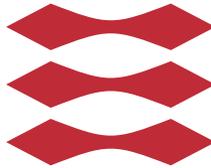


# Stochastic Optimization and Risk Management in the Production Optimization of Oil Reservoirs

Peter Edward Aackermann (s093066)

DTU



Kongens Lyngby 2015

Technical University of Denmark  
Department of Applied Mathematics and Computer Science  
Richard Petersens Plads, building 324,  
2800 Kongens Lyngby, Denmark  
Phone +45 4525 3031  
[compute@compute.dtu.dk](mailto:compute@compute.dtu.dk)  
[www.compute.dtu.dk](http://www.compute.dtu.dk)

# Summary (English)

---

Given uncertainty of oil reservoir properties, such as the permeability field, the net present value (NPV) received from oil reservoir production becomes stochastic. This encourages optimization strategies that focuses on maximizing the expected NPV while minimizing the risk of low outcomes.

The goal of this thesis is to investigate the potential of utilizing such optimization strategies. Especially the focus is on the performance achieved when using Robust Optimization (RO) over an ensemble of 100 reservoir models with a bi-criterion objective function including both Conditional Value at Risk (CVaR) as the risk measure and NPV as the profitability measure. This is compared to more conventional reactive strategies where producers are shut when they are no longer profitable and Certainty Equivalence (CE) optimization that only optimize over the expected reservoir model parameters. In the Mean-CVaR optimization the focus can be shifted between the parameters and create an efficient frontier that shows the level of risk associated with a given NPV which gives more options for a satisfying solution.

For the simulation we will use a oil reservoir with 3 injection wells on one side and 3 producer wells on the other side of the reservoir and simulate over 8 years. The simulations will be done in Matlab using the Reservoir Simulation Toolbox MRST.

The Mean-CVaR optimization greatly outperformed the CE optimization both in terms of expected NPV and CVaR. Compared to the the Reactive Strategy we generated an efficient frontier with up to 2.3% higher average NPV. The CVaR of the Reactive Strategy could however not be fully matched.



# Summary (Danish)

---

Grundet usikkerheder ved oliereservoir egenskaber, som f.eks. permeabiliteten, er den genererede Net Present Value (NPV) fra oliereservoirproduktion stokastisk. Dette betyder at optimeringsstrategier der maksimerer den forventede NPV mens risikoen for lave NPV minimeres.

Målet for denne afhandling er at undersøge potentialet af sådanne optimeringsstrategier. Der fokuseres specielt på effekten opnået ved brug af Robust Optimization (RO) over 100 permeabilitetsfelter med en objektfunktion både indeholdende risiko i form af Conditional Value at Risk (CVaR) og profitabilitet i form af forventet NPV. Denne metode kaldes Mean-CVaR optimering. Mean-CVaR optimering er sammenlignet med mere konventionelle metoder som Reactive Strategy hvor produktionsbrønde er lukket når de ikke længere er profitable. Der undersøges også Certainty Equivalence (CE) optimering hvor der kun optimeres over det forventede oliereservoir. I Mean-CVaR optimeringen er der mulighed for at rykke fokus mellem profitabilitet og risiko og dermed skabe den efficient frontier. Dette giver mulighed for at vælge en injektions profil med den profitabilitet/risiko der er mest fordelagtig.

Til simuleringen bruger vi et oliereservoir med 3 injektionsbrønde på den ene side og 3 produktionsbrønde på den anden side af reservoiret. Produktionen er simuleret over 8 år ved brug af MATLAB Reservoir Simulation Toolbox (MRST).

Vi fandt at Mean-CVaR optimering giver langt bedre resultater end CE i forhold til både NPV og CVaR over de 100 permeabilitetsfelter. Sammenlignet med den Reactive Strategy var vi i stand til at opnå 2,3% højere NPV. Vi kunne dog med Mean-CVaR optimeringen ikke fuldt nå CVaR'en fra Reactive Strategy.



# Preface

---

This thesis was created under the Department of Applied Mathematics and Computer Science (DTU Compute) at the Technical University of Denmark (DTU) in fulfilment of the requirements for acquiring an M.Sc. degree in applied mathematics.

The project was carried out from September 15 2014 to February 16 2015. Under supervision of John Bagterp Jørgensen and Andrea Capolei.

Lyngby, 16-February-2015

A handwritten signature in black ink that reads "Peter Aackermann". The signature is written in a cursive style with a long, sweeping tail on the final letter.

Peter Edward Aackermann (s093066)



# Contents

---

<b>Summary (English)</b>	<b>i</b>
<b>Summary (Danish)</b>	<b>iii</b>
<b>Preface</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Oil Reservoir Model</b>	<b>5</b>
2.1 MATLAB Reservoir Simulation Toolbox (MRST) . . . . .	9
2.1.1 Parallel Implementation . . . . .	12
2.2 Test Case . . . . .	14
<b>3 Optimization Strategies</b>	<b>19</b>
3.1 Profitability Measure (NPV) . . . . .	19
3.2 Risk Measure: CVaR . . . . .	20
3.3 Reactive Strategy . . . . .	22
3.3.1 Test Case . . . . .	22
3.4 Certainty Equivalence Optimization . . . . .	27
3.4.1 Test Case . . . . .	28
3.5 Mean-CVaR Optimization . . . . .	30
3.5.1 Test Case . . . . .	32
3.6 Comparing Optimization Performance . . . . .	40
<b>4 Conclusion</b>	<b>43</b>
<b>Bibliography</b>	<b>45</b>



# List of Figures

---

1.1	Illustration of closed loop reservoir management (CLRM) . . . . .	2
2.1	Logaritmic plot of the 100 permability fields used. The permeability values are in the range 6 mD to 23452 mD. . . . .	7
2.2	Logaritmic plot of the average permability field. Note that the permeability values are only in the range 53 mD to 406 mD. . . . .	8
2.3	Illustration of the well setup, shown in logaritmic plot of permability field 1. . . . .	8
2.4	Illustration of injection, production and well preassures in the oil reservoir over 8 year . . . . .	14
2.5	Illustration of cumulative injection and production in the oil reservoir over 8 year . . . . .	15
2.6	Oil and water saturation in the oil reservoir over time and avearge pressure throughout the oid field. . . . .	15
2.7	Illustration of NPV obtained from the reservoir. . . . .	16
2.8	Oil saturation in the reservoir for different time steps. . . . .	17
3.1	Illustration of injection, production and well preassures in the oil reservoir over 8 year for the Reactive Strategy . . . . .	23
3.2	Illustration of cumulative injection and production in the oil reservoir over 8 year for the Reactive Strategy . . . . .	24
3.3	Oil and water saturation in the oil reservoir over time and avearge pressure throughout the oid field for the Reactive Strategy. . . . .	24
3.4	Illustration of NPV obtained from the reservoir for the Reactive Strategy. . . . .	25
3.5	Oil saturation in the reservoir for different time steps without the Reactive Strategy. . . . .	26

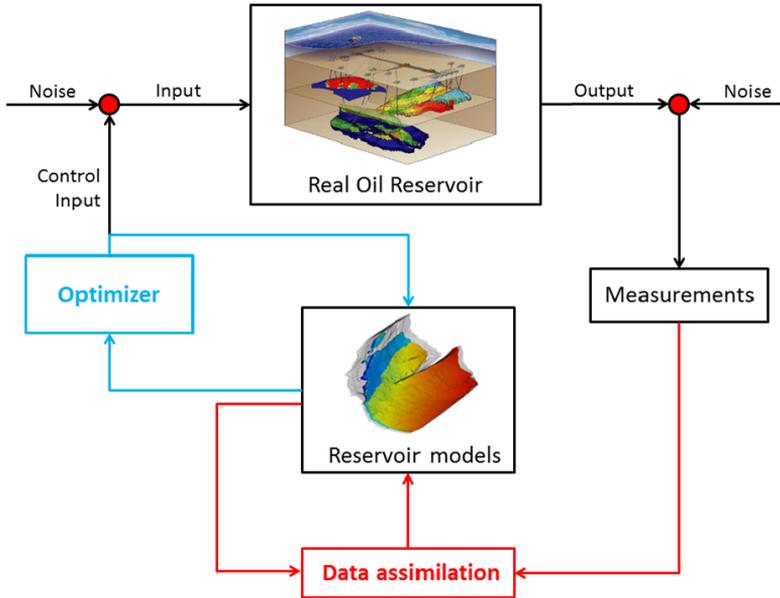
3.6	Oil saturation in the reservoir for different time steps for the Reactive Strategy. . . . .	26
3.7	Illustration of the resulting objective values $NPV_{E[\theta]}$ for the CE optimization plotted against their average performance over the 100 permeability fields $E[NPV_\theta]$ . The values of $N$ used are 1, 2, 4, 8, 16, 32, 50 and 100. The right plot is a zoom of the 4 best performing realizations.. . . .	28
3.8	Optimal injections found using CE optimization. . . . .	29
3.9	Efficient frontier found using MCVaR optimization for $N = 1$ . . . . .	32
3.10	Solutions found using MCVaR optimization for $\lambda = 0.125, 0.375, 0.625$ and $0.875$ for different $N$ values. The optimization was stopped due to the bad results that is why there are only few $\lambda$ values. . . . .	33
3.11	Efficient frontier found using MCVaR optimization for different $N$ values. . . . .	34
3.12	Example convergence plots for $\lambda = 0.0$ and $N = 50$ . . . . .	34
3.13	Efficient frontier found using MCVaR optimization for $N = 50$ . . . . .	35
3.14	Injection scheme for MCVaR 50 for each $\lambda$ value. . . . .	36
3.15	Objective function values for different $\lambda$ using the different MCVaR 50 solutions. . . . .	37
3.16	Efficient frontier found using MCVaR optimization for $N = 50$ and the improved MCVaR 50 frontier with better start guesses. . . . .	38
3.17	Injection scheme for MCVaR 100 for each $\lambda$ value. . . . .	39
3.18	Resulting optimal strategies plotted as a function of $E[NPV_\theta]$ and $CVaR_{5\%}[NPV_\theta]$ . . . . .	41
3.19	Resulting optimal strategies plotted as a function of $E[NPV_\theta]$ and $CVaR_{5\%}[NPV_\theta]$ with added relative runs for CE 100 and MCVaR 100. . . . .	42

# Introduction

---

In the water flooding phase of an oil field it is of high interest to have water injection schemes that maximizes the profitability of the oil field. Conventionally the industry use Reactive Strategies where producers are closed when the water separation cost exceeds the value of the oil recovered. It has been suggested that Optimal control technology and Nonlinear Model Predictive Control (NMPC) can be used to control the injection rates and generate higher profitability, often measured as net present value (NPV) [JBvdH08].

In such closed-loop applications an optimization scheme solves the nonlinear constrained optimal control problem in order to find injection profiles that maximizes a given objective function over the available reservoir models. These injection profiles are then used for the real oil reservoir. Whenever new data about the oil reservoir becomes available (e.g. from oil production or seismic surveys) the measurements are used in a data assimilation process to update the reservoir models and the process starts over again with the updated models. Such procedures are also referred to as closed-loop reservoir management (CLRM). The process is illustrated in Figure 1.1



**Figure 1.1:** Illustration of closed loop reservoir management (CLRM)

In this thesis the focus is on the optimization problem and especially how risk is handled in the objective function. It is an important part of any NMPC application to have an effective optimization routine that chooses appropriate control input based on the available models and since the optimization takes place on a fixed set of models it can be studied independently of data assimilation and real oil reservoirs tests. For this reason there will not be looked into any data assimilation and feedback effects other than that of the Reactive Strategy. Thus the problem investigated can be viewed as an open-loop optimal control problem which can be used as the optimization part of any CLRM application. Furthermore it perfectly resembles the situation when the production of an oil field has not yet begun and no feedback data is available yet.

When modelling physical systems it is always important to account for uncertainty and noise in the available data the model is build upon. This is especially true when modelling oil reservoirs since the data from seismic surveys, core samples and borehole logs are often very sparse and associated with significant noise. This leaves a large range of models that might satisfy the data available. For simplicity in the optimization it is common to use deterministic reservoir models that approximates the uncertainty. One way this can be achieved is by having the reservoir model as the expected parameter data and then maximizing the NPV of that single realization. This method is known as Certainty Equiva-

lence optimization (CE). This however could lead to significant errors compared to the real reservoir and no clear way of handling the risk of low outcomes. Another approach is to have a large ensemble of models that all satisfy the data and optimize the expected NPV over all of them [VEZVdH<sup>+</sup>09],[CSFJ14]. This method is known as Robust Optimization (RO). The RO approach encourages a more direct way of addressing the risk since a chosen injection scheme gives an ensemble of NPV outcomes and measures could be taken to for example reduce variance or increase minimum outcome. The drawback however is the added computational burden of running multiple simulations.

In this thesis both CE and RO are compared to the Reactive Strategy where an ensemble of 100 reservoir models with varying permeability fields are used for the RO and the average permeability field is used for CE. Finally a Mean-CVaR bi-criteria objective function is introduced where the Conditional Value at Risk (CVaR) serves as a risk measure so that the risk is directly addressed in the objective function. This approach is similar to the Mean-Variance bi-criteria objective function introduced in [CSFJ14] but CVaR is chosen instead of Variance due to its better properties as a risk measure as described in [CFJ14].



