

Generation of a Pareto front for a bi-objective water flooding optimization problem using approximate ensemble gradients

Fonseca, Rahul; Reynolds, Albert C.; Jansen, Jan Dirk

DOI

[10.1016/j.petrol.2016.06.009](https://doi.org/10.1016/j.petrol.2016.06.009)

Publication date

2016

Document Version

Final published version

Published in

Journal of Petroleum Science and Engineering

Citation (APA)

Fonseca, R., Reynolds, A. C., & Jansen, J. D. (2016). Generation of a Pareto front for a bi-objective water flooding optimization problem using approximate ensemble gradients. *Journal of Petroleum Science and Engineering*, 147(November), 249-260. <https://doi.org/10.1016/j.petrol.2016.06.009>

Important note

To cite this publication, please use the final published version (if applicable). Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.



ELSEVIER

Contents lists available at ScienceDirect

Journal of Petroleum Science and Engineering

journal homepage: www.elsevier.com/locate/petrol

Generation of a Pareto front for a bi-objective water flooding optimization problem using approximate ensemble gradients



Rahul M. Fonseca^a, Albert C. Reynolds^b, Jan Dirk Jansen^{a,*}

^a Delft University of Technology, Department of Geoscience and Engineering, The Netherlands

^b The University of Tulsa, McDougall School of Petroleum Engineering, United States

ARTICLE INFO

Article history:

Received 3 August 2015

Received in revised form

1 June 2016

Accepted 7 June 2016

Available online 9 June 2016

Keywords:

Water flooding

Bi-objective optimization

Pareto front

Ensemble optimization

Approximate gradient

Augmented Lagrangian

ABSTRACT

Conflicting objectives are frequently encountered in most real-world problems. When dealing with conflicting objectives, decision makers prefer to obtain a range of possible optimal solutions from which to choose. In theory, methods exist that can produce a range of possible solutions, some of which are “Pareto Optimal”. The application of these methods to solve bi-objective production optimization problems is increasing. A recent paper introduced a method to find points on the boundary of the objective function space by solving a constrained optimization problem using adjoint gradients. In this work, we investigate the applicability of using ensemble optimization (EnOpt) (which relies on approximate ensemble gradients instead of exact adjoint-based gradients) to generate points along a “Pareto” front with acceptable computational effort. Moreover, we investigate the applicability of this approximate gradient technique to solve constrained optimization problems using the augmented Lagrangian method. Finally, we compare the performance of this bi-objective optimization method to a traditional weighted sum method for bi-objective water flooding optimization of two different synthetic reservoir models. The two objectives used in this work are, undiscounted (0%) net present value (NPV), representing long-term targets and highly discounted (25%) NPV, representing short-term operational targets. The controls are inflow control valve (ICV) settings over time for one model and water injection rate controls for the other. The effect of different starting points and the computational efficiency of the constrained optimization method are also investigated.

© 2016 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

A majority of studies and applications of life-cycle water flooding optimization using a model-based approach have focused on a single objective optimization with emphasis being placed on the theoretical understanding and practical application of the optimization methodology. Life-cycle optimization essentially aims to find a strategy which optimizes long-term reservoir management targets, but life-cycle optimization is often at the expense of operationally significant short-term targets. Thus, there is a need to solve a bi-objective problem to obtain a strategy that accounts for the two objectives because the long-term perspective is usually in conflict with the short-term targets which are decided by operational constraints, contractual obligations etc. Van Essen et al. (2011) introduced a hierarchical optimization framework to solve such a multi-objective optimization problem. This was

motivated by the observation in, e.g., Jansen et al. (2009) that the objective function space consists of many redundant degrees of freedom which can be exploited to optimize a secondary objective. This hierarchical structure provides a single optimal strategy which incorporates multiple objectives. However, decision makers usually prefer to have multiple strategies to choose from, especially when dealing with conflicting objectives. Isebor and Durlofsky (2014) applied an evolutionary algorithm to generate points along a “Pareto” front for a bi-objective water flooding problem. The main pitfall of this approach was the computational effort required to obtain the points on a Pareto front. Also they did not compare the front generated with any other method used to generate Pareto fronts to check if the front obtained was Pareto optimal. Liu and Reynolds (2014) applied the normal boundary intersection method (NBI) first introduced in Das and Dennis (1998) to a bi-objective water flooding problem with and without geological uncertainty. Liu and Reynolds (2014) showed that the NBI method is computationally more efficient than the method of Isebor and Durlofsky (2014) and produces better solutions than the traditional weighted sum method. The NBI method involves

* Correspondence to: Delft University of Technology, Department of Geoscience and Engineering, Stevinweg 1, 2628 CN Delft, The Netherlands.

E-mail address: j.d.jansen@tudelft.nl (J.D. Jansen).

solving a series of constrained optimization sub-problems. In Liu and Reynolds (2014), these constrained optimization problems were solved using an augmented Lagrangian method using an adjoint formulation to compute the gradients. The adjoint formulation, an overview of which can be found in Jansen (2011) and references therein, is a computationally efficient method which requires access to the simulator source code to implement. Most commercial simulators either do not have a fully developed adjoint code or access to the source code is not permissible. This has led to an increase in the application of various approximate gradient based techniques which are computationally less efficient but use the simulator as a black-box, and are more flexible. Do and Reynolds (2013) provided theoretical connections between various existing approximate gradient techniques which use an ensemble of perturbed controls to estimate a gradient. One such approximate gradient technique introduced in Lorentzen et al. (2006) and thereafter in its current form by Chen et al. (2009) is the ensemble optimization (EnOpt) method. Recently many studies have used EnOpt for life-cycle production optimization problems. Fonseca et al. (2014) applied EnOpt to solve a bi-objective optimization problem using the hierarchical structure proposed by Van Essen et al. (2011). Additionally there has been an increase in the number of applications of different evolutionary algorithms to solve either a bi-objective optimization problem, Isebor and Durlofsky (2014) etc., or for history matching applications, as detailed in Liu and Reynolds (2014). In this work we investigate the applicability of the EnOpt technique to generate points along a ‘‘Pareto’’ front with acceptable computational effort. A secondary aim is the application of EnOpt to solve constrained optimization problems using the augmented Lagrangian method. Note that Fonseca et al. (2014) consider hierarchical optimization (using EnOpt), in which case an a-priory choice is made which of the two objectives is most important. Here we consider bi-objective optimization (using EnOpt) based on the Pareto front approach which provides freedom to the decision maker to choose the relative importance of each of the two objectives, as will be explained in more detail below.

2. Theory

This section investigates the applicability of the use of approximate ensemble gradients to calculate points on a Pareto front for bi-objective production optimization problems.

2.1. Objective functions

We first define the objective functions followed by an overview of EnOpt. We apply the usual expression for Net Present Value (NPV) as objective function J :

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

where $q_{o,k}$ is the average oil production rate in m³/day for time step k , $q_{wp,k}$ is the water production rate in m³/day for time step k , $q_{wi,k}$ is the water injection rate in m³/day for time step k , r_o is the sale price of oil in \$/m³, r_{wp} is the cost of water produced in \$/m³, r_{wi} is the cost of water injected in \$/m³, Δt_k is the length of the k^{th} time step in days, b is the discount factor, t_k is the cumulative time in days corresponding to time step k , and τ_t is the reference time period for discounting, typically one year (i.e. 365 days). In this work the two objective functions are:

– Undiscounted NPV, $b=0.0$ (0%) in Eq. (1), representing the long-

term objective (‘‘recovery optimization’’).

– Highly discounted NPV, $b=0.25$ (25%) in Eq. (1), representing the short-term objective (‘‘day-to-day production optimization’’).

2.2. Ensemble optimization (EnOpt)

In this section, we outline the standard formulation of the EnOpt algorithm as proposed by Chen et al. (2009). We take \mathbf{u} to be a single control vector containing all the control variables to be optimized. This vector has N components where N is equal to the product of the controllable well parameters (number of well settings like bottom hole pressures, rates or valve settings) and the number of control time steps. Chen et al. (2009) sample the initial mean control vector from a Gaussian distribution while, at later iteration steps, the final control vector of the previous iteration is taken as the mean control. However, the initial controls can also be chosen by the user, as will be done in our experiments. The vector of controls is given by,

$$\mathbf{u}_i = [u_1 \quad u_2 \quad \dots \quad u_N]^T \quad (2)$$

where the counter i preempts the use of multiple control vectors, and where \mathbf{u}_i is assumed to be a random vector which has a mean \mathbf{u} and covariance matrix $\tilde{\mathbf{C}}$, i.e. $\mathbf{u}_i \sim N(\mathbf{u}, \tilde{\mathbf{C}})$. Then an ensemble of M independent samples of $N(\mathbf{u}, \tilde{\mathbf{C}})$ are generated as,

$$\mathbf{u}_i = \mathbf{u} + \tilde{\mathbf{C}}^{1/2} \mathbf{z}_i, \quad (3)$$

with $i=1, 2, \dots, M$, where $\mathbf{z}_i \sim N(\mathbf{0}, \mathbf{I})$, i.e., each \mathbf{z}_i is a vector of independent standard random normal deviates, and $\tilde{\mathbf{C}}^{1/2}$ is any square root of $\tilde{\mathbf{C}}$. In our examples $\tilde{\mathbf{C}}^{1/2} = \mathbf{L}$, where \mathbf{L} is the lower triangular matrix in the Cholesky decomposition of $\tilde{\mathbf{C}}$. We truncate any element of the ensemble of controls outside of the set of bounds to the bound value. Then, the sample mean is computed as

$$\bar{\mathbf{u}} = \frac{1}{M} \sum_{i=1}^M \mathbf{u}_i. \quad (4)$$

To estimate the gradient, a mean-shifted ensemble matrix is defined as

$$\Delta \mathbf{U} = [\mathbf{u}_1 - \bar{\mathbf{u}} \quad \mathbf{u}_2 - \bar{\mathbf{u}} \quad \dots \quad \mathbf{u}_M - \bar{\mathbf{u}}]. \quad (5)$$

