

Ensemble-based hierarchical multi-objective production optimization of smart wells

R. M. Fonseca · O. Leeuwenburgh ·
P. M. J. Van den Hof · J. D. Jansen

Received: 25 February 2013 / Accepted: 30 December 2013 / Published online: 4 February 2014
© Springer Science+Business Media Dordrecht 2014

Abstract In an earlier study, two hierarchical multi-objective methods were suggested to include short-term targets in life-cycle production optimization. However, this earlier study has two limitations: (1) the adjoint formulation is used to obtain gradient information, requiring simulator source code access and an extensive implementation effort, and (2) one of the two proposed methods relies on the Hessian matrix which is obtained by a computationally expensive method. In order to overcome the first of these limitations, we used ensemble-based optimization (EnOpt). EnOpt does not require source code access and is relatively easy to implement. To address the second limitation, we used the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm to obtain an approximation of the Hessian matrix. We performed experiments in which water flood was optimized in a geologically realistic multilayer sector model. The controls were inflow control valve settings at predefined time intervals. Undiscounted net present value (NPV) and highly discounted NPV were the long-term and short-term objective functions used. We obtained an increase of approximately 14 % in the secondary objective for a

decrease of only 0.2–0.5 % in the primary objective. The study demonstrates that ensemble-based hierarchical multi-objective optimization can achieve results of practical value in a computationally efficient manner.

Keywords Ensemble optimization · Multi-objective optimization · Smart wells · Hierarchical optimization · Null-space · BFGS

1 Introduction

There are numerous methods for model-based optimization of hydrocarbon recovery. Some of them use gradient-based techniques, where the gradient information is obtained with the aid of an adjoint formulation; see, e.g., [1, 13, 14], or [9] and references therein. Note that much-earlier applications of the adjoint method in reservoir engineering have been made to compute gradients for computer-assisted history matching; see the text book of [12] for a review and relevant references. The adjoint approach is computationally very efficient but has the disadvantage that it requires access to the simulation code to be implemented. Chen [3, 4] 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 note that an earlier, somewhat different version of EnOpt was proposed by [10], and that there exist other ensemble-based methods such as the simultaneous perturbation stochastic approximation (SPSA; [15]) or the covariance matrix adaptation evolutionary strategy (CMA-ES; [7]). Here, we just use the EnOpt method as proposed by [3, 4], although our results may also be of relevance to other ensemble-based methods. Many recent model-based optimization studies are focused on optimizing

R. M. Fonseca (✉) · J. D. Jansen
Department of Geoscience and Engineering, Delft University of
Technology, Delft, The Netherlands
e-mail: r.m.fonseca@tudelft.nl

J. D. Jansen
e-mail: j.d.jansen@tudelft.nl

O. Leeuwenburgh
TNO, Utrecht, The Netherlands
e-mail: olwijn.leeuwenburgh@tno.nl

P. M. J. Van den Hof
Department of Electrical Engineering, Eindhoven University of
Technology, Eindhoven, The Netherlands
e-mail: p.m.j.vandenhof@tue.nl

over the producing life of the reservoir, as is appropriate, for example, for water flooding strategies that typically aim to prevent early water breakthrough at the production wells. However, operational decisions are generally based on short-term objectives of a project in terms of operational criteria, production contracts, etc., and strategies to optimize such objectives are often in conflict with optimal long-term strategies. Jansen et al. [8] observed that significantly different optimized long-term 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 redundant degrees of freedom (DOFs). Thus, there exist multiple solutions to the optimization problem, and different initial starting points may lead to different solutions in an optimal subset of the decision variable space. A similar observation was made by [12] for optimization problems that appear in computer-assisted history matching which are well known to be very ill-posed. The presence of redundant DOFs formed the basis of the multi-objective optimization approach of [18]. They suggested a hierarchical optimization scheme to include secondary objectives into the life-cycle optimization using the adjoint formulation. They observed a significant increase in short-term objectives with minimal change to the primary objective function, and similar results were obtained by [2, 17]. In this paper, we investigate the applicability of the EnOpt method instead of the adjoint method for multi-objective optimization. We also propose a modification of the hierarchical optimization scheme which improves the computational efficiency of the algorithm. This paper aims to provide a practical and relatively easy to implement alternative to adjoint-based multi-objective optimization. We first provide an overview of the theoretical aspects followed by application to a three-dimensional (3D) reservoir model.

2 Theory

Life-cycle optimization of hydrocarbon recovery 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. Water flooding is our choice of 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 accurately over long time intervals.

Recent improvements in technology have led to an increase in the application of downhole chokes or inflow control valves to regulate flow rates and maintain pressure in the

reservoir. Smart wells, i.e., wells with inflow control valves (ICVs), are important tools to increase oil recovery and delay water production in multi-zone reservoirs. ICVs can also be used to manipulate the stream lines in a single heterogeneous reservoir, thus sweeping the reservoir more efficiently [9]. Thus, it is important to find the optimum value for the settings of the ICVs so as to maximize their potential. In order to improve economic life-cycle performance, dynamic optimization has to be performed over the producing life of the reservoir due to the slowly changing nature of the saturation distribution. We consider a control problem where ICV settings can be manipulated to achieve the best possible objective function. The objective function J , the NPV is defined in the usual fashion as follows:

$$J = \sum_{k=1}^K \left(\frac{[(q_{o,k}) \cdot r_o - (q_{wp,k}) \cdot r_{wp}] - [(q_{wi,k}) \cdot r_{wi}]}{(1+b)^{\frac{t_k}{\tau_r}}} \right) \cdot \Delta t_k \quad (1)$$

where $q_{o,k}$ is the oil production rate in cubic meters per day, $q_{wp,k}$ is the water production rate in cubic meters per day, $q_{wi,k}$ is the water injection rate in cubic meters per day, r_o is the price of oil produced in \$/cubic meter, r_{wp} is the price of water produced in \$/cubic meter, r_{wi} is the price of water injected in \$/cubic meter, 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_r is the reference time for discounting (365 days).

2.1 Ensemble optimization

This technique, as proposed by [3, 4], 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 with respect to the controls. Distinct characteristics of the EnOpt method are [3, 4] as follows:

- The search direction (gradient) is obtained from the ensemble.
- It can be applied to maximize the expected objective function based on multiple geological realizations.
- It is largely independent of simulator specifics and requires minimal code development.
- It has been shown to work with high-dimensional control vectors.

Approximating the gradient from the sensitivity of the ensemble enables the use of any type of control variable without modification to the existing algorithm.

