Computational Geosciences (2023) 27:355–364
https://doi.org/10.1007/s10596-022-10185-z

ORIGINAL PAPER

A natural Hessian approximation for ensemble based optimization

Yiteng Zhang[1] · Andreas S. Stordal[2] · Rolf J. Lorentzen[2]

Received: 13 May 2021 / Accepted: 29 November 2022 / Published online: 16 March 2023

Abstract
A key challenge in reservoir management and other fields of engineering involves optimizing a nonlinear function iteratively. Due to the lack of available gradients in commercial reservoir simulators the attention over the last decades has been on gradient free methods or gradient approximations. In particular, the ensemble-based optimization has gained popularity over the last decade due to its simplicity and efficient implementation when considering an ensemble of reservoir models. Typically, a regression type gradient approximation is used in a backtracking or line search setting. This paper introduces an approximation of the Hessian utilizing a Monte Carlo approximation of the natural gradient with respect to the covariance matrix. This Hessian approximation can further be implemented in a trust region approach in order to improve the efficiency of the algorithm. The advantages of using such approximations are demonstrated by testing the proposed algorithm on the Rosenbrock function and on a synthetic reservoir field.

Keywords Stochastic optimization · Ensemble based · Natural gradient · Monte Carlo · Evolution strategy

Mathematics Subject Classification (2010) 65C05 · 65K05 · 90C15

1 Introduction

We consider the unconstrained optimization problem

$$\min_{x\in\mathbb{R}^n} J(x), \tag{1}$$

where $J(x)$ is a differentiable function in $\mathbb{R}^n$. The focus is on reservoir management where J typically represents the Net-Present-Value (NPV). However, the methodology presented here is applicable to any type of optimization problem of moderate size (number of controls not exceeding $O(10^2)$). Production optimization in petroleum science plays a significant role in the return on investment, being of two constituents of closed-loop reservoir management. This usually requires a model-based optimization technique since the workflow involves the prediction of future production [2, 5, 13]. There are a significant number of papers that use the gradient obtained through the adjoint technique [9]. Sarma et al. [13] proposed an approximate feasible-direction algorithm with the help of the adjoint method to handle production optimization problems with nonlinear path inequality constraints. Likewise, Zandvliet et al. [17] used the adjoint method for well placement optimization. This method calculates the gradients of the objective function and the gradients are used subsequently to approximate directions. Forouzanfar et al. [8] developed a gradient-based optimization algorithm, where the gradient is computed by combining the adjoint method and an analytical method for linear constraints to estimate the optimal well location and target rates in the reservoirs. Although the adjoint method has been proven as an efficient way to calculate the gradient, it is a nearly impractical method since it requires access to commercial simulator source code.

Over the last decade stochastic gradient approximations have gained a lot of interest in the petroleum community. The ensemble based optimization (EnOpt) was introduced into the petroleum community by Chen et al. [2] and

✉ Yiteng Zhang
yiteng.zhang@shell.com

✉ Andreas S. Stordal
asto@norceresearch.no

Rolf J. Lorentzen
rolo@norceresearch.no

[1] The National IOR Centre of Norway, University of Stavanger, A/S Norske Shell, Løkkeveien 103, 4007 Stavanger, Norway

[2] NORCE Norwegian Research Centre AS, Postboks 22 Nygårdstangen, 5838 Bergen, Norway

Lorentzen et al. [10] as an optimization equivalent to the ensemble Kalman filter [6], where correlations between random Gaussian input controls and their output are used to approximate a gradient. Do and Reynolds [4] applied a similar technique, the simultaneous perturbation stochastic approximation (SPSA), to estimate optimal well controls in production optimization. The efficiency of EnOpt was improved in [7] by deploying covariance matrix adaption [11], in which the covariance matrix is allowed to change according to the best samples from the ensemble of controls. A theoretical evaluation of EnOpt was presented in [15] where it was shown that EnOpt is a special case of a natural evolution strategy [1], an optimization framework where the standard gradient is replaced by a natural gradient w.r.t. parameters of a Gaussian search distribution. Furthermore, it was shown that for robust optimization (e.g. optimizing the mean NPV over an ensemble of reservoir models) the strategy of pairing one random control with one reservoir realization is theoretically sound in the natural evolution framework. This is a key point for real applications as the number of simulation runs is reduced to the number of reservoir models, and not multiple runs per reservoir model. This makes EnOpt as computationally efficient as the adjoint method for robust optimization.