Similarly, a mean-shifted objective function vector is defined as

$$\Delta \mathbf{j} = [J(\mathbf{u}_1) - \bar{J} \quad J(\mathbf{u}_2) - \bar{J} \quad \dots \quad J(\mathbf{u}_M) - \bar{J}]^T, \quad (6)$$

where the average of the objective function is given by

$$\bar{J} = \frac{1}{M} \sum_{i=1}^M J(\mathbf{u}_i). \quad (7)$$

In the present paper, we use as the search direction in a steepest ascent algorithm an approximation to the gradient, rather than the approximation of the smoothed gradient that is used in standard EnOpt. The approximate gradient is

$$\mathbf{g} = (\Delta \mathbf{U} \Delta \mathbf{U}^{\dagger}) \Delta \mathbf{U} \Delta \mathbf{j} = (\Delta \mathbf{U}^{\dagger}) \Delta \mathbf{j}, \quad (8)$$

where the superscript \dagger indicates the Moore-Penrose pseudo-inverse, which is conveniently computed using a singular value decomposition (SVD); see, e.g., Strang (2006). Do and Reynolds (2013) demonstrated that it is akin to what is known as a ‘simplex gradient’, Conn et al. (2009). They also provided theoretical connections between various ensemble methods such as simultaneous perturbation stochastic approximation (SPSA), simplex gradient and EnOpt. Moreover, they proposed a modification to the gradient formulation which uses the current control vector \mathbf{u}^c and the corresponding objective function value J^c to calculate

the control and objective function anomalies $\Delta \mathbf{U}$ and $\Delta \mathbf{j}$:

$$\Delta \mathbf{U} = [\mathbf{u}_1 - \mathbf{u}^\ell \quad \mathbf{u}_2 - \mathbf{u}^\ell \quad \dots \quad \mathbf{u}_M - \mathbf{u}^\ell], \quad (9)$$

$$\Delta \mathbf{j} = [J(\mathbf{u}_1) - J^\ell \quad J(\mathbf{u}_2) - J^\ell \quad \dots \quad J(\mathbf{u}_M) - J^\ell]^T, \quad (10)$$

where the superscript ℓ is the optimization iteration index. In this work, we have used (Eqs. (9) and 10) in combination with Eq. (8). We note that many authors also use single and double smoothed versions of Eq. (8) which can be obtained by pre-multiplication (either once or twice) of the gradient estimate by the covariance matrix used to generate the ensemble of controls. For the original derivation of EnOpt, including an extension to use the method under geological uncertainty (as represented by an ensemble of geologically different reservoir models), we refer to Chen et al. (2009). Here we restrict our application to “deterministic EnOpt”, i.e. to optimization under the assumption that the geology is known.

2.3. Update rules

The approximate gradient \mathbf{g} from Eq. (8) can be used in any gradient-based optimization algorithm. In our study we use the simple steepest ascent scheme given by

$$\mathbf{u}^{\ell+1} = \mathbf{u}^\ell + \alpha^\ell \frac{\mathbf{g}^\ell}{\|\mathbf{g}^\ell\|_\infty}, \quad (11)$$

where the superscript ℓ is the iteration index, and α^ℓ is a step length in the direction of the approximate gradient. Note that we scaled the gradient by its infinity norm and choose a step length to be 10% of the difference between the maximum and minimum values of the controls. We allowed for a maximum of five back-tracking steps, each time reducing the step size with a factor of one half if the objective function J decreases from one iteration to the next. If after the five back-tracking steps we still do not find an increase in J we accept the current control strategy and continue with the optimization until a convergence criteria is satisfied. Optimal update schemes and their corresponding parameters are typically case-dependent and more sophisticated line search algorithms are sometimes beneficial. However, we chose to use a relatively simple update strategy to facilitate the comparison of the various multi-objective optimization schemes which form the key subject of our paper.

3. Multi-objective optimization

Most real world problems have multiple objectives that need to be satisfied. Usually these objectives are in conflict with each other, i.e. one must accept decreases in one objective to achieve increases in another objective. The process of optimizing systematically and simultaneously a collection of objective functions is called multi-objective optimization. In theory, there exist many methods to solve a multi-objective problem and recently there has been an increased focus on finding methods to solve multi-objective problems in the reservoir simulation community. These objectives are usually defined as long-term (life-cycle) objectives from a reservoir engineering viewpoint and short-term objectives from a production engineering/operational constraints viewpoint. Van Essen et al. (2011) showed that these two objectives may be in conflict with each other and suggested the use of a hierarchical framework for multi-objective optimization. An alternative to hierarchical bi-objective optimization (in which the primary objective is considered more important than the secondary

objective), is regular bi-objective optimization in which there is no predefined preference for one of the objectives. Isebor and Durlofsky (2014), and Liu and Reynolds (2014) have introduced methodologies to generate the ‘Pareto front’ i.e. a range of possible solutions for a decision maker for a regular bi-objective reservoir optimization problem. Isebor and Durlofsky (2014) presented their methodology using a hybrid evolutionary algorithm, PSO-MADS, and reported results which were obtained with a significant computational effort. Liu and Reynolds (2014) presented a method using adjoint gradients which was shown to be computationally much more efficient. We use, in this work, the methods introduced in Liu and Reynolds (2014) and investigate their applicability in combination with the EnOpt method.

A point is defined as “Pareto optimal” if at that point the value of one objective function cannot be increased unless the value of a second objective function is decreased or, in other words, a control set is Pareto optimal if there does not exist any other control set which achieves better objective function solutions. Liu and Reynolds (2014) provide details of the commonly used theoretical definitions to determine whether points are non-dominated, i.e. Pareto optimal, and lie on a Pareto front.

3.1. Weighted sum method

The life-cycle waterflooding problem is inherently a long-term optimization problem as shown in Van Essen et al. (2011) and short-term goals are sacrificed to achieve the optimal long-term targets. A traditional technique to balance two conflicting objectives is the weighted sum method, see Marler and Arora (2004), which aims to optimize a weighted objective function that combines both objectives in a single function according to

$$J_{ws} = w_1 J_1 + w_2 J_2, \quad (12)$$

where J_{ws} is the weighted sum objective function constructed from the long-term and short-term objective functions J_1 and J_2 with w_1 and w_2 as weighting factors. Liu and Reynolds (2014), among others, showed that the biggest drawback of this method in finding solutions on a Pareto curve is that the solutions tend to be concentrated on one part of the curve, i.e., the solutions generated are not evenly distributed along the Pareto front. Another disadvantage is that the weighted sum method cannot obtain points on the concave part of the Pareto front, see, for example, Fig. 1 of Liu and Reynolds (2014).

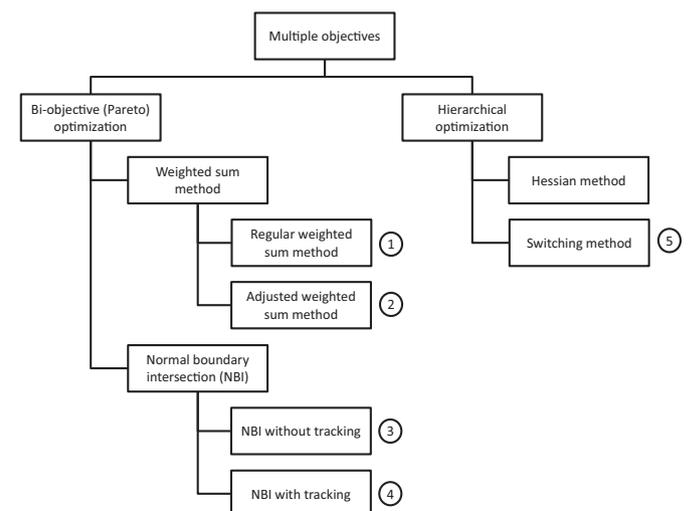


Fig. 1. Overview of various methods available to solve multi-objective reservoir optimization problems.

