AES/PE/11-17 **Robust Ensemble Based Multi-Objective Production Optimization: Application to Smart Wells**

July 28, 2011 **Rahul-Mark Fonseca**

Title	:	Robust Ensemble Based Multi-Objective Production Optimization: Application to Smart Wells

Author(s)	:	Rahul-Mark Fonseca

Date	:	July 28, 2011
Professor(s)	:	J.D. Jansen
Supervisor(s)	:	O. Leeuwenburgh (TNO)
TA Report number	:	AES/PE/11-17

Postal Address	:	Section for Petroleum Engineering Department of Applied Earth Sciences Delft University of Technology P.O. Box 5028 The Netherlands
Telephone	:	(31) 15 2781328 (secretary)
Telefax	:	(31) 15 2781189

ACKNOWLEDGEMENTS

I would like to express my deepest and most sincere gratitude and thanks to my parents and sister for their unconditional love and support, most of all for always encouraging me to do my Masters at Delft University. They have always been my greatest sources of inspiration without whose constant help and support I would have never ever been able to come here. Their everlasting patience and belief in my abilities has encouraged me always. I truly appreciate their efforts and contributions since I've been born.

I would also like to thank Dr. Olwijn Leeuwenburgh for his superb guidance and advice. I thank him for his kindness and tolerance, never once did he not make the time to answer my questions or have discussions about the thesis in spite of his busy schedule. I am truly fortunate to have had Olwijn supervise my thesis work and for being so patient with me and guiding me for the past 7 months.

I would also like to thank Professor Jan-Dirk Jansen for his supervision and giving me the opportunity to do my thesis at TNO. My thanks to the entire Oil and Gas division at TNO for making my time there very comfortable and knowledgeable and giving me the opportunity to write my thesis. I would like to thank Dr. Paul Egberts for critically reviewing and guiding me with the report. They left no stone unturned to make me feel a part of the group, which is greatly appreciated.

To all my professors at TU Delft for all their efforts and guidance during the past 2 years.

To all my fellow classmates at TU Delft for the wonderful 2 years we have spent together, the fun trips and friendships I will always cherish. I would like to thank all my friends for the awesome 2 years in Delft.

I would like to thank everyone for their support and guidance during my time in Netherlands!

ABSTRACT

Recent improvements in dynamic reservoir modeling have led to an increase in the application of model-based optimization of hydrocarbon bearing reservoirs. Numerous studies and articles have indicated the possibility of improving reservoir management using these dynamic models, coupled with methods to reduce uncertainties in the static models, to optimize reservoir performance. These studies have focused on maximizing the life-cycle performance of the project. Thus life cycle optimization is essentially a single-objective optimization problem. In reality, short-term targets usually drive operational decisions. The impact of short-term targets should be included in the optimization to achieve a more realistic solution. The process of optimizing these short-term targets constrained to life cycle targets is a form of multi-objective optimization. Several methods have been suggested to achieve multi-objective reservoir flooding optimization (Van Essen et al. 2011). These methods have been implemented with the adjoint formulation. This thesis proposes the use of an ensemble-based optimization technique (EnOpt) for multi-objective optimization. The optimization of smart wells or production schedules (inflow control valve (ICV) settings) is the objective of this work. We also propose variations to the existing multi-objective algorithms suggested by Van Essen et al. (2011). We propose the use of the BFGS algorithm to improve the computational efficiency. Undiscounted Net Present Value (NPV) and highly discounted NPV are the long-term and short-term objective functions used in this thesis. We also propose an extension of the optimization functionality to better cope with model uncertainties. This robust ensemble-based multi-objective production optimization framework has been applied and tested on a synthetic reservoir model. In our test cases, the ensemble-based multi-objective optimization methods achieved a 14.2% increase in the secondary objective at the cost of only a minor decrease between 0.2-0.5% in the primary objective.

TABLE OF CONTENTS

INTRODUCTION

1.1 Objective

The objective of this thesis is to extend the framework for multi-objective production optimization as proposed by (Van Essen *et al.* 2011), by alleviating the need for an adjoint formulation. We aim to use the ensemble-based optimization algorithm proposed by (Chen *et al*. 2008) to achieve this objective.

1.2 Literature Review

Optimization is the process of deriving the best possible value of a given objective function by manipulating a given set of control variables. Optimization when applied to the upstream petroleum industry is a means to maximize the recovery of hydrocarbons present in the structure of the earth. Optimization of hydrocarbon recovery has traditionally been a reactive process. Closed-loop reservoir management and model-based optimization aim to convert this practice of optimization into a proactive process. There are numerous methods that are applied to achieve model-based optimization of ultimate recovery. Most of them used gradient-based techniques, where the gradient information is obtained with the aid of an adjoint formulation see e.g. Brouwer and Jansen (2004); Sarma *et al.* (2005, 2006, 2008); Jansen (2011). The adjoint approach is computationally very efficient but has the disadvantage that it requires access to the simulation code to be implemented. (Chen *et al.* 2008) introduced an ensemble-based optimization method (EnOpt), which is computationally less attractive but does not require simulator access and has proven to achieve good results. We use this optimization technique for our proposed work. Life cycle optimization studies have shown favorable results. However; most studies have not considered the impact of short-term objectives on the life cycle optimization. Operational decisions are generally based on short-term objectives of a project in terms of operational criteria, production contracts etc. Jansen *et al.* (2009) observed that significantly different optimized water flooding strategies result in nearly equal values of the objective function, defined as net present value (NPV). They concluded that the life cycle optimization problem is ill-posed and contains many more control variables than necessary. Thus, there exist multiple solutions to the optimization problem; different initial starting points may lead to different solutions in an optimal subset of the decision variable space. This observation formed the basis of multi-objective optimization. Van Essen *et al.* (2011) suggested a hierarchical optimization algorithm to include secondary objectives into the life cycle optimization. They observed a significant increase in short-term objectives with minimal change to the primary objective function.

Recent improvements in technology have led to an increase in the use of chokes or inflow control valves to regulate flow rates and maintain pressure in the reservoir. Smart wells, i.e. wells with Inflow Control Valves (ICV's), are important tools to increase oil recovery and delay water production. Changes in ICV settings can be used to adjust the inflow from individual zones, reduce water or gas production, and increase the profitability of a reservoir. ICV's are used not only to delay water breakthrough but also to achieve a uniform water front thus sweeping the reservoir more efficiently; see e.g. Brouwer and Jansen (2004). Su & Oliver (2010) demonstrate the application of smart well production optimization using the EnOpt technique.

Optimization that does not take into account geological uncertainty assumes that the model used for optimization is the true geological model. However, this is rarely possible. There is always uncertainty

involved in geological modeling. Thus optimization based on a single model may not be very accurate for a field application. Robust optimization accounts for uncertainties in geological models; see van Essen et al. (2009). An ensemble of possible geological models is evaluated for which the resulting optimization strategy is an expected solution.

1.3 Problem Definition

The adjoint formulation used to evaluate the gradient for optimization requires access to the simulator source code, requires extensive coding, and is not easily available. Thus, although the adjoint formulation finds the most accurate gradient it is not easy to implement.

1.4 Hypothesis

“The application of multi-objective ensemble based optimization to smart wells produces results of practical value in a computationally efficient manner.”

1.5 Thesis Outline

A description of the mathematical formulations and proposed algorithms is given in section 2. An example of the implementation of the algorithm on a synthetic reservoir model is described and illustrated in section 3. Conclusions are given in section 4. The appendix consists of implementation of the multi-objective optimization methods on the Rosenbrock test function and an overview of gradient search techniques.

OPTIMIZATION FORMULATION

Life cycle optimization has been successfully applied to various hydrocarbon recovery mechanisms especially secondary and tertiary recovery mechanisms. This requires at least one decision variable as well as a model that provides relatively accurate long-term predictions. The most widely implemented secondary recovery mechanism in the petroleum industry is water flooding. We also choose water flooding as our recovery mechanism for the following reasons.

- There are many decision variables involved in a water flooding strategy
- The process is well understood and can be modeled efficiently over long time intervals.

In order to improve economic life cycle performance dynamic optimization has to be performed over the producing life of the reservoir due to the transient nature of the saturation distribution. We consider a control problem wherein choke settings are manipulated to increase oil recovery efficiency. The objective function *J* is the NPV for the life of the field defined as

$$J=\sum_{k=1}^{K}\left(\frac{\left\{\left[(Q_{O_k})\cdot r_O-(Q_{Wp_k})\cdot r_{Wp}\right]-\left[(Q_{Wi_k})\cdot r_{Wi}\right]\right\}\cdot\Delta t_k}{(1+b)^{t_k/\tau_t}}\right), \tag{1}$$

where Q_{O_k} is the oil production rate in bbl/day, Q_{Wp_k} is the water production rate in bbl/day, Q_{Wi_k} is the water injection rate in bbl/day, r_O is the price of oil produced in \$/bbl, r_{Wp} is the price of water produced in \$/bbl, r_{Wi} is the price of water injected in \$/bbl, Δt_k is the difference between consecutive time steps in days, b is the discount factor expressed as a fraction per year, t_k is the cumulative time in days corresponding to time step *k*, and τ_t is the reference time for discounting (365 days). A generic mathematical formulation of the constrained life cycle optimization problem is

$$\begin{aligned}&\max_{\mathbf{u}_{1:K}} J(\mathbf{u}_{1:K}),\\ &s.t.\ \ \mathbf{f}_{k+1}(\mathbf{u}_{k+1},\mathbf{x}_k,\mathbf{x}_{k+1})=\mathbf{0},\ \ k=0,.....,K-1,\\ &\mathbf{c}_{k+1}(\mathbf{u}_{k+1},\mathbf{x}_{k+1})\le\mathbf{0},\ \ \ k=0,.....,K-1,\end{aligned} \tag{2}$$

where **u** is the control vector (input vector), **x** is the state vector (grid block pressures, saturations), **f** is a vector valued function that represents the system equations, $\mathbf{x_0}$ is the state vector representing the initial state of the reservoir, the subscript *k* indicates discrete times and *K* is the total number of time steps. The vector inequality constraints **c** relate to the capacity limitation of the wells. The optimization problem is to find the optimal ICV settings $\mathbf{u}_k$ over all time steps 1,...,*K*, that maximize the objective function based on the dynamic model.

1.6 Ensemble Optimization (EnOpt)

This technique introduced by Chen *et al.* (2008) is a stochastic gradient-based optimization method, which utilizes an ensemble of control vectors to estimate a gradient. This method approximates the gradient based on the sensitivity of the ensemble averaged over the objective function *J* with respect to the control vector **u**.

Distinct characteristics of the EnOpt method are (Chen *et al.,* 2008):

- The search direction used in production optimization is obtained from the ensemble
- It aims at maximizing the expectation of the objective function rather than maximizing the objective function based on a single model.
- It is very flexible because it is independent of a simulator requiring minimal code development

- It can work with large control vectors (many elements)

Approximating the gradient from the sensitivity of the ensemble enables the use of any type of control variables without modification to the existing algorithm (Chen *et al.,* 2008). This is very applicable to smart wells and ICV settings as shown in (Su and Oliver, 2010)

Ensemble based sensitivity approximates a global search direction and has less chance of getting stuck at local maxima/minima than accurate local gradients (Chen *et al.,* 2008).

1.6.1 Mathematics of the EnOpt

The following is a general description of the mathematics governing this optimization algorithm taken from Leeuwenburgh *et al.* (2010). The dynamic well control variables form a control vector

$$\mathbf{u} = (\mathbf{u}_1^T, ..., \mathbf{u}_K^T)^T = (u_1, ..., u_N)^T,$$

where *N* is the number of controls which can be rather large. Thus **u** is a 'super vector' with a number of elements *N* that may be as large as the number of time steps *K* times the number of control variables (ICVs). The control vector **u** applied to a reservoir model with static reservoir property vector **m** yields after simulation a value $J = J(\mathbf{u}, \mathbf{m})$ depending on **u** and **m.** We will adopt and briefly discuss the ensemble-based optimization (EnOpt) method as proposed by Chen et al. (2008). It is assumed that the geological uncertainty is captured by a set $\mathbf{m} = [(\mathbf{m}^1)^T, ..., (\mathbf{m}^M)^T]^T$ of *M* equiprobable models. The optimization goal is to maximize the expected value of the objective function, i.e. maximize $\bar{J}$ given by

$$\bar{J} = \bar{J}(\mathbf{u}, \mathbf{m}) = \frac{1}{M} \sum_{j=1}^{M} J_j(\mathbf{u}, \mathbf{m}^j).$$

Note that when the models are identical the problem reduces to a deterministic optimization problem. A stochastic gradient of $\bar{J}$ with respect to the control vector **u** is obtained by performing *M* simulations with models $\mathbf{m}^j$, each with a perturbed control vector $\mathbf{u}^j$ where $(j = 1, ..., M)$. The perturbations are Gaussian with zero mean and some appropriate standard deviation σ. We can collect the means of the control vectors and the objective functions in ensemble matrices

$$\mathbf{U} = \begin{pmatrix} u_1^1 - \bar{u}_1 & \cdots & u_N^1 - \bar{u}_N \\ \vdots & \ddots & \vdots \\ u_1^M - \bar{u}_1 & \cdots & u_N^M - \bar{u}_N \end{pmatrix}, \text{ and } \mathbf{j}_j = \begin{pmatrix} J_j^1 - \bar{J}_j \\ \vdots \\ J_j^M - \bar{J}_j \end{pmatrix},$$

where $\bar{u}_l = \frac{1}{M} \sum_{j=1}^{M} u_l^j$ for $l = 1, ..., N$, $J_j^i = J_j^i(\mathbf{u}^i, \mathbf{m}^j)$, $\bar{J}_j = \frac{1}{M} \sum_{i=1}^{M} J_j^i$, for $i, j = 1, ..., M$. Applying linear regression through the *M* points $(u^i - \bar{u}, J_j^i - \bar{J}_j)$ results in a regression coefficient vector $\boldsymbol{\beta}_j = (\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T\mathbf{j}_j$ that approximates ∇J_j. Averaging the coefficients $\boldsymbol{\beta}_j$ leads to a stochastic gradient **g** which approximates the gradient $\nabla \bar{J}$. The resulting stochastic gradient **g** is $\mathbf{g} = \mathbf{C}_u^{-1} \cdot \mathbf{c}_{u,\bar{J}(u)}$,

with

$$\mathbf{c}_{u,\bar{J}(u)} \approx \frac{1}{M} \sum_{j=1}^{M} \frac{1}{M-1} (\mathbf{U}^T \mathbf{j}_j), \tag{3}$$

and $\mathbf{C}_u \approx \frac{1}{M-1} (\mathbf{U}^T\mathbf{U})$. When considering a deterministic case the above calculation reduces to

$$\mathbf{g} = \mathbf{C}_u^{-1} \frac{1}{M-1} \sum_{i=1}^{M} (\mathbf{u}^i - \mathbf{1}\bar{u})(J_i^i - \bar{J}), \tag{4}$$

where

$$\overline{J} = \frac{1}{M}\sum_{i=1}^{M} J_i^i \ ,$$

and where **1** is an M x 1 vector of ones. Note that equation (3) and hence the stochastic gradient **g** involves M^2 function evaluations J_j^i, for $i, j = 1,.....,M$, while equation (4) only involves M function evaluations J_i^i, for $i = 1,.....,M$. The straightforward use of the stochastic gradient in the steepest descent formulation is not optimal, hence a regularized gradient is used by pre-multiplication of **g** with a matrix **R**. Chen *et al.* (2008) suggest that the choice of $\mathbf{R} = \mathbf{C_u} \cdot \mathbf{C_u}$ provides both a pre-conditioning for the steepest ascent method as well as a smoothing to correct for the noise resulting from a smaller ensemble size. Thus using the steepest ascent formula a new update for the controls is calculated using the gradient information approximated using the EnOpt method. The steepest ascent formula is

$$\mathbf{u}_{l+1} = \frac{1}{\alpha_l}\mathbf{C}_u\mathbf{g}_l + \mathbf{u}_l \ . \tag{5}$$

The steepest decent formula for the updated set of controls is then

$$\mathbf{u}_{l+1} = \frac{1}{\alpha_l}\mathbf{C}_u\mathbf{c}_{u,g(u)} + \mathbf{u}_l \ ,$$

$$\mathbf{u}_{l+1} = \frac{1}{\alpha_l}\mathbf{C}_u\mathbf{C}_u\mathbf{g}_l + \mathbf{u}_l \ , \tag{6}$$

$$\mathbf{u}_{l+1} = \frac{1}{\alpha_l}\mathbf{R}_u\mathbf{g}_l + \mathbf{u}_l \ . \qquad \text{(Pre-conditioned version)}$$

The general optimization loop used in this thesis is shown in figure (1). A short description of the Armijo condition and gradient search techniques is given in the appendix. A detailed description of these methods and concepts can be referred to in Nocedal and Wright (2006).