In this article the work of Stordal et al. [15] is extended and a Hessian approximation is derived using second order natural gradient information. With this second order information we implement a trust-region strategy for EnOpt where we use a preconditioned conjugate gradient method to solve the subproblem of the trust-region algorithm known as Steihaug's approach [14]. Steihaug's approach is based on the preconditioned conjugate gradient method and may be regarded as a generalized dogleg technique where the quasi-Newton step is supplied.

The paper starts with background information about ensemble-based optimization and shows how EnOpt is a special case of a natural evolution strategy. Using the natural evolution strategy, the development of approximate Hessian is also presented in Section 2. As the Hessian matrix is needed in the trust-region approach, Section 3 gives a brief description of the approach. Next the practicalities for the proposed method are presented in Section 4 along with applications on the Rosenbrock function and a reservoir model. Concluding remarks are found in Section 5.

2 Ensemble-based optimization

The EnOpt algorithm starts, as any numerical optimization algorithm, with a vector $\mu_0 \in \mathbb{R}^n$ as the initial values for the control and a covariance matrix Σ that needs to be specified.

$$X = [x_1, x_2, x_3, \ldots, x_n]^\top \tag{2}$$

denotes the vector of the optimization variables at each step, where n is the total number. Initially, an ensemble of control vectors $\{X_0^i\}_{i=1}^N$, are drawn from a multivariate Gaussian density $\Phi(\mu_0, \Sigma)$ and their objective function values, $\{J(X_0^i)\}_{i=1}^N$, are evaluated. Originally, the EnOpt algorithm updated the mean control vector μ_k at each iteration as

$$\mu_{k+1} = \mu_k + \beta_k \Sigma \nabla_{\mu_k} J,\ k = 1, 2, \ldots \tag{3}$$

where k denotes the iteration number, μ_k is the current control, β_k is the step size and Σ is the covariance matrix of the Gaussian distribution where the ensemble is drawn from. The preconditioned gradient is then approximated as

$$\Sigma \nabla_{\mu_k} J \approx \frac{1}{N-1} \sum_{i=1}^{N} (X_k^i - \overline{X_k})(J(X_k^i) - \overline{J(X_k^i)}), \tag{4}$$

where $\{X_k^i\}_{i=1}^N$ is an i.i.d. sample from $\Phi(\mu_k, \Sigma)$. It was shown in [15] that asymptotically, as N goes to infinity, the right hand side of Eq. 4 converges to

$$\Sigma \mathbb{E}[\nabla_{\mu_k} J(X)],$$

which is the natural gradient [1] of the objective function

$$L(\mu) = \mathbb{E}[J(X)].$$

In general, natural evolution searches for the minimum of the expected objective function w.r.t. a location parameter of the search distribution. That is

$$L(\mu) = \mathbb{E}[J(X)] = \int J(x) f(x; \mu)\, dx, \tag{5}$$

where f is a probability density that depends on the parameter μ. Both [1] and [16] pointed out that the ordinary gradient of Eq. 5 w.r.t μ does not account for gradient uncertainty and depends on the particular parameterization of the distribution, leading to unstable updates. The essence of the natural gradient is to remove this dependence on the parameterization by multiplying the gradient with the inverse of the Fisher information matrix, which is Σ in our case. From the natural evoluation point of view, Stordal et al. [15] redefined the EnOpt algorithm, whithout Σ as a pre-conditioner in Eq. 3, and with an additional update equation for Σ. Given a parametric family of multivariate Gaussian distributions $\Phi(x|\mu, \Sigma)$, the objective function in the natural evolution is defined as

$$L(\mu, \Sigma) = \mathbb{E}[J(X)] = \int J(x) \Phi(x|\mu, \Sigma)\, dx,$$

with corresponding natural gradients

$$\widetilde{\nabla}_\mu L = \mathbb{E}[J(x)(x-\mu)], \tag{6}$$

$$\widetilde{\nabla}_\Sigma L = \mathbb{E}\left[J(x)\left((x-\mu)(x-\mu)^\top - \Sigma\right)\right], \tag{7}$$

where the natural gradient is defined as the inverse of the Fisher information matrix multiplied with the traditional gradient. For details of the derivation the reader is referred to [1].