3.2. Adjusted weighted sum method

To overcome the difficulties of the weighted sum method, Liu and Reynolds (2014) proposed an adjusted weighted sum formulation where the weights w_1 and w_2 are now replaced by

$$\tilde{w}_1 = \frac{\frac{w_1}{J_1(\mathbf{u}_1^*) - J_1(\mathbf{u}_2^*)}}{\frac{w_1}{J_1(\mathbf{u}_1^*) - J_1(\mathbf{u}_2^*)} + \frac{w_2}{J_2(\mathbf{u}_2^*) - J_2(\mathbf{u}_1^*)}}, \quad (13)$$

and

$$\tilde{w}_2 = 1 - \tilde{w}_1, \quad (14)$$

while Eq. (12) is replaced by

$$\tilde{J}_{ws} = \tilde{w}_1 J_1 + \tilde{w}_2 J_2. \quad (15)$$

Note that $w_1=1$ implies $\tilde{w}_1=1$ and $\tilde{J}_{ws}=J_1$ so maximizing \tilde{J}_{ws} with $\tilde{w}_1=1$ corresponds to maximizing J_1 . Similarly, $w_1=0$ implies $\tilde{w}_1=0$ and in this case maximizing \tilde{J}_{ws} corresponds to maximizing J_2 . Liu and Reynolds (2014) found that choosing decreasing w_1 from 1 to 0.1 in increments of 0.1, computing the corresponding values of \tilde{w}_1 and \tilde{w}_2 , and maximizing \tilde{J}_{ws} for each of these \tilde{w}_1 , \tilde{w}_2 values tended to result in points that were well distributed along the Pareto front when maximizing \tilde{J}_{ws} , whereas Eq. (12) with the same set of w_1 values did not generate a well-distributed Pareto front.

3.3. Normal Boundary Intersection (NBI) method

In order to overcome the disadvantages of the weighted sum method, Das and Dennis (1998) proposed a technique, the Normal Boundary Intersection (NBI) method, to find points on the boundary of a feasible set starting from points along the “utopia line” which is defined as the line in the objective function space that connects the optimum solutions for the individual objective functions. The boundary points are then found by optimizing the magnitude of a unit normal to the utopia line in the objective function space. A detailed description of the NBI method can be found in Das and Dennis (1998) and, for petroleum engineering applications, in Liu and Reynolds (2014). The NBI method is motivated by the fact that the Pareto front must coincide with a part of the boundary of the feasible region. The disadvantage of NBI is that boundary points may or may not be Pareto optimal, i.e., may or may not lie on the Pareto front. However once optimal design vectors \mathbf{u}_1^* , ..., \mathbf{u}_n^* are generated it is easy to check if each point is non-dominated by any other, which must be the case if $(J_1(\mathbf{u}_n^*), J_2(\mathbf{u}_n^*))$ is a point on the Pareto front; see definitions in Liu and Reynolds (2014, 2016). The following is a brief description of the method as described in Liu and Reynolds (2014) for bi-objective water flooding optimization problems. For the two objective functions denoted by J_1 and J_2 the NBI procedure is repeated for different points along the utopia line. The general formulation for NBI is given by

$$\begin{aligned} & \max_{\mathbf{u}, t} t, \\ & \text{s. t. } \mathbf{e}(\mathbf{u}, t) = \Phi\boldsymbol{\beta} + t\mathbf{n} - \mathbf{j}(\mathbf{u}) = \mathbf{0}; \\ & \boldsymbol{\beta} = (\beta_1, \beta_2)^T, \beta_1 > 0, \beta_2 > 0, \beta_1 + \beta_2 = 1, \end{aligned} \quad (16)$$

where

$$\Phi = \begin{bmatrix} J_1(\mathbf{u}_1^*) & J_1(\mathbf{u}_2^*) \\ J_2(\mathbf{u}_1^*) & J_2(\mathbf{u}_2^*) \end{bmatrix} \text{ and } \mathbf{j}(\mathbf{u}) = [J_1(\mathbf{u}), J_2(\mathbf{u})]^T, \quad (17)$$

and where \mathbf{u}_1^* and \mathbf{u}_2^* are the optimal control strategies obtained for the individual optimizations of J_1 and J_2 . The line segment that

connects $\mathbf{j}(\mathbf{u}_1^*)$ and $\mathbf{j}(\mathbf{u}_2^*)$ in the objective space as $[\mathbf{j}(\mathbf{u}_1^*)\mathbf{j}(\mathbf{u}_2^*)][\beta_1\beta_2]^T = \Phi\boldsymbol{\beta}$ is defined as the utopia line. To solve the equality constrained optimization problem as described in Eq. (16), Liu and Reynolds (2014) used the augmented Lagrangian method. While there exist several techniques to solve constrained optimization problems, we too have applied the augmented Lagrangian method because the main purpose of this work is to investigate the ability of approximate gradient techniques like EnOpt to generate solutions along a Pareto front. A by-product of this work is the demonstration of the applicability of an approximate gradient technique to solve constrained optimization problems. The augmented Lagrangian method (Nocedal and Wright, 2006), used to solve the different NBI sub-problems, is based on the augmented Lagrangian function which is defined by

$$J_{nbi}(\mathbf{u}, t, \lambda, \mu) = t - \lambda^T \mathbf{e} - \frac{1}{2\mu} \mathbf{e}^T \mathbf{e}, \quad (18)$$

where \mathbf{e} is defined in Eq. (16), λ is a vector of Lagrange multipliers and μ is a penalty parameter. The constraint violation is given by $\sigma_{cv} = \sqrt{\mathbf{e}^T \mathbf{e} / 2}$. Liu and Reynolds (2014), because they were using adjoint gradients, calculated the gradient of the Lagrangian function with respect to \mathbf{u} in terms of the gradients of objective functions J_1 and J_2 with respect to \mathbf{u} . In this work, because we use approximate ensemble gradients, we calculate the gradient of the Lagrangian function directly using Eq. (8). The following is a brief algorithmic description of the NBI method as implemented in our case.

- Calculate the initial optimization parameters: $\boldsymbol{\beta}, \mathbf{n}, \mathbf{u}, t, \mathbf{e}(\mathbf{u}), \lambda, \mu$ where $\mathbf{u} = \beta_1 \mathbf{u}_1^* + \beta_2 \mathbf{u}_2^*$, and where t , following Liu and Reynolds (2014), is initialized as

$$t = \frac{\mathbf{n}^T (\mathbf{j}(\mathbf{u}) - \Phi\boldsymbol{\beta})}{\mathbf{n}^T \mathbf{n}}.$$

The initial penalty parameter is then given by $\mu = \mathbf{e}^T \mathbf{e} / (0.1 \cdot t)$ and the Lagrange multipliers by $\lambda = \mathbf{e} / \mu$.

- While $\sigma_{cv} > 0.01$ (outer loop)
- Until stopping criteria is satisfied (inner loop)
 - Maximize the Lagrangian function given by Eq. (18) until convergence is achieved. Note: λ, μ are constant within the inner loop and can only change in the outer loop. Gradients are approximated using Eq. (8) in conjunction with (Eqs. (9) and 10)

End Inner Loop

- Check criteria to update λ and μ using formulas given in Liu and Reynolds (2014)
- Repeat until convergence of outer loop.

3.4. Tracking the Pareto front using NBI

The NBI method as implemented by Liu and Reynolds (2014) chooses as a starting point a combination of the optimal control sets \mathbf{u}_1^* and \mathbf{u}_2^* depending on the weight factors chosen. Due to the non-linearity of the problem these initial points usually have objective function values that do not lie exactly on the utopia line. The NBI problem does not necessarily require the starting points to be on or close to the utopia line, so we propose to generate points on the Pareto front by starting from a point on the front which has already been obtained with different values of β_1 and β_2 . This is akin to “tracking” a front. In the results section, we discuss the advantages/disadvantages of using this method of generating solutions on a front.

3.5. Hierarchical switching method

Van Essen et al. (2011) 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. The ordering of the different objective functions is the prerogative of the user, thus secondary objectives can be implemented as primary objectives and vice versa. This hierarchical scheme is especially attractive in the presence of redundant degrees of freedom in the primary objective function. Van Essen (2011) proposed two different varieties of the hierarchical scheme: one requiring the computation of the Hessian matrix of the objective function with respect to the controls, and one which, in a more pragmatic fashion, alternately optimizes the short and long-term objectives while maintaining the first objective function value close to its initial maximum. Based on the results in Fonseca et al. (2014), we use the hierarchical switching method with EnOpt for the optimization where details of the implementation are provided. This method optimizes the objectives alternately with the use of a switching function according to

$$J_{switch} = \Omega_1 J_1 + \Omega_2 J_2, \quad (19)$$

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:

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

$$\Omega_2(J_1) = \begin{cases} 0 & \text{if } J_1^* - J_1 > \varepsilon, \\ 1 & \text{if } J_1^* - J_1 \leq \varepsilon. \end{cases} \quad (20)$$

Here ε is the threshold value and J_1^* is the value of the primary objective at the optimal solution achieved during life cycle optimization. We will compare the results obtained from hierarchical switching optimization to the other methods presented above. The advantage of using a hierarchical switching method is that a user can decide the maximum allowable decrease in the primary objective value which is practically impossible to know when using the weighted sum method. However, with this hierarchical method, only a single control set is generated which may or may not be Pareto optimal since no other information is available for comparison.

Fig. 1 depicts an overview of various optimization methods

available to solve multi-objective reservoir optimization problems. In the current paper we compare the methods indicated with numbers 1–5 with the aid of two numerical examples.