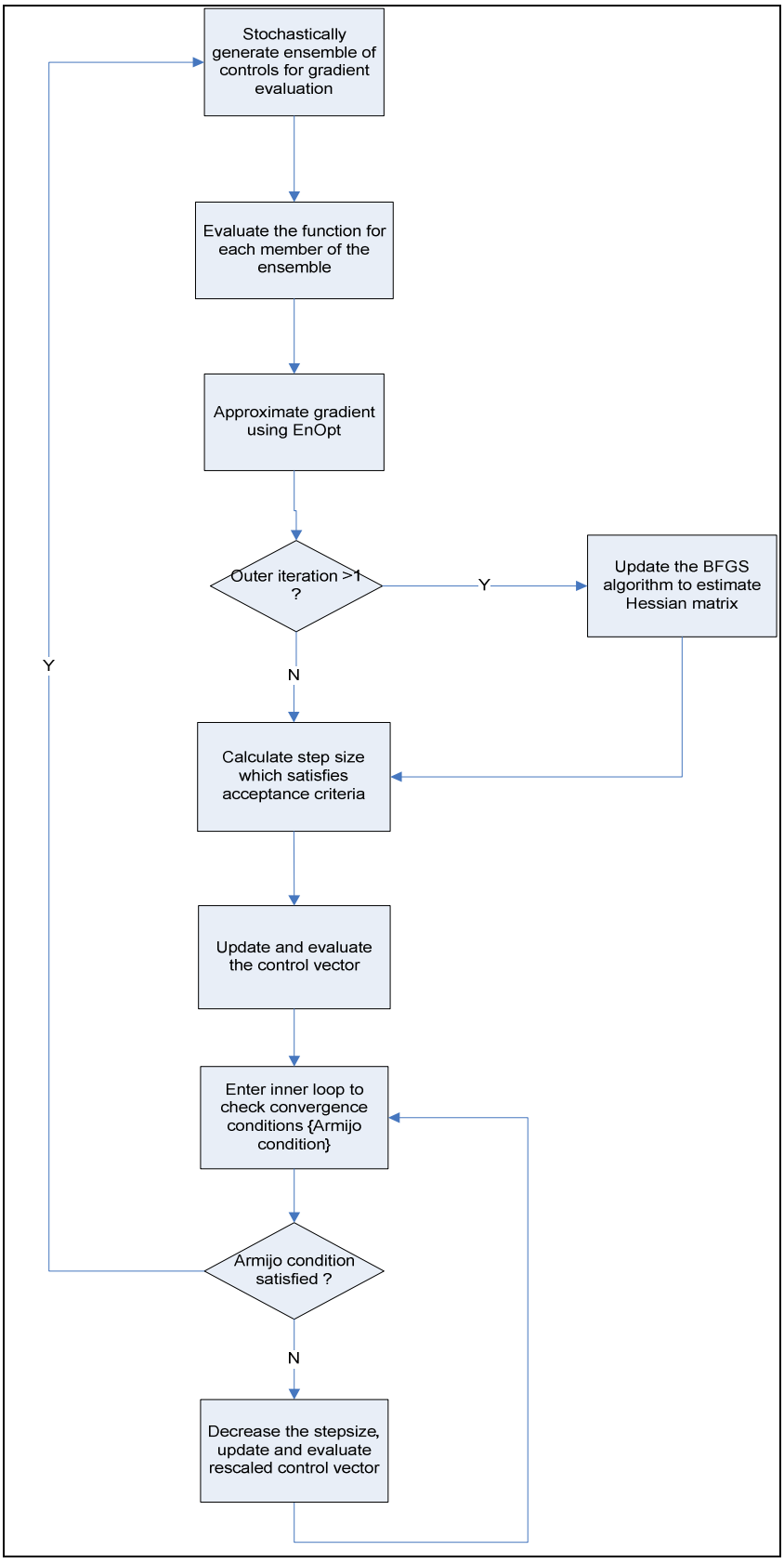

Figure 1: Flow chart illustrating the general optimization loop used in this thesis

1.6.2 Robust Optimization

Van Essen *et al.* (2009) have shown a significant improvement in optimal control strategies when geological uncertainty is accounted for in the optimization. Geological models have inherent uncertainties. Incorporating these uncertainties into the optimization framework reduces the uncertainty involved with the optimization. This is called robust optimization. Chen et al. (2008) have also used the concept of robust optimization with the ensemble optimization method. As shown in the flow chart below, the use of EnOpt for robust optimization as introduced by Van Essen *et al*. (2009) will require the generation of two ensembles. An ensemble for gradient evaluation and an ensemble of geological models. Assume the two ensembles consist of *M* members each. Thus to evaluate a gradient M^2 samples must be evaluated. The mathematical proof below proves the need to evaluate only *M* samples, thus EnOpt becomes computationally attractive when applied to a robust setting. The stochastic gradient $\mathbf{g} = \mathbf{C}_u^{-1}\mathbf{c}_{u,\bar{J}(u)}$, from equation (4) can be approximated by $\tilde{\mathbf{g}}$,

$$\tilde{\mathbf{g}} = \mathbf{C}_u^{-1}\frac{1}{M}\sum_{i=1}^{M}(\mathbf{u}^i - \mathbf{1}\bar{u})(J_i^i - \bar{J}) .$$

Derivation: *(Courtesy Paul Egberts)*

Chen (2008) has shown the ensemble optimization method to be computationally attractive when applied to an optimization that takes into account geological uncertainty. The following is a proof of the computational attractiveness of the EnOpt. We have shown that for *M* equiprobable models the gradient **g** is approximated by

$$\mathbf{g} \approx \mathbf{C}_u^{-1}\mathbf{c}_{u,\bar{J}(u)} ,$$

$$\text{with} \quad \mathbf{c}_{u,\bar{J}(u)} \approx \frac{1}{M}\sum_{j=1}^{M}\frac{1}{M-1}\mathbf{U}^T\mathbf{j}_j . \tag{7}$$

Since we generate an ensemble of control vectors $\mathbf{u}^j$ using a stochastic method by perturbing around the control vector **u** with zero mean we can express equation (7) by

$$\mathbf{c}_{u,\bar{J}(u)} \approx \frac{1}{M}\sum_{j=1}^{M}\frac{1}{M-1}\sum_{i=1}^{M}(\mathbf{u}^i - \mathbf{1}\bar{u})(J_j^i - \bar{J}_j) . \tag{8}$$

As the mean of $(\mathbf{u}^i - \mathbf{1}\cdot\bar{u})$, is approximately zero we can express $\mathbf{c}_u$ as

$$\mathbf{c}_{u,\bar{J}(u)} = \frac{1}{M}\sum_{j=1}^{M}\frac{1}{M-1}\sum_{i=1}^{M}(\mathbf{u}^i - \mathbf{1}\bar{u})J_j^i . \tag{9}$$

Expanding equation (9) we get

$$\mathbf{c}_{u,\bar{J}(u)} \approx \frac{1}{M}\sum_{j=1}^{M}\frac{1}{M-1}\sum_{i=1}^{M}\mathbf{u}^i J_j^i - \mathbf{1}\bar{u}\frac{1}{M}\sum_{j=1}^{M}\frac{1}{M-1}\sum_{i=1}^{M}J_j^i . \tag{10}$$

Thus we need M^2 samples to evaluate the covariance as shown in equation (10) and the covariance is approximated by the *mean* of the functions $\mathbf{u}^T J(\mathbf{u},\mathbf{m})$, and $\mathbf{1}\bar{u}J(\mathbf{u},\mathbf{m})$, . Since the ensembles for the control vector **u** and reservoir property **m** are independent, the mean of these two functions can also be approximated as

$$E(\mathbf{u}^i J_i^i) \approx \frac{1}{M}\sum_{i=1}^{M}\mathbf{u}^i J_i^i \ , \& \quad E(J_i^i) \approx \frac{1}{M}\sum_{i=1}^{M}J_i^i , \tag{11}$$

Substituting equation (11) in equation (10) we get

$$\mathbf{c}_{u,\bar{J}(u)} \approx \frac{1}{M}\sum_{i=1}^{M}\mathbf{u}^i J_j^i - \mathbf{1}\bar{u}\frac{1}{M}\sum_{i=1}^{M}J_j^i ,$$

Thus the covariance is now approximated using *M* samples instead of M^2 samples

$$\mathbf{C}_{u,\bar{J}(u)} \approx \frac{1}{M}\sum_{i=1}^{M}(\mathbf{u}^i - \mathbf{1}\bar{u})J_i^i \approx \frac{1}{M}\sum_{i=1}^{M}(\mathbf{u}^i - \mathbf{1}\bar{u})(J_i^i - \bar{J}) \,. \tag{12}$$

Thus for robust optimization the EnOpt needs to evaluate only M function evaluations to estimate the gradient instead of M^2 evaluations.

1.7 Multi-Objective Optimization Methods.

The process of optimizing systematically and simultaneously a collection of objective functions is called multi-objective optimization. The general definition of a multi-objective problem is

$$\max_{\mathbf{u}} J(\mathbf{u}) = [J_1(\mathbf{u}), J_2(\mathbf{u}), ..., J_k(\mathbf{u})]^T$$
$$\text{subject to } \mathbf{c}_j(\mathbf{u}) \leq \mathbf{0}, \ j = 1,2,.....,m$$

where k is the number of objective functions, m is the number of inequality constraints and $J(\mathbf{u})$ is a collection of the various objective functions. There are various methods of multi-objective optimization such as weighted sum method, Pareto optimality, goal programming etc. In this thesis we use a hierarchical optimization structure for multi-objective optimization.

1.7.1 Hierarchical Optimization

Van Essen *et al.* (2011) introduced a method to achieve multi-objective production optimization. The method proposes a hierarchical optimization structure, which prioritizes the multiple objective functions. A general formulation of this hierarchical optimization is described below. The optimization of the secondary objective function J_2 is constrained to a very small allowable change in the primary objective function. Thus the primary objective function J_1 will remain close to its optimal value.

$$\begin{aligned} &\max_{\mathbf{u}_{1:K}} J_2(\mathbf{u}_{1:K}) \,, \\ &s.t. \ \mathbf{f}_{k+1}(\mathbf{u}_k, \mathbf{x}_k, \mathbf{x}_{k+1}) = \mathbf{0}, \ k = 0,....,K-1, \mathbf{x}_0 = \overline{\mathbf{x}}_0 \,, \\ &\mathbf{c}_{k+1}(\mathbf{u}_{k+1}, \mathbf{x}_{k+1}) \leq \mathbf{0}, \quad k = 0,.....,K-1 \,, \\ &J_1^* - J_1(\mathbf{u}_{1:K}) \leq \varepsilon. \end{aligned} \tag{13}$$

where $\varepsilon > 0$ is an arbitrary small value compared to J_1^*. Solving the above equations requires the knowledge of J_1^* which is the optimized value of J_1 obtained from the life cycle optimization. Thus the hierarchical optimization constrains the optimization of the short-term objective with respect to the life cycle optimization. The ordering of the different objective functions is not unique thus secondary objectives can be implemented as primary objectives and vice versa. Some concepts essential to understanding the hierarchical optimization structure are briefly discussed below.

Null Space

In linear algebra the null space of a matrix **A** is the set of all vectors of **A**, which satisfy the condition $\mathbf{Ax} = \mathbf{0}$. The null space of an $m \times n$ matrix is a subspace of a collection of vectors $\mathbf{R}^n$. The set $Null(\mathbf{A})$ has the following three properties:

- $Null(\mathbf{A})$ always contains the zero vector.
- If $\mathbf{x} \in Null(\mathbf{A})$ and $\mathbf{y} \in Null(\mathbf{A})$, then $\mathbf{x} + \mathbf{y} \in Null(\mathbf{A})$
- If $\mathbf{x} \in Null(\mathbf{A})$ and c is a scalar, then $c \cdot \mathbf{x} \in Null(\mathbf{A})$

Degrees of freedom

When dealing with an optimization problem, there may exist multiple sets of control variables for which we achieve similar results. This set of different optimal control variables is an indication of the presence of redundant degrees of freedom (DOF) in the system. This was demonstrated by Jansen *et al.* (2009) who showed that different solutions exist for the optimal control problem of maximizing the economic objective function over the producing life of the reservoir. This existence of multiple solutions suggests, when the optimality of a primary objective function is reached not all DOFs of the control variable space

are fixed. This implies that there may exist redundant degrees of freedom in the optimization problem. This conclusion by Jansen *et al.* (2009) was the basis for multi-objective optimization.

How do we find these DOFs? Van Essen *et al* (2011) approximate a Hessian matrix to find these DOFs, which is an integral part of the hierarchical optimization. Below we argue as to why we need to estimate this matrix

Why do we need the Hessian?

A Taylor expansion is a representation of a function as a sum of terms calculated from the values of the function's derivatives at a point. Consider an objective function J and let $\mathbf{u}^*$ be a control vector. If $\Delta\mathbf{u}$ is a vector (with same length as $\mathbf{u}^*$) of small perturbations then a Taylor expansion around the vector **u*** is given by

$$J(\mathbf{u}^* + \Delta\mathbf{u}) \approx J(\mathbf{u}^*) + \nabla J(\mathbf{u}^*)^T \cdot \Delta\mathbf{u} + \frac{1}{2}(\Delta\mathbf{u})^T \cdot \nabla^2 J(\mathbf{u}^*)\Delta\mathbf{u} + O(\|\Delta\mathbf{u}\|^3) . \quad (14)$$

If $\mathbf{u}^*$ is a (local) optimum of J and $\mathbf{u}^*$ is in the interior (i.e. not on the boundary) of the feasible domain for a constrained optimization then we can conclude that

$$\nabla J(\mathbf{u}^*) = \mathbf{0} \quad (15)$$

Substituting equation (15) into equation (14) we get

$$J(\mathbf{u}^* + \Delta\mathbf{u}) \approx J(\mathbf{u}^*) + \frac{1}{2}(\Delta\mathbf{u})^T \cdot \nabla^2 J(\mathbf{u}^*)\Delta\mathbf{u} + O(\|\Delta\mathbf{u}\|^3) . \quad (16)$$

If we choose $\Delta\mathbf{u} \in Null[\nabla^2 J(\mathbf{u}^*)]$ then $\nabla^2 J(\mathbf{u}^*)\Delta\mathbf{u} = \mathbf{0}$. Thus equation (16) reduces to

$$J(\mathbf{u}^* + \Delta\mathbf{u}) \approx J(\mathbf{u}^*) + O(\|\Delta u\|^3) . \quad (17)$$

Thus equation (17) implies that for a small perturbation $\Delta\mathbf{u} \in Null[\nabla^2 J(\mathbf{u}^*)]$, the adjusted control vector $(\mathbf{u}^* + \Delta\mathbf{u})$ will have an objective function value very close to the objective function value $J(\mathbf{u}^*)$ which is an optimal value. Henceforth we denote $\mathbf{H} = \nabla^2 J(\mathbf{u}^*)$ where **H** is the *Hessian matrix*, a matrix of second order derivatives of the objective function. This means that we can make any update to the control vector that is in the null space of the primary objective function to improve the secondary objective function. Thus proving the need to find the Hessian matrix at the optimum of the primary objective function and the set of vectors that span its null space.

Why do we need an SVD ?

We have the Hessian matrix; the next question that arises is how we find the null space from this matrix. A powerful matrix decomposition method for any matrix irrespective of its dimension or rank is the singular value decomposition (SVD). A SVD represents every $m \times n$ matrix **A** as

$$\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^\mathsf{T} ,$$

where the columns of the $m \times m$ matrix **U** are known as the left singular vectors, columns of $n \times n$ matrix **V** are known as right singular vectors and $\mathbf{\Lambda}$ is an $m \times n$ matrix consisting of the singular values $\sigma_i, i = 1, \ldots\ldots, r$ of **A** which are the square roots of the non-zero eigenvalues of both $\mathbf{A}\mathbf{A}^\mathsf{T}$ and $\mathbf{A}^\mathsf{T}\mathbf{A}$. Singular value decomposition has many applications, [refer to Strang (2006)]. One important application of singular value decomposition is to obtain the orthonormal basis of the columns of the right singular values **V** corresponding to the null space of **A**.

Orthonormal Basis

Any set of vectors completely spanning a space is a basis of that space. When *n* canonical unit vectors of length *n* form the basis of a *n*-dimensional vector space and are mutually orthogonal. This basis is called an orthonormal basis.

A matrix consisting of second order derivatives of objective function *J* with respect to control vector **u** is a Hessian matrix. There exist redundant DOFs if the Hessian does not have full rank. The orthonormal basis for the undetermined directions of **H** is obtained through singular value decomposition. The orthonormal basis **B** consists of the columns of **V** related to the singular values of zero

$$\mathbf{B} \triangleq (\mathbf{V}_i \,|\, \sigma_i = 0, \quad i = 1, ..., N_u).$$

The criteria above may not always be satisfied. Thus we have defined a cutoff criterion to define the orthonormal basis, which has been described in section 3.

