight trade-off in the expected NPV (2% variation in expected NPV between the two cases), but a more significant trade-off in the worst-case NPV (12% variation between the two cases). By selecting the compromise solution indicated in Figure 6.13(b), we achieve E[NPV] of \$182.3 million and min[NPV] of \$152.0 million. This compromise development plan is thus near-optimal in terms of expected NPV (within 1% of the optimal expected NPV) while still mitigating the risk associated with the worst-case scenario (within 1.5% of the optimal worst-case NPV).

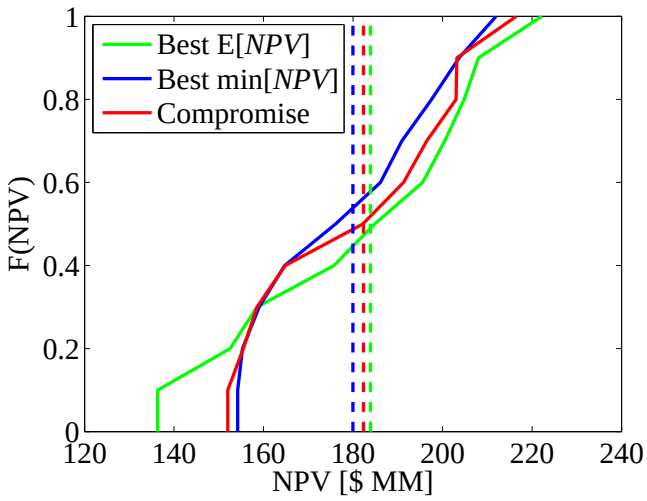


Figure 6.14: Comparison of NPV CDFs for the best expected NPV, best worst-case NPV and the compromise solutions (the dashed lines indicate the mean values), obtained from the ten realizations (Case 3).

Figure 6.14 presents the cumulative distribution functions (CDFs) of NPV over the ten realizations, for the different development plans indicated in Figure 6.13(b). From this figure we see that, by optimizing only expected NPV, we arrive at a field development plan that has a substantial downside risk (specifically, there is a 10% probability of achieving NPV of \$136.3 million). We also see that optimizing only the worst-case NPV leads to a decrease in expected performance compared to optimizing E[NPV], though the decrease is small for this particular case. The compromise development

plan provides better expected performance than the optimized min[NPV] plan, while still guarding against the worst-case scenario. This example demonstrates that, even though the expected NPV and worst-case NPV are not in general conflicting objectives, there is still benefit in approaching this problem in a biobjective fashion.

This example does not include nonlinear constraints. The handling of these constraints when optimizing under uncertainty requires some additional considerations. For example, if we have a field-level water injection constraint, we need to decide if this constraint must be satisfied by the development and production plan in all model realizations or only in some specific fraction. Once a decision is made on this issue, an aggregate constraint violation function h can be defined and then used in the filter treatment of PSO-MADS. Some additional penalty associated with the realizations that do not honor the nonlinear constraints could also be introduced into the formulation.

6.4 Summary

In this chapter, we presented approaches for biobjective optimization. We described the straightforward LWM approach and demonstrated its limitations. We then developed the BiPSOMADS procedure for the solution of optimization problems with two objectives. The LWM and BiPSOMADS approaches were applied to three biobjective field development example cases. The first case involved a well control optimization, with NPV and COP as objectives, where the optimization variables were the injection well BHPs. The second example entailed full field development, again with NPV and COP as objectives, where the goal was to determine the optimal number of wells, well types, locations and controls. The first two problems, which displayed conflicting objectives, were successfully addressed with both the LWM and BiPSOMADS approaches. The BiPSOMADS procedure, however, provided significantly better Pareto fronts than LWM. In the last example, we demonstrated the utility of the biobjective optimization approach for field development under geological uncertainty. Expected NPV and worst-case (P10) NPV were chosen as the two objectives. Although these objectives are not strictly conflicting, our example demonstrated that it can still be useful to apply a biobjective approach for this problem. This enabled us

to maximize the expected NPV while reducing the risk associated with the worst-case scenario.

Chapter 7

Summary, Conclusions and Future Work

This work entailed the development and implementation of a derivative-free optimization framework applicable to the solution of complex constrained mixed-integer nonlinear programming (MINLP) problems. The target application area for this framework is the optimization of oil field development and operation. These problems typically involve categorical, integer and/or continuous variables, as well as bound, linear and nonlinear constraints. Our general optimization framework includes the following main components:

- The derivative-free optimization methods comprising the core framework include MADS, a local search optimization method; PSO, a global search algorithm; a new PSO-MADS hybrid with both local convergence and global search characteristics; and B&B, an optimization approach for conducting a systematic tree-like search of the categorical variable combinations. We also introduced a heuristic extension of the MADS stencil used in the PSO-MADS hybrid for problems with categorical variables that have more than two categories.