Standard EnOpt uses an ensemble of control vectors to approximate the gradient of the objective function J with respect to the (average) control vector. A single control vector is defined as follows:

$$\mathbf{u} = [u_1 \ u_2 \ \dots \ u_N]^T, \tag{2}$$

where N is the number of control variables (e.g., bottom hole pressures, well rates, or valve settings) which can be rather large. Thus \mathbf{u} is a “super vector” with a number of elements N that may be as large as the number of control time steps times the number of control variables per time step. In standard EnOpt, an ensemble $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_M\}$ is chosen as multivariate Gaussian distributed with a predefined distribution mean $\tilde{\mathbf{u}}$ and a predefined distribution covariance matrix $\tilde{\mathbf{C}}$. During the iterative optimization process, $\tilde{\mathbf{u}}$ is updated until convergence, whereas $\tilde{\mathbf{C}}$ is kept fixed. To approximate the gradient, a mean-shifted ensemble matrix is defined as follows:

$$\bar{\mathbf{U}} = [\mathbf{u}_1 - \bar{\mathbf{u}} \ \mathbf{u}_2 - \bar{\mathbf{u}} \ \dots \ \mathbf{u}_M - \bar{\mathbf{u}}]^T,$$

where

$$\bar{\mathbf{u}} = \frac{1}{M} \sum_{i=1}^M \mathbf{u}_i$$

is the ensemble mean (i.e., the sample mean which is an estimator of the distribution mean $\tilde{\mathbf{u}}$). Similarly, a mean-shifted objective function vector is defined as follows:

$$\mathbf{j} = [J_1 - \bar{J} \ J_2 - \bar{J} \ \dots \ J_M - \bar{J}]^T,$$

where values J_i correspond to the simulated response to control vectors \mathbf{u}_i and where

$$\bar{J} = \frac{1}{M} \sum_{i=1}^M J_i.$$

If we would have an overdetermined case, i.e., for $M > N$, the approximate gradient with respect to the controls could be obtained as a least-squares solution

$$\mathbf{g} = (\bar{\mathbf{U}}^T \bar{\mathbf{U}})^{-1} \bar{\mathbf{U}}^T \mathbf{j}, \tag{3}$$

where in a practical implementation, we would solve a linear system of equations for \mathbf{g} rather than compute the inverse. For the derivation of Eq. 3, see any introductory linear algebra textbook, e.g., [16]. Equation 3 can also be expressed as follows:

$$\mathbf{g} = \mathbf{C}_{uu}^{-1} \mathbf{C}_{uJ}, \tag{4}$$

where

$$\mathbf{C}_{uu} = \frac{1}{M-1} (\bar{\mathbf{U}}^T \bar{\mathbf{U}}) \text{ and } \mathbf{C}_{uJ} = \frac{1}{M-1} (\bar{\mathbf{U}}^T \mathbf{j})$$

are ensemble (sample) covariance and cross-covariance matrices, respectively [3, 4]. Normally, however, we have an under-determined case, i.e., $M < N$. This implies that the

matrix product $\bar{\mathbf{U}}^T \bar{\mathbf{U}}$ is rank-deficient such that we cannot directly compute its inverse or solve the associated system of equations. Instead, we can compute the Moore-Penrose pseudo inverse using a singular value decomposition (SVD), if necessary with additional regularization; see, e.g., [16]. Alternatively, [3, 4] propose to simply use

$$\mathbf{g}' = \mathbf{C}_{uJ} = \mathbf{C}_{uu} \mathbf{g} \tag{5}$$

instead of \mathbf{g} , i.e., they approximate the gradient by the ensemble cross covariance \mathbf{C}_{uJ} . Moreover, they propose to use a second premultiplication with \mathbf{C}_{uu} as a preconditioning step which leads to

$$\mathbf{g}'' = \mathbf{C}_{uu} \mathbf{C}_{uJ} = \mathbf{C}_{uu} \mathbf{C}_{uu} \mathbf{g}. \tag{6}$$

Both expressions (5) and (6) can be interpreted as modified (approximate) gradients. In our study, we applied Eq. (3) with an SVD. Note that the EnOpt method requires a pooling step, i.e., the creation of a new ensemble, at every iteration and running the corresponding M forward simulations, which is why the method is computationally intensive.

2.1.1 Update rules

The approximate gradient \mathbf{g} from Eq. (3) can be used in any gradient-based optimization algorithm. In our study, we used a simple steepest-ascent scheme according to

$$\tilde{\mathbf{u}}^{\ell+1} = \tilde{\mathbf{u}}^\ell + \alpha^\ell \mathbf{g}^\ell, \tag{7}$$

where the superscript ℓ is the iteration counter, and α^ℓ is a step length in the direction of the gradient. We determined α^ℓ with the aid of an inexact line search and the Arjimo conditions [11]. We note that there is a scope to improve the computational efficiency of the optimization through the use of quasi-Newton methods, see, e.g., [5].

2.2 Hierarchical multi-objective optimization

The process of optimizing systematically and simultaneously a collection of objective functions is called multi-objective optimization. There are various methods of multi-objective optimization such as the weighted sum method, Pareto optimality, goal programming, etc. [18] introduced a hierarchical optimization scheme to solve the multi-objective production optimization problem. They argued that the weighted sum method (in which each objective function is assigned a weight factor) suffers from an arbitrariness in choosing the weight factors and therefore preferred a hierarchical method (in which the second objective is optimized while keeping the optimized value of the first objective (almost) fixed). They proposed two hierarchical methods in combination with the adjoint formulation, which are explained below. We have, in this work, used those

hierarchical methods to investigate their applicability in combination with the EnOpt method.

2.2.1 Null-space-based optimization

Van Essen et al. [18] introduced a hierarchical optimization scheme to achieve multi-objective production optimization, which prioritizes the objective functions. The optimization of the secondary objective function J_2 is constrained by a maximum allowable change in the primary objective function. Thus, the primary objective function J_1 will remain close to its optimal value. A general formulation for hierarchical optimization is as follows:

$$\begin{aligned} & \max_{\mathbf{u}_{1:K}} J_2(\mathbf{u}_{1:K}), \\ & \text{s.t. } \mathbf{f}_{k+1}(\mathbf{u}_k, \mathbf{x}_k, \mathbf{x}_{k+1}) = \mathbf{0}, k = 0, \dots, K-1, \\ & \mathbf{c}_{k+1}(\mathbf{u}_{k+1}, \mathbf{x}_{k+1}) \leq \mathbf{0}, k = 0, \dots, K-1, \\ & J_1^* - J_1(\mathbf{u}_{1:K}) \leq \varepsilon, \end{aligned} \quad (8)$$

where \mathbf{u} is the control vector or input vector (ICV settings), \mathbf{x} is the state vector (grid block pressures and saturations), \mathbf{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 of inequality constraints \mathbf{c} concerns production system capacity limitations or operational constraints. The parameter $\varepsilon > 0$ has an appropriately 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 primary objective optimization. Thus, the hierarchical optimization constrains the optimization of the secondary objective with respect to the primary objective function. The ordering of the different objective functions is not unique; thus, secondary objectives can be implemented as primary objectives and vice versa. This hierarchical scheme is attractive when there is a presence of redundant degrees of freedom in the primary objective function. To exploit these degrees of freedom, we require the Hessian of the primary objective function. Some concepts detailing the need for this Hessian and methods to exploit the redundant degrees of freedom are explained in the [Appendix](#).

Approximate Hessian Van Essen et al. [18] proposed the use of a finite difference scheme in combination with the adjoint formulation to approximate the second-order derivatives of the objective function. Without the adjoint formulation, in order to estimate a finite difference-based Hessian, we require $n \times (n + 1)$ function evaluations, where n is the number of controls. Thus, this method is computationally infeasible for realistic reservoir models and large numbers of controls, due to the high number of function evaluations needed. To alleviate this shortcoming, we propose to use the Broyden-Fletcher-Goldfarb-Shanno (BFGS)

algorithm which approximates and updates the Hessian during the optimization of the primary objective function. This leads to a significant reduction in overall computational costs incurred during optimization of the secondary objective function. We note that [5] use the BFGS Hessian in the context of constrained production optimization (as an alternative to steepest ascent), while we use the Hessian only to get information about the null-space of the primary objective.