Unfortunately, the expectations in Eqs. 6 and 7 are not available analytically in general. Instead, an approximation of the natural search gradients are obtained from Monte Carlo samples $\{X^i\}_{i=1}^N$ as

$$\widetilde{\nabla}_\mu L \approx \frac{1}{N}\sum_{i=1}^{N} J(X^i)(X^i-\mu), \tag{8}$$

$$\widetilde{\nabla}_\Sigma L \approx \frac{1}{N}\sum_{i=1}^{N} J(X^i)\left((X^i-\mu)(X^i-\mu)^\top-\Sigma\right). \tag{9}$$

The EnOpt in a natural evoluation setting evolves as

$$\mu_{k+1} = \mu_k + \beta_k^1 \widetilde{\nabla}_{\mu_k} L, \tag{10}$$

$$\Sigma_{k+1} = \Sigma_k + \beta_k^2 \widetilde{\nabla}_{\Sigma_k} L, \tag{11}$$

where β_k^1 and β_k^2 are the step sizes. We also note that often $J(X_k^i)$ is replaced by $J(X_k^i) - J(\mu_k)$ or $J(X_k^i) - \overline{J}$ for stability. Furthermore, it is also quite common to normalize the gradients in order to avoid tuning the step sizes on a case by case basis.

The above Monte Carlo estimates of the covariances suffer from sampling errors due to the limited sample size computationally available for reservoir simulations (typically 50-100). Unlike data assimilation methods, where a type of covariance localization is used to reduce the impact of sampling error (typically via a tapering function or domain localization), the NPV objective function used in the EnOpt algorithm does not have this "local" feature. This renders standard localization techniques from the data assimilation literature useless for EnOpt. Fortunately, the EnOpt gradient consists of a single vector of covariance estimates (not a matrix) of order $O(10^2)$ or less for most reservoir problems so the need for reducing the sampling error is not nearly as severe as for data assimilation problems where the covariance matrices can be of order $O(10^6)$ and higher. There are, however, a few approaches that can help reducing the sampling errors in Eqs. 8 and 9. For the gradient w.r.t. μ, a pre-conditioning matrix can be used to smooth the gradient estimate and hopefully average out some of the errors [2]. Another possible approach is to use truncation, where all estimates below a certain threshold in absolute value are set to zero. For the gradient w.r.t. Σ, a classical shrinkage towards the identity matrix is one approach. The second approach, which is implemented in the reservoir example presented later, is to only update the variances (the diagonal of Σ) while keeping the correlation structure fixed. The latter approach significantly reduces the number of covariances to be estimated.

We now extend the EnOpt by including a natural Hessian approximation that can be applied in both backtracking and trust-region algorithms to show later how the performance of an optimization can be improved by having second order gradient information.

2.1 A natural Hessian approximation

In view of the natural evolution theory presented above, a formulation of a natural Hessian w.r.t. μ is presented.

From Eq. 6

$$\begin{aligned}
&\widetilde{\nabla}_\mu(\widetilde{\nabla}_\mu L)^\top \\
&= \Sigma\nabla_\mu(\widetilde{\nabla}_\mu L)^\top \\
&= \Sigma\nabla_\mu(\mathbb{E}[J(x)(x-\mu)^\top]) \\
&= \Sigma\nabla_\mu \int J(x)\Phi(x|\mu,\Sigma)(x-\mu)^\top \mathrm{d}x
\end{aligned}$$

Then using the chain rule and the log trick ($\nabla \log f(x) = \nabla f(x)/f(x)$) we get

$$\begin{aligned}
&= \Sigma\int J(x)\left(\nabla_\mu \log\Phi(x|\mu,\Sigma)(x-\mu)^\top - \mathbf{I}\right)\Phi(x|\mu,\Sigma)\mathrm{d}x \\
&= \Sigma\int J(x)\left(\Sigma^{-1}(x-\mu)(x-\mu)^\top - \mathbf{I}\right)\Phi(x|\mu,\Sigma)\mathrm{d}x \\
&= \int J(x)\left((x-\mu)(x-\mu)^\top - \Sigma\right)\Phi(x|\mu,\Sigma)\mathrm{d}x \\
&= \mathbb{E}\left[J(X)\left((X-\mu)(X-\mu)^\top - \Sigma\right)\right].
\end{aligned}$$