Approximate Hessian

Since no reservoir simulator is capable of calculating the second order derivates, Van Essen *et al.* (2011) proposed the use of a finite difference scheme to approximate the second order derivatives of the function. We propose the use of the Broyden-Flecther-Goldfarb-Shanno (BFGS) algorithm [refer Nocedal and Wright 2006], which approximates and updates the Hessian during the optimization of the primary objective function. This could lead to a significant reduction in overall computational time during optimization of the secondary objective function. We approximate the matrix of second order derivatives using the BFGS algorithm, which is as follows.

1.7.2 BFGS Algorithm

From an initial guess $\mathbf{u}_0$ and an approximate Hessian matrix, $\mathbf{H}_0$ the following steps are repeated until **u** converges to the solution.

- Obtain a direction $\mathbf{p}_k$ by solving: $\mathbf{H}_k\mathbf{p}_k = -\nabla J(\mathbf{u}_k)$.
- Perform a line search/trust region to find an acceptable step size α_k in the direction found in the first step, then update $\mathbf{u}_{k+1} = \mathbf{u}_k + \alpha_k\mathbf{p}_k$.
- Set $\mathbf{s}_k = \alpha_k\mathbf{p}_k$.
- We also define $\mathbf{y}_k = \nabla J(\mathbf{u}_{k+1}) - \nabla J(\mathbf{u}_k)$.
- The updated Hessian is given by $\mathbf{H}_{k+1} = \mathbf{H}_k + \frac{(\mathbf{y}_k\mathbf{y}_k^T)}{\mathbf{y}_k^T\mathbf{s}_k} - \frac{\mathbf{H}_k^T\mathbf{s}_k^T\mathbf{s}_k\mathbf{H}_k}{\mathbf{s}_k^T\mathbf{H}_k\mathbf{s}_k}$.

$J(\mathbf{u})$ denotes the objective function to be minimized. Practically, $\mathbf{H_0}$ can be initialized with $\mathbf{H_0} = \mathbf{I}$, so that the first step will be equivalent to a gradient descent, but further steps are more and more refined by $\mathbf{H_k}$, the approximation to the Hessian. [Refer to Nocedal & Wright (2006)]

Hierarchical Optimization Algorithm

The proposed algorithm is a modification of the algorithm proposed by (Van Essen *et al.,* 2011). The modification is the implementation of the BFGS algorithm to approximate the Hessian matrix. The algorithm has been implemented within the EnOpt framework in MATLAB. The following iteration procedure is proposed to solve the optimization problem using the Hessian approximated at the optimal control vector of the primary optimization.

1. Find a single optimal strategy $\mathbf{u}^*$ that maximizes the primary objective function J_1 and use $\mathbf{u}^* = \mathbf{u}$ with n=0 as a starting point for the secondary optimization problem with n as the iteration index.

2. Use the approximated Hessian **H** at $\mathbf{u}^*$ and performing a singular value decomposition to obtain the orthonormal basis **B** for the null-space of **H**. The projection matrix **P** is calculated by
$$\mathbf{P} = \mathbf{B}\mathbf{B}^T .$$
3. Find the improving direction (gradient) **s** for the secondary objective function J_2
4. Project this improving direction **s** onto the orthonormal basis **B** to obtain the projected direction **d**, such that **d** is an improving direction for J_2 and does not affect J_1. Thus **d** is calculated by
$$\mathbf{d} = \mathbf{P} \cdot \mathbf{s} .$$
5. Update the control vector $\mathbf{u_n}$ using the projected direction **d** in the steepest-ascent method.
$$\mathbf{u}_{n+1} = \mathbf{u}_n + \tau \cdot \mathbf{d} .$$
where τ is an appropriately small step size in the direction of the improving gradient **d**
6. Update the Hessian **H** using BFGS for the new set of controls.
7. Perform steps 2 to 6 until the convergence of J_2

The EnOpt algorithm has been used to approximate the gradient of the secondary objective function **s**. However there exist other schemes, such as conjugate gradient and genetic algorithms, which can be implemented.

1.7.3 Switching Algorithm

Van Essen *et al*. (2011) presented an alternative method to the hierarchical optimization algorithm. The reason for this alternative was, the hierarchical structure is computationally cumbersome and not feasible for realistic reservoir models having large number of input parameters when implemented with the Adjoint formulation. Thus the switching algorithm was tested with the use of a balanced objective function as described below.

$$J_{bal} = \Omega_1 \cdot J_1 + \Omega_2 \cdot J_2 ,$$

where Ω_1 and Ω_2 are the switching functions of J_1 and J^* that take on values of 1 and 0 or vice versa:

$$\begin{aligned} \Omega_1(J_1) &= \begin{cases} 1 & \text{if } J_1^* - J_1 > \varepsilon \\ 0 & \text{if } J_1^* - J_1 \le \varepsilon \end{cases} \\ \Omega_2(J_1) &= \begin{cases} 0 & \text{if } J_1^* - J_1 > \varepsilon \\ 1 & \text{if } J_1^* - J_1 \le \varepsilon \end{cases} \end{aligned} \tag{18}$$

Here ε is the threshold value as defined in the inequality constraint in eq. (13). J_1^* is the value of the primary objective at the optimal solution achieved during life cycle optimization.

The gradient of J_{bal} with respect to the input parameters is then given by

$$\left.\frac{dJ_{bal}}{d\mathbf{u}}\right|_{n+1} = \Omega_1(J_{1,n}) \cdot \left.\frac{dJ_1}{d\mathbf{u}}\right|_{n+1} + \Omega_2(J_{1,n}) \cdot \left.\frac{dJ_2}{d\mathbf{u}}\right|_{n+1} . \tag{19}$$

The use of a balanced objective function sequentially for the optimization will give improving directions for either J_1 or J_2. Thus with each iteration the value of J_2 either increases while J_1 decreases or vice versa as the solution switches from the infeasible solution to a feasible solution in accordance with the inequality constraint. However, the convergence towards an optimal solution would be rather slow due to the switching between the different solutions.

In order to improve convergence speed Van Essen et al (2010) suggested a small adaptation may be made to the switching algorithm. The gradients of the secondary objective function are projected onto the null

space of the optimal primary objective function. The general definition for projection of any vector on to a space spanned by the columns of a matrix **A** is

$$\mathbf{P = A(A^T A)^{-1} A^T} . \tag{20}$$

Consider the case when **A** consists of one column which is a vector then equation (20) becomes

$$\mathbf{P = a(a^T a)^{-1} a^T} \tag{21}$$

When applied to the gradients of the primary objective function we get

$$\mathbf{P} = \frac{dJ_1}{d\mathbf{u}} \left(\frac{dJ_1}{d\mathbf{u}}^T \frac{dJ_1}{d\mathbf{u}} \right)^{-1} \frac{dJ_1}{d\mathbf{u}}^T , \tag{22}$$

In the neighborhood of the optimum, the complement of the gradient with respect to the primary objective function can be used as an approximation to the null space of the Hessian of this function. The orthogonal complement of any vector **a** is given by

$$\mathbf{a = (I - P)} \cdot \bar{\mathbf{a}}$$

Refer to Strang (2006) or Jansen (2009) for a detailed proof of this relation.

Thus the improved direction for the secondary objective is given by

$$\frac{dJ_2}{d\mathbf{u}}^T = [\mathbf{I - P}] * \frac{dJ_2}{d\mathbf{u}}^T , \tag{23}$$

Thus the alternative switching search direction **d** for solving the hierarchical problem is

$$\mathbf{d}_{n+1}^T = \Omega_1(J_n) \cdot \left. \frac{dJ_1}{d\mathbf{u}} \right|_{n+1}^T + \Omega_2(J_n) \cdot [\mathbf{I - P}] * \left. \frac{dJ_2}{d\mathbf{u}} \right|_{n+1}^T . \tag{24}$$

Van Essen *et al.* (2011) proposed a slightly different version of the projection matrix **P** in the formula above wherein the projection matrix is given by $\mathbf{P = aa}^T$.

RESERVOIR MODEL

The application of the multi-objective methods described with the ensemble based optimization algorithm was first tested on the well-known Rosenbrock function. Results of these tests are presented in the Appendix. In this chapter experiments are presented in which the methods were applied to two synthetic reservoir models.

1.8 2D Model

The following case is built in ECLIPSE 100. The geological and fluid properties of this model are depicted in Figure 2 and specified in table 1. The reservoir is divided into 5 layers having different horizontal permeabilities. There are 2 vertical multi-segmented wells which penetrate all the 5 layers: 1 injector and 1 producer. Each segment corresponds to a layer and inflow or outflow can be separately controlled by an inflow control valve (ICV). The producing life of the reservoir is 1500 days divided into 15 time intervals of 100 days each. Thus we will optimize 150 controls. Water is injected at a constant pressure of 210 bars and the production well is operated at 190 bars. A Corey model is used for the relative permeabilities with 20% connate water saturation and an end point relative permeability to oil and water are 0.8 and 1 respectively . No capillary pressures have been included in this model. The reservoir rock is incompressible.

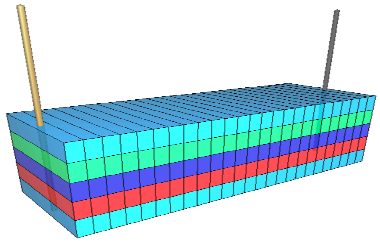

Figure 2: Illustration of the reservoir model to test multi-objective optimization methods

The geological and flow properties of this simple choke model are given in the table 1 below.

Property	**Values**	**Units**
Porosity	0.2	--
Permeability (layer 1 to layer 5)	100-200-50-400-100	mD
Reservoir pressure	200	bar
Oil density	800	kg/m^3
Water density	1000	kg/m^3
Oil compressibility	0.0002	1/bar
Water compressibility	$4e^{-5}$	1/bar
Viscosity of oil	2	cP at 1 bar
Viscosity of water	0.5	cP at 1 bar

Table 1: Geological and flow properties of the model.

1.8.1 Life Cycle Optimization

An optimal strategy of ICV settings for the individual layers is obtained by optimizing the NPV as described in equation 2, with r_o = 150 USD/m^3, r_{wp} = 25 USD/m^3, r_{wi} = 5 USD/m^3. The discount rate b was set to 0. Figure (3) is an illustration of the optimization with undiscounted NPV as the objective function. The optimal solution $\mathbf{u}^*$ was obtained using a steepest ascent scheme with the trust region algorithm a description of which can be found in the appendix. The optimization algorithm finds an optimal solution within a set of bounds. The inflow control valve settings are modeled as well productivity index (PI) multipliers in ECLIPSE. Thus the maximum bound applied equals unity, which corresponds to the valve being fully open. The minimum bound was set to 1e-4, ideally we would like to set the bound to 0, corresponding to the valve being fully closed, however 1e-4 is used since the simulator cannot accept 0. The starting point of this optimization was a control vector with all values equal to 1. Thus all the ICVs/chokes are open as a starting strategy. An ensemble size of 100 members was used. The optimization was allowed to run for 75 iterations although there was no significant improvement in objective function value after 30 iterations. This was done to allow the BFGS algorithm to estimate a Hessian matrix as close to the true Hessian as possible. The value of the objective function at the optimal strategy is USD 2.489x10^7.

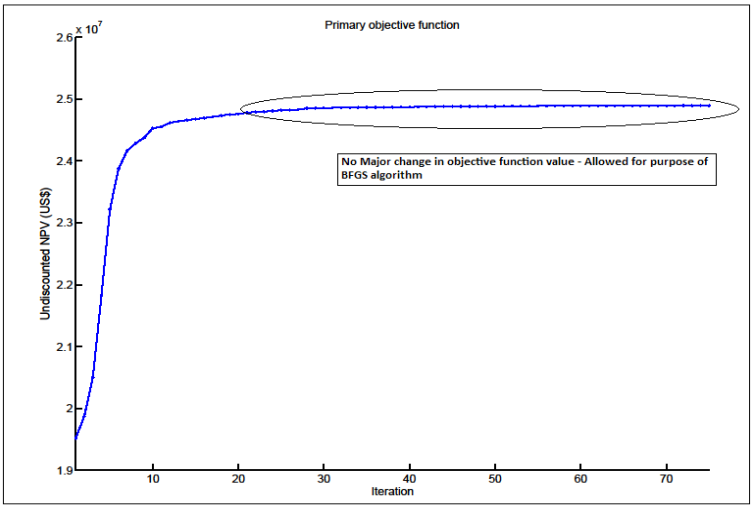

Figure 3: Life cycle optimization with primary objective function as undiscounted Net Present Value (NPV)

The set of controls i.e. the operating strategy for the ICV settings over the life of the reservoir, which maximizes the objective function (solid line), is shown in figure (4). For comparison the initial strategy of having all the ICVs kept fully open (blue dashed line) is also shown.

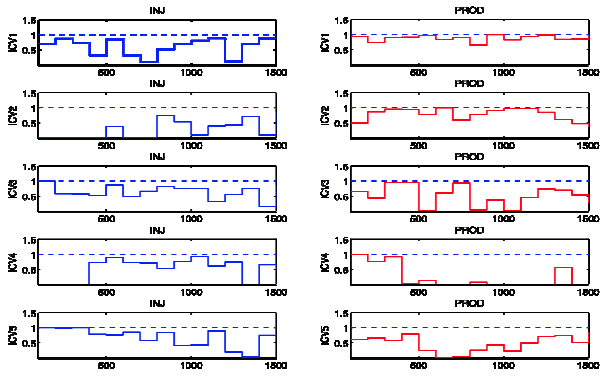

Figure 4: Comparison of the optimized control strategy (solid lines) with the initial control strategy (dashed lines)

This set of controls illustrates the presence of a high permeability layer in the reservoir. The ICV4 corresponds to this high permeability layer hence the optimal strategy does not require injection in the initial 500 days since the permeability is high enough for the oil to flow. The production from this layer is negligible after 500 days to avoid water breakthrough thus reducing water production.

Another important aspect of this exercise was to compare the gradients as obtained with the EnOpt with gradients obtained using a Finite Difference scheme. We do not deal with exact gradients like Adjoint gradients; we assume the Finite Difference gradient to be the best approximation of the exact gradients. The finite difference approximation to find the gradient is obtained using the following formula [Nocedal and Wright (2006)]

$$\mathbf{g}_i = \frac{J(\mathbf{u} + \Delta\mathbf{u}_i) - J(\mathbf{u})}{\Delta\mathbf{u}_i}$$

where **g** is the gradient of the control vector **u,** J(**u**) is the function evaluation and $\Delta\mathbf{u}_i$ is a control vector where element *i* has been perturbed.

Figure (5) is a comparison of the gradients estimated by the Finite difference method (blue) and EnOpt (red). The ensemble size to evaluate the gradient with EnOpt was 100 samples.

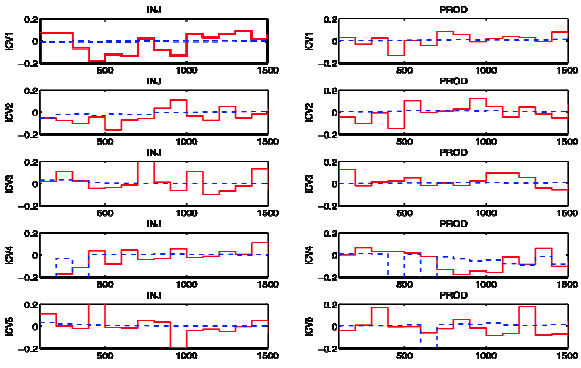

Figure 5: Comparison of the gradients approximated from EnOpt (red) and Finite Difference gradients (blue)

The finite difference based gradients are only determined at the optimal strategy achieved using the EnOpt gradients. We chose 100 samples for the EnOpt because the larger the number of samples the

better the gradient is approximated. For optimization with bounds (constraints) on the controls, the magnitude of the gradient is not necessarily 0 at the optimum. A condition for optimality in the constrained case, is that for those controls which are at the bounds, the gradient is pointing outward of the feasible domain. This is what is observed in Figure (4) and Figure (5). For ICV5 of the producer at time step 6 it is observed that the gradient is larger than 0, at the same time step the controls for ICV5 are at the bounds. Similarly for ICV4 of the producer at time steps 4 and 6 and time steps 1 and 3 for the injector similar trends are observed. Thus the conditions for optimality are thus met, confirming that we are indeed at or very close to an optimum.

Robust Optimization

A robust optimization scheme, which takes into account geological uncertainty, was implemented. As has been explained in the theory, EnOpt becomes especially computationally attractive when used for robust optimization in which the *expected* objective function value is optimized. This was tested on this simple 2D model. A set of geological models was created for the optimization. The geological properties described in Table 1 are used as the base case properties. A random Gaussian ensemble of multipliers was created with a fluctuation of 0.15 around 1. Thus maximum and minimum values of the multipliers will be 1.15 and 0.85 respectively. These multipliers were subsequently multiplied with the base case permeability.