BFGS algorithm

- From an initial guess \mathbf{u}^0 and an approximate Hessian matrix, \mathbf{H}^0 , the following steps are repeated until \mathbf{u} converges:
- Obtain a direction \mathbf{p}^ℓ by solving: $\mathbf{H}^\ell \mathbf{p}^\ell = -\nabla J(\mathbf{u}^\ell)$.
- Determine an acceptable step size α^ℓ in the direction found in the first step and then update

$$\mathbf{u}^{\ell+1} = \mathbf{u}^\ell + \alpha^\ell \mathbf{p}^\ell.$$

- Set $\mathbf{s}^\ell = \alpha^\ell \mathbf{p}^\ell$.
- Compute $\mathbf{y}^\ell = \nabla J(\mathbf{u}^{\ell+1}) - \nabla J(\mathbf{u}^\ell)$.
- Compute the updated Hessian as

$$\mathbf{H}^\ell = \mathbf{H}^\ell + \frac{\mathbf{y}^\ell (\mathbf{y}^\ell)^T}{(\mathbf{y}^\ell)^T \mathbf{s}^\ell} - \frac{(\mathbf{s}^\ell \mathbf{H}^\ell)^T \mathbf{s}^\ell \mathbf{H}^\ell}{(\mathbf{s}^\ell)^T \mathbf{H}^\ell \mathbf{s}^\ell}.$$

- $J(\mathbf{u})$ 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}^ℓ , the approximation to the Hessian. Note that because we typically take fewer iteration steps than there are DOFs, the BFGS approach will only lead to an approximate Hessian.

Null-space-based optimization algorithm The algorithm is a modification of the algorithm proposed by [18]. The modification is the implementation of the BFGS algorithm to approximate the Hessian matrix.

- 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 where n is the iteration index.
- Use the approximated Hessian \mathbf{H} at \mathbf{u}^* and perform a singular value decomposition to obtain the orthonormal basis \mathbf{B} for the null-space of \mathbf{H} .
- Form the projection operator \mathbf{P} according to

$$\mathbf{P} = \mathbf{B}\mathbf{B}^T.$$

- Find the gradient \mathbf{s} for the secondary objective function J_2 .
- Project this improving direction \mathbf{s} onto the orthonormal basis \mathbf{B} to obtain the projected direction \mathbf{d} , such that \mathbf{d}

is an improving direction for J_2 and does not affect J_1 . Thus, \mathbf{d} is

$$\mathbf{d} = \mathbf{P}\mathbf{s}.$$

- Update the control vector \mathbf{u} using the projected direction \mathbf{d} in the steepest-ascent method.

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \alpha^n \mathbf{d}^n$$

where α is an appropriately small step size.

- Update \mathbf{H} by using the BFGS algorithm for the new set of controls.
- Perform steps 2 to 6 until convergence of J_2 .

The EnOpt algorithm has been used to approximate the gradient of the secondary objective function s . Further details on the algorithm sketched above are provided in the [Appendix](#). Note that an infinitesimally small update of J_2 , projected on the null-space of J_1 , would result in no change in the value of J_1 (by definition). However, because we use an approximate Hessian and because we take finite updates of J_2 , small changes in the value of J_1 may occur.

2.2.2 Switching algorithm

The null-space-based hierarchical algorithm presented above is computationally cumbersome and not feasible for realistic reservoir models having a large number of input parameters when implemented using a finite difference-based Hessian. To overcome this shortcoming, [18] presented a practical alternative method to the null-space-based hierarchical algorithm with the use of a switching function according to

$$J_{bal} = \Omega_1 J_1 + \Omega_2 J_2, \tag{9}$$

where Ω_1 and Ω_2 are switching functions for J_1 and J_2^* that take on values of 1 and 0 or vice versa:

$$\begin{aligned} \Omega_1(J_1) &= \begin{cases} 1 & \text{if } J_1^* - J_1 > \varepsilon, \\ 0 & \text{if } J_1^* - J_1 \leq \varepsilon, \end{cases} \\ \Omega_2(J_1) &= \begin{cases} 0 & \text{if } J_1^* - J_1 > \varepsilon, \\ 1 & \text{if } J_1^* - J_1 \leq \varepsilon. \end{cases} \end{aligned} \tag{10}$$

Here, ε is the threshold value as defined in the inequality constraint in Eq. (8), and J_1^* is the value of the primary objective at the optimal solution achieved during life-cycle optimization. The (transposed) gradient of J_{bal} with respect to the input parameters is then given by the following:

$$\frac{dJ_{bal}^n}{d\mathbf{u}} = \Omega_1(J_1^n) \frac{dJ_1^n}{d\mathbf{u}} + \Omega_2(J_1^n) \frac{dJ_2^n}{d\mathbf{u}}. \tag{11}$$

The use of a balanced objective function in the optimization will give improving directions for either J_1 or J_2 , thus switching between feasible and infeasible solutions. However, the convergence towards an optimal solution may

be rather slow due to the switching between the different solutions. In order to improve convergence speed, [19] suggested the following adaptation in which gradients of the secondary objective function are projected onto the null-space of the optimal primary objective function:

$$\mathbf{P} = \frac{dJ_1}{d\mathbf{u}} \left(\frac{dJ_1^T}{d\mathbf{u}} \frac{dJ_1}{d\mathbf{u}} \right)^{-1} \frac{dJ_1^T}{d\mathbf{u}}. \tag{12}$$

In the neighborhood of the optimum, the complement of the gradient with respect to the primary objective function can be used as a first-order approximation to the null-space of the Hessian of this function. Thus, the improved direction for the secondary objective is given by

$$\mathbf{a} = (\mathbf{I} - \mathbf{P}) \frac{dJ_2^T}{d\mathbf{u}}, \tag{13}$$

Hence, the alternative switching search direction \mathbf{d} for solving the hierarchical problem is

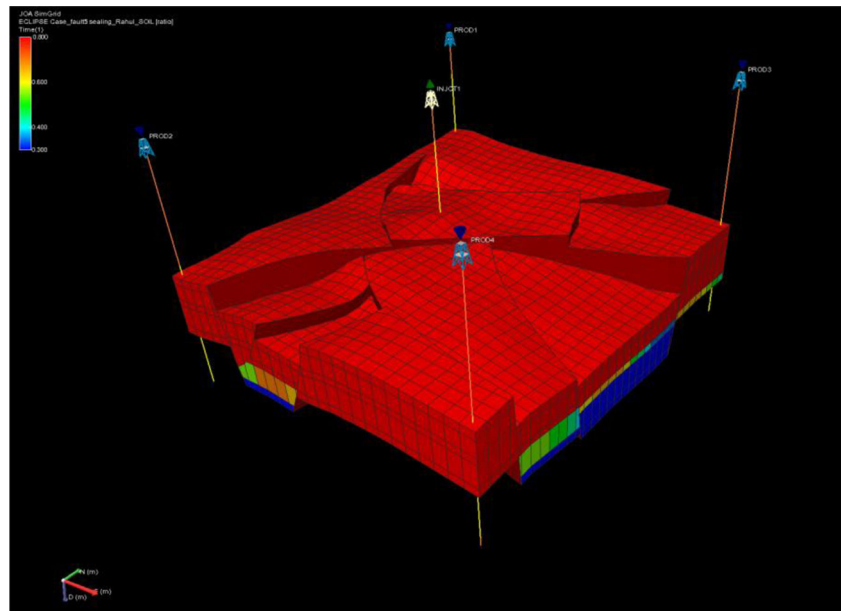
$$\mathbf{d}_{n+1} = \Omega_1(J_1^n) \left(\frac{dJ_1^n}{d\mathbf{u}} \right)^T + \Omega_2(J_1^n) (\mathbf{I} - \mathbf{P}) \left(\frac{dJ_2^n}{d\mathbf{u}} \right)^T. \tag{14}$$

The EnOpt method is used to approximate the gradients for both the primary and the secondary objective functions. The two hierarchical methods presented above are tested on a geologically realistic sector model explained below.