# Oil Reservoir Model

---

We model the oil reservoir in secondary recovery phase where the oil is pushed to the surface by the pressure of injected water. Modelling oil reservoirs is often associated with significant uncertainty due to the noisy and sparse nature of data obtained through seismic surveys, core samples and borehole logs. We therefore have a large number of uncertain model parameters  $\theta_u$  which complicates simulation of the reservoir. However for our optimal control problem it is desirable to have a deterministic reservoir model since we then have a scalar output for the objective function. The simplest way of achieving this is to have the deterministic model parameters as the expected value of the uncertain model parameters,  $E(\theta_u)$ . By choosing such a model we can however be vulnerable to the uncertainties regarding the parameters because we only look at the expected value. A more robust way of handling the uncertainty would be to discretize the uncertainty space giving a finite set of deterministic parameter values  $\theta = \{\theta_1, \theta_2, \dots, \theta_n\}$ . The NPV of a given injection scheme can then be found for each realization and we get the possibility of shaping the injections to also optimize risk measures such as variance or Conditional Value at Risk (CVaR). We will be using both approaches and compare how their performance compared to the model free Reactive Strategy over  $\theta$  (see Chapter 3).

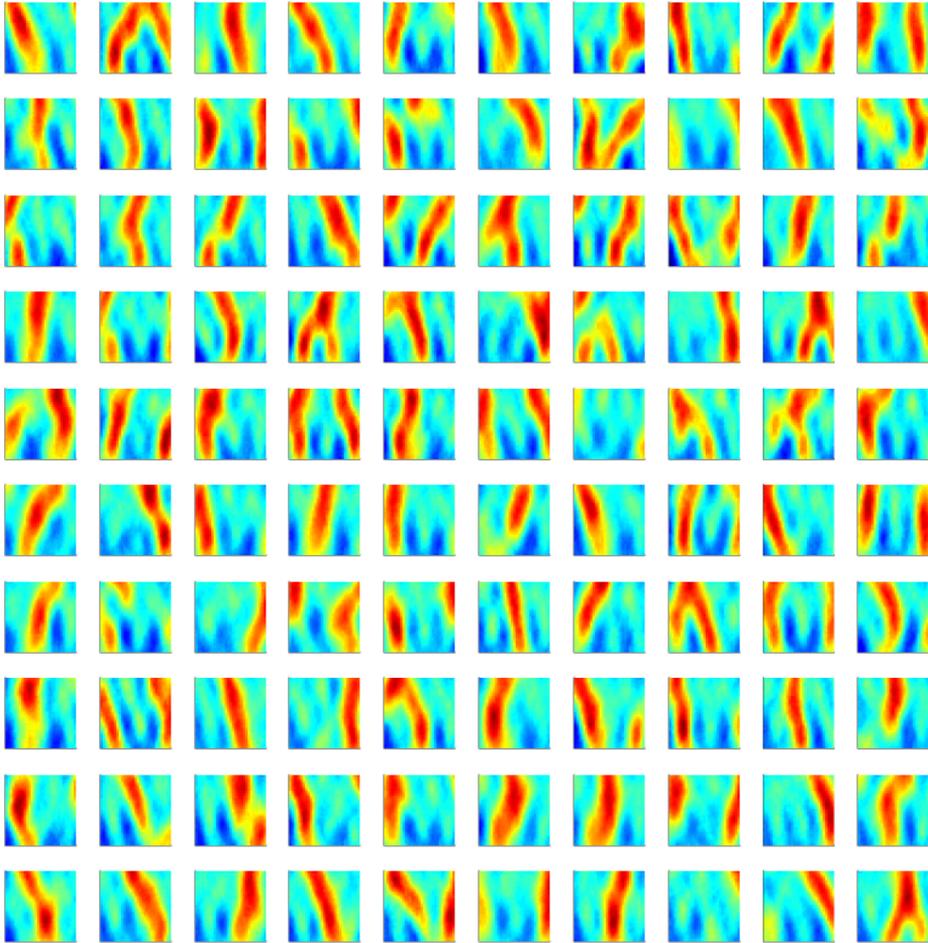
For simplicity we assume that the only uncertain parameters are the permeability field of the reservoir and all other variables are known and fixed for all models. The initial water saturation is 0.15 leaving the initial oil saturation

throughout the reservoir of 0.85. Other known reservoir parameters can be found in Table 2.1.

Symbol	Description	Value	Unit
$\phi$	Porosity	0.3	-
$c_r$	Rock compressibility	$3.0 \cdot 10^{-5}$	$\text{Pa}^{-1}$
$c_w$	Water compressibility	$4.28 \cdot 10^{-5}$	$\text{Pa}^{-1}$
$c_o$	Oil compressibility	$6.65 \cdot 10^{-5}$	$\text{Pa}^{-1}$
$P_{init}$	Initial reservoir pressure	234	atm
$S_{init}$	Initial water saturation	0.15	-
MaxInj	Maximum well injection	800	$m^3/day$
MinInj	Minimum well injection	0	$m^3/day$

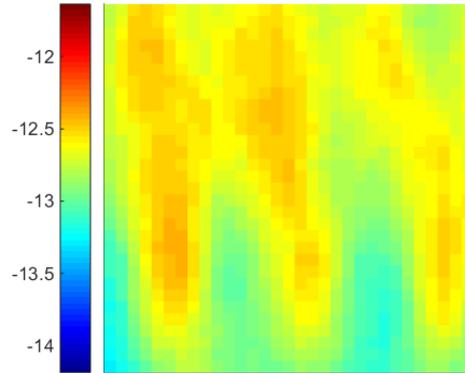
**Table 2.1:** Table of reservoir parameters

For the permeability fields we use an ensemble of 100 realizations of a 2D reservoir in a fluvial depositional environment with a known vertical main-flow direction and permeability values in the range 6 mD to 23452 mD. It is assumed that this ensemble represents the range of possible geological uncertainties. The reservoir size is set to  $620 \text{ m} \times 620 \text{ m} \times 50 \text{ m}$  which is divided into  $31 \times 31 \times 1$  equal sized grid blocks. The 100 permeability fields are illustrated in Figure 2.1.



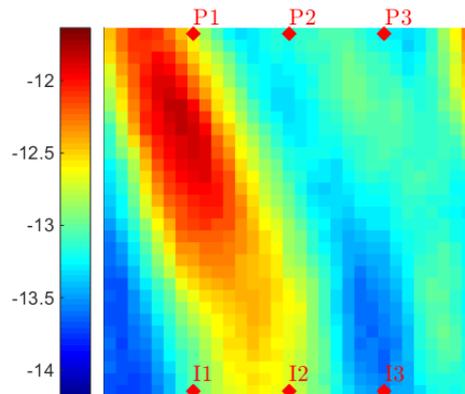
**Figure 2.1:** Logarithmic plot of the 100 permeability fields used. The permeability values are in the range 6 mD to 23452 mD.

For the expected model parameters we use the average of the 100 permeability fields  $E(\theta)$  as shown in Figure 2.2. Note that the expected permeability field looks very different from any of the 100 realisations and only has permeability values in the range 53 mD to 406 mD.



**Figure 2.2:** Logarithmic plot of the average permeability field. Note that the permeability values are only in the range 53 mD to 406 mD.

We use a simple well setup with 3 injection wells and 3 producer wells (denoted I1, I2, I3 and P1, P2 and P3). Due to the vertical main-flow direction they are placed with the injectors along the bottom boundary and the produces at the top boundary as shown in Figure 2.3. It should be noted that in the model flow rates from the injectors are set positive while flow rates from the producers are negative.



**Figure 2.3:** Illustration of the well setup, shown in logarithmic plot of permeability field 1.

The reservoir production is modelled over a time period of 8 years divided into 100 time steps of 30 days each (except in the first 4 time steps where smaller steps are taken to adjust the system to the added pressure from the injection). The manipulated variables are the injection rates over the life of the reservoir

where the maximum water injection is set to  $800 \text{ m}^3/\text{day}$  and the minimum of  $0 \text{ m}^3/\text{day}$ .

We use the MATLAB Reservoir Simulation Toolbox (MRST) for the simulation of the reservoir. For an description of how this is done see Section 2.1, note it is not necessary to read this section to understand the following chapters. It is only given as a guidance to how the system is setup in MATLAB and how to replicate our results using MRST.

## 2.1 MATLAB Reservoir Simulation Toolbox (MRST)

We use MATLAB Reservoir Simulation Toolbox (MRST) to simulate the reservoir over the 8 years. In order to use MRST we first need to activate the toolbox which is done by running the startup script

```
1 run ../mrst-2014a/startup
```

We are then ready to load our reservoir model form the an Eclipse file. For info on how to format the Eclipse file see [Pet06]. We use the MRST function `loadEclipseModel` to load the reservoir model.

```
1 current_dir = fileparts(mfilename('fullpath'));
2 fn = fullfile(current_dir, 'my_simple31x31x1.data');
3 [G, rock, fluid, scheduleEclipse, p0] = loadEclipseModel(fn);
```

Then we setup the system by calculating the geometry , the initial state and the permeability fields of the reservoir by using the MRST functions `computeGeometry` and `initResSol` and by specifying the permeability in `rock.perm`.

```
1 %% Compute constants
2 % Once we are happy with the grid and rock setup, we compute
3 % transmissibilities . For this we first need the centroids.
4 G = computeGeometry(G);
5
6 %% Set up reservoir
7 % We turn on gravity and set up reservoir and scaling factors .
8 gravity on
9
10 state = initResSol(G, p0, [.15, .85]);
11
```

```

12 %% load permeability field
13 load permEns
14 rock.perm = permEns(:,p);

```

The wells are setup using the MRST function `processWellsLocal` and we then set their injections rate and the time periods of the simulation according to the given injection profile.

```

1 %% convert Eclipse schedule to MRST
2 W = processWellsLocal(G, rock, scheduleEclipse.control(1) );
3
4 % store max well capacity (given from eclipse file )
5 maxInje = vertcat(W.val);
6
7 % Get pore volume
8 [OilInPlace, WaterInPlace, poreVol] = GetPoreAndOil(state, G, rock);
9
10 % Update well injects
11 numControl = size(x0,2);
12 schedule.control = [];
13 for i = 1:numControl
14     schedule.control(i).W = W;
15     for j=1:3
16         schedule.control(i).W(j).val = x0(j,i);
17     end
18 end
19
20 % Set time steps
21 numStep = 100;
22 schedule.step.val = ones(numStep,1)*30*day; % timestep: 30*day
23 schedule.step.val(1:4) = [1 ; 4 ; 9 ; 16]*day;
24
25 schedule.step.control = ones(numStep,1);%[1:numControl]';
26 for i = 1:numControl
27     for j = floor((i-1)*numStep/numControl+1) : floor(i*numStep/
28         numControl)
29         schedule.step.control(j)=i;
30     end
31 end

```

Now the oil/water system is setup using the function `initADIsystem` and the flow and pressure equations are solved implicitly using `runScheduleADI`.

```

1 %% Run the whole schedule
2 system = initADISystem({'Oil', 'Water'}, G, rock, fluid);
3
4 %% Use the schedule based on MRST wells
5 timer = tic;
6 [wellSols, states, scheduleOUT, iter, convergence] = ...
7     runScheduleADI(state, G, rock, system, schedule, 'verbose', false);
8 t_forward = toc(timer)

```

The simulation takes approximately 35 seconds to run with the given specifications. After the simulation we can use the MRST function `runAdjointADI` to compute the adjoint gradients for the schedule. This is done using the MRST fully implicit AD solvers and takes approximately 15 seconds to run with our settings. This gives a total simulation time of a single reservoir including calculation of the gradients of approximately 50 seconds.

```

1 %% set Prices
2 prices = {'OilPrice',      126.0 , ...
3          'WaterProductionCost', 19.0 , ...
4          'WaterInjectionCost',  6.0, ...
5          'DiscountFactor',      0.0 };
6
7 %% Adjoint Gradient
8 objective_adjoint = @(tstep)NPVOW(G, wellSols, schedule, ...
9     'ComputePartials', true, 'tStep', tstep, prices {});
10
11 timer = tic;
12 [adjointGradient] = runAdjointADI(G, rock, fluid, schedule,
13     objective_adjoint, ...
14     system, 'Verbose', verbose, 'ForwardStates', states);
15 t_adjoint = toc(timer)
16
17 gradAdj = horzcat(adjointGradient{:});
18 gradAdj = -gradAdj(1:3,:);
19 gradAdj = gradAdj(:);

```

And finally we calculate different KPIs

```

1 %% Calculate KPIs
2 [KPI] = CalKPIs(wellSols, scheduleOUT, states, G, rock);
3 obj = -KPI.NPV.stepCum(end);

```

This is all wrapped in a function `[obj, gradAdj] = SimMRST(x0, p)` that takes as input the injection scheme `x0` and the permeability field number which are to be used and gives as output the total NPV generated by the simulation and the adjoint gradient associated with the injection scheme.