Note that this is the exact same expression as the natural gradient w.r.t. Σ in Eq. 7. And so a Monte Carlo approximation of the natural Hessian matrix is given by Eq. 9 and is already available in the EnOpt algorithm. No extra computation is needed. A more detailed description of EnOpt with backtracking strategy and trust-region strategy utilizing the natural Hessian approximations are described later in Algorithms 1 and 2, respectively. The trust-region approach is described in more details in the next section.

3 Trust-region approach

We formulate the trust-region approach in terms of minimization, hence we seek to minimize the negative NPV instead of maximizing the NPV. In trust-region approaches, the objective function, J, is replaced by a quadratic form m_k at each iteration k using the first two terms of the Taylor-series expansion of J around μ_k. Let

$$m_k(p) = J_k + g_k^\top p + \frac{1}{2}p^\top B_k p, \tag{12}$$

where p is the trial step, $J_k = J(\mu_k)$, g_k is the gradient of J at the current point and B_k is a real symmetric $n \times n$ matrix

(typically the Hessian). At the current iterate point the trial step is computed by solving the sub-problem

$$\min_{p\in\mathbb{R}^n} m_k(p) = J_k + g_k^\top p + \frac{1}{2} p^\top B_k p, \quad \text{s.t. } \|p\| \le \Delta_k, \tag{13}$$

where $\Delta_k > 0$ is the trust-region radius. Having determined the trial step, the objective function is now computed at $\mu_k + p$ and then compared to the value predicted by the approximate model at this point. By doing so, this information is further used to adapt the trust region per iteration.

Here, the Steihaug approach [14] is used where the algorithm terminates when it exits the trust-region $\|p\| > \Delta$ or when it encounters a direction of negative curvature in B. The approximate derivatives are then used in the algorithm to update the search direction. In particular, after the first iteration, we have the Cauchy point,

$$p_1 = -\frac{g^\top g}{g^\top B g} g, \tag{14}$$

the point lying on the gradient that minimizes the quadratic model. By iteratively finding the Cauchy point the local minimum can be found. Having determined the trial step, p_k, the objective function is now computed at

$$\mu_{k+1} = \mu_k + p_k. \tag{15}$$

If the reduction predicted by the approximate model $m_k(\mu_k + p_k)$ is realized by the objective function $J(\mu_k + p_k)$, the trial point p_k is accepted and the trust-region is updated.

Up to this point all necessary theories are presented, we will demonstrate the advantages of using the proposed method in the next section.

Step 0: Initialization.
Set $k = 0$, a tolerance ε and maximum number of inner and outer iterations
Given an initial point x_0 and an initial step size β_0^1 and β_0^2.
Step 1: Sampling.
Sample N i.i.d. $\{X_k^i\}$ random variables from $\mathcal{N}(x|\mu_k, \Sigma_k)$.
Step 2: Calculate the gradient.
Calculate the gradients $\widetilde{\nabla}_\mu L$ and $\widetilde{\nabla}_\Sigma L$ from Eqs. 8 and 9.
Step 3: Update the mean ensemble:
$\mu_{k+1} = \mu_k + \beta_k^1 (\widetilde{\nabla}_{\Sigma_k} L)^{-1} \widetilde{\nabla}_{\mu_k} L$.
Step 4: Check update:
if $J(\mu_{k+1}) - J(\mu_k) > \varepsilon$, continue to step 5
else reduce β_k^1 by 1/2.
If maximum inner iterations reached: exit
Step 5: Update the covariance matrix:
$\Sigma_{k+1} = \Sigma_k + \beta_k^2 \widetilde{\nabla}_{\Sigma_k} L$.
Restore β_k^1 and to the initial step size β_0^1 .
Increment k by 1.
If $k >$maximum outer iterations: exit.
Else: go to Step 1.

Algorithm 1 Basic backtracking strategy with natural Hessian approximations.

4 Numerical experiments

In this section two examples to show the performance of the natural Hessian approximation for ensemble-based optimization are presented. The first example, a two-dimensional test function, is to demonstrate the improved convergence rate by comparing three different versions of EnOpt with a more conventional gradient based method; while the second example presents a large-scale reservoir optimization problem.