- Nonlinear constraint handling techniques for use in optimization problems with general constraints were introduced into the formulation. Filter methods were applied for this purpose because of their versatility and because they do not involve tuning parameters, as required by other constraint handling approaches

such as penalty methods. The filter method had previously been implemented for use with MADS. We extended this procedure to both the PSO and PSO-MADS algorithms.

- A biobjective optimization procedure that utilizes PSO-MADS to solve a sequence of single-objective product formulations of the biobjective problem was developed. The goal of this BiPSOMADS procedure is to generate an approximation of the Pareto front, which represents the optimal trade-off between the two objectives being optimized.

The new optimization framework described above was applied to generalized field development and reservoir management problems. These problems are true MINLP problems with categorical variables that determine the optimal number of wells to drill in addition to the well types and drilling sequence, integer variables that define the locations of the wells in the reservoir, and continuous variables that represent well controls (BHPs were used here but well controls could also be rates or choke settings). Field development problems also include bound constraints that ensure that the wells are located within the reservoir boundaries and that the BHPs are within specified pressure ranges, and nonlinear constraints including well-to-well distance, field-level water injection rate, oil production rate and total fluid production rate, and well water cut constraints.

The key findings from this study are as follows:

- The filter methods used in conjunction with MADS, PSO and PSO-MADS were shown to be effective constraint handling techniques that enable the generation of feasible optimal solutions, even starting from infeasible initial guesses. In addition, these approaches provide information on the sensitivity of the objective function to the specified constraints.

- For well control and well placement optimization, it was found that the joint solution of the combined problem (i.e., simultaneous optimization) outperformed procedures that address the two problems separately and in sequence for the problems considered.

- In all cases in which PSO-MADS was compared to standalone PSO and MADS, the PSO-MADS hybrid was shown to provide better solutions than its component PSO and MADS methods. This can be viewed as a validation of the search-poll paradigm on which the hybrid optimization procedure is based.

- The extended MADS categorical stencil implemented for use in MADS and PSO-MADS was also found to be quite effective. This procedure provided improved results compared to those with the standard stencil, though it is more expensive computationally.

- The BiPSOMADS optimization procedure was shown to be an effective solution technique for problems with two objectives, such as NPV and cumulative oil recovery. We demonstrated that it can also be used for performing optimization under geological uncertainty, where the goal is to maximize the expected performance of the reservoir development project while safeguarding against downside risk.

The methods developed in this work entail substantial computational requirements. However, because they naturally parallelize, the computational demands can be alleviated by distributed computing, as was done in this work. Our overall conclusion is that the new optimization framework developed in this thesis is indeed applicable to generalized field development and reservoir management problems.

The methodology introduced in this work is quite general but there are still issues that should be investigated further. The following areas are suggested for future research:

- Strategies should be developed to improve the computational efficiency of the framework, especially for problems involving optimization under uncertainty. These strategies could employ computationally efficient surrogate models to replace some of the required reservoir simulations, or upscaling techniques to reduce the run times of the reservoir simulations. Techniques for selecting subsets of representative realizations in problems with multiple realizations should also be implemented.

- The use of other optimization methods should be considered, including different PSO families [44] and the cloud PSO approach [47]. Since the framework presented here is quite general, a wide range of global and local optimization algorithms could be incorporated. Of particular interest are other MINLP approaches, besides Branch and Bound, for the rigorous treatment of problems with categorical variables. Techniques that might be considered include outer approximation approaches [49], methods based on generalized Benders decomposition [52], and extended cutting plane procedures [122].

- Other approaches for robust optimization under uncertainty should be assessed. This investigation should focus on procedures to minimize computational expense, and on optimization under uncertainty with general constraints.

- In reality, because the reservoir description is uncertain, as each new well is drilled, more information about the reservoir is obtained. This new geological information should be applied to update the reservoir description. This suggests the use of a closed-loop field development strategy that entails a sequence of wells and updated models. The use of the optimization algorithms developed in this work within such a framework should be studied.

- The optimization framework could be extended from biobjective to multiobjective problems to enable the treatment of cases with three or more objectives. Extension of the BiPSOMADS single-objective formulation might be feasible, though such an approach would likely entail a significant amount of computational effort. It could therefore be useful to investigate new strategies to generate the multiobjective Pareto surface, such as methods based on MOPSO [29] or NSGA-II [37].

- The applications considered in this work involved the development of fields under waterflood and the use of black-oil simulations. Application of the general optimization framework to reservoir problems involving other recovery processes, such as steam-assisted gravity drainage (SAGD), gas injection, etc., should be considered.

- The methods developed in this work should be tested on real field applications. This will identify areas where enhancements are required. The use of our optimization framework for MINLP problems in areas outside of petroleum engineering should also be explored.