4. Example 1: faulted five-spot

4.1. Reservoir model

Advances in technology have led to an increase in the application of inflow control valves (ICVs) to regulate flow rates and maintain pressure in the reservoir. We consider a control problem where ICV settings of injection and production wells in a 3D synthetic reservoir model, from JOA (2007), are manipulated to optimize waterflooding over the producing life of the reservoir. The model, illustrated in Fig. 2(a), consists of $25 \times 32 \times 5 = 4000$ grid blocks. The approximate size of each grid block is $110 \times 90 \times 20$ m, so that the reservoir volume is $2.5 \times 3.5 \times 0.1$ km³. The geological structure consists of uplifted blocks, separated by faults. The reservoir is produced using an inverted five-spot well pattern, i.e. four producers at the corners of the grid with an injector in the center. The reservoir is divided into five layers with different horizontal permeabilities. Fig. 2(b) is the top view of the transmissibility multipliers used for this model and the white cells are grid blocks that are inactive. There is a sealing fault on the North-Western side of the block, close to producer 1. Table 1 lists the reservoir and fluid properties of the model. 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 0.2, the residual oil saturation is 0.3 and the end point relative permeabilities to oil and water are 0.8 and 0.4 respectively. Capillary pressure effects are not included. The wells penetrate all five layers with one ICV in each layer. The producing life of the reservoir is divided into 15 optimization control steps, each of which is one year (365 days) in duration, and there are 25 controls per control step which results in a total of $15 \times 25 = 375$ controls to be optimized. Water is injected at a constant pressure of 300 bars and the production wells are operated at a minimum pressure of 15 bars. We used an oil price $r_o = 130$ \$/m³, water production costs $r_{wp} = 25$ \$/m³, and water injection costs $r_{wi} = 6$ \$/m³. Well index multipliers were used to model the ICVs in the simulator with bounds of 1×10^{-4} and 1, where the finite lower bound was chosen to avoid numerical problems in the simulator associated with a zero lower bound. For the simulation of the model we used a commercial fully implicit finite difference black oil simulator (Eclipse, 2011).

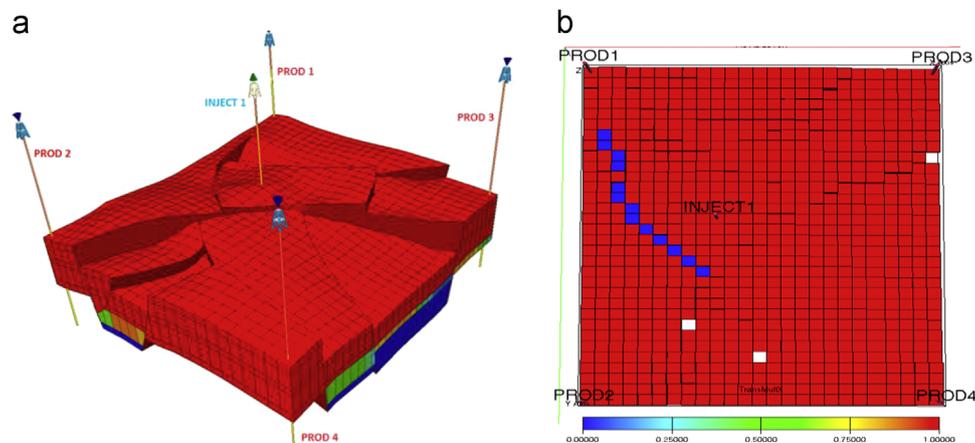


Fig. 2. (a) Five-spot reservoir model. The colors indicate the initial oil saturation. (b) Transmissibility multiplier values for the model. One fault has a zero transmissibility (blue) thus is sealing. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Reservoir and fluid properties.

Property	Values	Units
Porosity	0.2	–
Permeability (layer 1–layer 5)	100-300-50-600-100	mD
Reservoir pressure @1950 m	200	bar
Density of oil	800	kg/m ³
Density of water	1000	kg/m ³
Temperature	77	°C
Oil compressibility @200 bar	4e-5	1/bar
Water compressibility @200 bar	4e-5	1/bar
Rock compressibility	0	1/bar
Viscosity of oil @1 bar	2	cP
Viscosity of water @1 bar	0.5	cP

4.2. Results for NBI

In this section we compare the use of the NBI method without tracking (method 3 in Fig. 1) and with tracking (method 4 in Fig. 1). Following Liu and Reynolds (2014), we obtain the normal vector \mathbf{n} by setting the second component of \mathbf{n} to 1 and solving the following equation

$$\mathbf{n}^T [\mathbf{j}(\mathbf{u}_1^*) - \mathbf{j}(\mathbf{u}_2^*)] = 0 \quad (21)$$

In Eq. (21), $\mathbf{j}(\mathbf{u}_1^*) = [9.1060 \times 10^9, 3.3522 \times 10^9]^T$ and $\mathbf{j}(\mathbf{u}_2^*) = [8.7086 \times 10^9, 4.4759 \times 10^9]^T$. The optimization is not dependent on the choice of \mathbf{n} . The solution of this equation gives $\mathbf{n} = [2.822, 1]^T$, which is the same for all the different starting points used in this work. Solving multiple NBI sub-problems for different choices of weight combinations, we obtain the solutions shown in Fig. 3. The black circles are obtained for starting points based on the first step of the NBI algorithm presented previously. The objective of this initialization is to obtain a starting point on or close to the utopia line. Due to the non-linearity of the problem the objective function values achieved for this initial guess are never on the utopia line, but always slightly above the line. Using the solutions already obtained we also test the applicability of finding solutions which satisfy the constraints starting from points (control sets) that have previously satisfied the constraints. This is akin to “tracking” points along a front. The red circles in Fig. 3 are the points achieved when the tracking process begins from $\beta_1 = 0.1$ (point A in Fig. 3). We observe that for most of the weight combinations, the tracking procedure achieve solutions that dominate

Table 2
Objective function values of the black circles in Fig. 3.

β_1	β_2	Long-term objective ($\times 10^9$) \$	Short-term objective ($\times 10^9$) \$
1	0	9.1060	3.3522
0.9	0.1	9.0952	3.4757
0.8	0.2	9.0749	3.5938
0.7	0.3	9.0619	3.7161
0.6	0.4	9.0472	3.8363
0.5	0.5	9.0414	3.9645
0.4	0.6	9.0266	4.0854
0.3	0.7	9.0174	4.2075
0.2	0.8	9.0123	4.3293
0.1	0.9	8.9750	4.4430
0.08	0.92	8.8760	4.4336
0.02	0.98	8.7609	4.4692
0	1	8.7086	4.4759

the solutions represented by the black circles. Since there is no preference to choose from which end the tracking begins, we also began the tracking from $\beta_1 = 0.9$ (point B in Fig. 3), to obtain the solutions shown by blue circles in Fig. 3. We observe that in this case the tracking procedure achieves solutions that dominate the solutions from the other two initialization procedures for all the points. Additionally this tracking procedure is computationally more efficient as is discussed later. Thus, different initial guesses for a given value of β_1 can have a significant impact on the solutions achieved with the bi-objective optimization algorithm. Besides the different starting points, all other algorithmic details are exactly the same for the three different sets of points generated. The gradients are estimated with an ensemble size equal to 30 with a perturbation size equal to 0.001.

Table 2 provides the objective function values for 11 different optimum points (black circles) along a boundary front. We observe that for the $\beta_1 = 0.4$ case, we obtain a 0.8% decrease in the primary, long-term objective function from its optimal value (9.1060×10^9 \$) and an approximately 22% increase in the secondary, short-term objective function. For the solution obtained with initial guesses based on the front tracking procedure for $\beta_1 = 0.1$ we observe a 0.8% decrease in the primary objective to achieve a 33% increase in the secondary objective. There is only a 5% difference in the primary objective function values between the optimal strategies for the two objective functions J_1 and J_2 , i.e. the first and last points in Table 2. Thus, for the objective functions chosen in this study, we do not expect to observe major increases in primary (long-term) objective for minor decreases in the secondary (short-term) objective, indicating that there may exist fewer redundant degrees of freedom in the short-term objective function.

Independent of the method used to generate the initial guess of a given β_1 , the approximate Pareto front generated with NBI (Fig. 3) shows that one can obtain a sharp increase in the secondary objective function for a very minimal decrease in the primary objective. Moreover, it seems that for this case that the Pareto front consists of two branches; a near-horizontal one near the optimal secondary objective and a near-vertical one near the optimal primary objective. Fig. 4(a) is an illustration of the evolution of the Lagrangian function through the iteration process. The sharp drop in the value of the Lagrangian function corresponds to an update (decrease) of the penalty parameter μ in the augmented Lagrangian method. In most of the cases we observe that we generally perform 5 outer loop iterations in which we update the penalty parameter for 3 iterations and the vector of Lagrange multipliers λ for the remaining two iterations. Fig. 4(b), right-side plot, shows the constraint violation throughout the optimization process. Note that the constraint violation must be less than the given tolerance specified in the optimization algorithm to obtain convergence for the outer loop of the augmented Lagrangian algorithm. When the inner loop

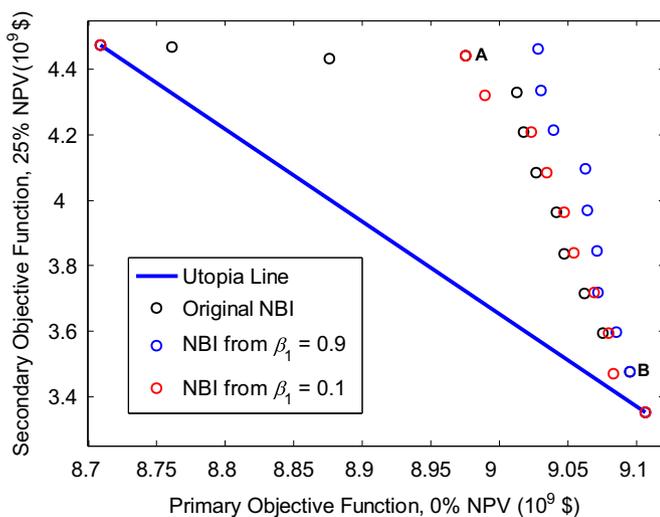


Fig. 3. Boundary points achieved using the NBI method starting from different initial guesses. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

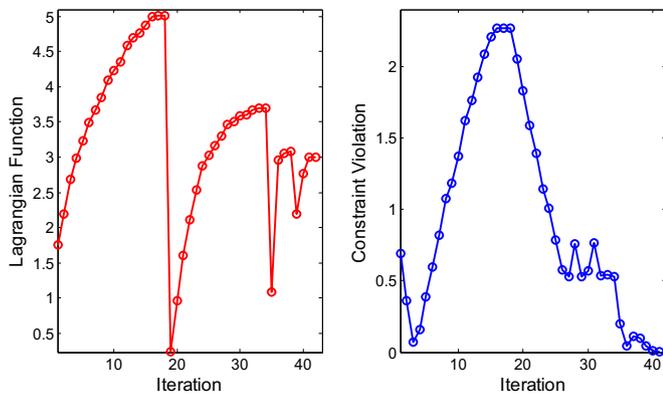


Fig. 4. Illustration of the evolution of the Lagrangian function and constraint violation at all the iterations.