3 Numerical example

A 3D synthetic reservoir model was used to test the multi-objective optimization methods. The life cycle of the reservoir covers 15 years, or 5,470 days. The model illustrated in Fig. 1 consists of $25 \times 32 \times 5 = 4,000$ grid blocks. The approximate size of the grid blocks is $110 \times 90 \times 20$ m. The model represents an area of 2.5 km by 3.5 km by 100 m. The geological structure consists of connected uplifted/offset blocks. There is a sealing fault on the north-western side of the block, close to producer 1. The initial average reservoir pressure is 200 bars. Table 1 lists the geological and fluid properties used to describe the model. The reservoir is produced using an inverted five-spot well pattern, i.e., four producers at the edges of the grid with an injector in the center of the grid. The reservoir is divided into five layers having different horizontal permeabilities, although the permeability in each individual layer is constant. The wells penetrate all the five layers with an ICV in every layer resulting in a total of 25 controls per time step.

Fig. 1 Initial saturation field



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 13 bars. A Corey model with exponents equal to 2 for both oil and water is used for the relative permeabilities where the connate water saturation is 20 %, the residual oil saturation is 30 %, and the end point relative permeabilities to oil and water are 0.8 and 0.4, respectively. No capillary pressures are included, and the reservoir rock is assumed to be incompressible. We used a commercial fully implicit black oil simulator [6].

3.1 Life-cycle optimization

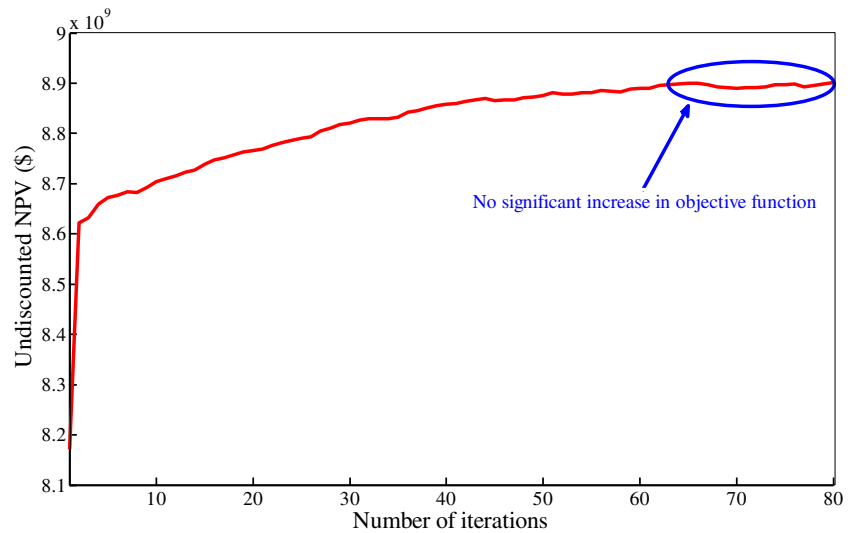
An optimal life-cycle strategy of ICV settings for the individual layers is obtained by optimizing the NPV as described in Eq. (1), with $r_o = 130\$/m^3$, $r_{wp} = 25\$/m^3$,

and $r_{wi} = 6\$/m^3$. The discount rate b was set to 0. Figure 2 is an illustration of the optimization with undiscounted NPV as the objective function, which is equivalent to cumulative cash flow over the producing life of the reservoir. The optimal solution \mathbf{u}^* was obtained using an ensemble size of 100. Well productivity index (PI) multipliers are used to model ICVs in the simulator with bounds of 10^{-4} and 1. The starting point for the optimization is an initial control vector having values equal to 1. Thus, all the ICVs 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 Fig. 2. 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 $8.902 \times 10^9\$$.

Table 1 Permeabilities and fluid properties of the synthetic reservoir model

Property	Values	Units
Porosity	0.15-0.2-0.1-0.25-0.15	–
Permeability (layers 1–5)	100-300-50-600-100	mD
Reservoir pressure	200	bar at 1,950 m
Density of oil	800	kg/m ³
Density of water	1,000	kg/m ³
Oil compressibility	4×10^{-5}	1/bar
Water compressibility	4×10^{-5}	1/bar
Oil viscosity	2	cP at 1 bar
Water viscosity	0.5	cP at 1 bar

Fig. 2 Life-cycle optimization with undiscounted NPV as the objective function



3.2 Objective functions

We use undiscounted NPV as the primary objective function. However, every project also aspires to recover the initial investments as soon as possible. In an ideal case, we would like to additionally meet such short-term economic objectives whilst still maintaining the life-cycle objectives. Thus, we choose a secondary objective function which highlights the importance of maximizing short-term production. The secondary objective function has the same cost structure as the primary objective function, but with a very high discount rate b of 25 %.

3.3 Unconstrained optimization

First, the secondary objective function J_2 is optimized without being constrained by the primary objective function J_1 . This case serves as a comparison to the hierarchical structure explained in the theory. The optimization was performed with an ensemble size of 75 members and was allowed to run for 65 iterations. The results are illustrated in Fig. 3. The starting point of the optimization is the optimal solution achieved during optimization of the primary objective. A decrease of 1.65 % is seen in the primary objective function to achieve an increase of 14.2 % in the

Fig. 3 Illustration of an unconstrained optimization of secondary objective (red) starting from the optimal strategy \mathbf{u}^* obtained from life-cycle optimization