Nomenclature

Abbreviations

B&B	branch and bound
BHP	bottomhole pressure
BiMADS	biobjective optimization framework using MADS to solve a sequence of single-objective formulations
BiPSOMADS	biobjective optimization framework using PSO-MADS to solve a sequence of single-objective formulations
CDF	cumulative distribution function
CMA-ES	covariance matrix adaptation evolution strategy
COP	cumulative oil production
FD	finite difference
GA	genetic algorithm
GPS	generalized pattern search
LC	linear constraint
LWM	linear weighting method for biobjective optimization
MADS	mesh adaptive direct search
MINLP	mixed-integer nonlinear programming
MM	million
NLP	nonlinear programming
NPV	net present value
<NPV>	average net present value
E[NPV]	expected value of net present value
min[NPV]	worst-case NPV
PSO	particle swarm optimization

PSO-MADS	hybrid of particle swarm optimization and mesh adaptive direct search
RB	reservoir barrels
SGeMS	Stanford Geostatistical Modeling Software
SOPF	single-objective product formulation
SPSA	simultaneous perturbation stochastic approximation
SQP	sequential quadratic programming
STB	stock tank barrels
WI	well index

Greek Symbols

Δ	MADS stencil size
Δ_k^m	MADS mesh size at iteration k
Δ_k^p	MADS poll size at iteration k
μ	viscosity
Ω	bounded set defining allowable values of optimization variables
ω	PSO inertial parameter
ϕ	poll size contraction parameter
ρ	density
σ	standard deviation
τ	characteristic time
θ	mesh size contraction parameter

Variables

B	formation volume factor
b	fractional discount rate
C	well drilling cost
c	cost per barrel
c_1	PSO cognitive parameter
c_2	PSO social parameter
$\mathbf{c}$	vector of nonlinear constraint functions
$\mathbf{d}$	MADS polling direction
$\mathcal{F}$	filter
f	optimization objective function

$\mathbf{f}$	vector of multiple objectives
$\mathbf{g}$	vector of reservoir simulation equations
h	aggregate normalized constraint violation function
I	set of indices for injection wells
k	iteration index or time interval index
k_r	relative permeability
L	number of objectives
M	underlying mesh in MADS
m	number of nonlinear inequality constraints
n	number of optimization variables
n_1	number of integer (pseudo-continuous) well placement variables
n_2	number of continuous well control variables
n_c	number of categorical variables
N_R	number of realizations
P	set of indices for production wells
p	price per barrel or pressure
$\mathbf{p}$	vector of dynamic state variables of reservoir model
$\mathbf{r}$	reference point in objective space, for use in BiPSOMADS algorithm
S	size of PSO swarm or saturation in grid block
U	bounded set defining allowable values for continuous well control variables
$\mathbf{u}$	vector of continuous well control variables
V	bounded set defining allowable values for integer (pseudo-continuous) well location variables
$\mathbf{v}$	vector of integer (pseudo-continuous) well location variables, or velocity in PSO update equation
X	set of nondominated solutions
$\mathbf{x}$	vector of general optimization variables
Y	mapping of set of nondominated solutions in objective space
$\mathbf{y}$	best position evaluated by PSO particle
z	categorical variable
$\mathbf{z}$	vector of categorical variables

Subscripts

0	initial
I	refers to set of injection wells
iw	injected water
k	iteration index or time interval index
l	lower bound
o	oil
P	refers to set of production wells
pw	produced water
t	refers to time intervals
tol	tolerance
u	upper bound
w	water

Superscripts

F	best feasible solution
I	least infeasible solution in filter
iw	injected water
m	mesh
o	oil
p	poll
pw	produced water
$*$	refers to an optimized variable

Bibliography

[1] M. A. Abramson, C. Audet, G. Couture, J. E. Dennis Jr., and S. Le Digabel. The NOMAD project, 2010. Software available at http://www.gerad.ca/nomad.

[2] I. Aitokhuehi. *Real-Time Optimization of Smart Wells*. Master's thesis, Department of Petroleum Engineering, Stanford University, 2004.

[3] I. Aitokhuehi and L. J. Durlofsky. Optimizing the performance of smart wells in complex reservoirs using continuously updated geological models. *Journal of Petroleum Science and Engineering*, 48(3-4):254–264, 2005.

[4] L. F. Almeida, Y. J. Tupac, J. G. Lazo Lazo, M. A. Pacheco, and M. M. B. R. Vellasco. Evolutionary optimization of smart-wells control under technical uncertainties. Paper SPE 107872 presented at the SPE Latin American & Caribbean Petroleum Engineering Conference, Buenos Aires, Argentina, 2007.

[5] V. Artus, L. J. Durlofsky, J. E. Onwunalu, and K. Aziz. Optimization of nonconventional wells under uncertainty using statistical proxies. *Computational Geosciences*, 10(4):389–404, 2006.