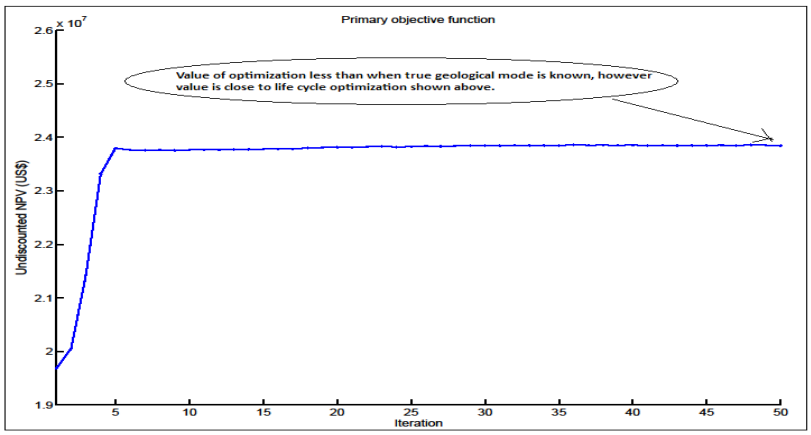

Figure 6: Illustration of a robust life cycle optimization with undiscounted NPV as the objective function.

Note: The robust methodology is applied when the true geological model of the reservoir is not known with certainty. As shown in figure (6), in this case the optimization leads to a strategy with a somewhat lower expected primary objective function value than if the true properties are exactly known.

1.8.2 Multi-Objective Optimization

The primary objective function used is undiscounted NPV which is equivalent to cumulative cash flow over the producing life of the reservoir. In an ideal case we would like to achieve maximum production in the initial life of the reservoir whilst still maintaining the life cycle objectives. Thus we choose a secondary objective function which highlights the importance of short-term production similar to van Essen et al. (2011). The secondary objective function is the primary objective function at a very high discount rate *b* of 25%. This has been chosen to emphasize the desire to increase production on a short-term basis whilst

maintaining the long-term optimization results. The two multi-objective methods proposed in the theory have been tested on the 2D model. A comparison between the results obtained by the two methods is also provided. As a means of comparison, an unconstrained optimization for the secondary objective function starting from the optimal strategy **u*** is performed. This will not ensure maintaining optimality of the primary objective function (J_1). The results of this optimization have been illustrated in comparison with the constrained optimization results in Figure (8) and Figure (12).

Hierarchical Optimization

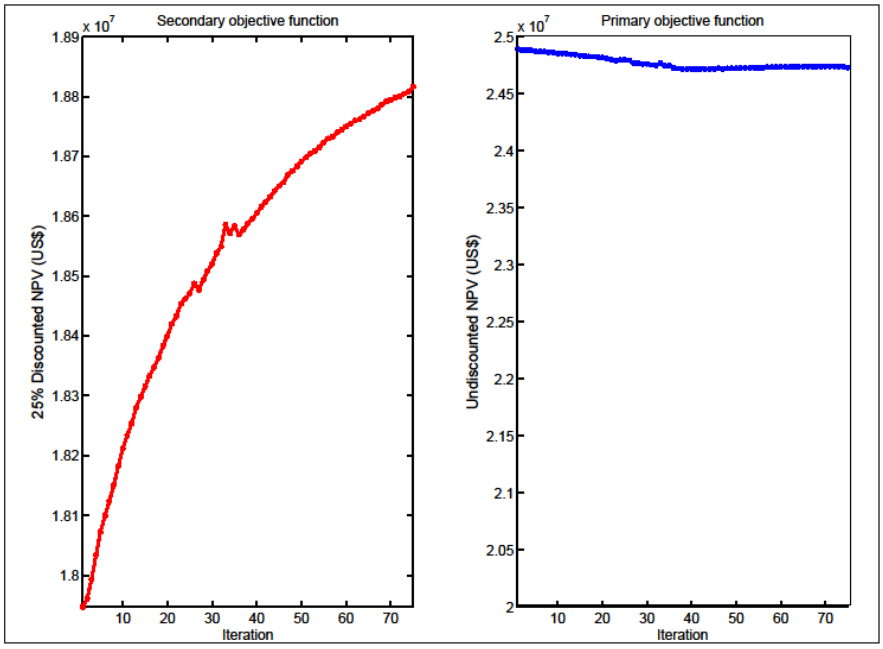

Figure 7: Comparison of the primary (red) and secondary (blue) objective functions obtained by the hierarchical optimization algorithm

Figure (7) illustrates the optimization of the secondary objective function within the null space of the primary objective function. We observe a 4.72% increase in the secondary objective function for a corresponding 0.6% decrease in the primary objective function. Since we are dealing with a small model, the changes are not spectacular, and on larger, more complex models one may expect to see higher increases in the secondary objective function. To illustrate the effect of the high discount factor a comparison of the cash flow over time between the hierarchical, unconstrained and life cycle optimizations is shown in Figure (8). We see an increase in the short-term cash flow compared to the life cycle optimization.

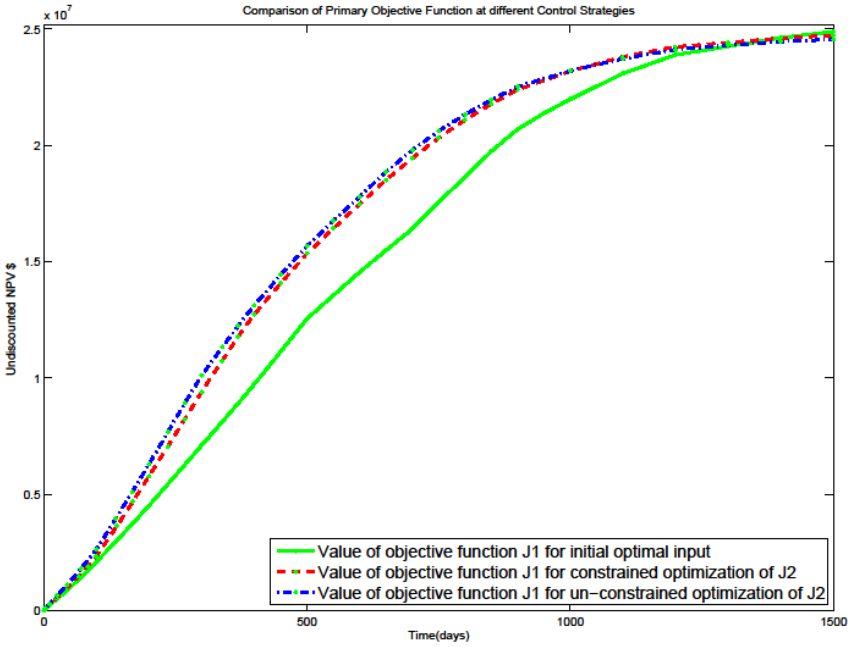

Figure 8: Plots showing the cash flow over time for the different optimization strategies, life-cycle optimized strategy (green), hierarchical optimized strategy (red) and unconstrained secondary optimized strategy (blue)

Figure (9) shows the controls for the individual ICVs during the life of the reservoir. This plot illustrates how the different set of controls for the different optimization strategies achieve similar results over the producing life of the reservoir.

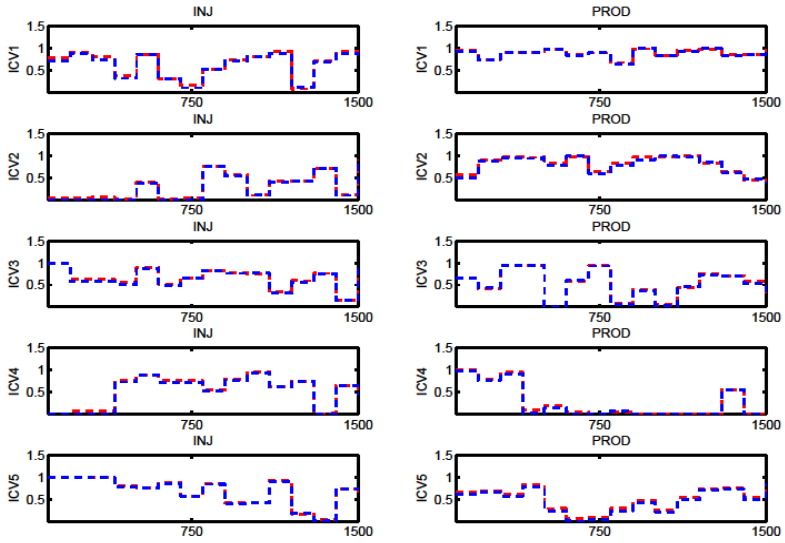

Figure 9: Comparison of the control strategies for the hierarchical (red) and life cycle (blue) optimization methods, different control sets achieving similar life cycle results

Discussion

Van Essen et al. (2011) proposed the use of a combined adjoint and finite difference scheme to approximate the Hessian matrix used in the Hierarchical optimization algorithm. He computed the first derivatives with an adjoint and the second derivatives with perturbations. However, because we do not

have access to an adjoint we would need to compute both the first and second derivatives with the aid of perturbations. For a large number of controls this is computationally extremely expensive. To estimate the Hessian at a control strategy we require *n*(n+1)* function evaluations where *n* are the number of controls. Thus for 150 controls this relates to 23,000 function evaluations. However, the BFGS algorithm proposed in this work estimates a Hessian during life cycle optimization without the need for additional function evaluations, thus making it computationally more attractive. As a means to compare the proposed BFGS-based Hessian with a perturbation-based Hessian we have also estimated the Hessian matrix using a finite difference scheme. We reduced the number of controls by 1/3rd to 50 controls to achieve faster results. Performing an SVD on both the Hessian matrices, we observed that the null space for the two Hessians was dependent on the cutoff criteria used to find the singular values close to zero. Van Essen et al (2011) had proposed to use the following criteria $[\sigma_i < 0.02\sigma_1]$ where σ_i is the singular values of **H**. If we employ this criterion then the number of vectors which span the null space (i.e. its dimension) of the two Hessians is the same. However, when we change this criterion the number of vectors spanning the null space is different. Thus the definition of this cutoff criterion is very important. The cut-off criterion used in our optimization is $[\sigma_i < 2e^{-6}\sigma_1]$. This criterion does not change throughout the optimization and has been chosen based on the magnitude of the largest singular value σ1. The null space consists of 77 vectors at the beginning of the secondary optimization and 68 vectors at the end. Thus throughout the optimization the number of vectors spanning the null space decreases.

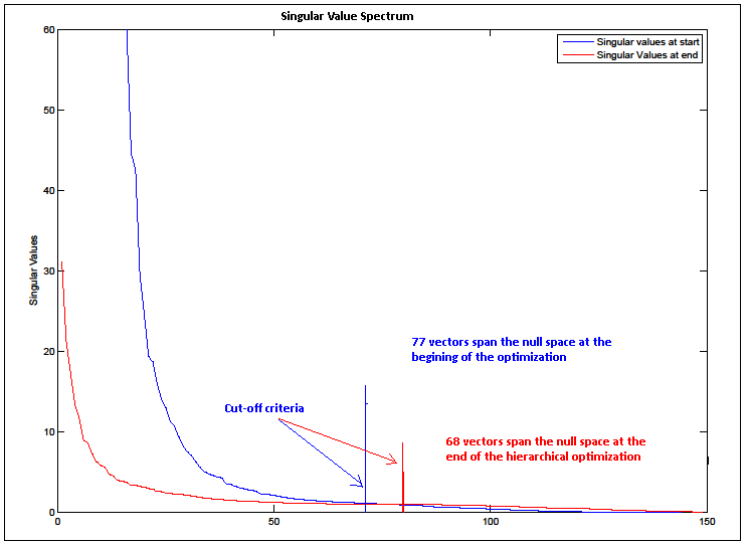

Figure 10: Singular value spectrum showing the singular value spectrum at the beginning (blue) and end (red) of the hierarchical optimization

Figure (10) is an illustration of the singular value spectrum for the 2D model. The dimension of the null space based on the cut-off criterion used is shown in figure (10).

Switching Algorithm

The mathematical formulation of the switching algorithm described in the theory requires the definition of a criterion ε. We have defined the criterion as $\{\varepsilon = 0.003\, J^*\}$. The switching algorithm has two variations, as discussed in the theory. An ensemble size of 100 samples was used and the optimization required 100

iterations to achieve the results shown in figure (11). At this point it appears as no optimum has been reached yet for the secondary objective.

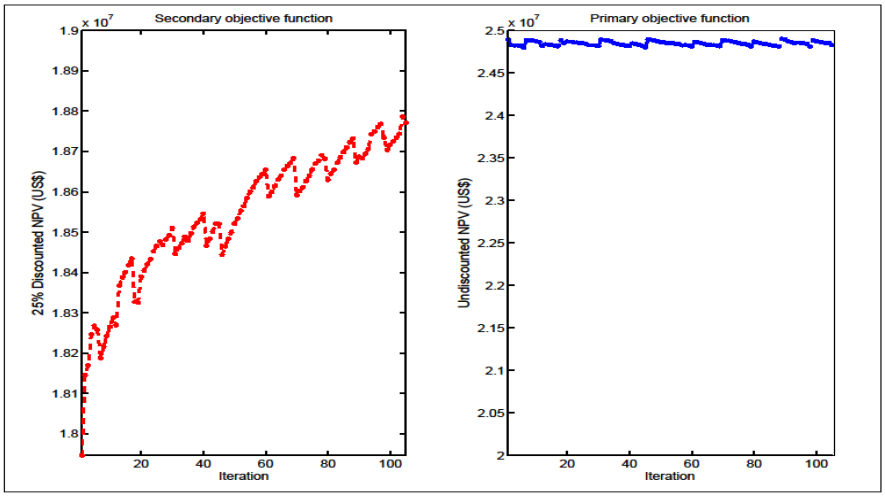

Figure 11: Illustration of switching algorithm optimization with projection applied to the gradient

Figure (11) illustrates a decrease in the primary objective function with a corresponding increase in the secondary objective function. Unlike the profile shown in Van Essen et al (2011), we do not observe switching between feasible and unfeasible solutions. This is due to the projection of the secondary gradient onto the primary gradient. The minor decreases, which are observed in the profile of the secondary optimization, could be due to the approximate EnOpt gradients being used for the optimization. The switching algorithm with projection applied achieves an increase of 4.75% for the secondary function for a corresponding 0.2% decrease in primary objective function.

The set of controls comparing the strategies obtained during life cycle optimization (blue) and multi-objective optimization using switching with projection (red) is illustrated in Figure (12). Minor changes in the control set as shown in Figure (12) achieve superior results on the short term whilst maintaining the long term goals.

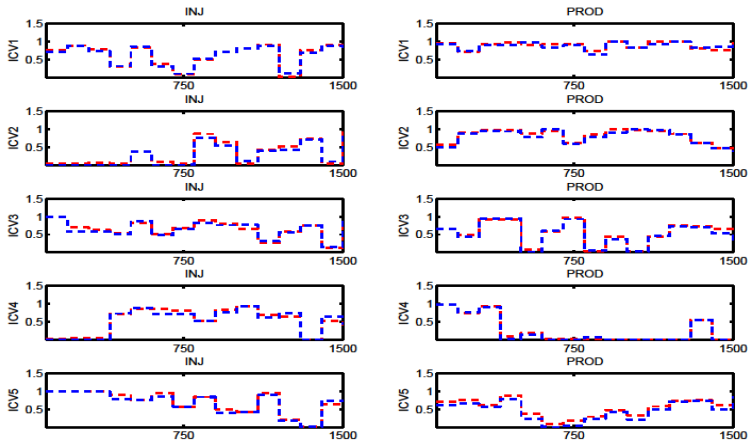

Figure 12: Comparison of the controls of the different ICVs obtained from the switching (red) and life cycle (blue) optimizations.

Figure (13) illustrates the impact on the cash flow over time when the strategy obtained using the

switching algorithm is used. We see an increase in the cash flow, which indicates a better strategy in terms of break even times, thus making the project economically attractive. Although the unconstrained optimization (blue) shows the maximum increase over the short term, the NPV after 1500 days is lower when compared to the hierarchical optimization (red).

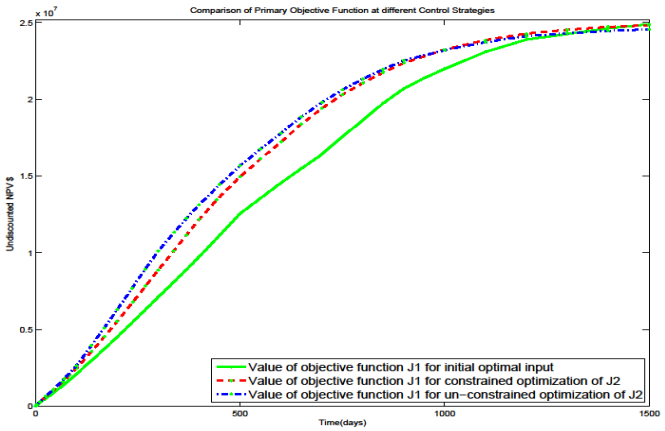

Figure 13: Comparison of the cash flow over time for the multi-objective switching (blue), unconstrained multi-objective (red) and optimized life-cycle (green) strategies

Unconstrained Optimization

The unconstrained optimization shows an increase of 4.85% in the secondary objective function. However this is achieved at a decrease of 1.5% in the primary objective function, which is much larger than the decrease resulting from the constrained optimization methods.