### 2.1.1 Parallel Implementation

In order to optimize over the 100 permeability fields we need a function that simulates the 100 reservoirs in parallel. We do this by creating a new function `[obj, gradAdj] = ParSimMRST(x0, Lambda, numP)` that uses Matlabs `spmd` to run the simulations in parallel with the number of cores specified by `numP`. It gives as output the MCVaR objective function and gradient with the  $\lambda$  value specified by `Lambda`. The function is shown in the following listing.

```

1 function [obj, gradAdj] = ParSimMRST(x0, Lambda, numP)
2
3 iterPrCPU = 100/numP;
4 spmd
5     objP = zeros(1,iterPrCPU);
6     gradAdjP = zeros(length(x0(:)),iterPrCPU);
7
8     for i = 1:iterPrCPU;
9         p = (labindex()-1)*iterPrCPU + i
10        [objP(i), gradAdjP(:,i)] = SimMRST(x0, p);
11    end
12 end
13
14 NPV = [];
15 gradAdjL = [];
16 for i = 1:numP
17     NPV = [NPV objP{i}];
18     gradAdjL = [gradAdjL gradAdjP{i}];
19 end
20
21 AvgNPV = mean(NPV);
22 AvgNPVGrad = mean(gradAdjL,2);
23
24 q = 5;
25 [B,I] = sort(NPV);
26 I = I(end-(q-1):end);
27
28 CVAR = mean(NPV(I));

```

```

29 CVARGrad = mean(gradAdjL(:,I),2);
30
31 obj      = Lambda * AvgNPV + (1-Lambda) * CVAR;
32 gradAdj = Lambda * AvgNPVGrad + (1-Lambda) * CVARGrad;
33
34 gradAdj = reshape(gradAdj,size(x0));
35 end

```

This function can then be used by `fmincon` using anonymous functions to specify `Lambda` and `numP`. We use the HPC cluster at DTU to run the optimization but it could be done on any parallel machine. We use 50 cores which we get access to by editing the `DTUcluster.settings`<sup>1</sup> to 50 workers. We can then run the simulation over `ThinLinc` to the DTU server and use the `DTUcluster` profile. The `fmincon` call along with the options we use are shown in following listing.

```

1  s = 100;
2
3  lb = zeros(3,s);
4  ub = 0.0093*ones(3,s);
5  x0 = ub;
6
7  Lambda = 1;
8
9  numP = 50; %% Has to be multible of 100!! So 2, 4, 10, 20, 25 or 50
10 %c = parcluster('local') % alternative to the DTUcluster
11 c = parcluster('DTUcluster')
12 poolobj = parpool(c,numP)
13
14 options = optimoptions('fmincon','GradObj','on' ...
15     , 'MaxFunEvals',1500,'TolFun',1e-3,'TolX',1e-5, ...
16     'PlotFcns',{@optimplotx,@optimplotfval,@optimplotfunccount,...
17     @optimplotstepsize,@optimplotfirstorderopt});
18
19
20 MCVAR_fun = @(inject)SPMD_Opt_simple(inject, Lambda, numP);
21
22 [x,fval, exitflag ,output,lambda,grad] = ...
23     fmincon(MCVAR_fun, x0, [], [], [], [], lb, ub, [], options);

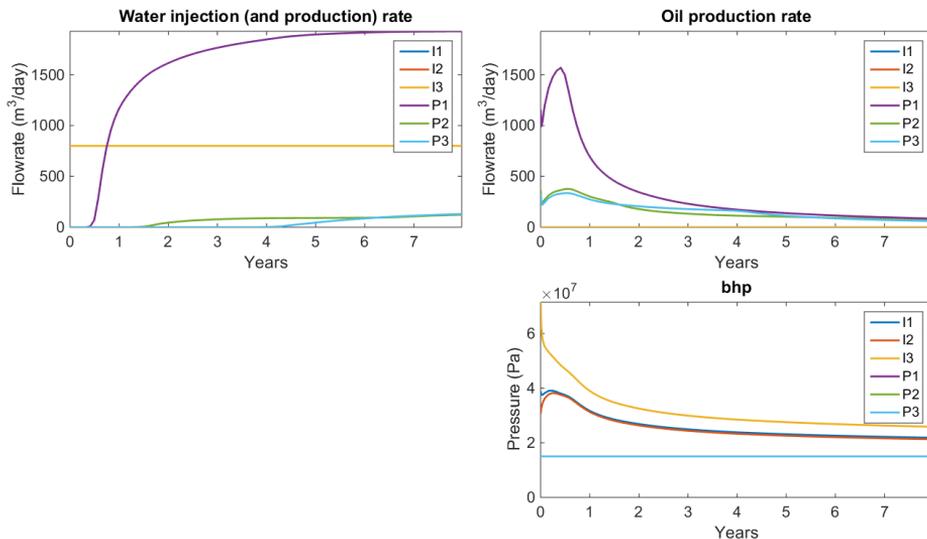
```

<sup>1</sup>the file can be downloaded at: [http://www.hpc.dtu.dk/?page\\_id=1284](http://www.hpc.dtu.dk/?page_id=1284)

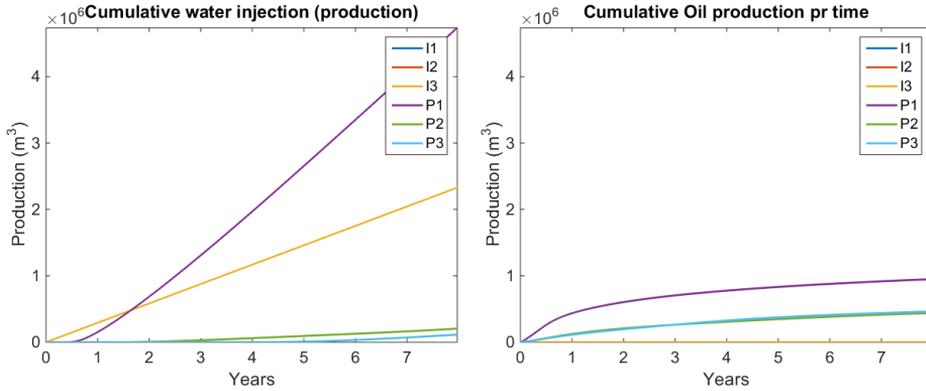
## 2.2 Test Case

In this section we investigate the results obtained when using MRST to simulate the oil reservoir. For this simulation we use the first of the 100 permeability fields (the permeability field, with wells, are shown in Figure 2.3). It should be noted that the permeability field has a channel towards the top left where the fluids can flow easier than in the rest of the reservoir, hence we expect the water to reach P1 quicker than P2 and P3. The 3 injection wells are all set to constantly inject water at maximum capacity ( $800 \text{ m}^3/\text{day}$ ) over the 8 years and the prices used for oil revenue  $r_0$  is  $126 \text{ \$/m}^3$ , water separation cost  $r_w$  is  $19 \text{ \$/m}^3$  and water injection cost  $r_i$  is  $6 \text{ \$/m}^3$

In Figure 2.4 we start by investigating the water injection and oil and water production. As expected we see that a lot of the flow goes to P1, and it doesn't take more than 6 month before we see a substantial rise in the water production from P1. While the oil production decreases from all producers after approximately 6 month the water production continues to rise throughout the period. By looking at the cumulative injections and productions in Figure 2.5 we can see that P1 ends up producing twice as much water than one of the injectors inject. This means that all the water from 2/3 of the injection has gone straight through to P1 and been pulled up again. Also P1 has produced 5 times as much water as oil.

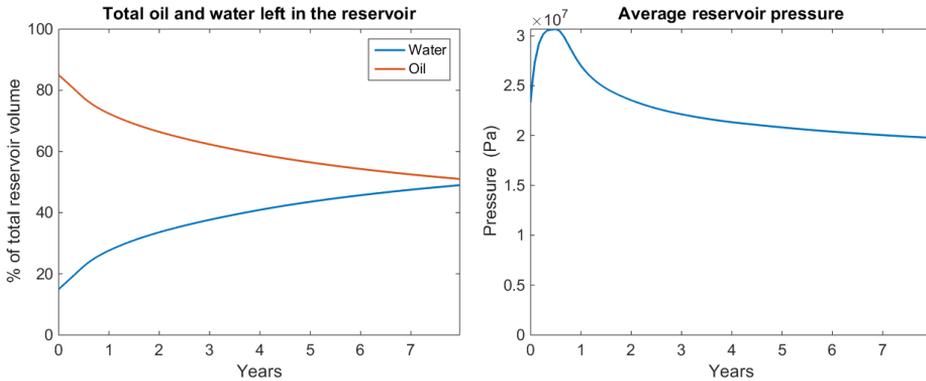


**Figure 2.4:** Illustration of injection, production and well pressures in the oil reservoir over 8 year



**Figure 2.5:** Illustration of cumulative injection and production in the oil reservoir over 8 year

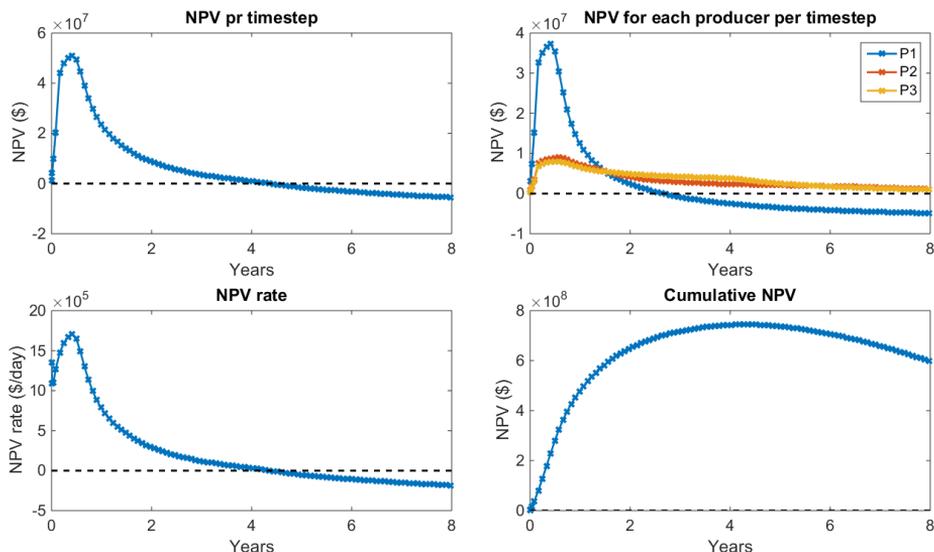
When looking at the total oil and water in the reservoir in Figure 2.6 we see that the oil saturation in the oil field has gone from 0.85 to 0.51 giving a production of 40% of the available oil. But it is clear that the production was largest in the first years and then slowly decreasing which we also found from Figure 2.4.



**Figure 2.6:** Oil and water saturation in the oil reservoir over time and average pressure throughout the oil field.

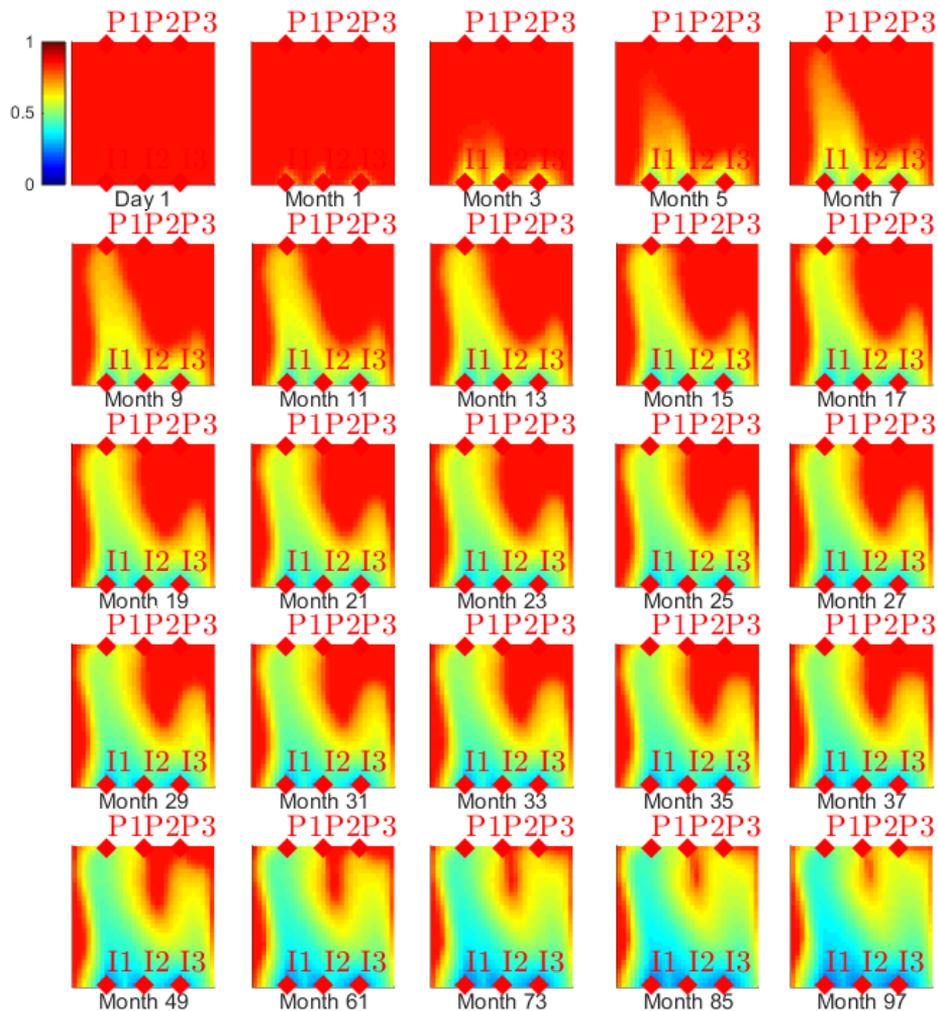
Finally we look at how profitable the production has been in terms of the Net Present Value (NPV) generated by the reservoir (see section 3.1 for calculation of NPV). In Figure 2.7 we see that just as the oil production the NPV is increased a lot in the first 6-12 month after which the rate starts declining rapidly. P2 and P3 does manage to stay profitable throughout the time period while P1 actually starts losing NPV after 2.7 years. After the 4 year the negative NPV generated by P1 is so large that it outweighs the positive NPV from P2 and P3

combined and we see a drop in the cumulative NPV. Here it becomes clear how the Reactive Strategy could improve this scenario by cutting off the produces when they are no longer profitable. We will look more at this in Section 3.3.