[6] H. Asheim. Maximization of water sweep efficiency by controlling production and injection rates. Paper SPE 18365 presented at the SPE European Petroleum Conference, London, United Kingdom, 1988.

[7] C. Audet and J. E. Dennis Jr. Analysis of generalized pattern searches. *SIAM Journal on Optimization*, 13(3):889–903, 2002.

[8] C. Audet and J. E. Dennis Jr. A pattern search filter method for nonlinear programming without derivatives. *SIAM Journal on Optimization*, 14(4):980–1010, 2004.

[9] C. Audet and J. E. Dennis Jr. Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on Optimization*, 17(1):188–217, 2006.

[10] C. Audet, G. Savard, and W. Zghal. Multiobjective optimization through a series of single-objective formulations. *SIAM Journal on Optimization*, 19(1):188–210, 2008.

[11] A. A. Awotunde and N. Sibaweihi. Consideration of voidage replacement ratio in well placement optimization. Paper SPE 163354 presented at the SPE Kuwait International Petroleum Conference and Exhibition, Kuwait City, Kuwait, 2012.

[12] K. Aziz and A. Settari. *Petroleum Reservoir Simulation*. Applied Science Publishers, 1979.

[13] O. Badru and C. S. Kabir. Well placement optimization in field development. Paper SPE 84191 presented at the SPE Annual Technical Conference and Exhibition, Denver, Colorado, USA, 2003.

[14] S. Bandyopadhyay, S. Saha, U. Maulik, and K. Deb. A simulated annealing-based multiobjective optimization algorithm: AMOSA. *IEEE Transactions on Evolutionary Computation*, 12(3):269–283, 2008.

[15] W. Bangerth, H. Klie, M. F. Wheeler, P. L. Stoffa, and M. K. Sen. On optimization algorithms for the reservoir oil well placement problem. *Computational Geosciences*, 10(3):303–319, 2006.

[16] B. L. Beckner and X. Song. Field development planning using simulated annealing – optimal economic well scheduling and placement. Paper SPE 30650 presented at the SPE Annual Technical Conference and Exhibition, Dallas, Texas, USA, 1995.

[17] M. C. Bellout, D. Echeverría Ciaurri, L. J. Durlofsky, B. Foss, and J. Kleppe. Joint optimization of oil well placement and controls. *Computational Geosciences*, 16(4):1061–1079, 2012.

[18] P. Bonami, L. T. Biegler, A. R. Conn, G. Cornuéjols, I. E. Grossmann, C. D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization*, 5(2):186–204, 2008.

[19] Z. Bouzarkouna, D. Y. Ding, and A. Auger. Well placement optimization with the covariance matrix adaptation evolution strategy and meta-models. *Computational Geosciences*, 16(1):75–92, 2011.

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

[21] D. A. Cameron and L. J. Durlofsky. Optimization of well placement, CO_2 injection rates, and brine cycling for geological carbon sequestration. *International Journal of Greenhouse Gas Control*, 10:100–112, 2012.

[22] D. A. Cameron and L. J. Durlofsky. Optimization and data assimilation for geological carbon storage. In R. Al-Khoury and J. Bundschuh, editors, *Computational Models for CO_2 Sequestration and Compressed Air Energy Storage*, Studies in Computational Intelligence. Taylor & Francis Group/CRC Press, 2013. In review.

[23] H. Cao. *Development of Techniques for General Purpose Simulators*. PhD thesis, Department of Petroleum Engineering, Stanford University, 2002.

[24] J. A. Carroll III. *Multivariate Production Systems Optimization*. Master's thesis, Department of Petroleum Engineering, Stanford University, 1990.

[25] M. M. Chaudhri, H. A. Phale, N. Liu, and D. S. Oliver. An improved approach for ensemble-based production optimization. Paper SPE 121305 presented at the SPE Western Regional Meeting, San Jose, California, USA, 2009.

[26] M. Christie, D. Eydinov, V. Demyanov, J. Talbot, D. Arnold, and V. Shelkov. Use of multi-objective algorithms in history matching of a real field. Paper SPE 163580 presented at the SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, 2013.

[27] M. Clerc. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In *Proceedings of the 1999 Congress on Evolutionary Computation*, pages 1951–1957, 1999.

[28] M. Clerc. *Particle Swarm Optimization*. ISTE, 2006.

[29] C. A. Coello Coello and M. S. Lechuga. MOPSO: A proposal for multiple objective particle swarm optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation*, volume 2, pages 1051–1056, 2002.

[30] A. R. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to Derivative-Free Optimization*. MPS-SIAM Series on Optimization. MPS-SIAM, 2009.

[31] A. S. Cullick, D. Heath, K. Narayanan, J. April, and J. Kelly. Optimizing multiple-field scheduling and production strategy with reduced risk. Paper SPE 84239 presented at the SPE Annual Technical Conference and Exhibition, Denver, Colorado, USA, 2003.