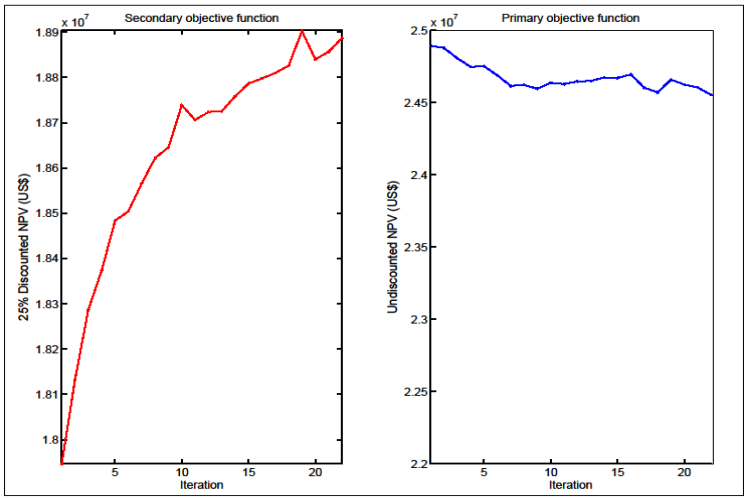

Figure 14: Illustration of unconstrained optimization of the secondary objective function

Hessian Comparison

The BFGS algorithm requires the gradients between successive iterations to approximate the Hessian. Using a finite difference method to calculate gradients as well as the EnOpt with an ensemble size of 100 members we obtain two Hessians. The singular values of these two Hessians are very similar. The number of vectors which span the null space for these Hessians is also found to be very similar for any cut-off

criteria. Thus we may conclude that for the 2D model the Hessian approximated with EnOpt gradients is very similar to the Hessian approximated with finite difference gradients which are assumed to be a better approximation to the exact gradients than the EnOpt gradients.

1.8.3 Conclusions

The 2D model was used to test two multi-objective optimization methods with an ensemble based gradient estimation scheme on a simple reservoir model. The results for multi-objective optimization showed improvements in the secondary objective function without large decreases in the primary objective function value. An overview of the results obtained is shown in Table 2

Optimization Method	Primary Objective (10^6)	Secondary Objective (10^6)	% Increase in second objective	% Decrease in primary objective
Hierarchical	2.474	1.88	4.72	0.6
Switching	2.484	1.882	4.75	0.2
Unconstrained	2.455	1.885	4.85	1.5

Table 2: Overview of multi-objective optimization results with the 2D model

The choke model was used a stepping stone towards the application of the multi-objective optimization code to larger, more representative synthetic reservoir models which is illustrated in the next sub-section.

1.9 3D Synthetic Reservoir Model

A 3D synthetic reservoir model was used to test the multi-objective optimization methods. The life cycle of the reservoir covers 15 years, or 5470 days. The model as shown in Figure (15) consists of 4000 grid blocks, having dimensions of 25x32x5. The approximate size of the grid blocks is 110x90x20 meters. The model represents an area of 2.5 km by 3.5 km and is 100 m thick. The geological structure consists of connected uplifted/offset blocks. There is presence of a sealing fault on the north-western side of the block, close to producer 1. The initial average reservoir pressure is 200 bars. Table 3 lists the geological and fluid properties used to describe the model.

The reservoir is produced using an inverted 5 spot well pattern, i.e. 4 producers at the edges of the grid with an injector in the centre. The reservoir is divided into 5 layers having different horizontal permeabilities. The wells penetrate all the 5 layers with an ICV in every layer resulting in a total of 25 controls per time step. The producing life of the reservoir is divided into 15 time intervals of 1 year (365 days) each, which results in a total of 375 controls that are to be optimized. Water is injected at a constant pressure of 300 bars and the production wells are operated at a minimum of 15 bars. A Corey model with exponents of 2 for both oil and water are used for the relative permeabilities where connate water saturation is 20% and end point relative permeability to oil and water of 0.8 and 0.4 respectively. No capillary pressures have been included in this model. The reservoir rock is incompressible. The simulator used in this study is ECLIPSE 100.

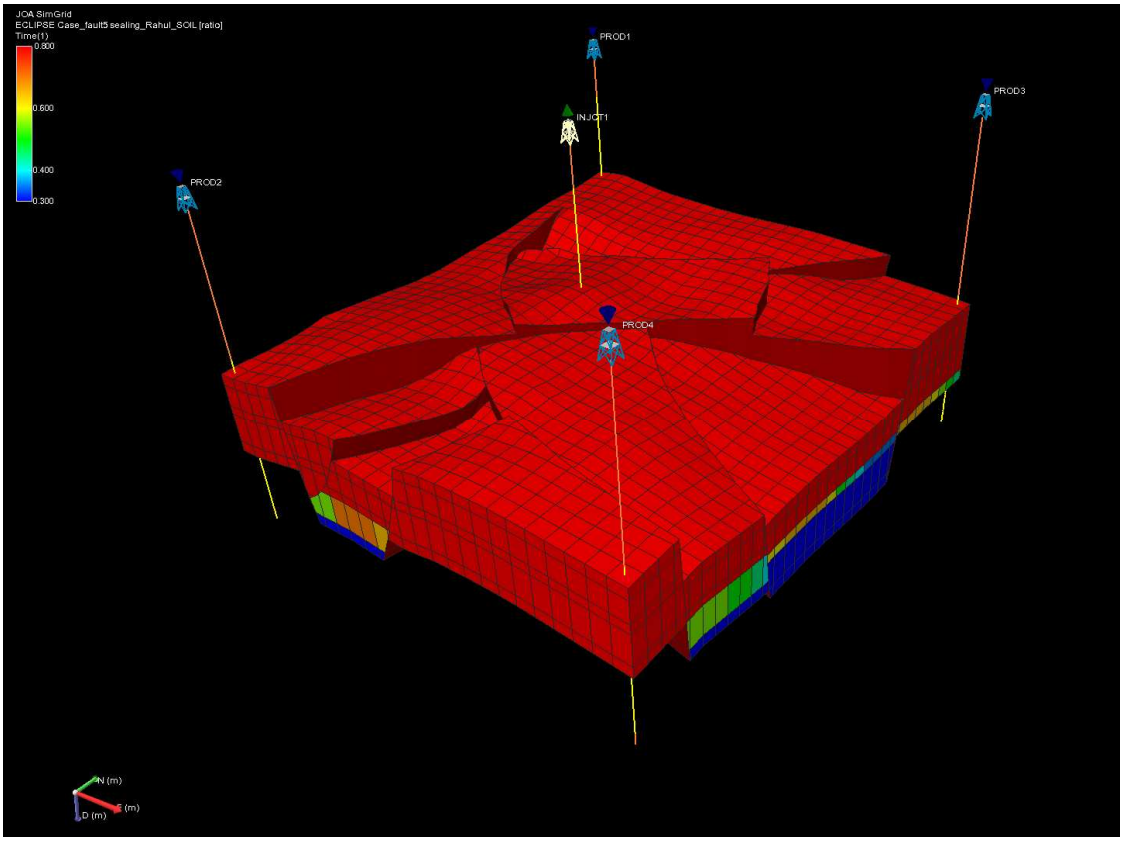

Figure 15: Illustration of the reservoir model to test multi-objective optimization methods.

Property	Values	Units
Porosity	20%	--
Permeability- (layer 1 – layer 5)	100-200-50-400-100	mD
Reservoir Pressure	200	bar at 1950 m
Density of Oil	800	kg/m^3
Density of Water	1000	kg/m^3
Temperature	77	^0F
Oil compressibility	$4e^{-5}$	1/bar
Water compressibility	$4e^{-5}$	1/bar
Viscosity of Oil	2	cP at 1 bar
Viscosity of water	0.5	cP at 1 bar

Table 3: Table containing the geological and flow properties of the synthetic reservoir model

1.9.1 Life Cycle Optimization

An optimal life-cycle strategy of ICV settings for the individual layers is obtained by optimizing the NPV as described in equation 2, with r_o = 130 USD/m^3, r_{wp} = 25 USD/m^3, r_{wi} = 6 USD/m^3. The discount rate b was set to 0. Figure 16 is an illustration of the optimization with undiscounted NPV as the objective function. The optimal solution $\mathbf{u}^*$ was obtained using a steepest ascent scheme with the line search method and an ensemble size of 100 samples. As explained with the 2D model above, well productivity index (PI) multipliers are used to model ICVs in ECLIPSE. These multipliers have bounds between 1e-4 and 1.

The starting point for the optimization is an initial control vector having values equal to 1. Thus all the ICVs/chokes are open as a starting strategy. The optimization was allowed to run for 80 iterations although there was no significant improvement in objective function value after 65 iterations as indicated in Figure (16). Additional iterations were performed to allow the BFGS algorithm to estimate a Hessian matrix that is as close to the true Hessian at the optimum as possible. The optimized value of the objective function is USD 8.902x10^9 $.

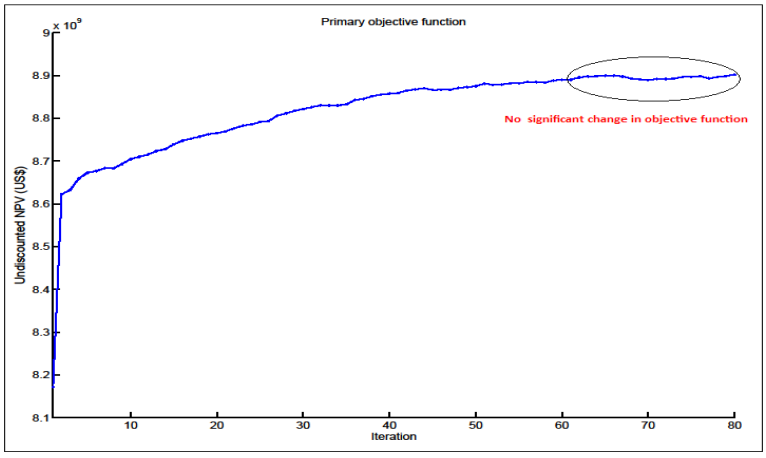

Figure 16: Life cycle optimization with undiscounted NPV as the objective function

The set of controls, i.e. the operating strategy for the ICV settings over the life of the reservoir, which maximizes the objective function, is shown in Figure (17). A comparison with the initial strategy of having all the ICVs kept fully open is also illustrated (blue line).

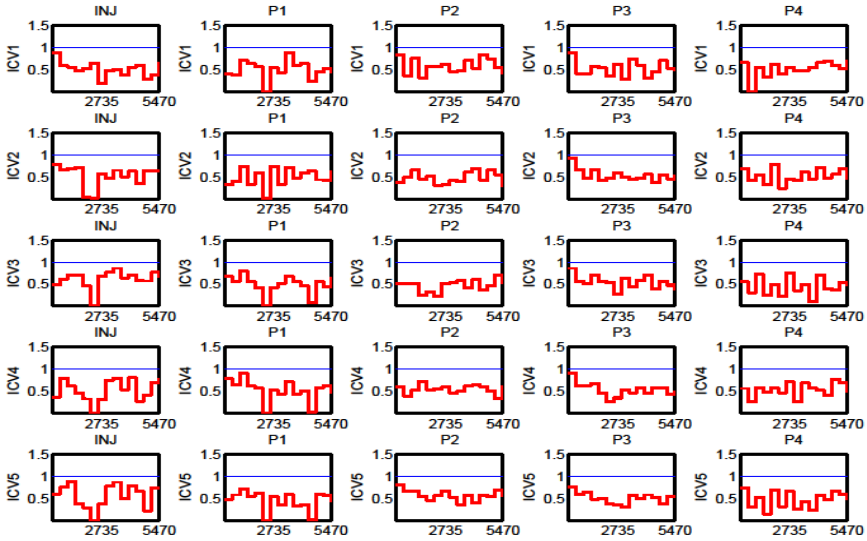

Figure 17: Comparison of the optimized control strategy (red solid lines) with the initial base case starting strategy (blue solid lines) for all the 5 wells and ICVs.

The blue line in each of the graphs in Figure (17) shows the starting strategy of the optimization, the ICVs are fully open. The red line is the control strategy obtained after optimization where undiscounted NPV is used as the objective function. ICV 4 corresponds to the high permeability layer (600 mD) in the reservoir. We observe that the average ICV settings of this layer for the producers are higher during the initial life of the reservoir and decreases with time. The layer being the most permeable will be the most susceptible to water breakthrough. The ICV settings of layer 5 also show similar behavior even though the permeability is much lower at 100mD compared to layer 4. This is because sections of this layer are below the Oil Water Contact; hence this zone also waters out fairly rapidly.

1.9.2 Multi-Objective Optimization

The objective functions used in multi-objective optimization for this model are similar to the functions used for the choke model. Following are the results obtained for the multi-objective optimization with this model.

Hierarchical Optimization

The hierarchical optimization was performed with an ensemble size of 75 members. The optimization was allowed to run for 55 iterations after which no significant improvements in the secondary objective function were observed. Figure (18) illustrates the optimization of the secondary objective function within the null space of the primary objective function.

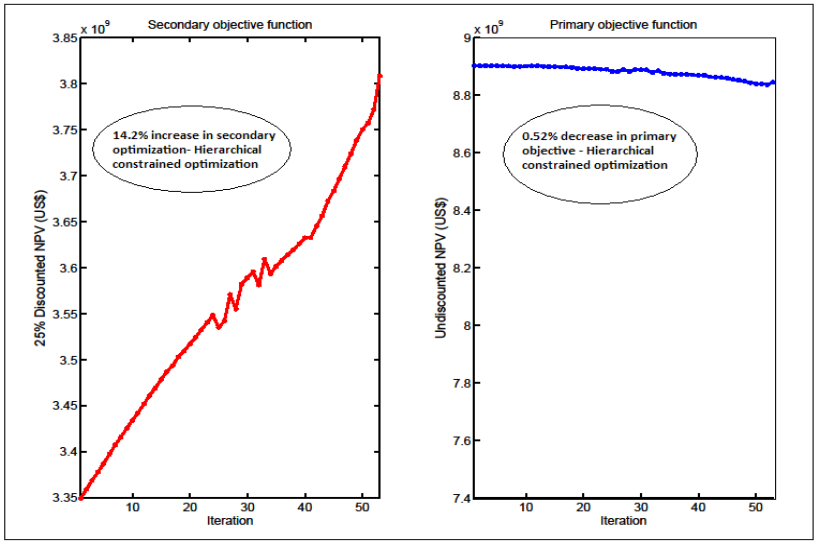

Figure 18: Comparison of the primary (blue) and secondary (red) objective functions obtained by the hierarchical optimization algorithm.

The value of the secondary objective function at the optimum of the primary objective is 3.35×10^9 \$, and this is the starting point of the hierarchical optimization. We achieve a value of 3.823×10^9 \$ after completing the hierarchical optimization. Thus we observe a 14.2% increase in the secondary objective function for a corresponding 0.52% decrease in the primary objective function. To illustrate the effect of the high discount factor a comparison of the cash flow over time between the hierarchical, unconstrained and life cycle optimizations is shown in Figure (19). An increase in the short to medium term cash flow is observed with the strategy obtained by the hierarchical optimization when compared to the life cycle optimization strategy.

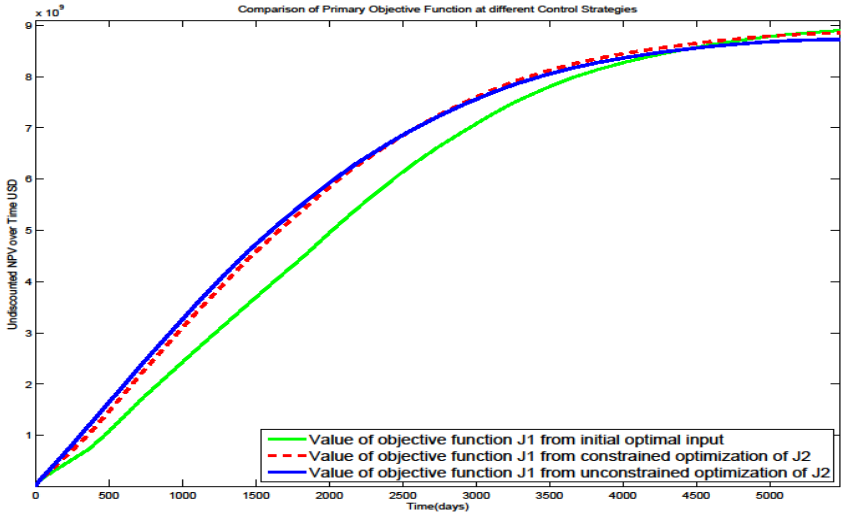

Figure 19: Plot showing the different cash flows over time for the different optimization strategies: life-cycle optimized strategy (green), hierarchical optimized strategy (red) and unconstrained secondary optimized strategy (blue)

The multi-objective optimization is seen to be useful to increase cash flow in the initial stages of the project whilst maintaining the life cycle goals of the project. Every project aspires to recover the initial investments as soon as possible. Hierarchical optimization in our case achieves such results.