**Figure 2.7:** Illustration of NPV obtained from the reservoir.

To fully understand how the water is water is moving through the reservoir we can plot the oil saturation for different time steps as done in Figure 2.8. Here we clearly see how the water is moving through the channel in the left side and quickly finding its way to P1. Actually already after 9 month the oil saturation near P1 has dropped to around 0.6 while it is still at 0.85 near P2, P3 and the whole top right quarter of the reservoir. We do however see the water starting to break through on the right side but in a much slower rate. At the last time step we see that a lot of the oil has been extracted. The remaining oil is mainly at the left boundary of the reservoir, in the top right and in a small pocket near P2. The oil on the left side and top right would properly be very difficult to extract with the current well setup. The pocket next to P2 however is more reasonable to get a hold off. By closing P1 and P3 more water would have to go to P2 and help extracting some more of the remaining oil.



**Figure 2.8:** Oil saturation in the reservoir for different time steps.

We have now shown how the oil reservoir model is set up and how an example simulation could run. In the next chapter we will look into the optimization strategies and how to improve the profitability of the oil reservoir.



# Optimization Strategies

---

In this chapter we look into different strategies to improve the reservoir production both in terms of increasing profitability and minimizing risk and compare how they perform next to the Reactive Strategy.

## 3.1 Profitability Measure (NPV)

When speaking about the profitability of a reservoir it is common to use the Net Present Value (NPV) as the profitability measure [BJ04],[VEZVdH+09],[CSFJ14]. This is intuitive since it does not only account for the amount of oil produced but also the cost of injecting water and separating water from oil after production. The generated NPV at any given time  $t$  can be expressed in the following way

$$NPV(u(t), x(t)) = \frac{-\sum_{j \in \mathcal{P}} (r_o q_{o,j} - r_{wp} q_{wp,j}) - \sum_{l \in \mathcal{I}} r_{wi} q_{wi,l}}{(1+d)^{\frac{\tau(t)}{365}}} \quad (3.1.1)$$

Where the oil price, water separation cost and water injection cost are given by  $r_o$ ,  $r_{wp}$  and  $r_{wi}$  respectively.  $q_{o,j}$  and  $q_{wp,j}$  are the volumetric oil and water flow rate at producer  $j$  and  $q_{wi,l}$  is the volumetric water flow rate at injector  $l$ .  $q_{o,j}$

and  $q_{wp,j}$  and  $q_{wi,l}$  are all functions of the state vector  $x(t)$  and the control input  $u(t)$ . Finally we have the yearly discount factor  $d$  and the time in days  $\tau(t)$ . The discount factor  $(1 + d)^{\frac{-\tau(t)}{365}}$  accounts for a daily compounded value of the capital. Recall that in our model producer flow rates are negative and injection flow rates are positive. This is the reason for the minus in front of the producer sum. In the special case where there are no discounting and no water injection or separation cost ( $d = r_{wp} = r_{wi} = 0$ ) we have that the NPV is equivalent to the quantity of produced oil. For our test we will have water separation and injection costs but we do not account for discounting. Thus (3.1.1) simplifies to the term shown in (3.1.2).

$$NPV(u(t), x(t)) = - \sum_{j \in \mathcal{P}} (r_o q_{o,j} - r_{wp} q_{wp,j}) - \sum_{l \in \mathcal{I}} r_{wi} q_{wi,l} \quad (3.1.2)$$

In table 3.1 are shown the parameters used in this study.

Symbol	Description	Value	Unit
$d$	Discount factor	0	-
$r_o$	Oil Price	126	\$/m <sup>3</sup>
$r_{wp}$	water separation cost	19	\$/m <sup>3</sup>
$r_{wi}$	water injection cost	6	\$/m <sup>3</sup>

**Table 3.1:** Table of economic parameters

When optimizing the production we are interested in maximizing the total NPV generated by the reservoir. The NPV of a given oil reservoir is a function of the control input  $\{u_k\}_{k=0}^{N-1}$ , the reservoir starting conditions  $x_0$  and the used permeability field  $\gamma$ . For simplicity throughout the report we will use following notation when referring to the total NPV of a simulation

$$NPV_\gamma = NPV(\{u_k\}_{k=0}^{N-1}, x_0, \gamma) \quad (3.1.3)$$

Hence  $NPV_{\theta_1}$  is a scalar value representing the total NPV generated when using the first of the 100 permeability fields  $\theta$ ,  $NPV_{E(\theta)}$  is a scalar representing the total NPV generated when using the average of the 100 permeability fields and  $NPV_\theta$  is a vector containing the total NPV generated for each of the 100 permeability fields.

## 3.2 Risk Measure: CVaR

Due to the high uncertainty of the model parameters in oil reservoirs it is highly relevant to look at methods that reduces the risk. In classical Markovitz portfolio

optimization the variance of the portfolio return is used as a measure of the risk of the investment. In a similar way [CSFJ14] introduces a Mean-Variance bi-criterion objective function to find injections that reduce the risk of low NPV outcomes from the oil reservoir. In [CFJ14] it is however shown that the variance of the NPV is not the most appropriate measure of the risk associated with an injection scheme for the reservoir.

For the oil reservoir we are only concerned about the risk of getting low NPVs. The variance however evaluates both tails of the NPV distribution equally. So when minimizing the variance it might as well be the possibility of getting high NPVs that are reduced. Also if our only objective is to minimize the variance a very simple solution come to mind: do not inject any water and do not produce any oil. In that way there will be generated 0\$ NPV independently of the uncertainty parameters and the variance will always be 0. This is of course not an attractive solution. In [CFJ14] they instead find the Conditional Value at Risk (CVaR) to be the most attractive risk measure.

CVaR is also know as Mean Shortfall, Tail VaR and expected tail loss. The CVaR at  $\alpha\%$  ( $CVaR_{\alpha\%}$ ) is calculated as the the expected return in the  $\alpha\%$  worst cases. This means only the lower tail of the distribution is addressed which is exactly what we want. Note by this definition  $CVaR_{100\%}$  is simply the average of the portfolio. Formally CVaR is calculated as

$$CVaR_{\alpha}(NPV) = \frac{1}{\alpha} \int_0^{\alpha} VaR_s(NPV) ds, \quad \alpha \in [0, 1] \quad (3.2.1)$$

Where

$$VaR_{\alpha} = \inf(z | P(NPV > z) \leq 1 - \alpha) \quad (3.2.2)$$

In the special case where we use the  $\alpha$  as a integer fraction of the ensemble realizations, i.e.  $\alpha = \frac{m}{N}$  with  $m = 5$ ,  $N = 100$  our calculation of  $CVaR_{5\%}$  simplifies into the average of the 5 lowest performing realizations. Let  $N\hat{P}V = \{N\hat{P}V_1, N\hat{P}V_2, \dots, N\hat{P}V_{100}\}$  be the NPV of the 100 realizations sorted from smallest to largest then we can calculate the  $CVaR_{5\%}$  by

$$CVaR_{5\%}(NPV) = \frac{1}{5} \sum_{i=1}^5 N\hat{P}V_i \quad (3.2.3)$$

For the reasons described above we will use CVaR as a risk measure with  $\alpha = 5$ . Do however note this method also differs from the variance in that we are interested in maximizing the CVaR in order to reduce the risk of low outcomes.

### 3.3 Reactive Strategy

The Reactive Strategy is very intuitive and differs substantially from the other strategies in this thesis. The principle is that with no prior information about the reservoir, a simple injection scheme is chosen and then producers are shut when they are no longer profitable. This means whenever the water separation cost exceeds the oil revenue of a producer, it is shut and production continues from the remaining producers until they become unprofitable or the time limit is reached. This method has two strong benefits compared to Certainty Equivalence Optimization and Robust Optimization:

- 1) It does not require any mathematical model or optimization to be implemented
- 2) It uses feedback to ensure positive NPV effects

The drawback however is that there are no clever way of picking an injection scheme to optimize the production. For the comparison to the other strategies we have chosen a constant injection scheme where water corresponding to the entire volume of the reservoir is injected over the 8 years. This corresponds to  $658 \text{ m}^3/\text{day}$  pr. injector which is 82% of the total capacity. To illustrate how this strategy affects the production we have in the following replicated the Test Case from Section 2.2.

#### 3.3.1 Test Case

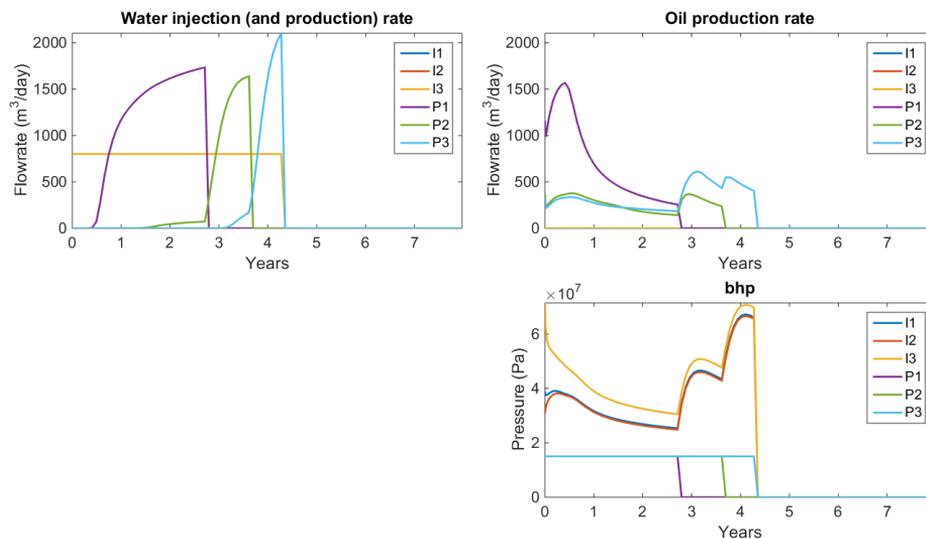
For this test we use exactly the same parameters and injections as in Section 2.2, the only difference is that we now use the Reactive Strategy. The results obtained are shown in Figure 3.1 to 3.6.

Immediately we see a big difference in the productions. P1 is no longer allowed to have a negative NPV rate and is shut off after only 2.7 years. This in turn increases the oil and water production from P2 and P3 and increases the pressure in the system and after only 4.3 years all producers are shut. Looking at the cumulative injections we now see that the oil production are spread a lot more homogeneously between the producers compared to before. Also a lot less water is produced.

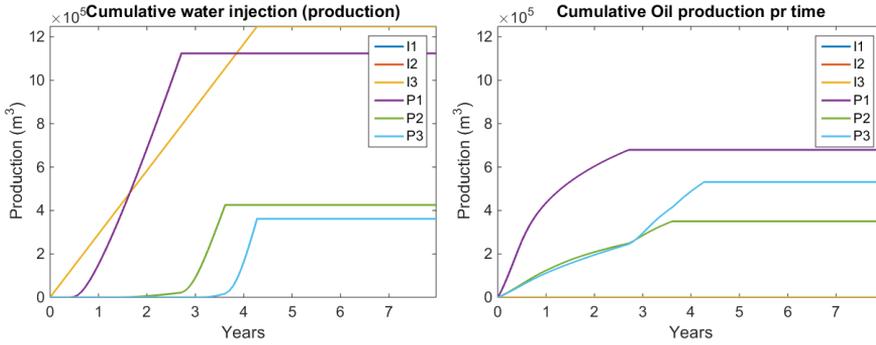
From Figure 3.3 we can see that with the Reactive Strategy more oil are left behind in the reservoir. Now the end average oil saturation is 54% in the

reservoir compared to 51% without the reactive strategy. The rate at which the oil is extracted is however substantially higher since after 4.3 years the oil saturation was at 58%

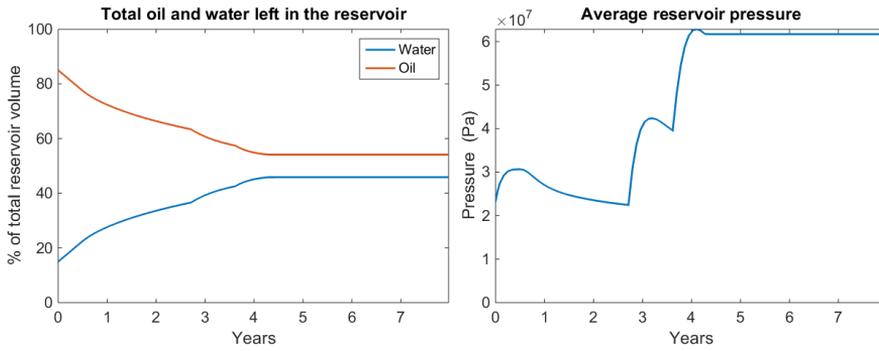
From Figure 3.4 the positive effect of the Reactive Strategy becomes immediately clear. Up until 2.7 years they are exactly the same but now when P1 is shut we actually get a substantial increase in NPV. In total we now generate an NPV of 868 M\$ compared to 598 M\$ before. That is an increase of 45%! This is in part due to the non reactive simulation loosing a lot of NPV at the end of the simulation, but even at it highest it only achieved 744 M\$. This means that the Reactive Strategy yields 16% higher NPV than the non reactive at its highest.