[32] A. L. Custódio, J. F. A. Madeira, A. I. F. Vaz, and L. N. Vicente. Direct multisearch for multiobjective optimization. *SIAM Journal on Optimization*, 21(3):1109–1140, 2011.

[33] C. Daniel. One-at-a-time plans. *Journal of the American Statistical Association*, 68(342):353–360, 1973.

[34] I. Das and J. E. Dennis Jr. A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural and Multidisciplinary Optimization*, 14(1):63–69, 1997.

[35] I. Das and J. E. Dennis Jr. 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.

[36] K. Deb. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186:311–338, 2000.

[37] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[38] R. C. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *Proceedings of the Sixth International Symposium on Micromachine and Human Science*, pages 39–43, 1995.

[39] D. Echeverría Ciaurri, A. R. Conn, U. T. Mello, and J. E. Onwunalu. Integrating mathematical optimization and decision making in intelligent fields. Paper SPE 149780 presented at the SPE Intelligent Energy Conference and Exhibition, Utrecht, The Netherlands, 2012.

[40] D. Echeverría Ciaurri, O. J. Isebor, and L. J. Durlofsky. Application of derivative-free methodologies for generally constrained oil production optimization problems. *International Journal of Mathematical Modelling and Numerical Optimisation*, 2(2):134–161, 2011.

[41] D. Echeverría Ciaurri, T. Mukerji, and L. J. Durlofsky. Derivative-free optimization for oil field operations. In X. S. Yang and S. Koziel, editors, *Computational Optimization and Applications in Engineering and Industry*, Studies in Computational Intelligence, pages 19–55. Springer, 2011.

[42] T. Ertekin, J. H. Abou-Kassem, and G. R. King. *Basic Applied Reservoir Simulation*. SPE Textbook Series, 2001.

[43] J. L. Fernández Martínez and E. García Gonzalo. The generalized PSO: a new door to PSO evolution. *Journal of Artificial Evolution and Applications*, 2008:1–15, 2008.

[44] J. L. Fernández Martínez and E. García Gonzalo. The PSO family: Deduction, stochastic analysis and comparison. *Swarm Intelligence*, 3(4):245–273, 2009.

[45] J. L. Fernández Martínez and E. García Gonzalo. Stochastic stability analysis of the linear continuous and discrete PSO models. *IEEE Transactions on Evolutionary Computation*, 15(3):405–423, 2011.

[46] J. L. Fernández Martínez, E. García Gonzalo, and J. P. Fernández Alvarez. Theoretical analysis of particle swarm trajectories through a mechanical analogy. *International Journal of Computational Intelligence Research*, 4(2):93–104, 2008.

[47] J. L. Fernández Martínez, E. García Gonzalo, Z. Fernández Muñiz, and T. Mukerji. How to design a powerful family of particle swarm optimizers for inverse modeling. *Transactions of the Institute of Measurement and Control*, 34(6):705–719, 2012.

[48] P. Ferraro and F. Verga. Use of evolutionary algorithms in single and multiobjective optimization techniques for assisted history matching. Paper OMC 2009-079 presented at the Offshore Mediterranean Conference and Exhibition, Ravenna, Italy, 2009.

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

[50] R. Fletcher, S. Leyffer, and P. Toint. A brief history of filter methods. Technical report, Mathematics and Computer Science Division, Argonne National Laboratory, 2006.

[51] F. Forouzanfar, G. Li, and A. C. Reynolds. A two-stage well placement optimization method based on adjoint gradient. Paper SPE 135304 presented at the SPE Annual Technical Conference and Exhibition, Florence, Italy, 2010.

[52] A. M. Geoffrion. Generalized Benders decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260, 1972.

[53] M. G. Gerritsen and L. J. Durlofsky. Modeling fluid flow in oil reservoirs. *Annual Review of Fluid Mechanics*, 37:211–238, 2005.

[54] H. Gross. Response surface approaches for large decision trees: Decision making under uncertainty. In *Proceedings of the 13th European Conference on the Mathematics of Oil Recovery, Biarritz, France*, 2012.

[55] V. Gunnerud and B. Foss. Oil production optimization – A piecewise linear model, solved with two decomposition strategies. *Computers & Chemical Engineering*, 34(11):1803–1812, 2010.

[56] B. Guyaguler and R. N. Horne. Uncertainty assessment of well-placement optimization. *SPE Journal*, 7(1):24–32, 2004.

[57] B. Guyaguler, R. N. Horne, L. Rogers, and J. J. Rosenzweig. Optimization of well placement in a Gulf of Mexico waterflooding project. *SPE Reservoir Evaluation & Engineering*, 5(3):229–236, 2002.

[58] Y. Hajizadeh, M. Christie, and V. Demyanov. Towards multiobjective history matching: Faster convergence and uncertainty quantification. Paper SPE 141111 presented at the SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, 2011.