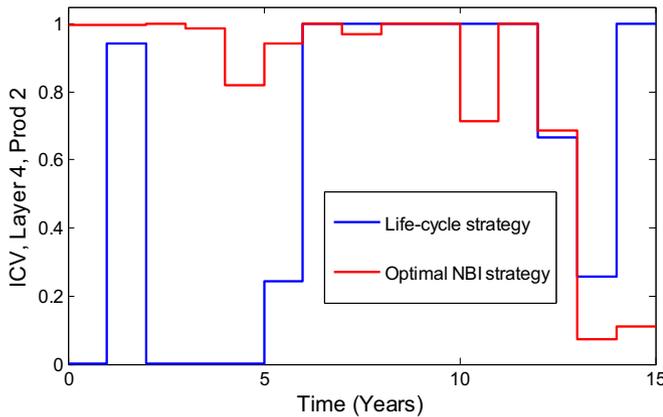


Fig. 5. Comparison of the controls (ICV settings) for layer 4 (highest permeability) for producer 2 over time for the optimal life-cycle strategy (blue) and the optimal NBI strategy (red) for $\beta_1=0.1$ using the front tracking procedure. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

converges and the constraint violation is sufficiently small the algorithm converges. Thus it is possible that there are multiple points at which the constraint violation is satisfied, however there is only one point at which both the inner and outer loop’s stopping criteria are satisfied.

Fig. 5 depicts a comparison of the optimal control settings for two different strategies for the highest permeability layer in producer 2. The blue line is the life-cycle strategy, i.e. one end point of

the utopia line, while the red line is the NBI strategy (blue circles in Fig. 3) obtained for weight combinations of $\beta_1=0.1$ and $\beta_2=0.9$, i.e. the strategy that achieved a 33% increase in the short-term objective for a 0.8% decrease in the long-term objective. From Fig. 5 we observe that significantly different strategies can be achieved by performing bi-objective optimization. For the red curve, the ICV setting is almost fully open for the first 10 years with lower setting values towards the end of the producing time period, which is in line with the emphasis on increasing the short-term increase in NPV. For the optimal life-cycle strategy, the same ICV is almost closed for four or the first five years and then is fully open through most of the remaining producing life, in order to virtually maintain the goal of life-cycle optimization. The optimized control settings for other valves are similar to the trend shown in Fig. 5. Fig. 6 shows the saturation distribution in layer 4 after 4 years of production for the different optimal strategies whose controls are compared in Fig. 5. We see that the optimal life-cycle strategy, being less aggressive, sweeps a much smaller area with less water being injected, while the optimal NBI strategy, i.e. the one for $\beta_1=0.1$ using the front tracking procedure, is more aggressive, i.e. more is water injected and more oil is displaced and produced.

4.3. Comparison of weighted sum techniques

In this section we compare the adjusted sum method (method 1 in Fig. 1) and the adjusted weighted sum method (method 2 in Fig. 1). Liu and Reynolds (2014) showed cases where the adjusted weighted sum method produces a significantly better spread of solutions compared to the traditional weighted sum technique. Tables 3 and 4 provide the solutions for the various weight combinations used where we observe, as reported in Liu and Reynolds (2014), that the adjusted weighted sum technique provides a better spread of solutions, and in particular gives a better representation of the front near the optimal long-term NPV which is the most important part of the front.

Fig. 7 provides a visual comparison of the solutions obtained with the two different methods. Note: The stopping criterion used to achieve this set of points is exactly the same as the stopping criterion used for the inner loop in the augmented Lagrangian-based NBI method albeit the objective functions are different.

4.4. Comparison of weighted sum and NBI

Fig. 8 is a comparison of the solutions achieved from the adjusted weighted sum method and the best results achieved with

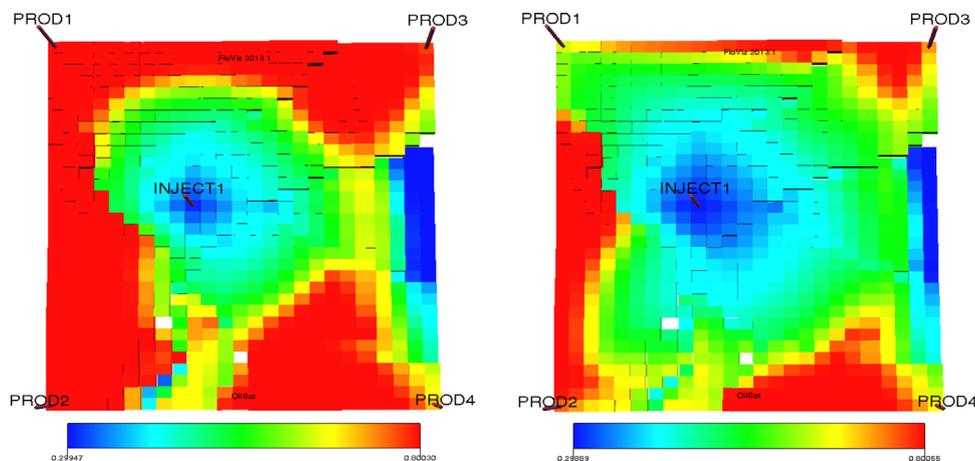


Fig. 6. Saturation distribution in layer 4 after 4 years of production for (a) (left), optimal life-cycle strategy and (b) (right) optimal NBI strategy for the controls shown in Fig. 5. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

Table 3
Solutions for different weight combination using the weighted sum method.

w_1	w_2	Long-term objective ($\times 10^9$) \$	Short-term objective ($\times 10^9$) \$
1	0	9.1060	3.3522
0.9	0.1	9.0851	3.6872
0.8	0.2	9.0419	3.9719
0.7	0.3	8.9720	4.1763
0.6	0.4	8.9156	4.3591
0.5	0.5	8.8696	4.4106
0.4	0.6	8.7873	4.4313
0.3	0.7	8.7538	4.4476
0.2	0.8	8.7403	4.4448
0.1	0.9	8.7243	4.4492
0	1	8.7086	4.4759

Table 4
Solutions for different weight combination using the adjusted weighted sum method.

w_1	w_2	Long-term objective ($\times 10^9$) \$	Short-term objective ($\times 10^9$) \$
1	0	9.1060	3.3522
0.9	0.1	9.1020	3.4502
0.8	0.2	9.0918	3.5996
0.7	0.3	9.0758	3.7522
0.6	0.4	9.0433	3.9326
0.5	0.5	8.9927	4.1400
0.4	0.6	8.9515	4.2268
0.3	0.7	8.8956	4.3905
0.2	0.8	8.8040	4.4384
0.1	0.9	8.7454	4.4423
0	1	8.7086	4.4759

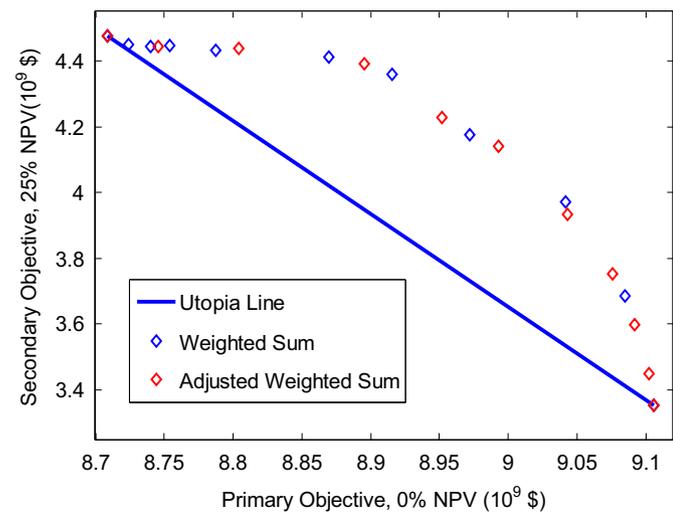


Fig. 7. Comparison of the spread in points along the Pareto front for the two variants of the weighted sum method.