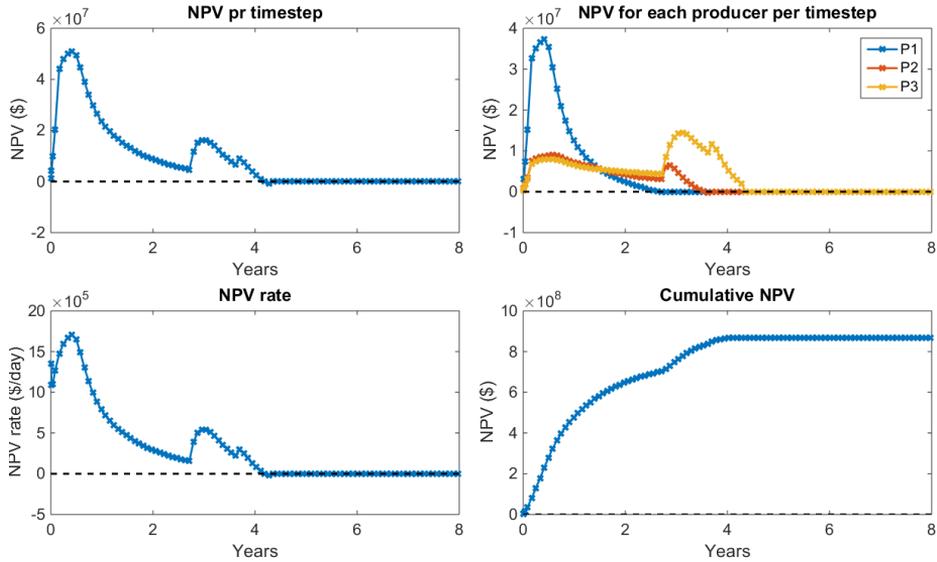
**Figure 3.1:** Illustration of injection, production and well pressures in the oil reservoir over 8 year for the Reactive Strategy



**Figure 3.2:** Illustration of cumulative injection and production in the oil reservoir over 8 year for the Reactive Strategy



**Figure 3.3:** Oil and water saturation in the oil reservoir over time and average pressure throughout the oil field for the Reactive Strategy.



**Figure 3.4:** Illustration of NPV obtained from the reservoir for the Reactive Strategy.

To illustrate how the Reactive Strategy influences the flow through the reservoir we look closer at how the Oil saturation in the reservoir changes with and without the Reactive Strategy. This is illustrated in Figure 3.5 and 3.6. We start from month 30 (2.5 years) since this is about when the first producer is shut and we therefore begin to see differences. It is clear that with the Reactive Strategy the right side of the reservoir is flooded a lot quicker and the oil pocket next to P2 gets drained more. However we notice that since the production is shut after 4.3 years (52 month) and a lot less water is injected we are left with a higher oil saturation near the injectors. This is the reason that less oil is produced with the Reactive Strategy.

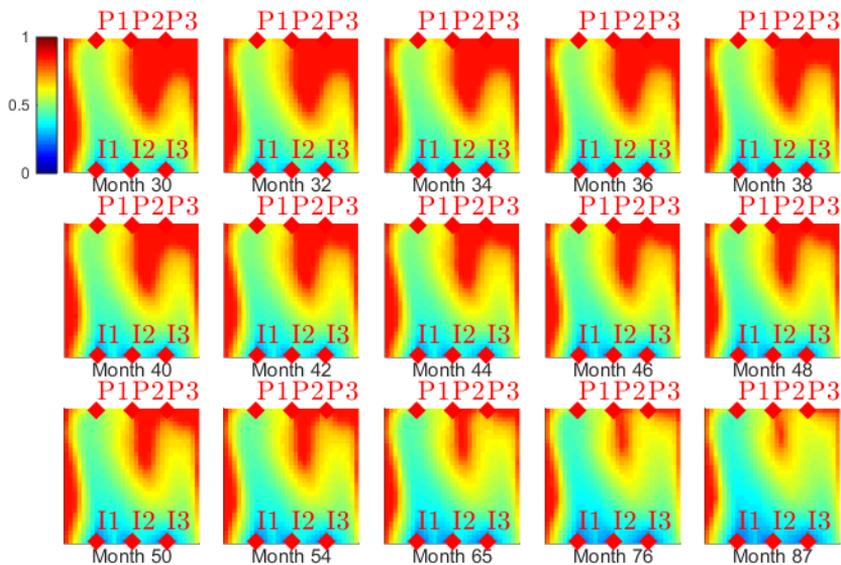


Figure 3.5: Oil saturation in the reservoir for different time steps without the Reactive Strategy.

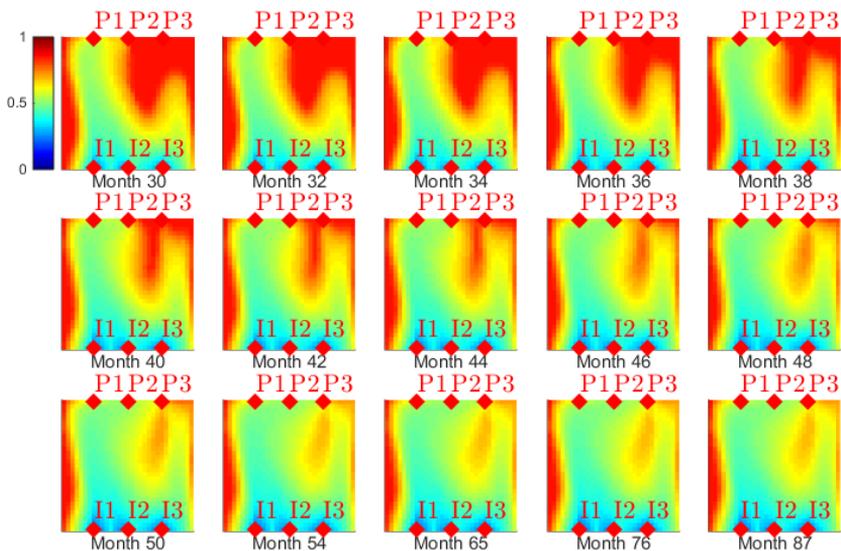


Figure 3.6: Oil saturation in the reservoir for different time steps for the Reactive Strategy.

We have now shown the clear benefits of the Reactive Strategy compared to a simple injection scheme. It does however not give a way to find a more profitable injection scheme. This is what we will look at in the following sections.

### 3.4 Certainty Equivalence Optimization

The Certainty Equivalence (CE) optimization aim to maximise the the NPV over the expected value of the uncertain parameters where the control input is the well injections. The expected permeability field,  $E[\theta]$ , used in this thesis is shown in Figure 2.2. The CE objective function therefore becomes

$$\psi_{\text{CE}} = NPV_{E[\theta]} \quad (3.4.1)$$

and the optimization problem

$$\max_{\{u_k\}_{k=0}^{N-1}} \psi_{\text{CE}} \quad (3.4.2)$$

$$\text{s.t.} \quad \text{MinInj} \leq \{u_k\}_{k=0}^{N-1} \leq \text{MaxInj} \quad (3.4.3)$$

This problem is a nonlinear constrained optimal control problem and it should be noted that due to this nonlinearity we have

$$NPV_{E[\theta]} \neq E[NPV_{\theta}] \quad (3.4.4)$$

This means we have no grantee that a control input that increases the value of the objective function also increases the average NPV over the ensemble of permeability fields. Compared to the Mean-CVaR optimization (see Section 3.5) it does however have the advantage that only a single reservoir simulation has to be made to evaluate the objective function.

To solve the optimization problem we use the MATLAB function `fmincon` with a user supplied gradient, maximum 1500 function evaluations, function value tolerance of  $10^{-3}$  and a step size tolerance of  $10^{-5}$ . The gradient is obtained using the MRST function `runAdjointADI` which computes the adjoint gradients for a schedule using the fully implicit AD solvers. Since `fmincon` finds the minimum of a function we are technically solving the problem

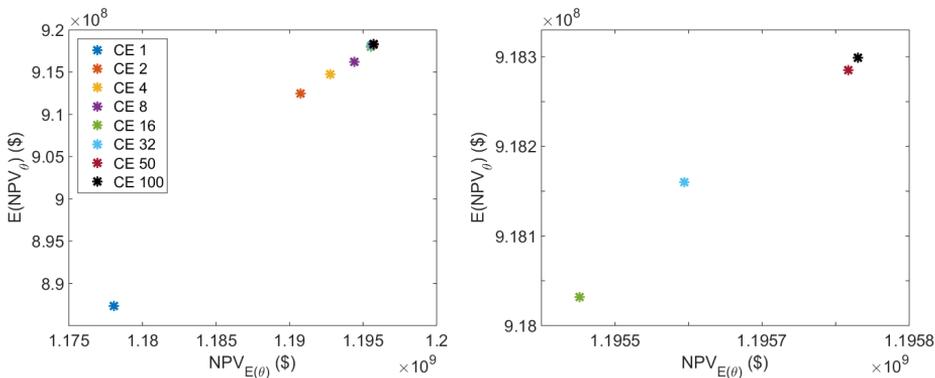
$$\min_{\{u_k\}_{k=0}^{N-1}} -\psi_{\text{CE}} \quad (3.4.5)$$

$$\text{s.t.} \quad \text{MinInj} \leq \{u_k\}_{k=0}^{N-1} \leq \text{MaxInj} \quad (3.4.6)$$

Which is equivalent to (3.4.2).

### 3.4.1 Test Case

For testing the CE strategy we solve the optimal control problem (3.4.5) to find the optimal control input  $\{u_k\}_{k=0}^{N-1}$ . We do this multiple times with different  $N$  values to show how more freedom in changing the injection over time increases profitability. This is also done to slowly increase the complexity of the optimization problem, since for  $N = 1$  the injection from each injector is constant over the 8 years resulting in only 3 variables, while when  $N = 100$  the injection rate can change every 30 days giving 300 variables. We solve (3.4.5) for  $N = 1, 2, 4, 8, 16, 32, 50$  and 100. Each control input is then used for each of the 100 permeability fields ( $\theta$ ) to see how the found injection schemes performs compared to each other. We use maximum constant injection as our starting guess for the optimization. In Figure 3.7 the resulting objective values  $NPV_{E[\theta]}$  for each  $N$  is plotted against their average performance over the 100 permeability fields  $E[NPV_\theta]$ .

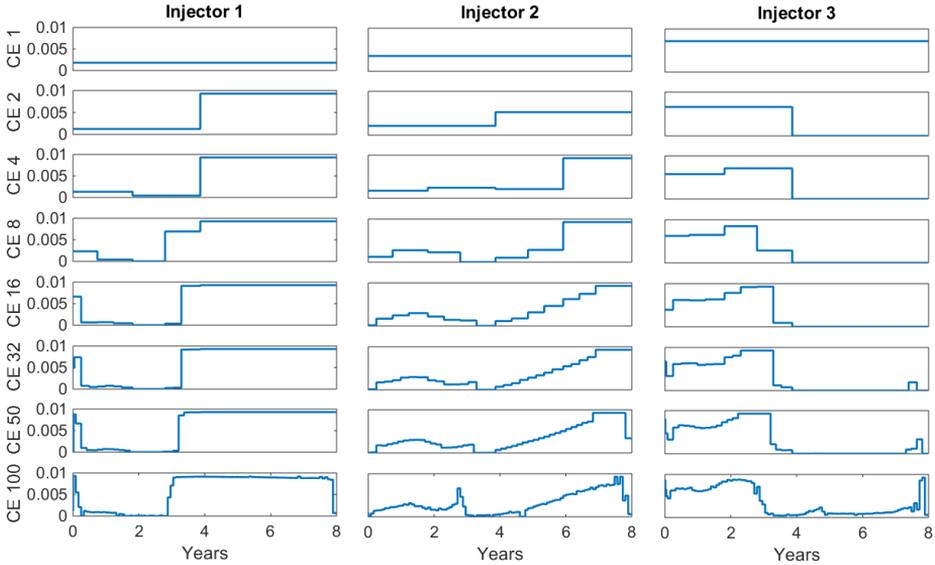


**Figure 3.7:** Illustration of the resulting objective values  $NPV_{E[\theta]}$  for the CE optimization plotted against their average performance over the 100 permeability fields  $E[NPV_\theta]$ . The values of  $N$  used are 1, 2, 4, 8, 16, 32, 50 and 100. The right plot is a zoom of the 4 best performing realizations..

We see that the CE optimization does manage to increase the  $E[NPV_\theta]$  as it finds better solutions to the objective function. It should however be noted that the objective function value  $NPV_{E[\theta]}$  generated far surpasses what the schemes can perform over the 100 permeability fields. In fact an interesting observation is that the data falls on a straight line indicating that there could be a linear relationship between  $NPV_{E[\theta]}$  and  $E[NPV_\theta]$ , this does however seem very unlikely due to the nonlinearity of the problem and it has therefore not been investigated further. Also we clearly see how increasing the freedom of the

injectors gives higher  $E[NPV_\theta]$ . As might be expected the benefits are greatest when increasing from small values of  $N$  and becomes less and less significant to the point where the result for  $N = 50$  and  $N = 100$  are almost identical.