4.1 Two-dimensional valley-shaped function

The objective function to be minimized is the two-dimensional Rosenbrock function

$$J(x, y) = (1 - x)^2 + 100\left(y - x^2\right)^2,$$

where the global minimum, $J(x^*, y^*) = 0$ at $(x^*, y^*) = (1, 1)$, is inside a long, narrow, parabolic shaped almost flat valley. This valley is trivial to find, however, convergence to the global minimum is difficult.

Four different optimization scenarios are: (i) trust-region strategy with true gradient and Hessian of Rosenbrock function (TR-grad); (ii) trust-region strategy with natural gradient and Hessian (TR-ens); (iii) backtracking strategy with natural gradient and Hessian (BT_H); (iv) simple backtracking without Hessian, i.e., steepest descent line search method without Hessian (BT). Results are in Fig. 1. Each scenario is simulated 100 times with 100 different starting points uniformly distributed in the function domain interval $[-5, 10]$. The covariance matrix Σ is initially set as a diagonal matrix with 0.01 on the diagonal. The steps sizes are set to $\beta_0 = 0.001$ for the trust-region approach and $\beta_0^1 = 0.1$ and $\beta_0^2 = 0.001$ for the backtracking. The trust-region parameters are set to $\gamma_1 = 0.5, \gamma_2 = 2, \eta_1 = 0.25, \eta_2 = 0.75$ and $\Delta_0 = 3$. The algorithms are run until convergence, that is until $\|\mu_k - (1, 1)^\top\| < 10^{-3}$. All ensemble based methods use the same initial ensemble.

Not surprisingly, among these four scenarios, the one using the analytic gradients has the best result. The least favorable scenario turns out to be the traditional EnOpt that only uses simple backtracking. The results are presented in Figs. 1 and 2. The figures clearly see the advantage of

Step 0: Initialization.

Set $k = 0$. Set maximum number of iterations. Set covariance step size β_0

Given an initial point x_0 and an initial trust-region radius Δ_0.

Let $0 < \eta_1 \leq \eta_2 < 1$, $0 < \gamma_1 < 1$, and $\gamma_2 > 1$.

Step 1: Sampling.

Sample N i.i.d. $\{X_k^i\}$ random variables from $\mathcal{N}(x|\mu_k, \Sigma_k)$.

Step 2: Calculate the gradients.

Calculate $\widetilde{\nabla}_{\mu_k} L$ from Eq. 8 and $\widetilde{\nabla}_{\Sigma_k} L$ from Eq. 9.

Step 3: Step calculation.

Let $g_k = \widetilde{\nabla}_{\mu_k} L$ and $B_k = \widetilde{\nabla}_{\Sigma_k} L$.

Solve p_k from Eq. 13 using Steihaug's approach.

Step 4: Acceptance of the trial point.

Compute $J(\mu_k + p_k)$ and define

$$\rho_k = \frac{J(\mu_k) - J(\mu_k + p_k)}{m_k(\mu_k) - m_k(\mu_k + p_k)}.$$

If $\rho_k \geq \eta_1$, then define $\mu_{k+1} = \mu_k + p_k$;

Else $\mu_{k+1} = \mu_k$. Exit.

Step 5: Update the covariance matrix.

Update the covariance matrix:

$\Sigma_{k+1} = \Sigma_k + \beta_k \widetilde{\nabla}_{\Sigma_k} L$.

Step 6: Trust-region radius update.

Set

$$\Delta_{k+1} = \begin{cases} \max[\gamma_2 \|p_k\|_k, \Delta_k] & \text{if } \rho_k \geq \eta_2, \\ \Delta_k & \text{if } \rho_k \in [\eta_1, \eta_2), \\ \gamma_1 \|p_k\|_k & \text{if } \rho_k \leq \eta_1. \end{cases}$$

Increment k by 1.

If $k >$maximum iterations: exit.

Else: go to Step 1.

Algorithm 2 Basic trust-region strategy with natural Hessian approximations.

including second order derivative information in EnOpt. In terms of iterations, the trust-region strategy significantly outperforms backtracking. Figure 2 shows the summary of the results in a box plot.