So how different are the sets of controls between the optimization strategies? Figure (20) shows the controls for the individual ICVs during the life of the reservoir. This plot illustrates that the optimized strategies obtained from life cycle optimization (black) and hierarchical optimization (blue) are very similar. Only ICV3, which corresponds to the low permeability layer (50mD), shows some variation in the control set.

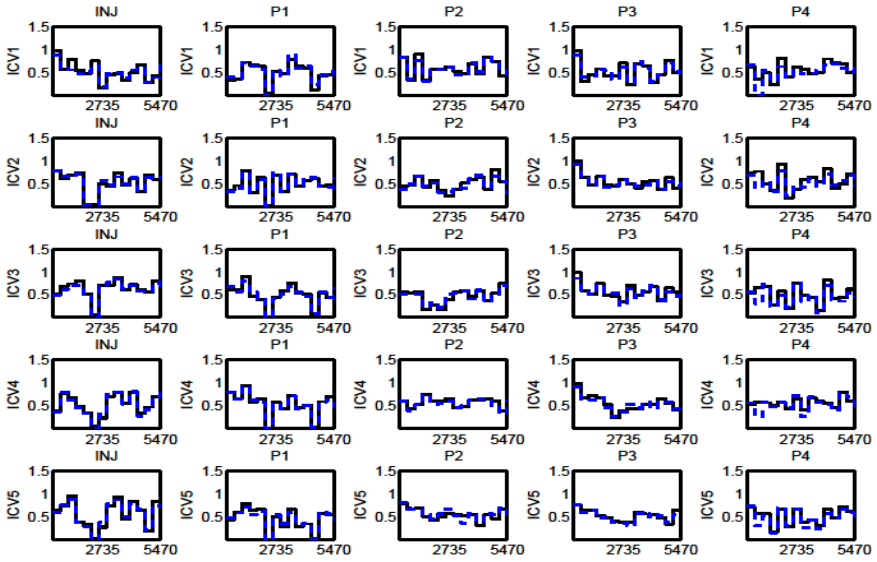

Figure 20: Comparison of the control strategies obtained from the hierarchical (blue) and life cycle (black) optimization methods, showing very minor differences in the control vector achieve similar life cycle results

Cutoff Criteria for SVD:

A cutoff criterion must be defined to estimate the null space of the Hessian matrix. We have used a cutoff criteria of $[\sigma_i / \sigma_1 < 2e^{-9}]$, where σ_i are the singular values of the Hessian matrix, with the values arranged from largest to smallest. This cutoff criterion has been chosen based on the magnitude of the largest singular value. The dimension of the resulting null space is 187 vectors which remains almost constant throughout the optimization. Figure (21) illustrates the singular value spectrum of this model.

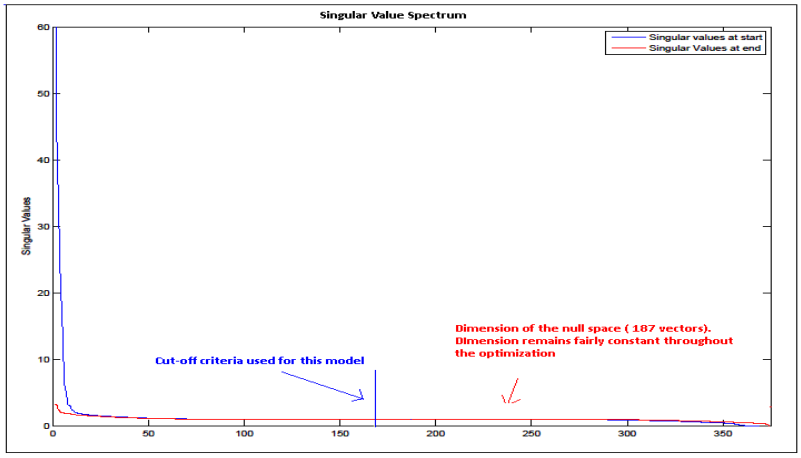

Figure 21: Singular value spectrum showing the number of vectors spanning the null space based on the cut-off criteria used at the beginning (blue) and end(red) of the hierarchical optimization

Switching Algorithm

The mathematical formulation of the switching algorithm described in the theory requires the definition of a criteria ε. We have defined the criteria $\{\varepsilon = 0.003J^{*}\}$. An ensemble size of 75 samples was used and the optimization is allowed to run for 50 iterations; however after 35 iterations the improvements in the secondary objective are minimal. Results are shown in Figure (22).

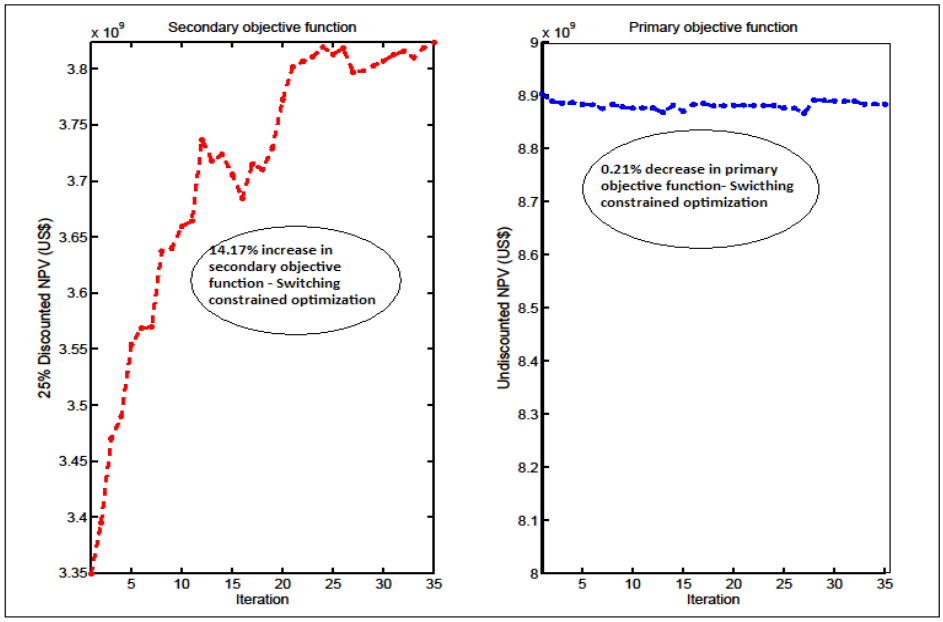

Figure 22: Illustration of switching algorithm optimization with projection applied to the gradient showing a 14.3% increase in secondary objective function (red) for a corresponding 0.2% decrease in primary objective (blue)

Similar to the 2D model, for the switching algorithm used in this thesis projection is applied to the gradient of the secondary objective function. The method achieves an increase of 14.17% for the secondary function for a corresponding 0.21% decrease in the primary objective function. The switching algorithm performs very favorably when applied to this model. The algorithm performance is dependent

on parameters such as ensemble size used for gradient evaluation, step length, size of the trust region etc. Thus fine-tuning of these parameters may lead to an improved performance of the switching algorithm.

In Figure (23) the set of controls obtained by the optimization using the switching algorithm (blue) is compared to the optimal set of control obtained after life cycle optimization (black). Similar to the results obtained by the hierarchical optimization changes in the control set are rather subtle.

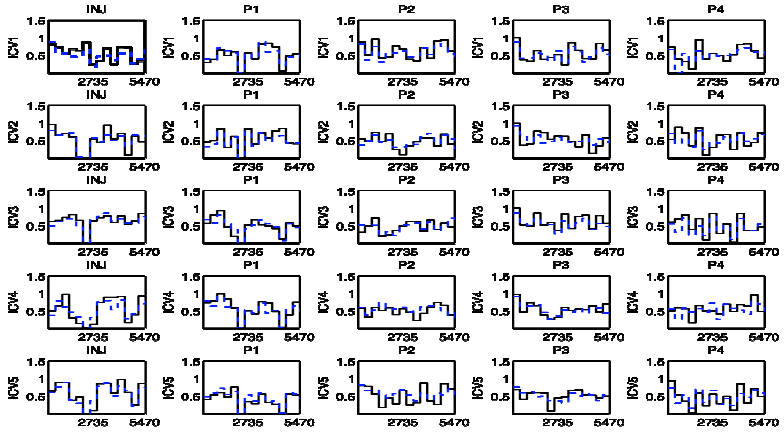

Figure 23: Comparison of the control strategy for the switching algorithm (blue) with optimal control strategy obtained during life cycle optimization (black), for the individual ICVs

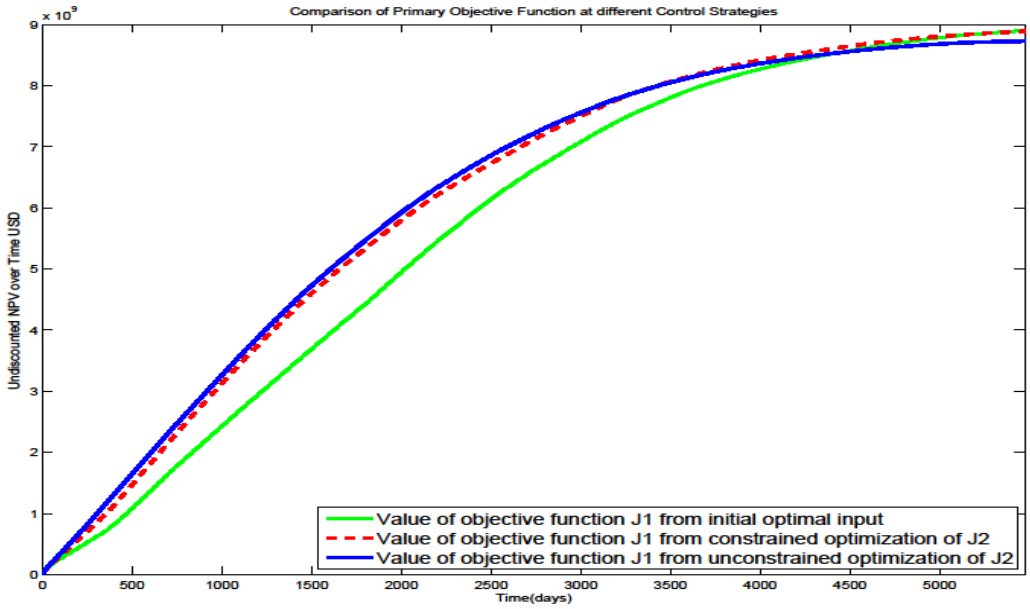

Figure 24: Comparison of the cash flow over time for the switching (red), unconstrained (blue) and optimal life-cycle (green) strategies

The attractiveness of multi-objective optimization is clearly seen in the Figure (24). One of the primary reasons for multiobjective optimization is to improve the production and in turn the cash flow during the initial life of the reservoir, thus making the project economically attractive. It is observed that after 500

days the cash flow with life cycle optimization is approximately 1 x10^9 USD. However the control strategy obtained with switching optimization achieves a cash flow of 1.5 x10^9 USD. This is an increase of 0.5x10^9 $ over 500 days which will enable the project to achieve the break even point faster, thus making this strategy attractive. We also observe that, although the unconstrained optimization performs slightly better at short to intermediate times compared to the switching algorithm, the NPV at the end time of the NPV of the unconstrained optimization (blue curve) is decreased by 1.6% compared to the NPV obtained by the switching algorithm (red curve) which decreases by only 0.2%. Thus the switching algorithm proves the importance of applying constrained optimization.

Algorithm Comparisons

How do the hierarchical and switching algorithms compare with each other? Can we conclude in favor of either of the algorithms? The answer to these questions may never be definitive. It will be model dependent, dependent on the optimization scheme being used, and on the parameters used for optimization, etc. It will be impossible to predict at this point which method will do better compared to the other. However for our model we examine the performance of the two methods.

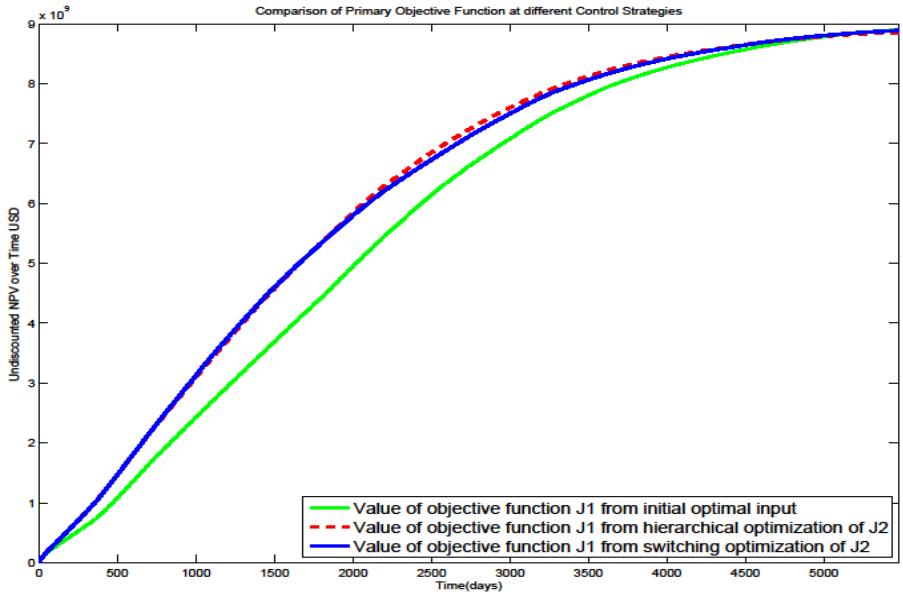

Figure 25: Comparison of the cash flow over time from the optimized strategies obtained from the Switching (red) and Hierarchical (blue) optimizations, the cash flow from the optimized primary objective (green) is also illustrated

We compare the cash flow over the producing life of the reservoir for the optimization strategies obtained by the two methods (Figure 25). We observe that the switching algorithm shows slightly better results than the hierarchical method. The decrease in the primary objective is 0.21% for the switching algorithm while the hierarchical optimization allows a decrease of 0.5% in the primary objective function. The cash flow over time for the switching and hierarchical methods are very similar. The switching algorithm for this model needed less iteration to reach the optimum as compared to the hierarchical method. On the other hand, Figure (17) suggests that there is significant scope for further improvement of the secondary objective with the hierarchical method when more iteration is allowed.

Unconstrained Optimization

The unconstrained optimization results are illustrated in Figure (26). A decrease of 1.65 % is seen in the primary objective function to achieve a similar increase of 14.2% in the secondary objective function. This decrease is much larger compared to the constrained optimization results.

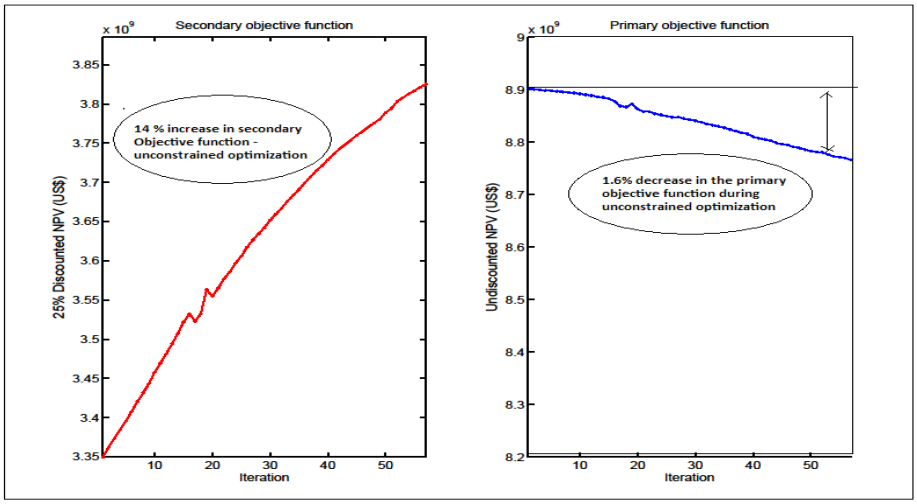

Figure 26: Illustration of an unconstrained optimization of secondary objective (red) starting from the optimal strategy **u*** obtained from life cycle optimization.