We now look at the injection schemes found in the optimizations. They are shown for each  $N$  in Figure 3.8.



**Figure 3.8:** Optimal injections found using CE optimization.

We see how the injections get more detailed as  $N$  increases and that all solutions look very much alike and that all solutions therefore have found the same optimum.

Finally we look at the computational effort the optimization required. Table 3.2 shows the amount of function evaluations and time taken for each optimization. In total more than 16 hours were needed for all the optimizations.

	Function evaluations	Time taken	$\frac{NPV_{E[\theta]}}{E[NPV_\theta]}$
CE 1	38	0.55 h	1.328
CE 2	34	0.47 h	1.305
CE 4	167	2.36 h	1.304
CE 8	62	0.88 h	1.304
CE 16	190	2.68 h	1.302
CE 32	153	2.19 h	1.302
CE 50	189	2.73 h	1.302
CE 100	332	4.80 h	1.302
Total	1165	16.71 h	-

**Table 3.2:** Table of computational effort needed for the CE optimizations.

### 3.5 Mean-CVaR Optimization

We now introduce the Mean-CVaR (MCVaR) optimization. MCVAR works similar to the Mean-Variance optimization introduced in [CSFJ14] but instead of using variance we use *CVaR* as the risk measure as described in Section 3.2. The fundamental idea is to have a bi-criterion objective function containing both risk and profitability and then have a scalar  $\lambda$  to switch the emphasis on each term. By doing this we can obtain an efficient frontier of profitability vs. risk and have a better foundation for choosing an injection scheme. The objective function for the MCVaR optimization is shown in (3.5.1)

$$\psi_{\text{MCVaR}} = \lambda \cdot E[NPV_\theta] + (1 - \lambda) \cdot \text{CVaR}_{5\%}[NPV_\theta], \quad \lambda \in [0, 1] \quad (3.5.1)$$

And the optimization problem becomes

$$\min_{\{u_k\}_{k=0}^{N-1}} - \psi_{\text{MCVaR}} \quad (3.5.2)$$

$$s.t. \quad \text{MinInj} \leq \{u_k\}_{k=0}^{N-1} \leq \text{MaxInj} \quad (3.5.3)$$

Note that for  $\lambda = 1$  only the average portfolio NPV is maximized (known as robust optimization) and for  $\lambda = 0$  only the CVaR is maximized.

The biggest complication arising using this method is the substantial computational power needed for the optimization. Since we are optimizing over all 100 realizations of the permeability field we have to make 100 simulations for each objective function evaluation. Combining this with multiple optimizations for varying  $\lambda$  (we use 9 different  $\lambda$  values) the problem requires 900 times as much computational power compared to the CE optimization although for only a single  $\lambda$  value the factor is only 100. The good thing however is that the 100

reservoir simulations required in each function call is completely independent and thus can be performed in parallel. For our simulations we therefore utilize the High Performance Computing Cluster at DTU<sup>1</sup>. We run our code in parallel using MATLABs `smvd` functions on the HPC cluster with 50 CPU cores available.

As was the case with the CE optimization we again use the MATLAB function `fmincon` with a user supplied gradient, maximum 1500 function evaluations, function value tolerance of  $10^{-3}$  and a step size tolerance of  $10^{-5}$ . The gradient is obtained by a linear combination of the gradients for the 100 realizations. More accurately, if we let  $\nabla_{u_k} NPV_\theta$  be the ensemble of gradients for each of the 100 realizations and let  $N\hat{P}V = \{N\hat{P}V_1, N\hat{P}V_2, \dots, N\hat{P}V_{100}\}$  be the NPV of the 100 realizations sorted from smallest to largest, we can calculate the gradient of  $\psi_{\text{MCVaR}}$  by

$$\nabla_{u_k} \psi_{\text{MCVaR}} = \frac{\lambda}{100} \sum_{i=1}^{100} [\nabla_{u_k} NPV_{\theta_i}] + \frac{(1-\lambda)}{5} \sum_{i=1}^5 [\nabla_{u_k} N\hat{P}V_j], \quad \lambda \in [0, 1] \quad (3.5.4)$$

In MATLAB we perform this by

```

1 AvgNPV = mean(NPV);
2 AvgNPVGrad = mean(gradAdjL,2);
3
4 q = 5;
5 [B,I] = sort(NPV);
6 I = I(end-(q-1):end);
7
8 CVAR = mean(NPV(I));
9 CVARGrad = mean(gradAdjL(:,I),2);
10
11 obj = Lambda * AvgNPV + (1-Lambda) * CVAR;
12 gradAdj = Lambda * AvgNPVGrad + (1-Lambda) * CVARGrad;
```

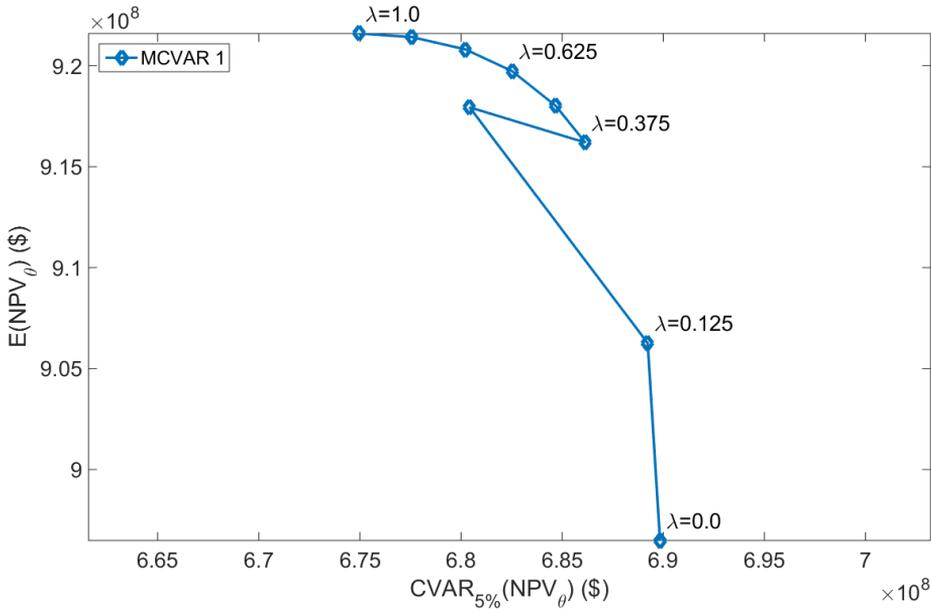
In the case where we use the  $\alpha = 5\%$  and have 100 NPV realizations  $CVaR_{5\%}$  simplifies into the average of the 5 lowest performing realizations. Let  $N\hat{P}V = \{N\hat{P}V_1, N\hat{P}V_2, \dots, N\hat{P}V_{100}\}$  be the NPV of the 100 realizations sorted from smallest to largest then we can calculate the  $CVaR_{5\%}$  by

$$CVaR_{5\%}(NPV) = \frac{1}{5} \sum_{i=1}^5 N\hat{P}V_i \quad (3.5.5)$$

<sup>1</sup>For more information on how to access the cluster go to <http://www.cc.dtu.dk/>

### 3.5.1 Test Case

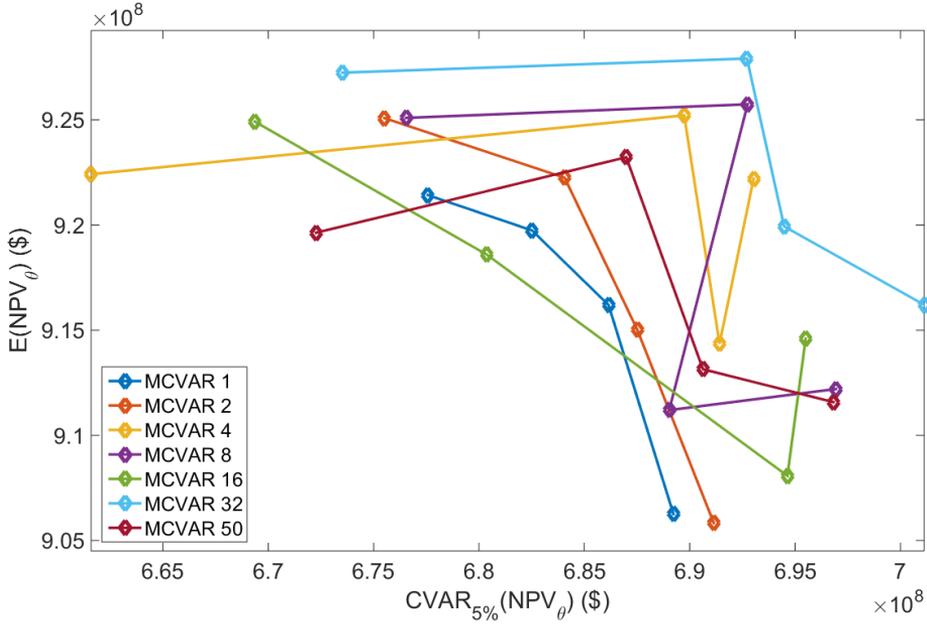
To test the MCVaR optimization we solve the optimal control problem (3.5.2) to find the optimal control input  $\{u_k\}_{k=0}^{N-1}$ . As for the CE optimization we do this for varying  $N$  in order to see the effect of more precise control through the period. We use  $N$  values 1,2,4,8,16,32,50 and 100. Now we also solve the problem for 9 equally spaced  $\lambda$  values between 0 and 1 (0.0, 0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875 and 1.0). We start using maximum constant injection as our starting guess. For  $N = 1$  we get the efficient frontier shown in Figure 3.9.



**Figure 3.9:** Efficient frontier found using MCVaR optimization for  $N = 1$ .

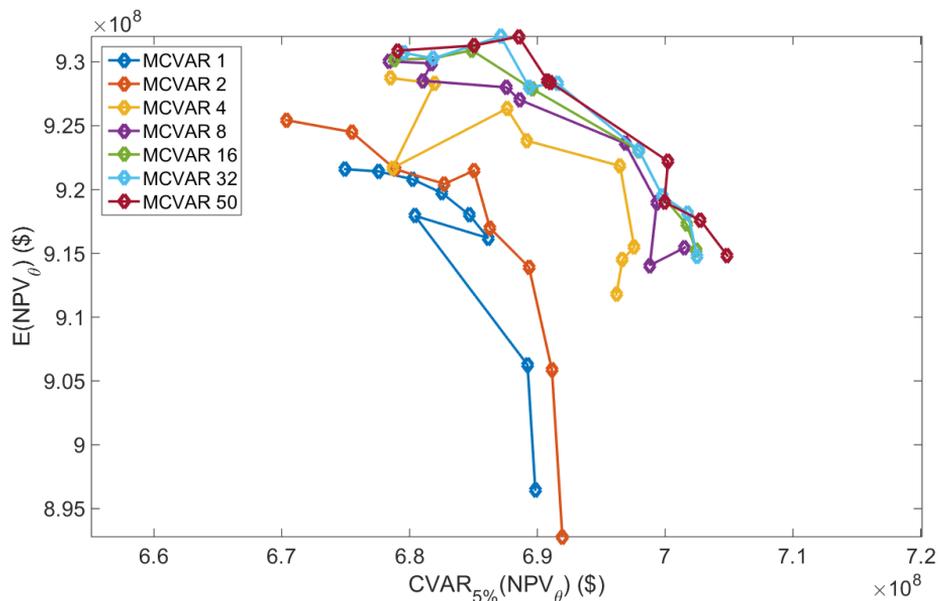
The plot shows the tradeoff between risk (CVaR) and return (expected NPV). We see that it is possible for the optimization to find injection schemes that maximise each term. By choosing  $\lambda = 1$  we achieve 2.8% higher  $E[NPV_\theta]$  than  $\lambda = 0$  while  $\lambda = 0$  has 2.2% higher  $CVaR_{5\%}$ . The frontier looks really smooth except when it comes to the point for  $\lambda = 0.25$ . We know that this solution is not optimal since many of the other found injection schemes would have yielded a higher objective value also for  $\lambda = 0.25$  since they have both higher NPV and CVaR. That the solver was not able to find a better solution is because we are dealing with a highly non-linear problem and thus we cannot be guaranteed to find the global optimum but only a local one.

Not being able to find good optimums also happens when trying to increase  $N$  while keeping the start guess at maximum constant injection. This is illustrated in Figure 3.10.