the NBI method. The results here are very interesting: for $w_1=0.9$ the solutions obtained with either method do not dominate each other while for $w_1=0.8$ and $w_1=0.7$ we observe that the adjusted weighted sum method achieves solutions which slightly dominate the solutions obtained with NBI. However, for the other weight combinations the solutions obtained with NBI dominate. It is difficult to know why this behavior is observed and it could be either case dependent or gradient quality dependent. However, Liu and Reynolds (2016) also find that the NBI method generally gives a better representation of the front than is obtained with the weighted sum method. Fig. 9 is a comparison of the optimization path for the different methods with the weight combination $w_1=0.7$ and $w_2=0.3$. The original NBI and the weighted sum have

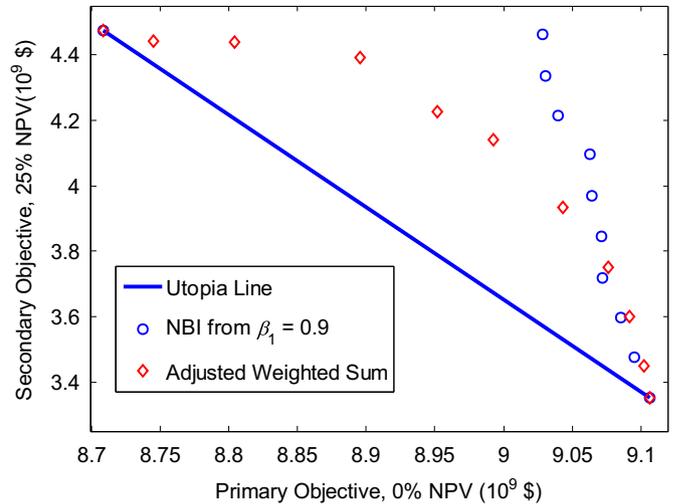


Fig. 8. Comparison of the points achieved from the adjusted weighted sum method and the NBI-based tracking method.

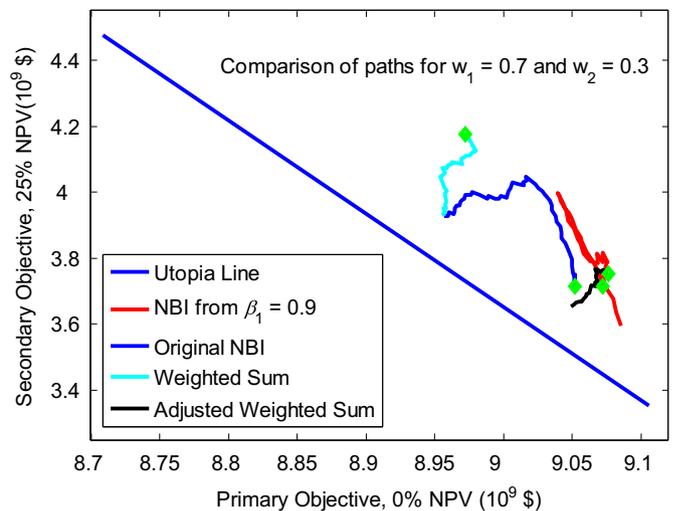


Fig. 9. Comparison of the optimization paths for the different methods described above. The green diamonds are the end points, i.e. the optimum solutions obtained by each of the methods. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the same starting point, however they have very different paths. The adjusting of the weights in the adjusted weighted sum method leads to a significantly different starting point for the optimization. All the optimization results shown here are influenced not only by the gradient quality, but for the NBI method, also by the choices of the initial penalty parameter μ and Lagrange multipliers λ . Using a larger ensemble size for the gradient estimate (Fonseca et al., 2015) could lead to smoother optimization paths to solve the individual sub-problems and possibly better solution points, however for computational reasons this has not been investigated.

Fig. 10 is an illustration of the total number of simulations performed for each of the methods including the two initial optimization runs to obtain the utopia line. The original NBI method was computationally the most expensive one with approximately 28,000 total simulations, while for the NBI tracking method about half the number of simulations required for the original NBI method were needed to achieve better solutions. Both the weighted sum variants were computationally much more efficient, similar to the results obtained in Liu and Reynolds (2014). We note that ensemble methods to generate approximate gradients, such

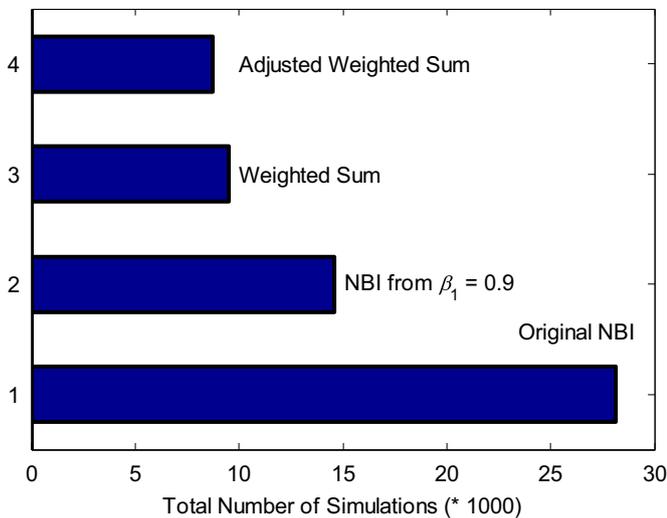


Fig. 10. Comparison of the total number of simulations taken to generate the Pareto curve for the different methods.

as EnOpt, are well suited for embarrassingly parallel processing. Moreover, when introducing geological uncertainty, in the form of different geological realizations, the computational load of ensemble methods becomes relatively less disadvantageous as compared to adjoint-methods; see Fonseca et al. (2015). We also note that the use of ensemble methods to compute approximate gradients involves various user-defined choices such as the ensemble size and the definition of the co-variance matrix (which may be used to enforce temporal correlations between the controls). Moreover a wide variety of ensemble gradient formulations is available; see, e.g., Fonseca et al. (2015) for an overview. In the present study we did not address these aspects, and we refer to Fonseca et al. (2015) for further information on the effects of the various formulations and user-defined parameters.

4.5. Hierarchical switching optimization method

In this section we compare the results of the NBI and weighted sum methods with those obtained with hierarchical switching (method 5 in Fig. 1). The switching method optimizes the objectives alternately, while staying within a maximum allowable decrease ε in the primary objective. The choice of ε is user dependent. Thus, the user has to a-priori decide the maximum allowable acceptable decrease in the optimal primary objective function value. Fig. 11 plots the optimization path where a maximum decrease of 0.3% in the primary objective is allowed (red curve). We see that we achieve approximately a 10% increase in the secondary objective. The values obtained are similar to using a weight combination of $w_1=0.7$ and $w_2=0.3$ for either the NBI method or the adjusted weighted sum method. However the solution is a non-dominated point when compared to all solutions obtained with the NBI and adjusted weighted sum methods. If the optimization is repeated for a 1% allowable decrease in the primary objective we observe that we achieve a 20% increase in the secondary objective function (black dotted curve). This solution however is dominated by the solutions from the other two methods. The hierarchical method only provides a single strategy which may or may not be Pareto optimal. A tracking procedure like the one implemented for the NBI method could be used with this method to generate a front. Alternatively we can use the primary objective function as a constraint while optimizing a secondary objective, i.e., a lexicographic approach; for details see Liu and Reynolds (2016).

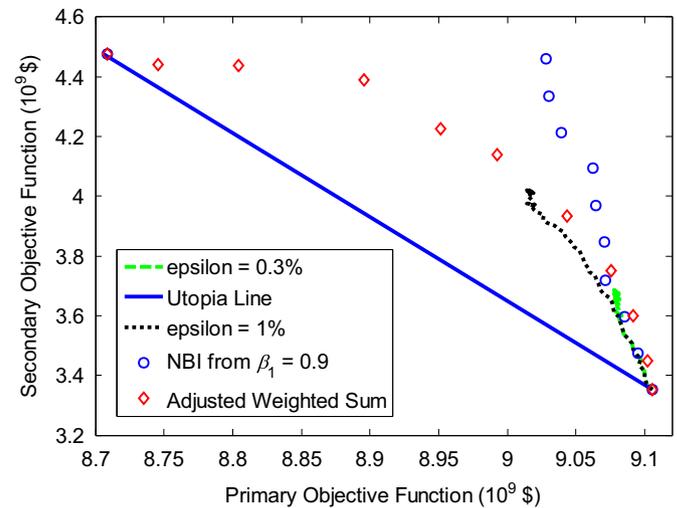


Fig. 11. Paths obtained from hierarchical switching optimization for different maximum allowable decreases in the primary objective function compared to the optimal solutions achieved by the other methods. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

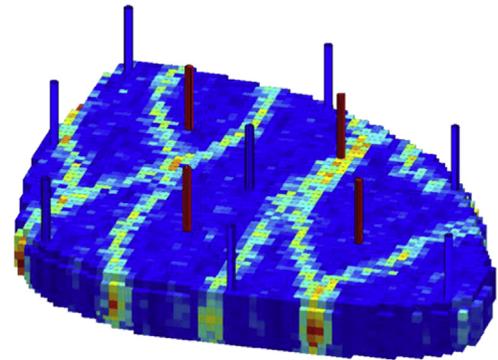


Fig. 12. Permeability field of the reservoir model used for nominal optimization.