[59] T. J. Harding, N. J. Radcliffe, and P. R. King. Optimization of production strategies using stochastic search methods. Paper SPE 35518 presented at the European 3-D Reservoir Modelling Conference, Stavanger, Norway, 1996.

[60] R. Hooke and T. A. Jeeves. Direct search solution of numerical and statistical problems. *Journal of the ACM*, 8(2):212–229, 1961.

[61] X. Hu and R. Eberhart. Solving constrained nonlinear optimization problems with particle swarm optimization. In *Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics*, 2002.

[62] T. D. Humphries, R. D. Haynes, and L. A. James. Simultaneous optimization of well placement and control using a hybrid global-local strategy. In *Proceedings of the 13th European Conference on the Mathematics of Oil Recovery, Biarritz, France*, 2012.

[63] O. J. Isebor. *Constrained Production Optimization with an Emphasis on Derivative-free Methods*. Master's thesis, Department of Energy Resources Engineering, Stanford University, 2009.

[64] O. J. Isebor, L. J. Durlofsky, and D. Echeverría Ciaurri. A derivative-free methodology with local and global search for the joint optimization of well location and control. In *Proceedings of the 13th European Conference on the Mathematics of Oil Recovery, Biarritz, France*, 2012.

[65] O. J. Isebor, D. Echeverría Ciaurri, and L. J. Durlofsky. Generalized field development optimization using derivative-free procedures. Paper SPE 163631 presented at the SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, 2013.

[66] J. D. Jansen, D. R. Brouwer, G. Naevdal, and C. P. J. W. van Kruijsdiik. Closed-loop reservoir management. *First Break*, 23:43–48, 2005.

[67] J. D. Jansen, S. D. Douma, D. R. Brouwer, P. M. J. Van den Hof, O. H. Bosgra, and A. W. Heemink. Closed-loop reservoir management. Paper SPE 119098 presented at the SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, 2009.

[68] Y. Jiang. *Techniques for Modeling Complex Reservoirs and Advanced Wells*. PhD thesis, Department of Energy Resources Engineering, Stanford University, 2007.

[69] D. Jones and T. Merhdad. *Practical Goal Programming*. International Series in Operations Research & Management Science. Springer, 2012.

[70] I. Y. Kim and O. de Weck. Adaptive weighted sum method for multiobjective optimization: A new method for Pareto front generation. *Structural and Multidisciplinary Optimization*, 31:105–116, 2006.

[71] T. G. Kolda, R. M. Lewis, and V. Torczon. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Review*, 45(3):385–482, 2003.

[72] V. D. Kosmidis, J. D. Perkins, and E. N. Pistikopoulos. A mixed integer optimization formulation for the well scheduling problem on petroleum fields. *Computers & Chemical Engineering*, 29(7):1523–1541, 2005.

[73] D. Kumar. *Optimization of Well Settings to Maximize Residually Trapped CO_2 in Geologic Carbon Sequestration*. Master's thesis, Department of Energy Resources Engineering, Stanford University, 2007.

[74] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence properties of the Nelder-Mead simplex method in low dimensions. *SIAM Journal on Optimization*, 9(1):112–147, 1998.

[75] A. H. Land and A. G. Doig. An automatic method of solving discrete programming problems. *Econometrica*, 28(3):497–520, 1960.

[76] S. Le Digabel. Algorithm 909: NOMAD: Nonlinear optimization with the MADS algorithm. *ACM Transactions on Mathematical Software*, 37(4):44:1–44:15, 2011.

[77] O. Leeuwenburgh, P. J. P. Egberts, and O. A. Abbink. Ensemble methods for reservoir life-cycle optimization and well placement. Paper SPE 136916 presented at the SPE/DGS Saudi Arabia Section Technical Symposium and Exhibition, Al-Khobar, Saudi Arabia, 2010.

[78] L. Li and B. Jafarpour. A variable-control well placement optimization for improved reservoir development. *Computational Geosciences*, 16(4):871–889, 2012.

[79] L. Li, B. Jafarpour, and M. R. Mohammad-Khaninezhad. A simultaneous perturbation stochastic approximation algorithm for coupled well placement and control optimization under geologic uncertainty. *Computational Geosciences*, 17(1):167–188, 2013.

[80] R. Li, A. C. Reynolds, and D. S. Oliver. History matching of three-phase flow production data. *SPE Journal*, 8(4):328–340, 2003.

[81] M. L. Litvak and P. F. Angert. Field development optimization applied to giant oil fields. Paper SPE 118840 presented at the SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, 2009.

[82] M. L. Litvak, L. A. Hutchins, R. C. Skinner, B. L. Darlow, R. C. Wood, and L. J. Kuest. Prudhoe Bay E-field production optimization system based on integrated reservoir and facility simulation. Paper SPE 77643 presented at the SPE Annual Technical Conference and Exhibition, San Antonio, Texas, USA, 2002.