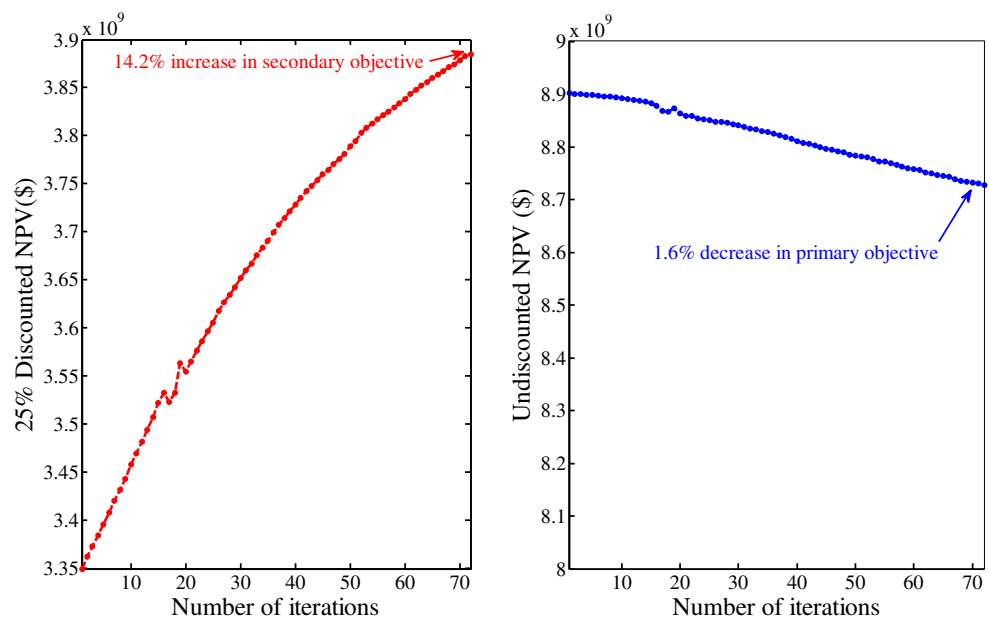
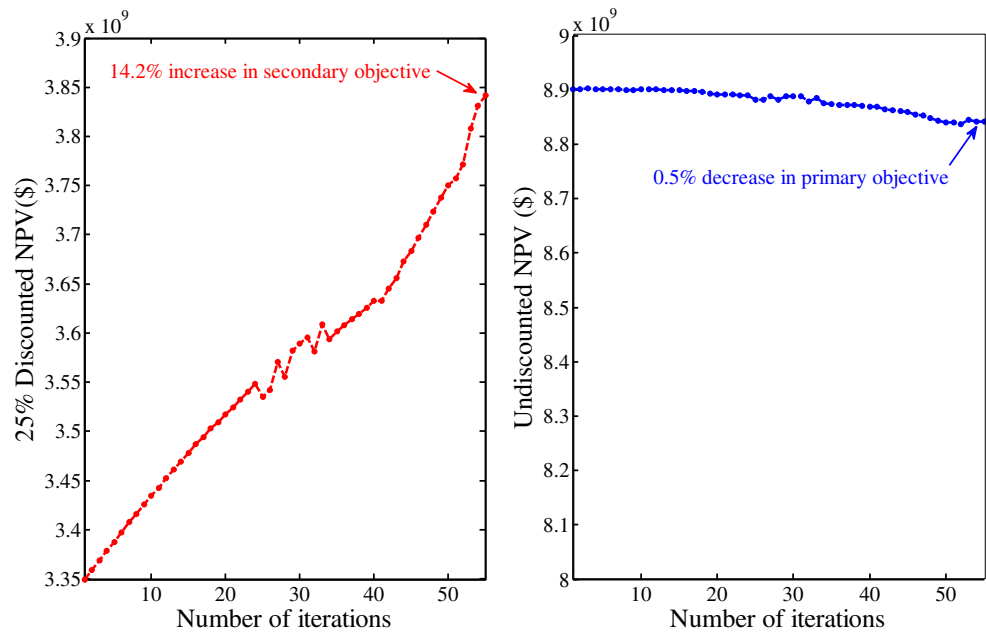


Fig. 4 Comparison of the primary (blue) and secondary (red) objective functions obtained by the null-space-based optimization algorithm



secondary objective function. Note that we did not optimize J_2 until convergence. However, the increase of J_2 with 14.2 % is, by design, just equal to the increase obtained with the hierarchical optimizations discussed below.

3.4 Null-space-based optimization

Figure 4 illustrates the optimization of the secondary objective function within the null-space of the primary objective function. Since, typically, none of the singular values of the Hessian are exactly equal to zero, a cutoff criterion must be defined to estimate the null-space of the Hessian matrix. We have used a cutoff criterion of $\sigma_i/\sigma_1 < 2 \times 10^{-9}$, where σ_i are the singular values of the Hessian matrix, with the values arranged from the largest to

smallest. The resulting null-space consists of 187 vectors, and its dimension remains almost constant throughout the optimization. The value of the secondary objective function at the optimum of the primary objective is 3.35×10^9 \$, which is the starting point for the null-space-based optimization. We achieve a value of 3.823×10^9 \$ after completing the null-space-based optimization equivalent to a 14.2 % increase in the secondary objective function at the price of a 0.52 % decrease in the primary objective function. This decrease is much smaller than that obtained with the unconstrained optimization approach and clearly illustrates the advantage of using the hierarchical multi-objective optimization approach. Another illustration of the impact of the null-space-based optimization is provided in Fig. 5 which shows a comparison of the cumulative cash flow over

Fig. 5 Comparison of cumulative cash flows over time for the different optimization strategies: life-cycle optimized strategy (green), null-space-based optimized strategy (red), and unconstrained secondary optimized strategy (blue)

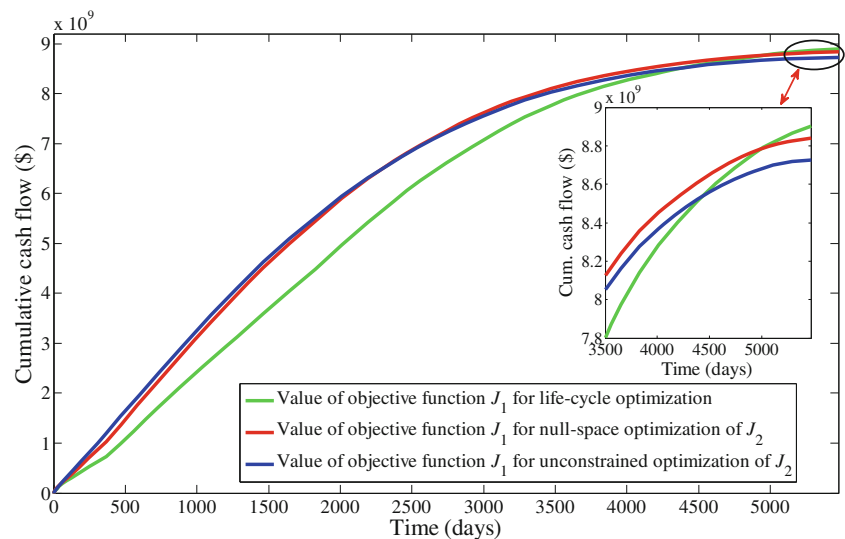
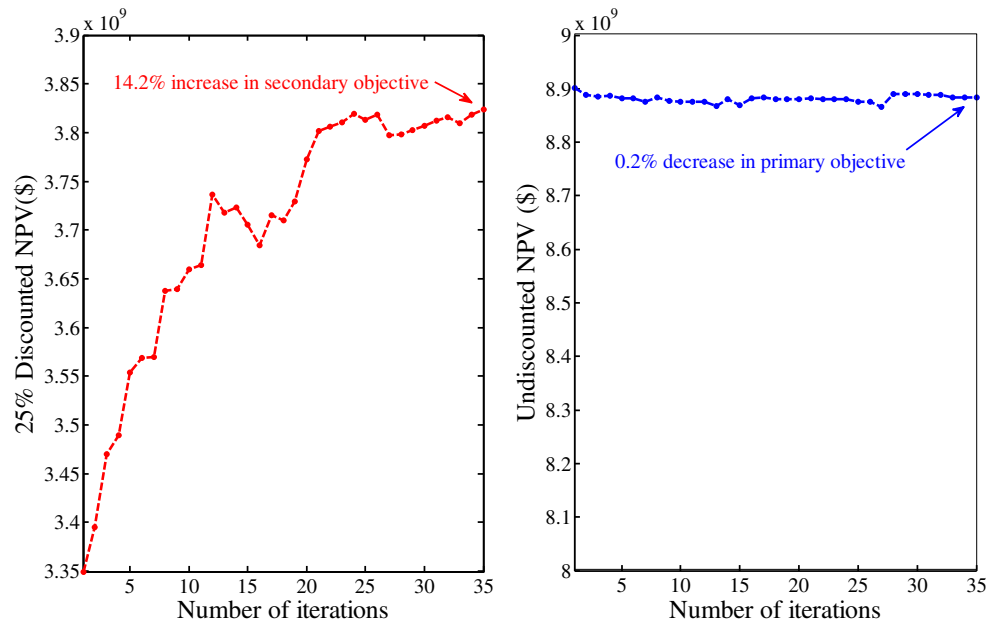