1.9.3 Conclusions

The two multi-objective methods have shown an improvement in the optimization of the secondary objective constrained to the primary objective using the EnOpt method for gradient evaluation. The results obtained with the hierarchical optimization and switching algorithms are very similar. The results also show the attractiveness of constrained optimization relative to unconstrained optimization. A brief summary of the results obtained for multi-objective ensemble based optimization of this model is given below.

Optimization Method	Primary Objective (10^9)	Secondary Objective (10^6)	% Increase in second objective	% Decrease in primary objective
Hierarchical	8.854	3.827	14.23	0.52
Switching	8.882	3.823	14.17	0.21
Unconstrained	8.727	3.825	14.2	1.65

CONCLUSIONS

Compromises made to short-term targets during life cycle optimization can be corrected for with multi-objective optimization methods.

In our numerical simulation examples, two constrained multi-objective methods roughly showed a 15% improvement in the secondary objective function (NPV at 25% discount rate) constrained to the primary objective function (NPV at 0% discount rate). The results obtained with the hierarchical optimization method are very similar to those obtained with the switching algorithm, although for the cases investigated here, the switching algorithm shows somewhat less decrease in the primary objective compared to the hierarchical optimization for both the models.

Multi-Objective Optimization of inflow control valve settings shows significant scope for improvement in short term goals constrained to life cycle targets.

The success of multi-objective optimization is model-dependent. A simplistic model, or a model without many degrees of freedom after life cycle optimization, may not be a good candidate for multi-objective optimization.

The choice for the different objective functions is very important; in order for significant scope to exist for multi-objective optimization the objectives should be in conflict with each other.

The EnOpt method estimates an approximate gradient; the larger the ensemble size the better the approximation of the gradient. A larger ensemble means longer computational times. Thus a trade-off must be made to ensure the best possible gradient is achieved whilst maintaining computational efficiency.

The hierarchical method is computationally more challenging than the switching algorithm, since it requires the computation of a Hessian matrix. The BFGS algorithm proposed to estimate the Hessian is computationally attractive compared to previous methods used to estimate the Hessian. The BFGS algorithm requires sufficient iterations to estimate a good approximation to the Hessian. However, when dealing with large control sets the BFGS is more efficient than a finite difference method to estimate the Hessian.

This work illustrates the scope to bridge the gap between life cycle and short-term production targets through multi-objective ensemble based optimization.

The results of the multi-objective ensemble-based optimization confirm our hypothesis.

REFERENCES

Asheim, H. 1988. Maximization of water sweep efficiency by controlling production and injection rates. Paper SPE 18365 presented at the *SPE European Petroleum Conference*, London, UK, October 16-18. DOI: 10.2118/18365-MS.

Brouwer, D.R. and Jansen, J.D. 2004. Dynamic Optimization of Water Flooding With Smart Wells Using Optimal Control Theory. SPEJ 9 (4): 391–402. SPE-78278-PA. DOI: 10.2118/78278-PA.

Chaudari, M.M., Phale, H.A., Liu, N. and Oliver, D.S., 2009. An Improved Approach for Ensemble-Based Production Optimization. Paper SPE 121305 presented at the 2009 SPE Western Regional Meeting, San Jose, California, USA, 24-26 March 2009.

Chen, C., Li, G., Reynolds, A.C. 2011. Robust Constrained Optimization of Short and Long-Term NPV for Closed-Loop Reservoir Management. Paper SPE 141314 presented at 2011 SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, 21-23 February 2011.

Chen, Y., Oliver, D.S. and Zhang, D. 2008. Efficient ensemble-based closed-loop production optimization. Paper SPE 112873 presented at the SPE/DOE Improved Oil Recovery Symposium, Tulsa, USA, 19-23 April, DOI: 10.2118/112873-MS.

Chen, Y., 2008. Efficient ensemble based reservoir management. PHD thesis, University of Oklahoma

Haimes, Y.Y. and Li, D. 1988. Hierarchical multiobjective analysis for large-scale systems: Review and current status. *Automatica* **24** (1): 53-69. DOI: 10.1016/0005-1098(91)90120Q

Jansen, J.D., Douma, S.G., Brouwer, D.R., Van den Hof, P.M.J., Bosgra, O.H. and Heemink, A.W. Closed-loop reservoir management. Paper SPE 119098 presented at the *2009 SPE Reservoir Simulation Symposium*, The Woodlands, Texas, U.S.A, 2-4 February. DOI: 10.2118/119098-MS.

Jansen, J.D., Bosgra, O.H. and Van den Hof, P.M.J. 2008. Model-based control of multiphase flow in subsurface oil reservoirs. *Journal of Process Control* **18**: 846-855. DOI: 10.1016/j.jprocont.2008.06.011.

Jansen, J.D., 2010. *Systems theory for Reservoir Management*, Lecture Notes AES 1490, Delft University of Technology

Jansen, J.D., 2011: Adjoint-based optimization of multiphase flow through porous media – a review. *Computers and Fluids* **46** (1) 40-51. DOI: 10.1016/j.compfluid.2010.09.039.

Leeuwenburgh, O., Egberts, P.J.P., and Abbink, O.A. 2010. Ensemble methods for reservoir life-cycle optimization and well placement. Paper SPE 136916, presented at 2010 SPE/DGS Annual Technical Symposium and Exhibition, Al-Khobar, Saudi Arabia, 04-07 April 2010.

Marler, R.T. and Arora, J.S. 2004. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization* **26** (6): 369-395. DOI: 10.1007/s00158-003-0368-6.

Naevdal, G., Brouwer, D.R. and Jansen, J.D. 2006. Water flooding using closed-loop control. *Computational Geosciences* **10** (1) 37-60. DOI: 10.1007/s10596-005-9010-6.

Nocedal, J; Wright, S. J. (2006), *Numerical Optimization* (2nd ed.), Berlin, New York

Sarma, P., Aziz, K., and Durlofsky, L.J. 2005. Implementation of Adjoint Solution for Optimal Control of Smart Wells. Paper SPE 92864 presented at the SPE Reservoir Simulation Symposium, The

Woodlands, Texas, USA, 31 January–2 February. DOI: 10.2118/92864-MS.

Sarma, P., Chen, W.H., Durlofsky, L.J., and Aziz, K. 2006. Production Optimization With Adjoint Models Under Nonlinear Control-State Path Inequality Constraints. SPEREE 11 (2): 326–339. SPE-99959-PA. DOI: 10.2118/99959-PA.

Sarma, P., Durlofsky, L.J. and Aziz, K. 2008. Computational techniques for closed-loop reservoir modeling with application to a realistic reservoir. *Petroleum Science and Technology* **26** (10 & 11): 1120-1140. DOI: 10.1080/10916460701829580.

Su, H-J., Oliver, D.S. 2009. Smart-Well Production Optimization using an Ensemble based method. Paper SPE 126072 presented at SPE Saudi Arabia Section Technical Symposium, AlKhobar, Saudi Arabia, 9-11 May 2009

Strang, G., 2006. *Linear Algebra and its applications, 4th ed.,* Thomson Brooks/Cole, Pacific Grove.

Van Essen, G.M., Van den Hof, P.M.J and Jansen, J.D. 2011. Hierarchical Long and Short-Term Production Optimization. *SPE Journal* **16** (1) 191-199. DOI: 10.2118/124332-PA.

Van Essen, G.M., Van den Hof, P.M.J and Jansen, J.D. 2010. Determination of lower and upper bounds of predicted production from history-matched models. Proc. 12th European Conference on the Mathematics of Oil Recovery ECMOR XII, Oxford UK, 6-9 September 2010.

Van Essen, G.M., Zandvliet, M.J., Van den Hof, P.M.J., Bosgra, O.H. and Jansen, J.D. 2009. *SPE Journal* **14** (1) 202-210. DOI: 10.2118/102913-PA.

Yeten, B., Durlofsky, L.J., and Aziz, K. 2002. 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, 4–7 November. DOI: 10.2118/79031-MS.

FIGURES

APPENDIX

Numerical Experiments to test Single and Multi-Objective Optimization Techniques

The Rosenbrock function is a non-convex function used as a performance test problem in mathematical optimization for optimization algorithms introduced by Rosenbrock in 1960. It is also known as Rosenbrock's valley or Rosenbrock's banana function.

The global minimum is inside a long narrow parabolic shaped flat valley, finding the minimum is trivial. Convergence to the global optimum is challenging. This function is used to test the implementation of the hierarchical optimization algorithms proposed in the paper.

The Rosenbrock function is as follows:

$$f(x,y) = 100(y - x^2)^2 + (1 - x)^2$$

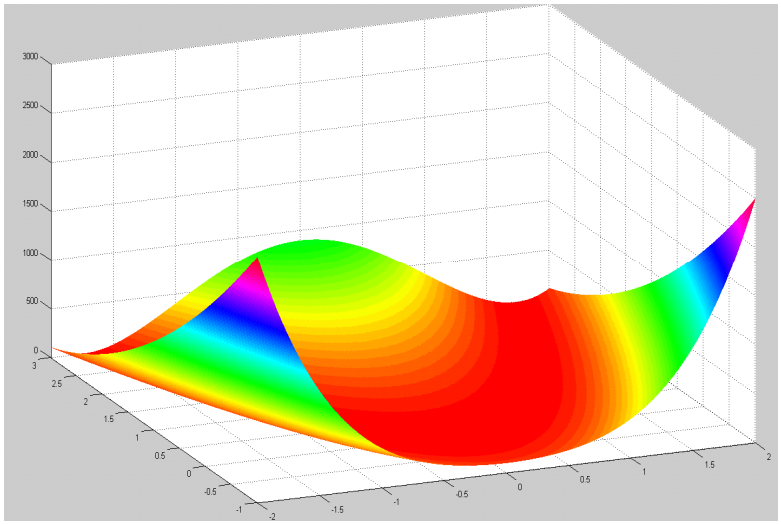

Figure AA: Rosenbrock function in 3D

The Rosenbrock function has a global minimum at [1,1] where the function evaluation $f(x,y) = 0$. The Rosenbrock's function has a shadow minimum at [0.76 0.68] which causes the problems whilst approaching the minimum.

We consider the negative of the Rosenbrock since our mathematical formulation of the optimization problem is defined to find a maximum. The negative of the Rosenbrock is

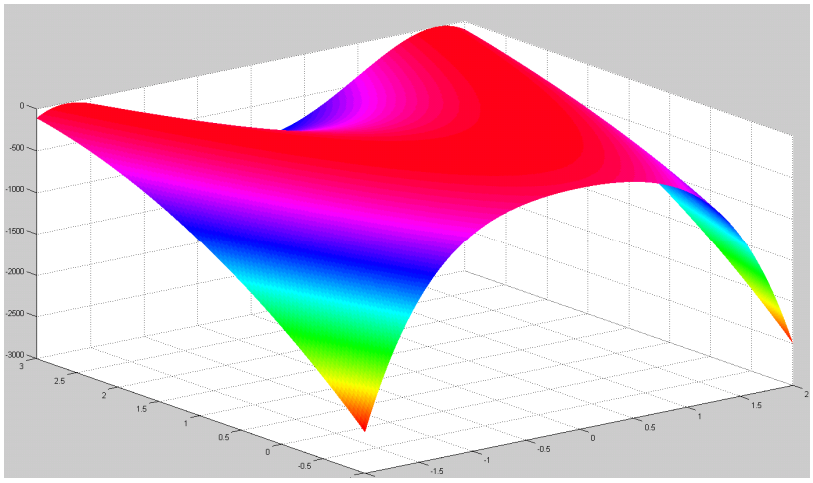

Figure BB: Negative of Rosenbrock considered in our optimization

The line search and trust region algorithms are used as direct search techniques. A 2D contour plot illustrating the optimization path is

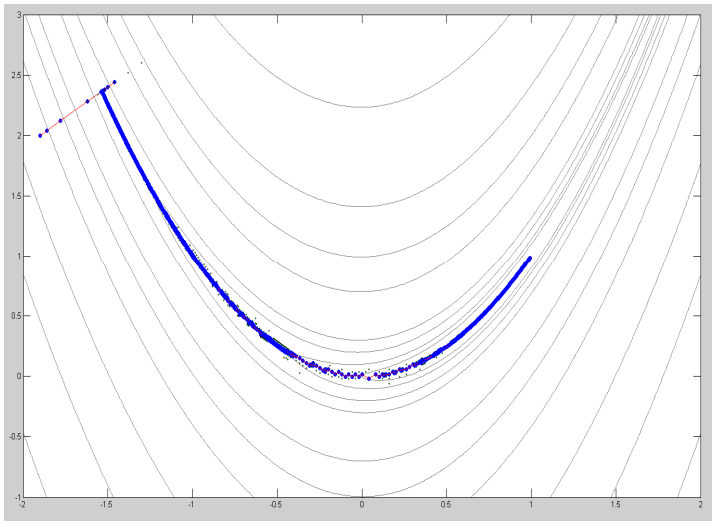

Figure CC: 2D contour plot illustrating the optimization path of Rosenbrock function

The optimization path illustrated above is the result of an optimization we have implemented in MATLAB. The starting point of the optimization scheme is [-1.9, 2]. The Rosenbrock is used to test the implementation of the BFGS algorithm. The BFGS algorithm is implemented as a means to approximate the Hessian matrix. The Rosenbrock is used to compare the Hessian matrices obtained from the BFGS algorithm with the true Hessian which are calculated analytically.

The optimization of the Rosenbrock function is tested with the ensemble optimization (ENOPT) algorithm. The Hessian approximation from the BFGS algorithm and ENOPT gradients are compared with the calculated true Hessian.

Case 1 : EnOpt- Trust Region, Starting Point: [-1 0]

True Hessian @ [0.95, 0.9]:

-725.1193	380.3477
380.3477	200.0000

The Approximate Hessian @ [0.95, 0.9] using the BFGS

-741.1303	409.7534
409.7534	269.9589

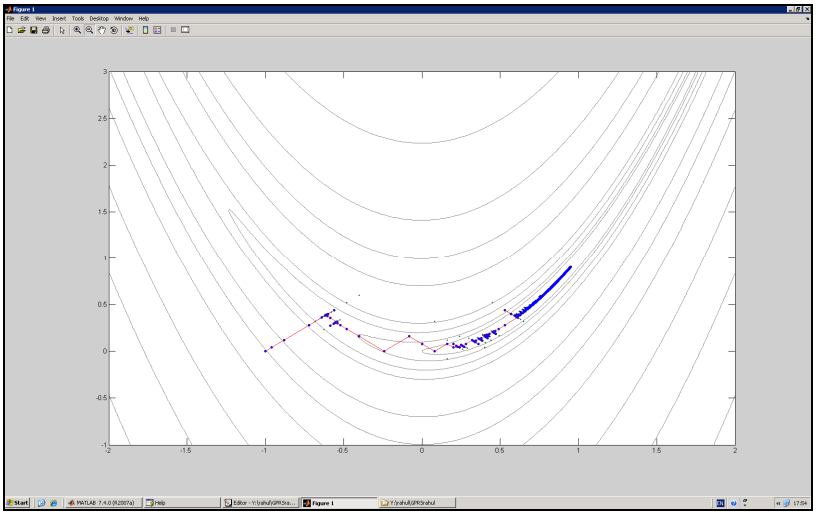

Figure DD: Illustration of the optimization path using trust region

Case 2: EnOpt- Line Search, Starting Point: [-1 0]

True Hessian @ [0.97, 0.95]:

-769.8054	391.7312
391.7312	-200.0000

The Approximate Hessian @ [0.97, 0.95] using the BFGS

-813.5818	410.4587
410.4587	-207.1014

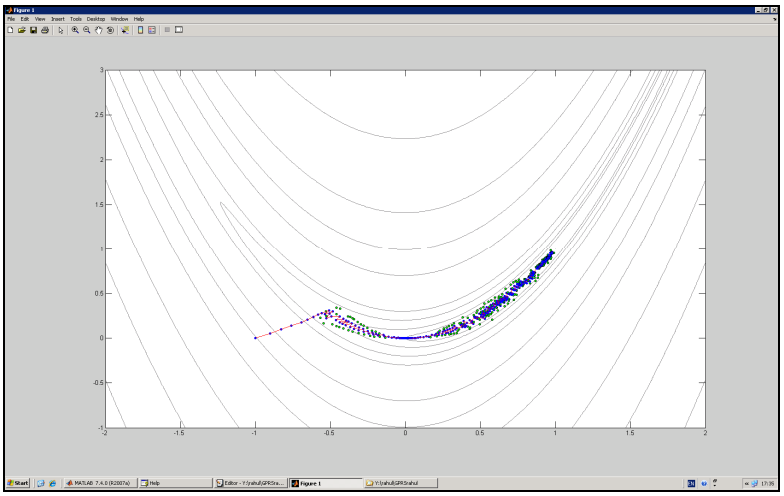

Figure EE: Illustration of optimization same using line search technique

Conclusion:

The approximation to the Hessian using the BFGS algorithm is close to the true Hessian

Tests for Multi-Objective Methods

We first studied the null space & orthonormal basis of the Hessian matrix which is shown below

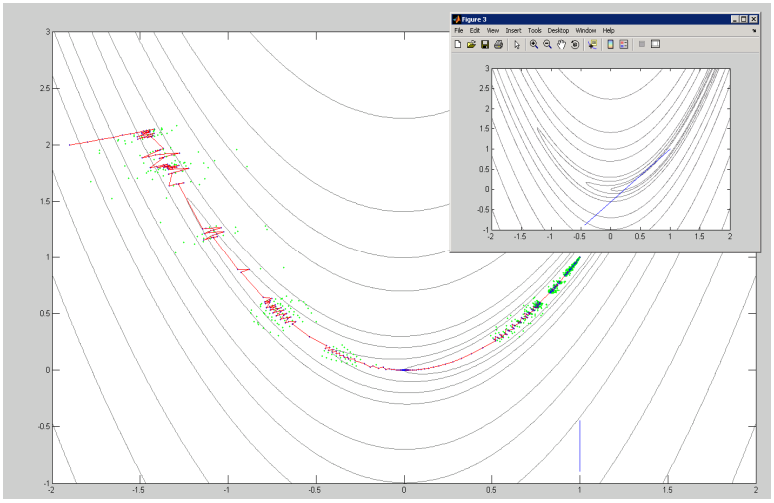

Figure FF: Optimization path for Rosenbrock & inset direction of orthonormal basis indicating null space of Rosenbrock

If we apply a secondary function to be optimized with respect to the Rosenbrock function, then the optimization will follow the blue line of the inset figure (orthonormal basis of the Hessian matrix).