[83] W. Liu and W. F. Ramirez. Optimal control of three-dimensional steam flooding process. *Journal of Petroleum Science and Engineering*, 11(2):137–154, 1989.

[84] A. L. Marsden, J. A. Feinstein, and C. A. Taylor. A computational framework for derivative-free optimization of cardiovascular geometries. *Computational Methods in Applied Mechanics and Engineering*, 197:1890–1905, 2008.

[85] A. L. Marsden, M. Wang, J. E. Dennis Jr., and P. Moin. Trailing-edge noise reduction using derivative-free optimization and large-eddy simulation. *Journal of Fluid Mechanics*, 572:13–36, 2007.

[86] Mathworks. *Optimization Toolbox^{TM}: User's Guide (R2012b)*. The Mathworks, Inc., 2012.

[87] G. Mehos and W. F. Ramirez. Use of optimal control theory to optimize carbon dioxide miscible flooding enhanced oil recovery. *Journal of Petroleum Science and Engineering*, 2(4):247–260, 1989.

[88] L. Mohamed, M. Christie, and V. Demyanov. History matching and uncertainty quantification: Multiobjective particle swarm optimisation approach. Paper SPE 143067 presented at the SPE EUROPEC/EAGE Annual Conference and Exhibition, Vienna, Austria, 2011.

[89] J. A. Nelder and R. Mead. A simplex for function minimization. *The Computer Journal*, 7(4):308–313, 1965.

[90] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, second edition, 2006.

[91] E. Nwankwor, A. K. Nagar, and D. C. Reid. Hybrid differential evolution and particle swarm optimization for optimal well placement. *Computational Geosciences*, 17(2):249–268, 2013.

[92] J. Onwunalu. *Optimization of Nonconventional Well Placement Using Genetic Algorithms and Statistical Proxy*. Master's thesis, Department of Petroleum Engineering, Stanford University, 2006.

[93] J. E. Onwunalu. *Optimization of Field Development Using Particle Swarm Optimization and New Well Pattern Descriptions*. PhD thesis, Department of Energy Resources Engineering, Stanford University, 2010.

[94] J. E. Onwunalu and L. J. Durlofsky. Application of a particle swarm optimization algorithm for determining optimum well location and type. *Computational Geosciences*, 14(1):183–198, 2010.

[95] J. E. Onwunalu and L. J. Durlofsky. A new well-pattern-optimization procedure for large-scale field development. *SPE Journal*, 16(3):594–607, 2011.

[96] U. Ozdogan and R. N. Horne. Optimization of well placement under time-dependent uncertainty. *SPE Reservoir Evaluation & Engineering*, 9(2):135–145, 2006.

[97] U. Ozdogan, A. Sahni, B. Yeten, B. Guyaguler, and W. H. Chen. Efficient assessment and optimization of a deepwater asset development using fixed pattern approach. Paper SPE 95792 presented at the SPE Annual Technical Conference and Exhibition, Dallas, Texas, USA, 2005.

[98] Y. Pan and R. N. Horne. Improved methods for multivariate optimization of field development scheduling and well placement design. Paper SPE 49055 presented at the SPE Annual Technical Conference and Exhibition, New Orleans, Louisiana, USA, 1998.

[99] K. E. Parsopoulos and M. N. Vrahatis. Particle swarm optimization method for constrained optimization problems. In *Proceedings of the Euro-International Symposium on Computational Intelligence*, 2002.

[100] W. F. Ramirez. *Application of Optimal Control Theory to Enhanced Oil Recovery*. Elsevier, 1987.

[101] S. S. Rao. *Engineering Optimization: Theory and Practice*. John Wiley & Sons, fourth edition, 2009.

[102] N. Remy, A. Boucher, and J. Wu. *Applied Geostatistics with SGeMS: A User's Guide*. Cambridge University Press, 2009.

[103] V. Rigot. *New Well Optimization in Mature Fields*. Master's thesis, Department of Petroleum Engineering, Stanford University, 2003.

[104] P. Sarma. *Efficient Closed-loop Optimal Control of Petroleum Reservoirs Under Uncertainty*. PhD thesis, Department of Petroleum Engineering, Stanford University, 2006.

[105] P. Sarma, K. Aziz, and L. J. Durlofsky. Implementation of adjoint solution for optimal control of smart wells. Paper SPE 92864 presented at the SPE Reservoir Simulation Symposium, Houston, Texas, USA, 2005.

[106] P. Sarma and W. H. Chen. Efficient well placement optimization with gradient-based algorithm and adjoint models. Paper SPE 112257 presented at the SPE Intelligent Energy Conference and Exhibition, Amsterdam, The Netherlands, 2008.