Fig. 6 Switching optimization showing a 14.2 % increase in secondary objective function (red, left) for a corresponding 0.2 % decrease in the primary objective (blue, right)



time resulting from the optimal life-cycle strategy (green), the strategy resulting from null-space-based optimization (red), and from unconstrained optimization of the secondary objective (blue). A considerable increase in the short- to medium-term cumulative cash flow can be observed in the results of the null-space-based hierarchical optimization compared to those of the life-cycle optimization.

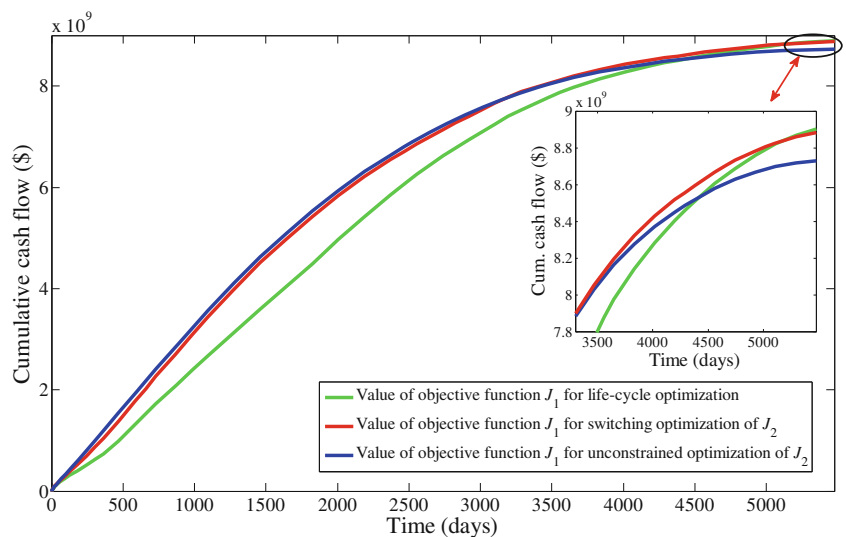
The multi-objective optimization is seen to be useful to increase cash flow in the initial stages of the project whilst maintaining the life-cycle goals. The unconstrained optimization, as expected, has the best cumulative cash flow in the early years of the economic life. However, as shown in the inset figure, compromises are made to the life-cycle

target. The inset plot in Fig. 5 also shows that the null-space-based optimization achieves a solution which performs better in the long-term compared to the unconstrained optimization.

3.5 Switching algorithm

The mathematical formulation of the switching algorithm described in the theory requires the definition of a criterion ϵ . The criterion used is $\epsilon = 0.003 J^*$, i.e., we allow a maximum decrease of 0.3 % in the primary objective function value. An ensemble size of 75 samples was used, and the optimization was allowed to run for 50 iterations, although after 35 iterations, the improvements in the secondary objective became minimal. The results are shown in Fig. 6. The

Fig. 7 Comparison of cumulative cash flow over time for the switching (red), unconstrained (blue), and optimal life-cycle (green) strategies



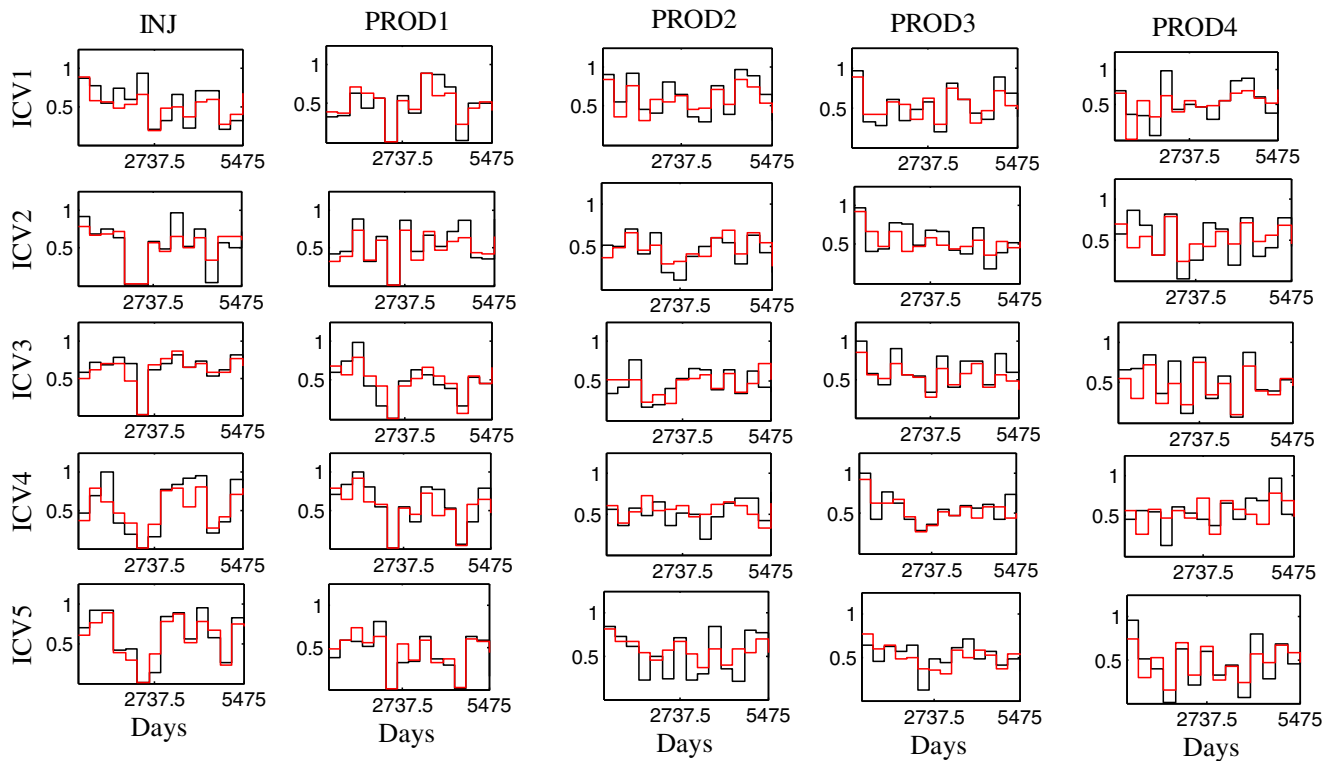


Fig. 8 Comparison of the control strategy for the switching algorithm (*red*) with the control strategy obtained during life-cycle optimization (*black*), for the individual ICVs

algorithm achieves an increase of 14.17 % in the secondary objective for a corresponding 0.21 % decrease in the primary objective. The switching algorithm thus performs very well when applied to this model. Since the performance is dependent on parameters such as ensemble size used for gradient evaluation, step length, and line search parameters, fine-tuning of these parameters may lead to further improved performance of the switching algorithm.