Another interesting study was the direction of the orthonormal basis at different points of the optimization of the Rosenbrock function, which is illustrated below

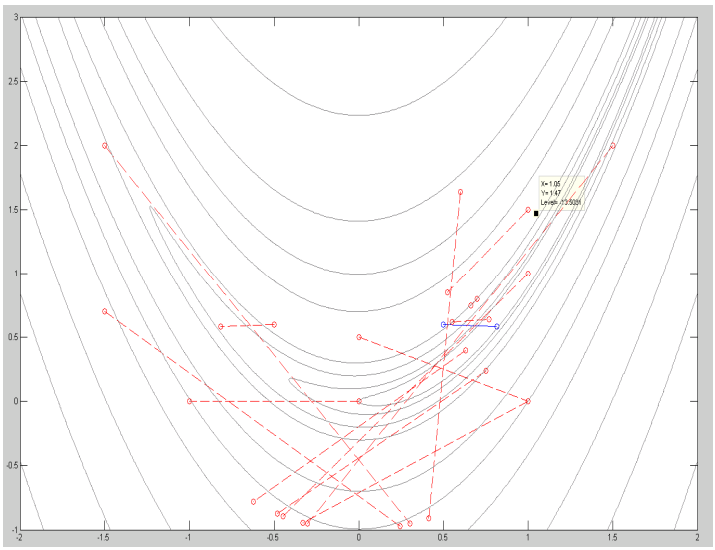

Figure GG: Illustration of orthonormal basis of null space of Hessian calculated at different points to understand the path along which the optimization will proceed

The directions of the orthonormal basis of the null space are evaluated to get an understanding of the projections of the secondary function onto the null space of the primary function**.** This has been plotted to illustrate that the basis of the null space at different points have different directions, this is the reason why the optimization path leaves the high value ridge after a certain point as shown in figure I.

Hierarchical Optimization

Variation1: The null space calculated from the Hessian is updated as described in the algorithm

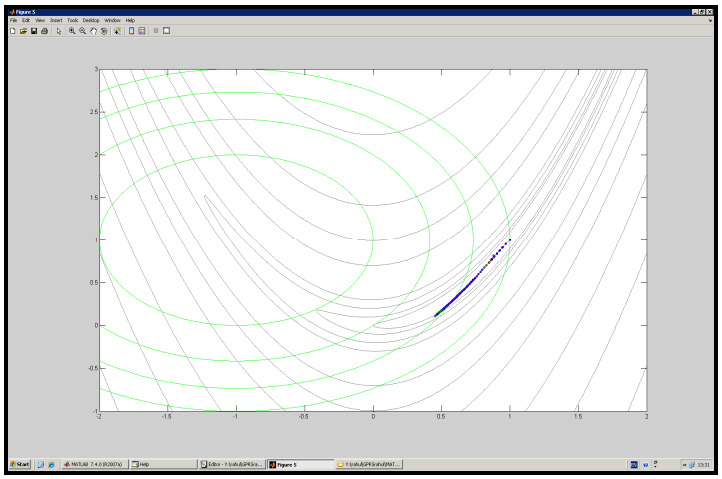

Figure HH: Hierarchical optimization of secondary objective function (green contour lines) with respect to the Rosenbrock function wherein the null space is updated at every iteration

Variation 2: The null space calculated from the Hessian is fixed.

A Singular Value Decomposition (SVD) is done to calculate the null space from the Hessian which has been approximated at the optimum of the primary objective function. This is then used throughout the secondary optimization.

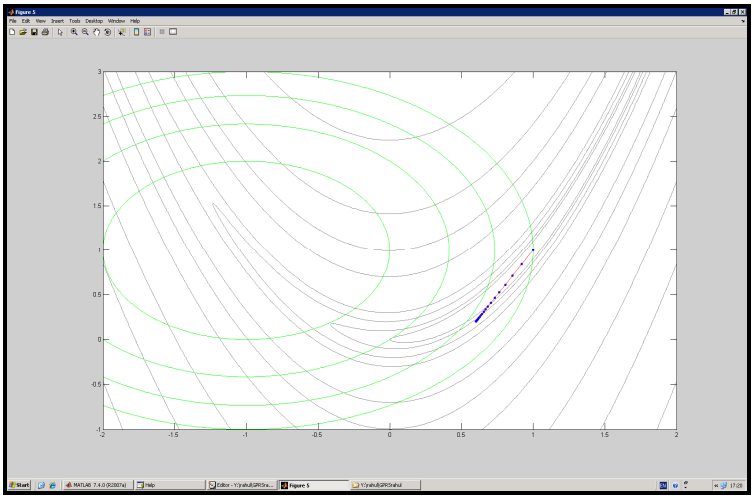

Figure II: Hierarchical optimization when the null space is kept fixed. i.e. the null space has been calculated for the optimum of the Rosenbrock and then not updated during the secondary objective optimization

Note: The variation 2 was implemented to better understand the hierarchical optimization. We observe that we exit the ridge of the Rosenbrock much higher than if we update the Hessian at every iteration. The figures in the variations show above are as a result of trust region based optimization. The figure below is an illustration with the line search method.

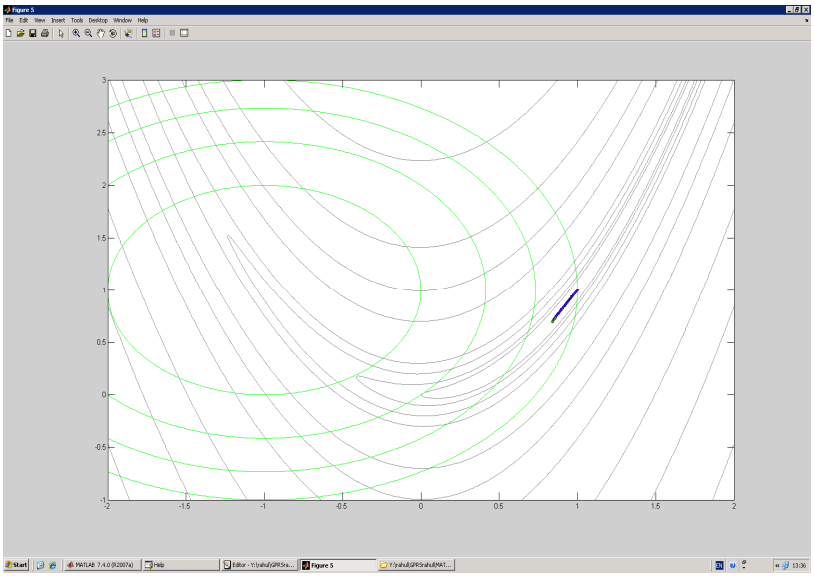

Figure JJ: Hierarchical optimization using the Line Search technique

Test for Switching Algorithm

Two variations of the switching algorithm have been tested on the Rosenbrock function

Variation 1: No projection applied to the gradient of secondary objective function

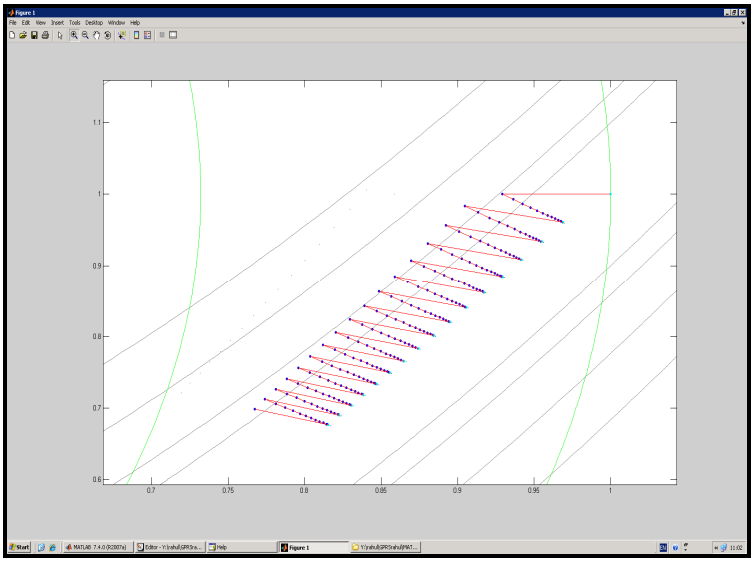

Figure KK: Illustration of switching algorithm optimization

Variation 2: Projection applied to the gradient

In this example we are dealing with a special case for the initial/starting point of the optimization. The gradient at the starting point is zero hence if we apply this to the formula described above we only multiply the gradient of the secondary function by an identity matrix.

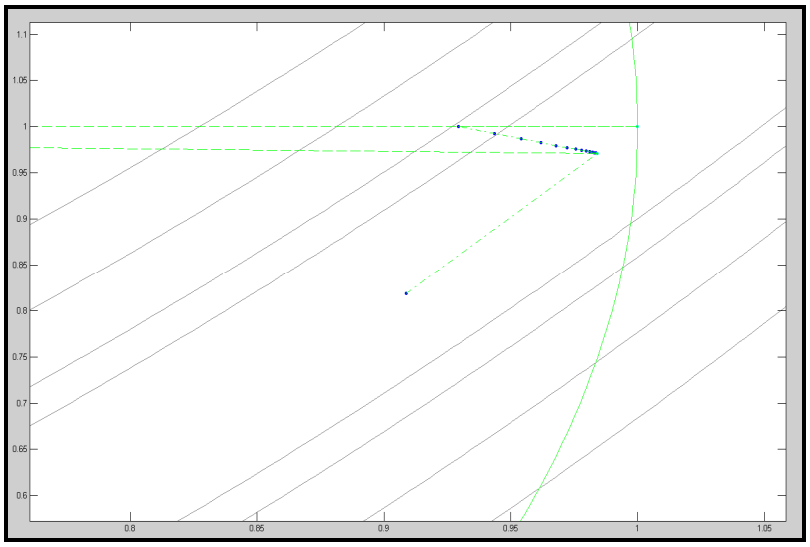

Figure LL: Application to projection to the switching algorithm

Note: However at the second point along the optimization we observe that we have successfully implemented the projection. The figure shows the gradient (dashed green line) and the projected point (dashed Green line to the blue point). The gradients used here are EXACT gradients.

Unconstrained Optimization

How does the constrained optimization results above compare with the unconstrained optimization of the secondary objective. The following figure is a good illustration of the differences between the unconstrained and constrained optimizations.

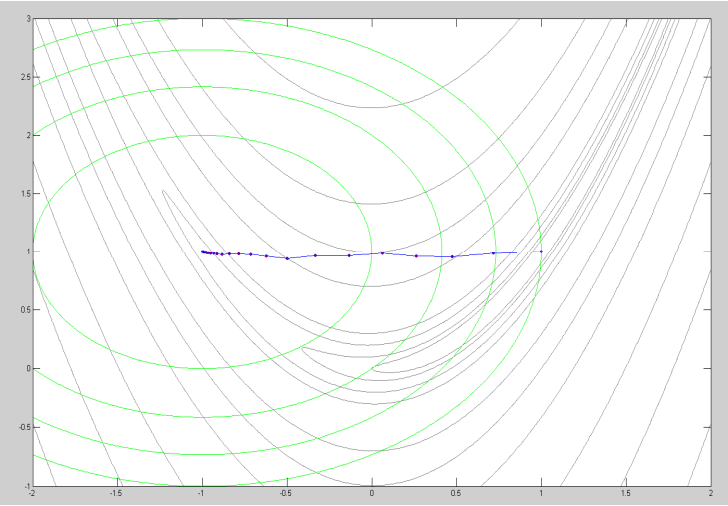

Figure MM: Unconstrained Optimization of the secondary function (green contours) starting from the optimum [1, 1] of the primary Rosenbrock function.

Robust Optimization

Robust optimization has been tested on the Rosenbrock function. A set of Rosenbrock models was created by multiplying the constants in the Rosenbrock equation with stochastically generated multipliers. These multipliers have deviations of 0.15 around 1 i.e. between 0.85 and 1.15.

Case: Robust EnOpt with Trust Region

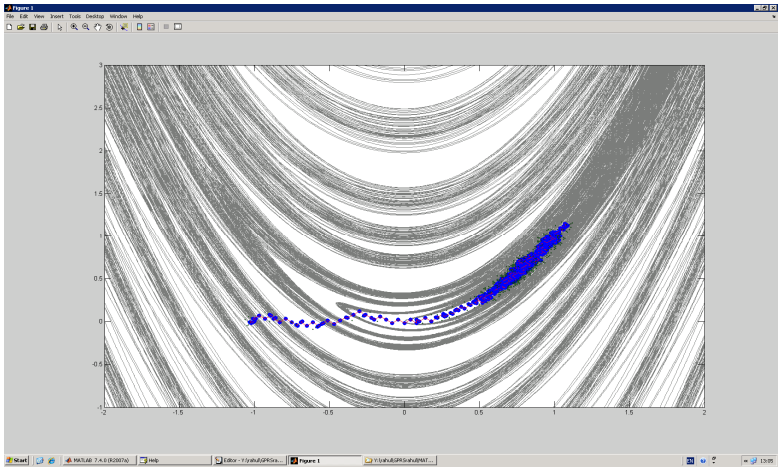

Figure NN: Robust Ensemble optimization using Trust region method

Conclusions

The Rosenbrock test function has been used to test the multi-objective optimization methods and get an understanding of the general optimization code. Hierarchical and switching algorithms have been implemented and tested along with an unconstrained optimization. These results were used as a stepping stone towards applying multi-objective optimization methods to synthetic reservoir models.

Direct Search Techniques

We have implemented 2 different search techniques in our optimization code namely the Line Search technique and the Trust Region method. A brief description of the algorithms for these two methods is provided below. Refer Nocedal & Wright (2006) for a detailed version of these algorithms

Trust Region:

Trust region is a mathematical optimization technique used in various optimization solutions. Starting with a control vector $\mathbf{u_k}$ and knowing the gradient at the control vector the algorithm attempts to find a region around $\mathbf{u_k}$ wherein the linear approximation for the objective function J can be "trusted". This region must be contained within the feasible domain of the objective function. The trust region is thus an area within this feasible domain. When an adequate model of the objective function is found within the trust region then the region is expanded. Conversely, if the approximation is poor then the region is contracted. Trust region methods are also known as **restricted stem methods**.

The trust region method first chooses a step size (the size of the trust region) and then a step direction while line search methods first choose a step direction and then a step size.

Line Search Method:

In optimization, the **line search** technique is one of the basic iterative approaches used in finding a minimum/maximum $\mathbf{u^*}$of an objective function f(u).

The line search approach first finds a descent direction along which the objective function f(u) will be reduced and then computes a step size that decides how far **u** should move along that direction. The update for the control vector can be computed by various methods, such as gradient descent, Newton's method and Quasi-Newton method.

The step size can be determined either exactly or inexactly. We have implemented bound criteria for the step size based on the maximum allowable change in objective function and change in the controls between successive outer iterations. The minimum between the bound criteria is accepted as the step size.

The exact method for determination of the step size is usually computationally expensive thus we have used a common inexact method based on the Armijo rule for the line search. The Armijo rule is one of several inexact line search methods, which guarantees a sufficient degree of accuracy to ensure convergence. For a detailed description of the line search technique and Armijo conditions refer to Nocedal and Wright 2006.