[107] P. Sarma, W. H. Chen, L. J. Durlofsky, and K. Aziz. Production optimization with adjoint models under nonlinear control-state path inequality constraints. *SPE Reservoir Evaluation & Engineering*, 11(2):326–339, 2008.

[108] P. Sarma, L. J. Durlofsky, K. Aziz, and W. H. Chen. Efficient real-time reservoir management using adjoint-based optimal control and model updating. *Computational Geosciences*, 10(1):3–36, 2006.

[109] M. Sayyafzadeh, M. Haghighi, and J. N. Carter. Regularization in history matching using multi-objective genetic algorithm and Bayesian framework. Paper SPE 154544 presented at the SPE EUROPEC/EAGE Annual Conference, Copenhagen, Denmark, 2012.

[110] J. C. Spall. An overview of the simultaneous perturbation method for efficient optimization. *Johns Hopkins Applied Physics Laboratory Technical Digest*, 19(4):482–492, 1998.

[111] R. E. Steuer. *Multiple Criteria Optimization: Theory, Computation and Application*. John Wiley & Sons, 1986.

[112] H. Su and D. S. Oliver. Smart well production optimization using an ensemble-based method. *SPE Journal*, 13(6):884–892, 2010.

[113] A. Suman. *Joint Inversion of Production and Time-Lapse Seismic Data: Application to Norne Field*. PhD thesis, Department of Energy Resources Engineering, Stanford University, 2013.

[114] Y. J. Tupac, L. Faletti, M. A. C. Pacheco, and M. M. B. R. Vellasco. Evolutionary optimization of oil field development. Paper SPE 107552 presented at the SPE Digital Energy Conference and Exhibition, Houston, Texas, USA, 2007.

[115] A. I. F. Vaz and L. N. Vicente. A particle swarm pattern search method for bound constrained global optimization. *Journal of Global Optimization*, 39(2):197–219, 2007.

[116] G. A. Virnovsky. Waterflooding strategy design using optimal control theory. Paper presented at the 6th European IOR Symposium, Stavanger, Norway, 1991.

[117] C. Wang, G. Li, and A. C. Reynolds. Optimal well placement for production optimization. Paper SPE 111154 presented at the SPE Eastern Regional Meeting, Lexington, Kentucky, USA, 2007.

[118] C. Wang, G. Li, and A. C. Reynolds. Production optimization in closed-loop reservoir management. Paper SPE 109805 presented at the SPE Annual Technical Conference and Exhibition, Anaheim, California, USA, 2007.

[119] H. Wang, D. Echeverría Ciaurri, L. J. Durlofsky, and A. Cominelli. Optimal well placement under uncertainty using a retrospective optimization framework. *SPE Journal*, 17(1):112–121, 2012.

[120] P. Wang. *Development and Applications of Production Optimization Techniques for Petroleum Fields*. PhD thesis, Department of Petroleum Engineering, Stanford University, 2003.

[121] P. Wang, M. Litvak, and K. Aziz. Optimization of production operations in petroleum fields. Paper SPE 77658 presented at the SPE Annual Technical Conference and Exhibition, San Antonio, Texas, USA, 2002.

[122] T. Westerlund and F. Pettersson. An extended cutting plane method for solving convex MINLP problems. *Computers & Chemical Engineering*, 19(Suppl. 1):131–136, 1995.

[123] B. Yeten, D. R. Brouwer, L. J. Durlofsky, and K. Aziz. Decision analysis under uncertainty for smart well deployment. *Journal of Petroleum Science and Engineering*, 44(1-2):175–191, 2004.

[124] B. Yeten, L. J. Durlofsky, and K. Aziz. Optimization of smart well control. Paper SPE 79031 presented at the SPE International Thermal Operations and Heavy Oil Symposium and International Horizontal Well Technology Conference, Calgary, Canada, 2002.

[125] B. Yeten, L. J. Durlofsky, and K. Aziz. Optimization of nonconventional well type, location and trajectory. *SPE Journal*, 8(3):200–210, 2003.

[126] P. L. Yu. Cone convexity, cone extreme points, and nondominated solutions in decision problems with multiobjectives. *Journal of Optimization Theory and Applications*, 14(3):319–377, 1974.

[127] M. Zandvliet, M. Handels, G. van Essen, R. Brouwer, and J. D. Jansen. Adjoint-based well-placement optimization under production constraints. *SPE Journal*, 13(4):392–399, 2008.

[128] K. Zhang, G. Li, A. C. Reynolds, J. Yao, and L. Zhang. Optimal well placement using an adjoint gradient. *Journal of Petroleum Science and Engineering*, 73(3-4):220–226, 2010.

[129] Y. Zhou. *Parallel General-Purpose Reservoir Simulation with Coupled Reservoir Models and Multisegment Wells*. PhD thesis, Department of Energy Resources Engineering, Stanford University, 2012.