Similar to Fig. 5 for the null-space-based optimization, Fig. 7 compares the cumulative cash flow over time for the switching algorithm with the unconstrained and life-cycle-only optimization results. It is observed that after 500 days, the cumulative cash flow with life-cycle optimization is approximately 1×10^9 \$. However, the control strategy obtained with switching optimization achieves a cash flow of 1.5×10^9 \$. This increase of 0.5×10^9 \$ over 500

days will enable the project to achieve the break-even point faster. Similar to the results obtained by [18] and the results shown above, the NPV at the end time of the unconstrained optimization (blue curve) is decreased by 1.6 %. In comparison, the NPV obtained by the switching algorithm (red curve) decreases by only 0.2 %. Finally, in Fig. 8, the set of controls obtained by the optimization using the switching algorithm (red) is compared to the optimal set of controls obtained after life-cycle optimization (black). The control sets are fairly different, so it can be concluded that rather different control sets may achieve very similar results for the primary optimization while drastically improving the secondary optimization.

Computational aspects In our example, we use an ensemble size of 100 members. The primary objective function J_1 was maximized in 80 iterations, which therefore would

Table 2 Summary of objective function values and changes for the two multi-objective optimization algorithms

Optimization method	Primary objective (10^9 \$)	Secondary objective (10^6 \$)	% Increase in secondary objective	% Decrease in primary objective
Null-space	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

require at least $(100 + 1) \times 80 = 8,080$ forward simulations. The use of back-tracking when the value of J_1 did not increase during a specific iteration increased the actual number of forward simulations to 8,127. For the optimization of the secondary objective J_2 , we used 75 ensemble members and 55 iterations for the null-space method, corresponding to a theoretical minimum number of $(75 + 1) \times 55 = 4,180$ forward simulations, while we actually used 4,215 simulations because of back-tracking. For the switching method, we only used 35 iterations, corresponding to $(75 + 1) \times 35 = 2,660$ (theoretical minimum) and 2,698 (actual) forward simulations.

Discussion The results for the model used here show that the switching method achieves better results compared to the null-space-based method in that the same improvement in secondary objective was obtained with fewer iterations and with a smaller decrease in the primary objective. It is not possible at this point, however, to draw any definitive general conclusion regarding the comparative performance of these methods for more complex models. The choice for the different objective functions may have a large impact and significantly affect the scope for multi-objective optimization. The Hessian approximation with the BFGS algorithm has shown to achieve good results for this case, but may perform differently in another case. The cutoff criterion used to define the dimension of the null-space, as discussed in Section 3.4, was chosen through trial and error in this work. However, this criterion was found to be very important for the success of the null-space optimization (not shown). Another parameter that influences the results is ε , i.e., the maximum allowed decrease in the primary objective. Note that, here, we used ε only for the switching method, but that it could also be used for the null-space method.

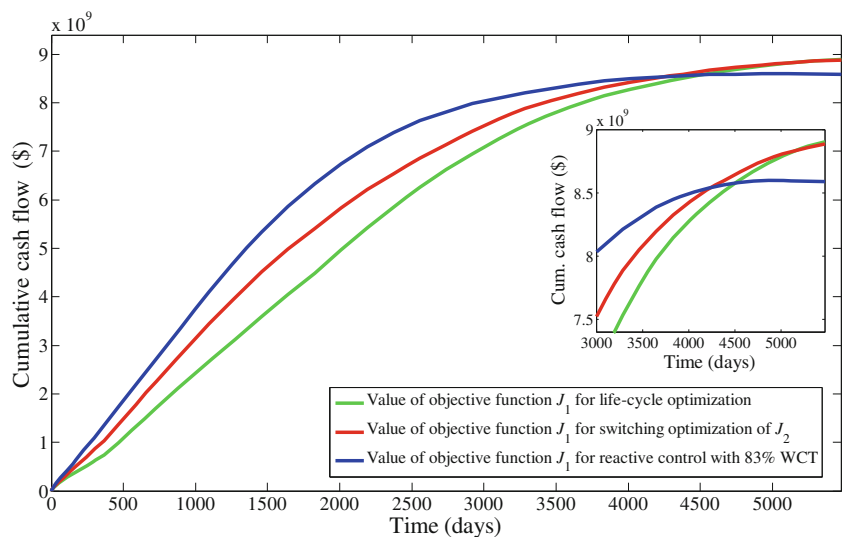
3.6 Results summary

The two hierarchical multi-objective methods using the EnOpt method for gradient evaluation have shown an improvement in a secondary objective approximately constrained to (a maximum specified decrease in) the primary objective. The results obtained with the null-space optimization and switching algorithms are very similar. Table 2 highlights the comparison between the different methods used in this paper to achieve multi-objective optimization.

3.7 Reactive control

For the given oil price of 130\$/bbl and water production costs of 25\$/bbl, the economic feasibility threshold will be achieved at a water cut (WCT) of 83 %. Thus, a reactive control strategy can be defined with a well shut-off limit of 83 % WCT. In such a reactive strategy, the wells are initially operated with all the ICVs fully open, and whenever the WCT limit is reached in a well, it is shut in. As expected, the reactive control strategy has a much lower NPV at the end of the life of the reservoir. Figure 9 is a comparison of the cumulative cash flow for the reactive control strategy to the life-cycle and the switching-based optimized strategies. The strategy obtained with the switching method (red) achieves an improved short-term performance compared to optimized life-cycle strategy (green line), but it is not as good as the reactive control strategy (blue), which gives the best short-term performance. Thus, the two multi-objective optimization methods presented improve the short-term/secondary objectives, but do not truly recover the best possible short-term/secondary objectives as obtained with the reactive strategy. (Note that the unconstrained optimization of J_2 also leads to short-term results worse than those obtained by the reactive strategy,

Fig. 9 Comparison of the cash flow over time for the switching (red), reactive control (blue), and optimal life-cycle (green) strategies



which is probably because we did not iterate to convergence, and/or because we were heading towards a local optimum different from the optimum found in the reactive strategy.) However, Fig. 9 does confirm the advantage of optimal life-cycle strategies in comparison to a reactive control-based operational strategy when long-term objectives are important.