5. Example 2: the egg model

5.1. Reservoir model

The model, illustrated in Fig. 12, which was introduced in Van Essen et al. (2009), represents a channelized depositional system in the form of a discrete permeability field modeled with $60 \times 60 \times 7 = 25,200$ grid cells of which 18,553 cells are active. A detailed description along with fluid and geological properties of the standardized version of this “egg model” is given in Jansen et al. (2014). The model is produced using eight peripheral water injection wells (blue) and four production wells (red) which are completed in all seven layers. No capillary pressures are included, the reservoir rock is assumed to be incompressible, and the liquids slightly compressible. The controls to be optimized are the water injection rates with a maximum allowable injection rate per well fixed at $79.5 \text{ m}^3/\text{day}$ and a minimum rate of $0 \text{ m}^3/\text{day}$. The producers are operated at a minimum bottom hole pressure of 395 bars without rate constraints. The producing life of the reservoir is divided into 40 control steps of 90 days each, and therefore the control vector \mathbf{u} has $N=8 \times 40=320$ elements. The NPV parameters used are $r_o=126 \text{ \$/m}^3$, $r_{wp}=19 \text{ \$/m}^3$, and $r_{wi}=5 \text{ \$/m}^3$. The two objectives used are the same as in the previous example. We use a commercial fully implicit black oil simulator (Eclipse, 2011) for the reservoir simulations.

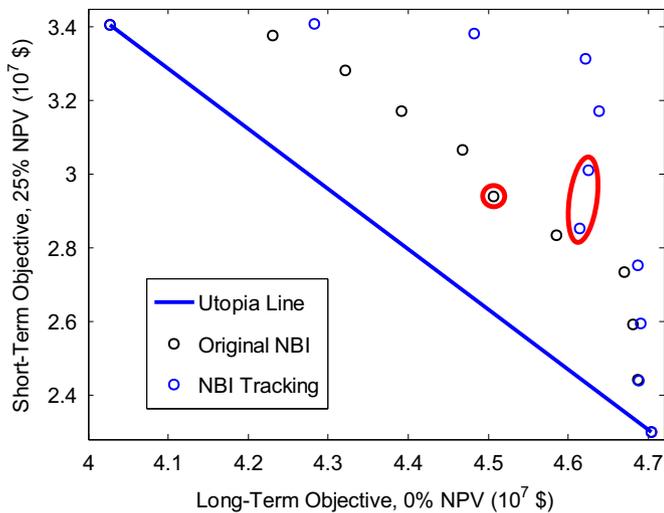


Fig. 13. Boundary points achieved using the NBI method starting from different initial guesses. Note that the points indicated with the red circle and ellipse are not Pareto dominant. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5.2. Results and comparison

For this somewhat larger and more complex example we only compare the use of the NBI methods without and with tracking (methods 3 and 4 in Fig. 1). The normal vector \mathbf{n} is obtained, similar to the example reported above, by setting the second component of \mathbf{n} to 1 and solving Eq. (21) which gives $\mathbf{n}=[1.6329 \ 1]^T$. Where $\mathbf{j}(\mathbf{u}_1^*)=[4.7035 \times 10^7 \ 2.3004 \times 10^7]^T$ and $\mathbf{j}(\mathbf{u}_2^*)=[4.0269 \times 10^7 \ 3.4053 \times 10^7]^T$. For different linearly varying combinations of β we solve multiple NBI sub-problems to find solutions for the two objective functions, as shown in Fig. 14, which satisfy the stopping criteria of the augmented Lagrangian function and the constraint violation. The black circles are obtained for starting points which aim to start on the utopia line. Again, due to the non-linearity of the problem, the objective function values of the starting points are never on the utopia line. The spread in the points found using NBI is more continuous compared to the solutions achieved in the previous example. Again instead of solving the sub-problems from a starting point close to or on the utopia line we aim to “track” points along a front. The blue circles in Fig. 13 are the points achieved when the tracking process begins from $\beta_1=0$. The results illustrated in Fig. 13 seem to suggest that there exists different fronts in the objective function space. We observe that till $\beta_1=0.7$ the points seem to lie on a line with a certain slope for both the original NBI method as well as the NBI tracking method. From $\beta_1=0.6$ onwards the points seem to align themselves along a line with a completely different slope for both the methods. The points obtained with NBI tracking seem to always dominate the solutions obtained with the original NBI method. Thus, similar to the previous example, different starting points of the optimization have significant impact on the solutions achieved. Besides the different starting points, all other algorithmic details are exactly the same for the three different sets of points generated. The gradients are estimated with an ensemble size equal to 30 with a perturbation size equal to 0.01.

With only a 0.7% decrease in the primary (long-term) objective function, we have found a solution that achieves an approximately 19% increase in the secondary (short-term) objective function for the black open circles, i.e. the original NBI method for $\beta_1=0.7$. For the same weight combination, the NBI tracking method finds a solution for which a 0.3% decrease in the long-term objective leads to a 20% increase in short-term gains. Additionally, for the blue

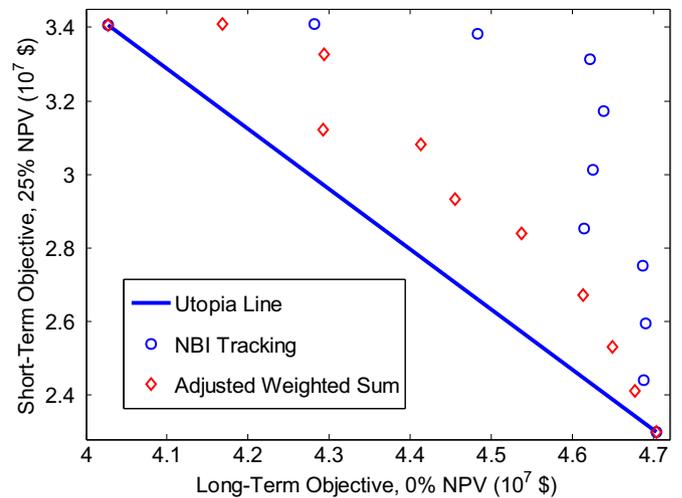


Fig. 14. Comparison of solutions achieved by the adjusted weighted sum method (blue stars) and the NBI tracking method (red circles). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

circles, i.e., NBI with tracking, we observe that for a 1.3% decrease in the primary objective, we can achieve an even more significant increase of 38% in the secondary objective. This last result corresponds to the $\beta_1=0.4$ solution.

The results from the previous example illustrated that the adjusted weighted sum method produced a much better spread in solution points compared to the weighted sum method. Thus for this example we compare the NBI solutions with solutions from the adjusted weighted sum technique. We observe, as shown in Fig. 14, that the solutions achieved by the NBI tracking method dominate the solutions from the adjusted weighted sum method. The solutions from the original NBI method also dominate the solutions from the adjusted weighted sum method. A comparison of the plot of remaining oil saturation for the top layer after 3 years of production illustrates the difference in the strategies; see Fig. 15. The optimal long-term strategy is less aggressive, as significantly less area is swept by injected water, while the NBI tracking solution for $\beta_1=0.4$, i.e. a 38% increase in short-term gains for a 1.3% decrease in long-term gains, is a more aggressive strategy as larger areas of the reservoir have been swept.

The control strategies that resulted in the saturation plots shown in Fig. 15 are illustrated in Fig. 16. The control settings have only been displayed for the first 3 years to highlight the differences between the strategies. We observe that the optimal NBI strategy injects much more water compared to the optimal long-term strategy. A similar trend in the control strategies is seen in the other wells.

Fig. 17 is a comparison of the computational efficiency of the different methods. The original NBI method requires the highest computational effort similar to the results reported for the previous example. We needed about 21,000 simulations to achieve the 11 points for the original NBI method while we needed approximately 14,000 simulations when using the NBI to track the boundary front. The adjusted weighted sum method required less than 8000 simulations to find the 11 points, and is thus the computationally most efficient one, though the solutions achieved are far from Pareto optimal compared to the solutions achieved by the NBI method.

5.3. Discussion

The differences between the results of the adjusted weighted sum and the traditional weighted sum methods for this example

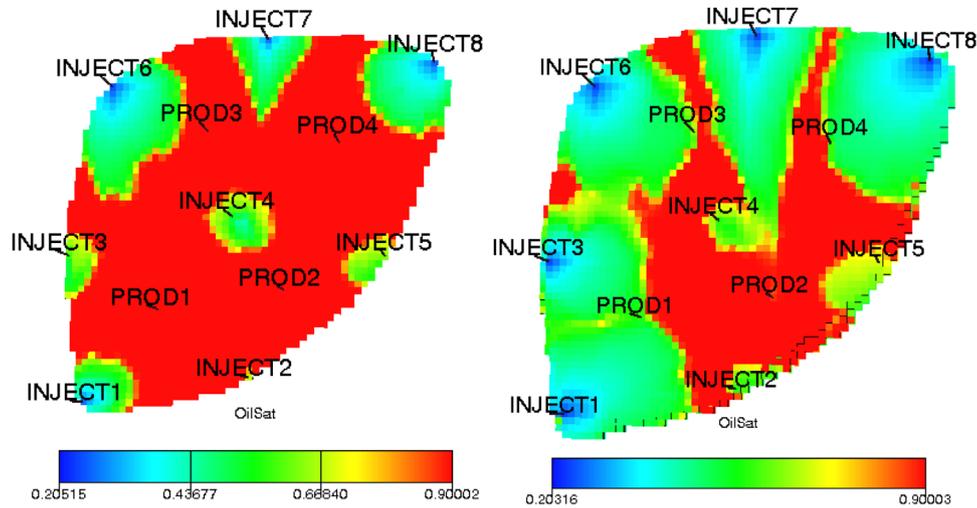


Fig. 15. Comparison of oil saturation distributions for the top layer after 3 years of production for (a) optimal long-term solution (left side) and (b) NBI tracking solution for $\beta_1=0.4$ (right side).