In order to address the sensitivity of the initial choice for Σ, the EnOpt with trust region and EnOpt with backtracking including the Hessian were rerun for ten different starting values. For each value, the starting point for μ was kept fixed and each method was run one hundred times to avoid any Monte Carlo effects. The average number of iterations are reported in Fig. 3 along with the value for the variance. The results are similar to the discussion in [15], i.e. the variance should not be set too large initially. However, since the variance is updated at each iteration, the initial value is not as sensitive as for the original EnOpt where the variance is kept fixed.

4.2 Synthetic reservoir model

A synthetic field designed by TNO, Brugge, is used as a benchmark study to test the combined use of waterflooding optimization and history matching methods in a closed-loop workflow [12]. The model is discretized on a $139 \times 48 \times 9$ grid lattice with a total of 60048 cells. The model has a complex geological structure that contains five facies types formed in different depositional environments. The field is equipped with 30 wells, 20 producers and 10 injectors as shown in Fig. 4.

The objective function is defined as the Net-Present-Value (NPV), which can be calculated as

$$J(x) = \sum_{j=1}^{N_t} \frac{\left(v_o \cdot q_{o,j}(x) - v_{wp} \cdot q_{wp,j}(x) - v_{wi} \cdot q_{wi,j}(x)\right) \Delta t_j}{(1+r)^{t_j}}, \tag{16}$$

where the oil production, $q_{o,j}$, water production, $q_{wp,j}$, and water injection rates, $q_{wi,j}$, at time t_j change as a function of the control variable x. The oil price is denoted v_o. The costs of water disposal and water injection are denoted v_{wp} and v_{wi}, respectively. The discount factor is denoted by r and Δt_j is the difference between two timesteps (measured in days).

In this example, the control variables are the sum of the entering reservoir volumes of oil, gas, water for each producer, and the injected water rates. The price of oil is set to \$80/stb. The cost of both water injection and water disposal is \$5/stb. The discount rate is 10% per year. The time frame for the optimization is 20 years and the control settings are modified every year, so the number of control steps is 20. Thus, the total number of control parameters is 600, which is the product of the number of wells and the number of control steps. As a reference case, the producers are set to produce at a rate of 2500 stb/day, with a minimum bottom hole pressure of 725 psia. The injectors have an upper limit in bottom hole pressure of 2610 psia and are operated at a rate of 3500 stb/day.

The performance of EnOpt using Hessian approximations in both trust-region (TR) method and backtracking (BT) method are compared with the original EnOpt method in which the steepest decent direction is applied without using Hessian approximations. In addition, we show two