4 Conclusions

- Compromises made to short-term targets during life-cycle optimization can be partly corrected for with an ensemble-based hierarchical multi-objective optimization method.
- The EnOpt method is a good alternative to achieve practical results when the adjoint formulation is not available for hierarchical multi-objective optimization.
- In our numerical simulation examples, two hierarchical multi-objective methods showed a 14.2 % improvement in the secondary objective function (NPV at 25 % discount rate) approximately constrained to the primary objective function (NPV at 0 % discount rate). The results obtained with the null-space-based optimization algorithm are similar to those resulting from the switching algorithm, although for the case investigated here, the switching algorithm was found to be computationally somewhat more efficient.
- The BFGS algorithm, used to estimate the Hessian for the null-space method, is computationally attractive compared to a finite difference method especially when dealing with large control sets and led to good results for the case reported here.
- Hierarchical multi-objective optimization of ICV settings shows significant scope for improvement in short- to medium-term goals approximately constrained to life-cycle targets.

Acknowledgments This research was carried out within the context of the ISAPP Knowledge Centre. ISAPP (Integrated Systems Approach to Petroleum Production) is a joint project of TNO, Delft University of Technology, ENI, Statoil, and Petrobras. The authors would like to acknowledge Paul Egberts (TNO) for his advice and contributions during this work. Moreover, they thank Schlumberger for the donation of multiple academic Eclipse licenses.

Appendix

Degrees of freedom

When dealing with an optimization problem, there may exist multiple sets of control variables for which we achieve similar results. Those sets of different optimal control variables

are an indication of the presence of redundant degrees of freedom (DOFs) in the system. This existence of multiple solutions suggests that, when the optimum of a primary objective function has been reached, not all DOFs of the control variable space are fixed. This implies that there may exist redundant DOFs in the optimization problem. This conclusion formed the basis for development of the multi-objective optimization algorithm of [18].

Null-space

Van Essen et al. [18] approximate a Hessian matrix to find these DOFs, which is an integral part of the null-space-based hierarchical optimization method. Henceforth, we denote $\mathbf{H} = \nabla^2 J(\mathbf{u}^*)$ where \mathbf{H} is the *Hessian matrix*, a matrix of second-order derivatives of the objective function. 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 \mathbf{u}^* is given by the following:

$$J(\mathbf{u}^* + \Delta\mathbf{u}) \approx J(\mathbf{u}^*) + \nabla J(\mathbf{u}^*)^T \cdot \Delta\mathbf{u} + \frac{1}{2} (\Delta\mathbf{u})^T \nabla^2 J(\mathbf{u}^*) \Delta\mathbf{u} + O(\|\Delta\mathbf{u}\|^3). \quad (15)$$

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

$$\nabla J(\mathbf{u}^*) = \mathbf{0}. \quad (16)$$

Substituting Eq. (16) into Eq. (15), we obtain

$$J(\mathbf{u}^* + \Delta\mathbf{u}) \approx J(\mathbf{u}^*) + \frac{1}{2} (\Delta\mathbf{u})^T \nabla^2 J(\mathbf{u}^*) \Delta\mathbf{u} + O(\|\Delta\mathbf{u}\|^3). \quad (17)$$

If we choose $\Delta\mathbf{u} \in \text{null}[\nabla^2 J(\mathbf{u}^*)]$, then $\nabla^2 J(\mathbf{u}^*) \Delta\mathbf{u} = \mathbf{0}$, and Eq. (17) reduces to

$$J(\mathbf{u}^* + \Delta\mathbf{u}) \approx J(\mathbf{u}^*) + O(\|\Delta\mathbf{u}\|^3). \quad (18)$$

Thus, Eq. (18) implies that for a small perturbation $\Delta\mathbf{u} \in \text{null}[\nabla^2 J(\mathbf{u}^*)]$, 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. This means that we can make an update to the control vector that is in the null-space of the primary objective function to improve the secondary objective function. This proves 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. A singular value decomposition is used to obtain the null-space and orthonormal basis \mathbf{B} used in the algorithm.

References

- Brouwer, D.R., Jansen, J.D.: Dynamic optimization of water flooding with smart wells using optimal control theory. *SPE J* **9**(4), 391–402 (2004)
- Chen, C., Li, G., Reynolds, A.C.: Robust constrained optimization of short- and long-term net present value for closed-loop reservoir management. *SPE J* **17**(3), 849–864 (2012)
- Chen, Y.: Efficient ensemble based reservoir management. PhD thesis, University of Oklahoma (2008)
- Chen, Y., Oliver, D.S., Zhang, D.: Efficient ensemble-based closed-loop production optimization. *SPE J* **14**(4), 634–645 (2009)
- Dehdari, V., Oliver, D.S.: Sequential quadratic programming for solving constrained production optimization—case study from Brugge field. *SPE J* **17**(3), 874–884 (2012)
- Eclipse. www.software.slb.com/products/foundation/Pages/eclipse-2013.aspx (2013)
- Hansen, N.: The CMA evolution strategy: a comparing review. In: Lozano, J. A., Larranaga, P., Inza, I., Bengoetxea, E. (eds.) *Towards a new evolutionary computation. Advances on estimation of distribution algorithms*, pp 75–102. Springer, Berlin (2006)
- Jansen, J.D., Douma, S.G., Brouwer, D.R., Van den Hof, P.M.J., Bosgra, O.H., Heemink, A.W.: Closed-loop reservoir management. Paper SPE 119098 presented at 2009 SPE Reservoir Simulation Symposium. The Woodlands, Texas (2009)
- Jansen, J.D.: Adjoint-based optimization of multiphase flow through porous media—a review. *Comput. Fluids* **46**(1), 40–51 (2011)
- Lorentzen, R.J., Berg, A.M., Naevdal, G., Vefring, E.H.: A new approach for dynamic optimization of waterflooding problems. Paper SPE 99690 Presented at the SPE Intelligent Energy Conference and Exhibition, Amsterdam (2006)
- Nocedal, J., Wright, S.J.: *Numerical optimization*. 2nd edn. Springer, New York (2006)
- Oliver, D.S., Reynolds, A.C., Liu, N.: *Inverse theory for petroleum reservoir characterization and history matching*. Cambridge University Press, Cambridge (2008)
- Sarma, P., Aziz, K., Durlofsky, L.J.: Implementation of adjoint solution for optimal control of smart wells. Paper SPE 92864 Presented at the SPE Reservoir Simulation Symposium, The Woodlands, Texas (2005)
- Sarma, P., Chen, W.H., Durlofsky, L.J., Aziz, K.: Production optimization with adjoint models under nonlinear control-state path inequality constraints. *SPE Reserv. Eng. Eval* **11**(2), 326–339 (2006)
- Spall, J.C.: Implementation of the simultaneous perturbation algorithm for stochastic optimization. *IEEE Trans. Aerosp. Electron. Syst* **34**(3), 817–823 (1998)
- Strang, G.: *Linear algebra and its applications*. 4th edn. Thomson Brooks/Cole, Pacific Grove (2006)
- Suwartadi, E., Krogstad, S., Foss, B.: Nonlinear output constraints handling for production optimization of oil reservoirs. *Comput. Geosci* **16**(2), 499–517 (2012)
- Van Essen, G.M., Van den Hof, P.M.J., Jansen, J.D.: Hierarchical long and short-term production optimization. *SPE J* **16**(1), 191–199 (2011)
- Van Essen, G.M., Van den Hof, P.M.J., Jansen, J.D.: Determination of lower and upper bounds of predicted production from history-matched models. In: *Proceedings of 12th European Conference on the Mathematics of Oil Recovery ECMOR XII*, Oxford (2010)