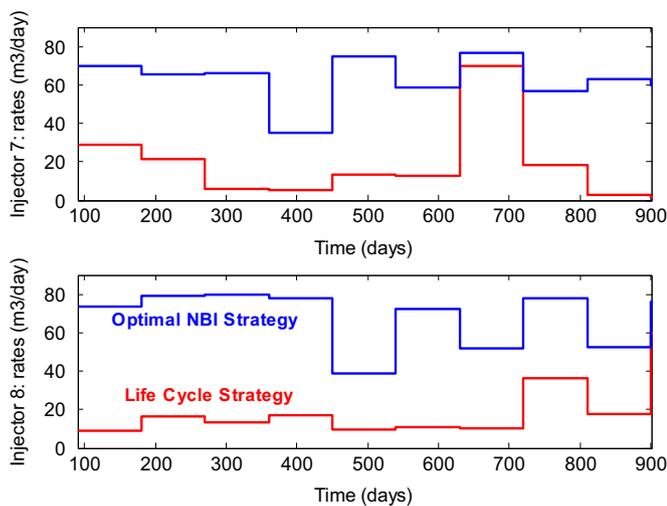


Fig. 16. Comparison of injection rate controls for injector 7 (top) and injector 8 (bottom) for the first 3 years of production resulting from the different strategies, life-cycle strategy (red line) and NBI strategy (blue line). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

are far from significant because the adjustment of the weights does not lead to very different weight combinations, i.e., scaling of the problem is not as important for this problem as it was for the 5 spot ICV problem and the problems investigated in Liu and Reynolds (2014). For this example, the difference between the optimal primary long-term objective function values is 14%, which is much higher than in the previous example, while the difference between the short-term objective function values is 48%.

6. Conclusions

- Approximate gradient techniques like EnOpt can be used to generate solutions for a bi-objective optimization problem which may lie on a Pareto front although the computational costs are significant.
- Tracking the Pareto front using NBI is a computationally more efficient method and produces better solutions for the decision maker to choose from compared to the original NBI method.

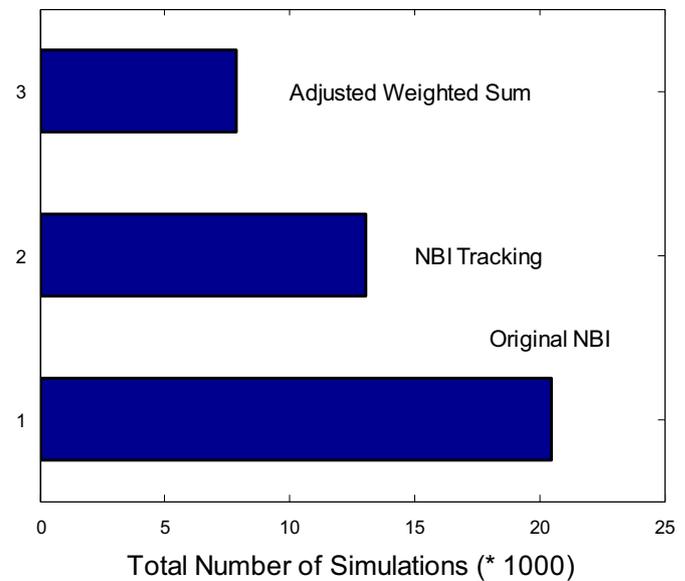


Fig. 17. Comparison of the computational effort required to achieve solutions using the different methods.

Different starting point have a significant impact on the optimal solutions achieved. This is observed for the two different example problems investigated.

- The adjusted weighted sum produces a more even distribution of solutions and is marginally computationally more efficient compared to the traditional weighted sum technique for the ICV control problem.
- For some weight combinations, the NBI method produces solutions which dominate solutions obtained by the weighted sum variants and vice versa for the ICV control problem.
- A hierarchical switching method provides a single solution which satisfies the maximum allowable decrease in the objective function value. For the ICV problem, the single solution point was non-dominated when compared to all the points obtained from the NBI and weighted sum method.

Acknowledgements

We thank the member companies of TUPREP for their support.

This research was carried out within the context of the Integrated Systems Approach to Petroleum Production (ISAPP) knowledge centre. ISAPP is a joint project of TNO, Delft University of Technology, ENI, Statoil and Petrobras. The authors acknowledge Schlumberger for providing academic Eclipse licenses for this work.

References

- Chen, Y., Oliver, D.S., Zhang, D., 2009. Efficient ensemble-based closed-loop production optimization. *SPE J.* 14 (4), 634–645. <http://dx.doi.org/10.2118/112873-PA>.
- Conn, A.R., Scheinberg, S., Vicente, L.N., 2009. *Introduction to derivative-free optimization*. SIAM, Philadelphia.
- Das, I., Dennis, E., 1998. Normal boundary intersection: a new method for generating the pareto surface in nonlinear multi-criteria optimization problems. *SIAM J. Optim.* 8, 631–657.
- Do, S.T., Reynolds, A.C., 2013. Theoretical connections between optimization algorithms based on an approximate gradient. *Comput. Geosci.* 17 (6), 959–973.
- Eclipse. 2011. (<http://www.slb.com/services/software/reseng/eclipse>).
- Fonseca, R.M., Leeuwenburgh, O., Van den Hof, P.M.J., Jansen, J.D., 2014. Ensemble based hierarchical multi-objective production optimization of smart wells. *Comput. Geosci.* 18 (3–4), 449–461.
- Fonseca, R.M., Kahrobaei, S., Van Gastel, L.J.T., Leeuwenburgh, O., Jansen, J.D., 2015. Quantification of the impact of ensemble size on the quality of an ensemble gradient using principles of hypothesis testing. Paper SPE 173236-MS presented at the SPE Reservoir Simulation Symposium, Houston, USA, 23–25 February.
- Isebor, O.J., Durlafsky, L.J., 2014. Bi-objective optimization for general oil field development. *J. Pet. Sci. Eng.* 119, 123–138. <http://dx.doi.org/10.1016/j.petrol.2014.04.021>.
- Jansen, J.D., Douma, S.G., Brouwer, D.R., Van den Hof, P.M.J., Bosgra, O.H., Heemink, A.W., 2009. Closed-loop reservoir management. Paper SPE 119098 presented at the SPE Reservoir Simulation Symposium, The Woodlands, Texas, U.S.A, 2–4 February.
- Jansen, J.D., 2011. Adjoint-based optimization of multi-phase flow through porous media: a review. *Comput. Fluids* 46 (1), 40–51. <http://dx.doi.org/10.1016/j.compfluid.2010.09.039>.
- Jansen, J.D., Fonseca, R.M., Kahrobaei, S., Siraj, M.M., Van Essen, G.M., Van den Hof, P.M.J., 2014. The egg model: a geological ensemble for reservoir simulation. *Geosci. Data J.* 1, 192–195.
- JOA Jewel Suite 2006 Training Manual, version 2.2, August 2007, JOA Oil & Gas B.V., Delft, Netherlands.
- Liu, X., Reynolds, A.C., 2014. Gradient based multi-objective optimization with applications to water flooding optimization. In: Proceedings of the 14th European Conference on the Mathematics of Oil Recovery (ECMOR XIV), Catania, Italy, 8–11 September.
- Liu, X., Reynolds, A.C., 2016. Gradient-based multi-objective optimization for maximizing expectation and minimizing uncertainty or risk with application to optimal well control with only bound constraints. *SPE J.* . <http://dx.doi.org/10.2118/173216-PA>, Published on-line
- Lorentzen, R.J., Berg, A., Naevdal, G., Vefring, E.H., 2006. A New Approach for Dynamic Optimization of Waterflooding Problems. Paper SPE-99690-MS presented at the Intelligent Energy Conference and Exhibition, Amsterdam, The Netherlands, 11–13 April 2006. <http://dx.doi.org/10.2118/99690-ms>.
- Marler, R.T., Arora, J.S., 2004. Survey of multi-objective optimization methods for engineering. *Struct. Multidiscip. Optim.* 26 (6), 369–395. <http://dx.doi.org/10.1007/s00158-003-0368-6>.
- Nocedal, J., Wright, S.J., 2006. *Numerical Optimization*, 2nd ed. Springer, New York.
- Strang, G., 2006. *Linear Algebra and Its Applications*, 4th ed. Thomson Brooks/Cole, Pacific Grove.
- Van Essen, G., Van den Hof, P.M.J., Jansen, J.D., 2011. Hierarchical long-term and short-term production optimization. *SPE J.* 16 (1), 191–199. <http://dx.doi.org/10.2118/124332-PA>.
- Van Essen, G.M., Zandvliet, M.J., Van den Hof, P.M.J., Bosgra, O.H., Jansen, J.D., 2009. Robust waterflooding optimization of multiple geological scenarios. *SPE J.* 14 (1), 202–210. <http://dx.doi.org/10.2118/102913-PA>.