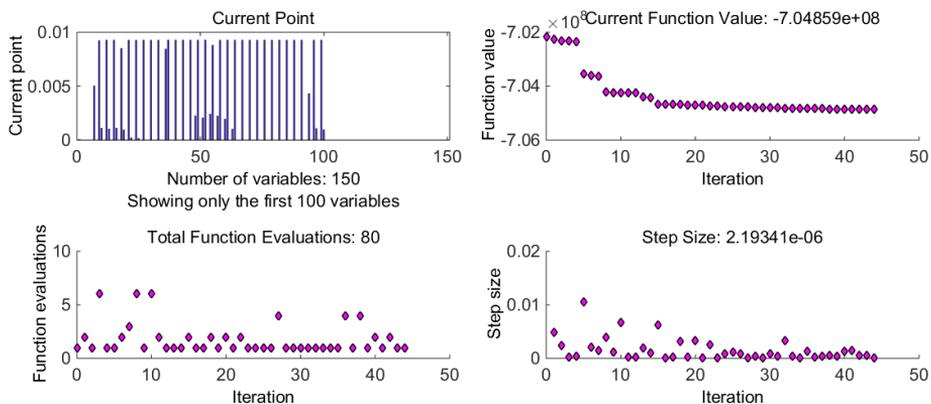
**Figure 3.10:** Solutions found using MCVaR optimization for  $\lambda = 0.125, 0.375, 0.625$  and  $0.875$  for different  $N$  values. The optimization was stopped due to the bad results that is why there are only few  $\lambda$  values.

The solutions found does not seem to make much sense. For instance MCVaR 16 performs worse than MCVaR 1 when  $\lambda = 0.625$  and MCVaR 50 performs worse than MCVaR 32 for all  $\lambda$ . This is again due to the optimizer finding local minimums. This is very undesirable so instead we switch strategy for the start guesses. Instead of a start guess on maximum capacity we use the previously obtained solutions as start guess for the next optimization. This means the injections schemes found by MCVaR 1 is used as start guess for MCVaR 2 and so on. By doing this we obtain the results shown in Figure 3.11.



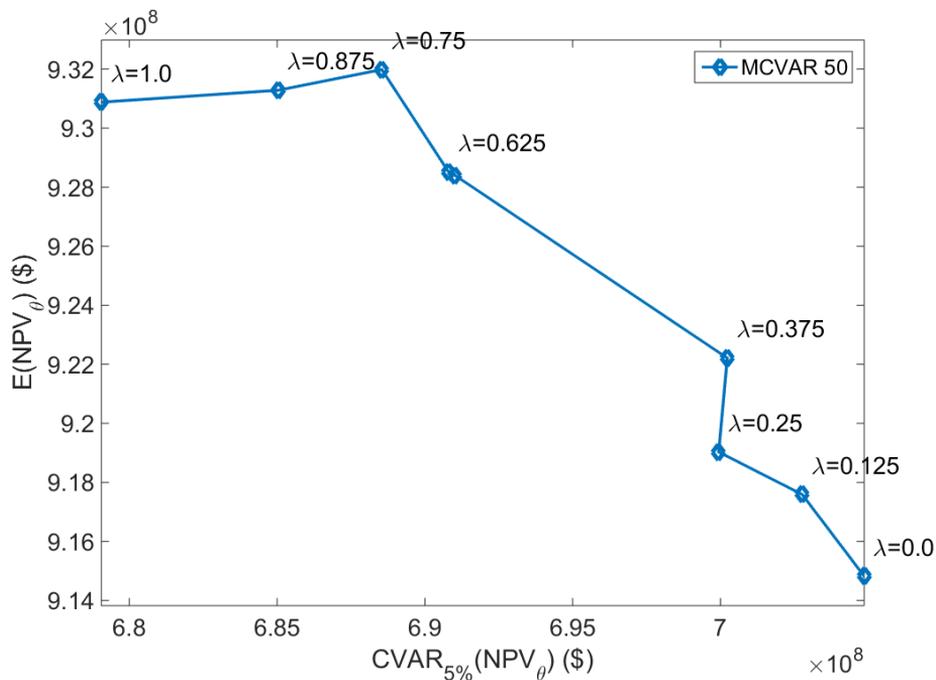
**Figure 3.11:** Efficient frontier found using MCVaR optimization for different  $N$  values.

As shown this greatly helped the optimizer finding appropriate solutions and the frontier improves as  $N$  is increased. There are still occasionally some not so optimal solutions found like for MCVaR 4,  $\lambda = 0.75$  that has improved almost nothing compared to MCVaR 1 and 2. An example convergence plot for the optimization are shown in Figure 3.12



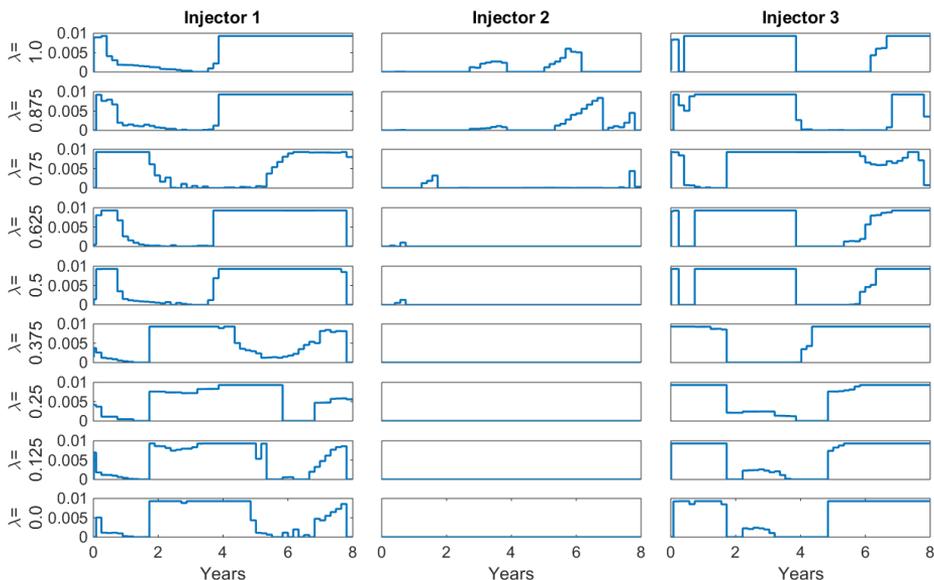
**Figure 3.12:** Example convergence plots for  $\lambda = 0.0$  and  $N = 50$ .

The computations get heavier and heavier as  $N$  increases so we want to make sure we have a good starting guess before solving with  $N = 100$ . In Figure 3.13 we take a closer look at the frontier for  $N = 50$ .



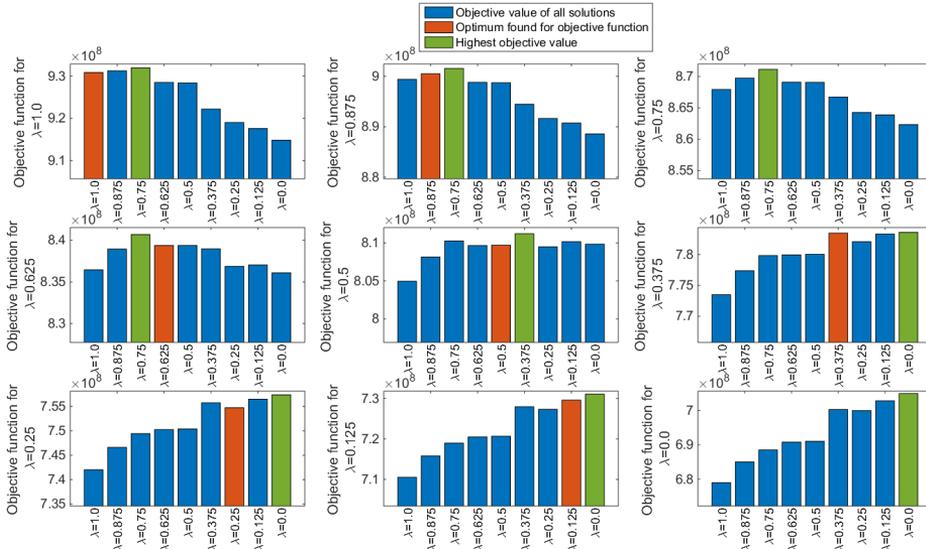
**Figure 3.13:** Efficient frontier found using MCVaR optimization for  $N = 50$ .

It becomes immediately clear that some of the solutions are not optimal. For instance  $\lambda = 0.75$  has higher  $E[NPV_\theta]$  than  $\lambda = 0.875$  and 1.0 and  $\lambda = 0.375$  has higher  $CVaR_{5\%}[NPV_\theta]$  than  $\lambda = 0.375$ . We get more insight by looking at the actual injection schemes as shown in Figure 3.14



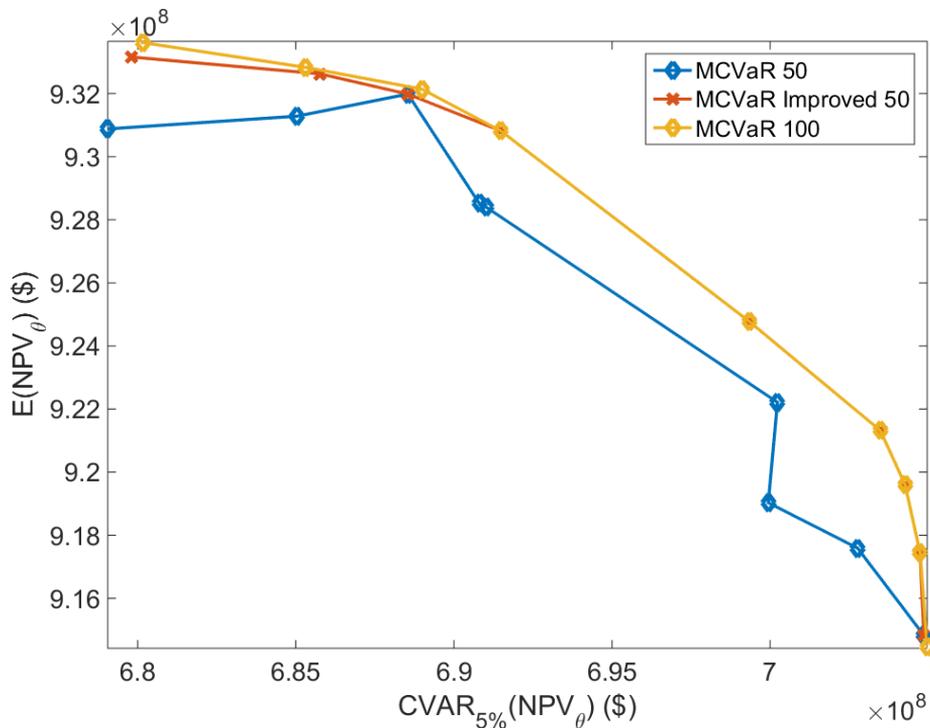
**Figure 3.14:** Injection scheme for MCVaR 50 for each  $\lambda$  value.

It can be seen that there are some solutions that clearly differ from the others. The injection schemes for  $\lambda = 1.0, 0.875, 0.625$  and  $0.5$  lies very close to each other while for  $\lambda = 0.75$  it is a significantly different solution. This indicates that there are several minimums found where we saw from Figure 3.13 that the one for  $\lambda = 0.75$  seems to be the better one. In order to see which injections schemes are good for which  $\lambda$  value we evaluate each injection scheme in the objective function for each  $\lambda$  to see where the highest objective value is found. This is done in Figure 3.15.



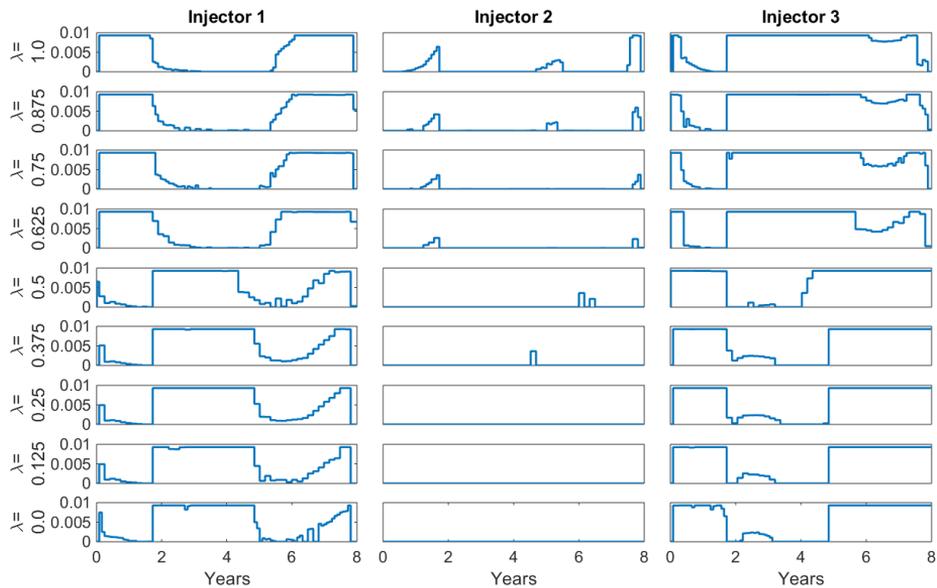
**Figure 3.15:** Objective function values for different  $\lambda$  using the different MC-VaR 50 solutions.

Here we see that for  $\lambda \geq 0.625$  the solution found using  $\lambda = 0.75$  is the best solution and for  $\lambda \leq 0.375$  the solution found using  $\lambda = 0.0$  is best. By solving MCVaR 50 again using these injection schemes as start guess we can improve on the solution as shown in Figure 3.16.



**Figure 3.16:** Efficient frontier found using MCVaR optimization for  $N = 50$  and the improved MCVaR 50 frontier with better start guesses.

We see that the improved start guesses greatly improved the shape of the frontier and gives better performance. Furthermore the efficient frontier for MCVaR 100, which is found by again using the injection scheme from the improved MCVaR 50 as starting guess, keeps the shape we would expect while increasing performance a little bit. In Figure 3.17 we show the resulting injection schemes for MCVaR 100.



**Figure 3.17:** Injection scheme for MCVaR 100 for each  $\lambda$  value.

The injection schemes lie very close to each other for  $\lambda \geq 0.625$  and for  $\lambda \leq 0.5$ . As mentioned earlier this is non-linear optimization so we cannot be certain that the solutions found are globally optimal, but only that it is the best local minimum we have seen so far.

Finally we look at the computational effort required to perform these simulations. The number of function evaluations and time taken is shown in Table [3.3](#)

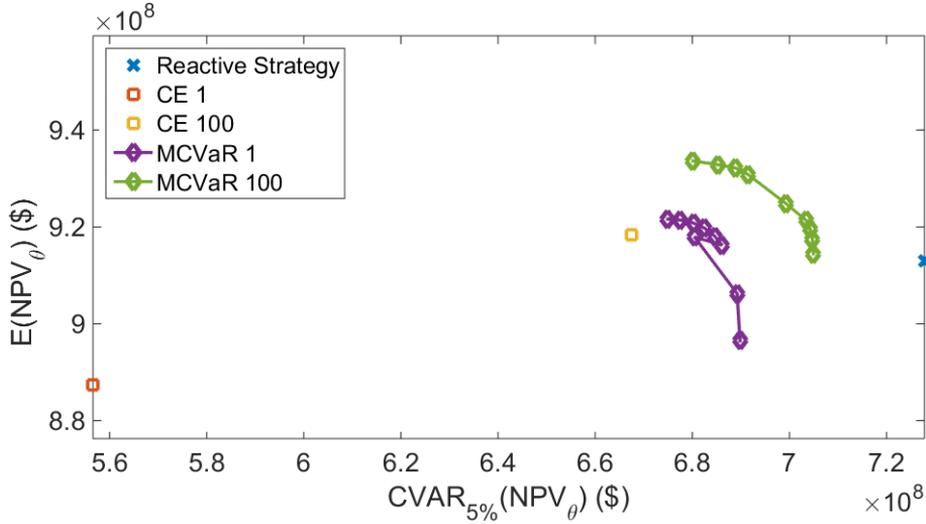
	Function evaluations	Time taken
Average MCVaR 1	28.1	0.76 h
Average MCVaR 2	36.3	1.02 h
Average MCVaR 4	32.0	0.97 h
Average MCVaR 8	44.3	1.29 h
Average MCVaR 16	50.1	1.55 h
Average MCVaR 32	29.2	0.91 h
Average MCVaR 50	55.3	1.58 h
Average MCVaR 100	72.1	2.09 h
Average Total	347.6	10.21 h
Total for all $\lambda$	3128	91.89 h

**Table 3.3:** Table of computational effort needed for the MCVaR optimizations.

It can be seen that the average function evaluations needed for a given  $N$  is significantly lower than for the CE optimization. This might be due to the more intelligently chosen starting points. The time pr. function evaluations is however doubled since we simulate 100 reservoirs using 50 cores instead of 1 reservoir using 1 core. In total the simulation time used to get the results for all  $\lambda$  values are almost 92 hours or equivalent to 3.8 days. Note however that this is when utilizing 50 parallel cores. Without the parallelization the time spent would have been more than 6 months! Hence it can be concluded that performing operations in parallel is crucial for the optimization to be performed.

### 3.6 Comparing Optimization Performance

In this section we look into how the different strategies perform compared to each other. In Figure 3.18 is shown CE 1, CE 100, MCVaR 1, MCVaR 100 and the Reactive Strategy in a return vs. risk plot.

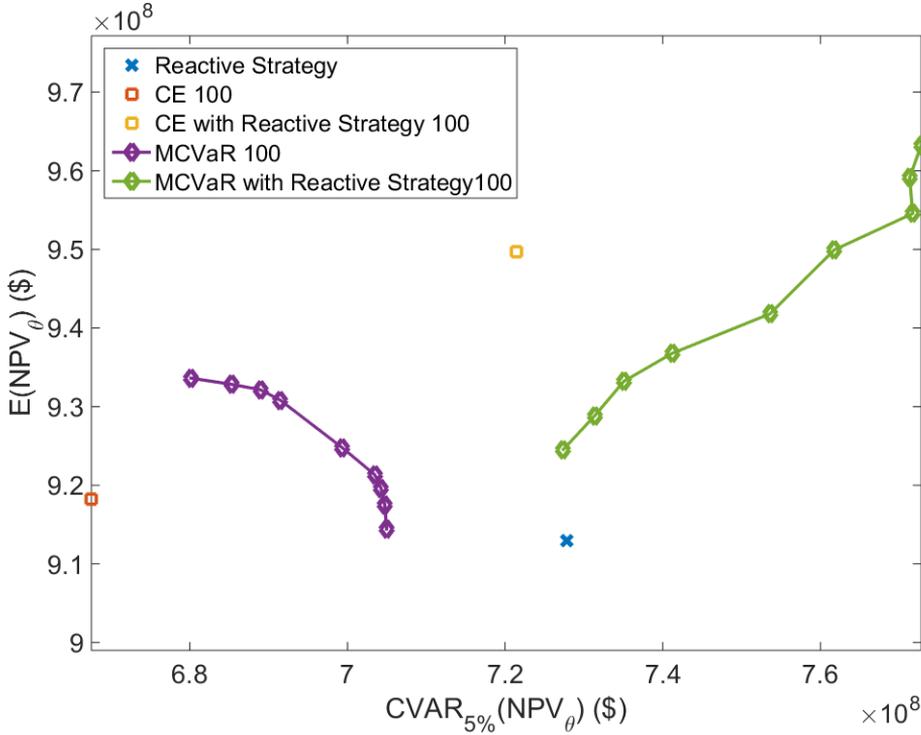


**Figure 3.18:** Resulting optimal strategies plotted as a function of  $E[NPV_\theta]$  and  $CVaR_{5\%}[NPV_\theta]$ .

Here we see that the MCVaR optimization strongly outperforms CE optimization. Even the MCVaR 1 is able to generate both higher  $E[NPV_\theta]$  and CVaR than CE 100. CE 100 does however manage to achieve a 0.6% higher  $E[NPV_\theta]$  than the Reactive Strategy although it comes at the cost of 8.3% lower CVaR. For  $\lambda = 1.0$  MCVaR 100 manages to achieve a 2.3% higher  $E[NPV_\theta]$  than the Reactive Strategy while losing 6.4% CVaR. As expected the MCVaR 100 with  $\lambda = 0.0$  achieves higher CVaR values than for other  $\lambda$  values but it is not enough to reach the Reactive Strategy.

We have shown for this test case that MCVaR optimization is an effective way to improve the average NPV performance while attaining lower risk than CE optimization. We were however not able to reduce the risk as efficiently as the Reactive Strategy.

In practice however it would be very unlikely that this type of reservoir production would be performed without any feedback at all (hence keeping unproductive wells open) as was the case for our MCVaR and CE optimizations. We therefore also investigate what happens if the injection schemes for CE 100 and MCVaR 100 are run with a Reactive Strategy (close wells when not profitable) as shown in Figure 3.19.



**Figure 3.19:** Resulting optimal strategies plotted as a function of  $E[NPV_\theta]$  and  $CVaR_{5\%}[NPV_\theta]$  with added reactive runs for CE 100 and MCVaR 100.

We see that the Reactive Strategy greatly improves the performance of both CE 100 and MCVaR 100. Interestingly the MCVaR for  $\lambda = 1.0$  is by far the superior compared to all other strategies both in terms of  $E[NPV_\theta]$  and CVaR. In fact it increases  $E[NPV_\theta]$  by 5.5% and CVaR with 6.2% compared to the normal Reactive Strategy!

It is expected that even better results could be achieved had we optimized for the best injection scheme while using a Reactive Strategy and not just taking the scheme found and implementing it with a Reactive Strategy. This has however not been further investigated.

# Conclusion

---

In this Thesis, we investigated a Mean-CVaR approach for risk mitigation in an open-loop optimal control problem for oil reservoir production. To our knowledge this has not previously been done in an oil reservoir setting. The control input was chosen as the injection schemes for the wells.

By using MATLAB Reservoir Simulation Toolbox (MRST) and the MATLAB optimization function `fmincon` we were able to demonstrate the effect of the Mean-CVaR approach compared to Certainty Equivalence (CE) optimization and a Reactive Strategy. We found that the Mean-CVaR optimization could significantly reduce the risk compared to CE optimization while also increasing the mean NPV over an ensemble of 100 permeability fields. Compared to the Reactive Strategy we were able to find solutions with as high as 2.3% higher average NPV but at the cost of 6.4% lower CVaR.

Finally we implemented the found control input using a Reactive Strategy and was able to achieve 5.5% higher NPV and 6.2% higher CVaR compared to the Reactive Strategy with a constant injection scheme. These results show the importance of feedback for the performance and encourages future studies to the Mean-CVaR performance in a closed-loop setting with moving horizon. Future studies should also investigate Mean-CVaR optimization for different permeability fields ensembles and different well location and setups in order to have a broader base for validating the approach.



# Bibliography

---

- [BCW05] W. J. Bailey, B. Couet, and D. Wilkinson. Framework for field optimization to maximize asset value. *SPE Reservoir Evaluation and Engineering* 8, 1:7–21, 2005.
- [BJ04] D.R Brouwer and J.D. Jansen. Dynamic optimization of waterflooding with smart wells using optimal control theory. *SPE Journal*, 9(4):391–402, 2004.
- [BJVdH14] E.G.D. Barros, J.D. Jansen, and P.M.J. Van den Hof. Value of information in closed-loop reservoir management. *14th European Conference on Mathematics in Oil Recovery (ECMOR XIV)*, pages 8–11, 2014.
- [Cap13] A. Capolei. *Nonlinear Model Predictive Control for Oil Reservoirs Management*. DTU, Lyngby, DK, 2013.
- [CBW04] B. Couet, R. Burrige, and Wilkinson. Optimization of oil well production with deference to reservoir and financial uncertainty. *US Patent 6,775,578*, 2004.
- [CFJ14] A. Capolei, B. Foss, and J.B. Jørgensen. Profit and risk measures in oil production optimization. *Preprint accepted to 2nd IFAC Workshop on Automatic Control in Offshore Oil and Gas Production.*, 2014.
- [CR09] B. Couet and R. Raghuraman. Tools for decision-making in reservoir risk management. *US Patent 7,512,543*, 2009.

- [CSF13] A. Capole, E. Suwartad, and J. B. Foss, B. Jørgensen. Waterflooding optimization in uncertain geological scenarios. *Computational Geosciences*, 17(6):991–1013, 2013.
- [CSFJ14] A. Capolei, E. Suwartadi, B. Foss, and J.B. Jørgensen. A mean-variance objective for robust production optimization in uncertain geological scenarios. *Preprint submitted to Journal of Petroleum Science and Engineering.*, 2014.
- [FSL<sup>+</sup>14] R.M. Fonseca, A. Stordahl, O. Leeuwenburgh, P.M.J. Van den Hof, and J.D. Jansen. Robust waterflooding optimization of multiple geological scenarios. *ECMOR XIV: Proceedings 14th European Conference on Mathematics in Oil Recovery*, 2014.
- [JBVdH08] J.D. Jansen, O.H. Bosgra, and P.M.J. Van den Hof. Model-based control of multiphase flow in subsurface oil reservoirs. *Journal of Process Control.*, 18:846–855., 2008.
- [MCM<sup>+</sup>14] Juan Miguel Morales González, Antonio J. Conejo, Henrik Madsen, Pierre Pinson, and Marco Zugno. *Integrating Renewables in Electricity Markets: Operational Problems*. Springer, 2014.
- [Pet06] O. Pettersen. *Basics of Reservoir Simulation With the Eclipse Reservoir Simulator*. Dept. of Mathematics, Univ. of Bergen, 2006.
- [SDMJ14] Leo Emil Sokoler, Bernd Dammann, Henrik Madsen, and John Bagterp Jørgensen. A mean-variance criterion for economic model predictive control of stochastic linear systems. *Proceedings of the 53rd IEEE Conference on Decision and Control*, 2014.
- [VEZVdH<sup>+</sup>09] G.M. Van Essen, M.J. Zandvliet, P.M.J. Van den Hof, O.H. Bosgra, and J.D. Jansen. Robust waterflooding optimization of multiple geological scenarios. *SPE Journal* 2009, 14(1):202–210, 2009.
- [VTFE13] D.M. Valladao, R.R. Torrado, B. Flach, and S. Embid. On the stochastic response surface methodology for the determination of the development plan of an oil and gas field. *paper SPE-167446-MS presented at the SPE Middle East Intelligent Energy Conference and Exhibition*, 2013.
- [WBC12] D. Wilkinson, W. Bailey, and B. Couet. Method for consistent valuation of assets with multiple sources of uncertainty. *SPE Economics and Management*, 4(4), 2012.

- 
- [YDA03] B. Yeten, L. J. Durlofsky, and K. Aziz. Optimization of non-conventional well type location and trajectory. *SPE Journal* 8, 3:200–210, 2003.
- [YPK<sup>+</sup>13] E. Yasari, M. R. Pishvaie, F. Khorasheh, K. Salahshoor, and R. Kharrat. Application of multi-criterion robust optimization in water-flooding of oil reservoir. *Journal of Petroleum Science and Engineering*, 2013.