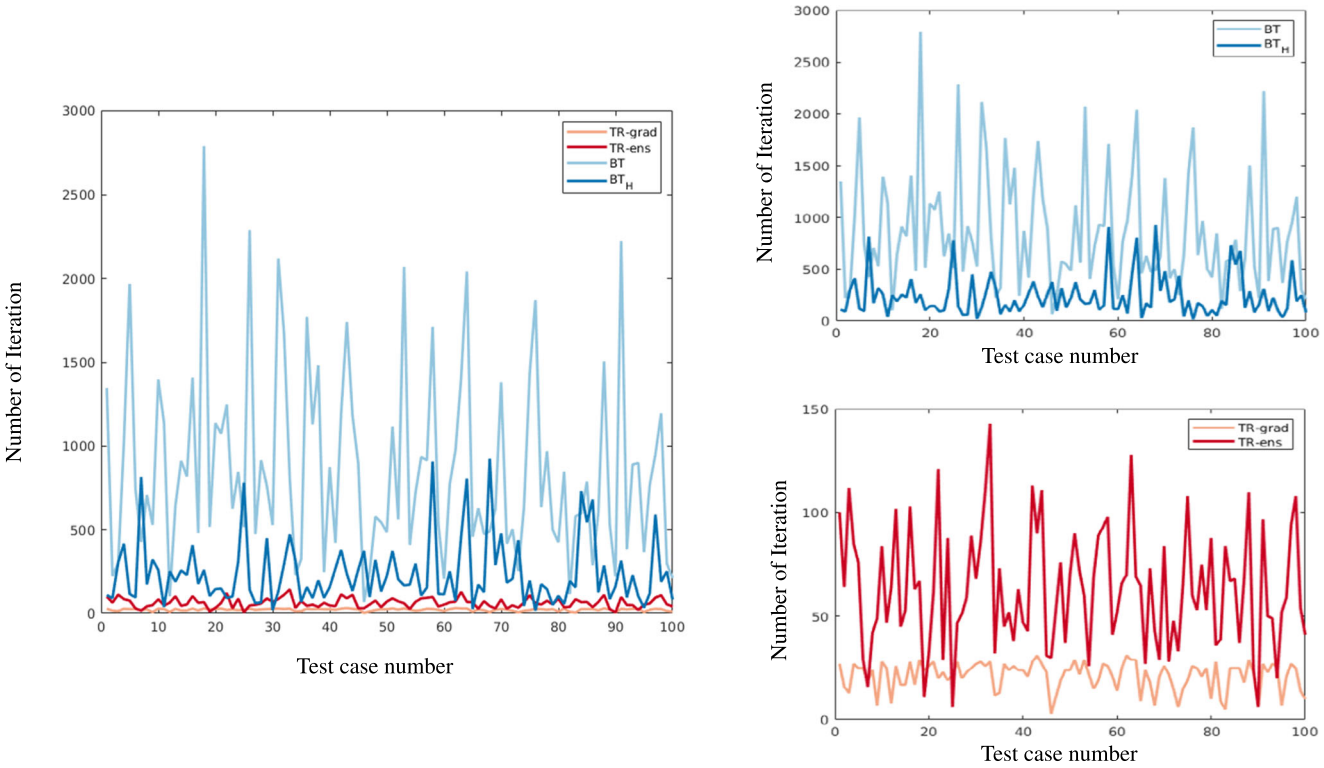

Fig. 1 Number of iterations for 100 simulation runs with the two-dimensional Rosenbrock function: (left) all four scenarios; (right) zoomed in comparison grouped by two optimization strategies

different scenarios for the original EnOpt method by varying the initial step size. To summarize briefly, we show four scenarios in this section:

(a) Trust-region method with Hessian approximations;
(b) Backtracking method with Hessian approximations;
(c) Backtracking method without Hessian approximations (initial step size = 0.05);
(d) Backtracking method without Hessian approximations (initial step size = 0.025).

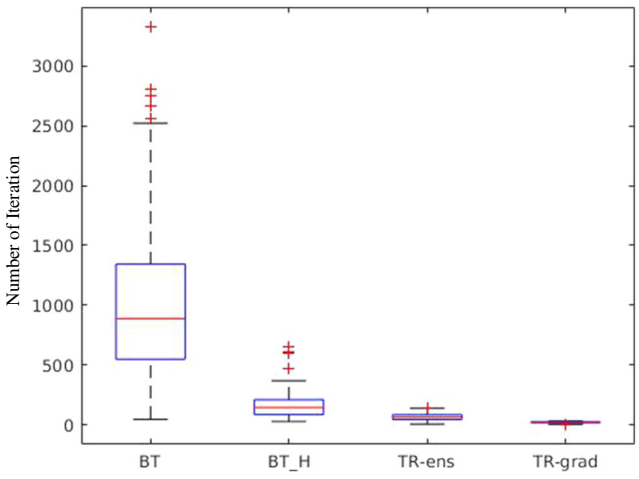

Fig. 2 Box plots of four different scenarios

The same initial ensemble is used for all algorithms.

We construct Σ using an autoregressive model of order 1 for each well. The autocorrelation function for each well at time t is given by:

$$\text{Corr}(Q[t], Q[t+h]) = \rho^{|h|}, \tag{17}$$

where h is an interger s.t. $h \in [0, T-t]$ and T is the total number of control steps and $\rho = 0.5$. There is no correlation between producers and injectors. The variance is given by $\sigma_P^2 = 0.1$ and $\sigma_I^2 = 0.1$ for producers and injectors, respectively. The correlation structure is kept fixed during the optimization procedure, so that only the diagonal of Σ is updated.

One hundred initial ensemble members are generated from a Gaussian distribution with mean given by the reference case. The average NPV for the 100 initial ensemble members is $\$2.8227 \times 10^9$.

Figure 5 shows the change of the NPV with iterations for the four abovementioned scenarios. For the final simulated

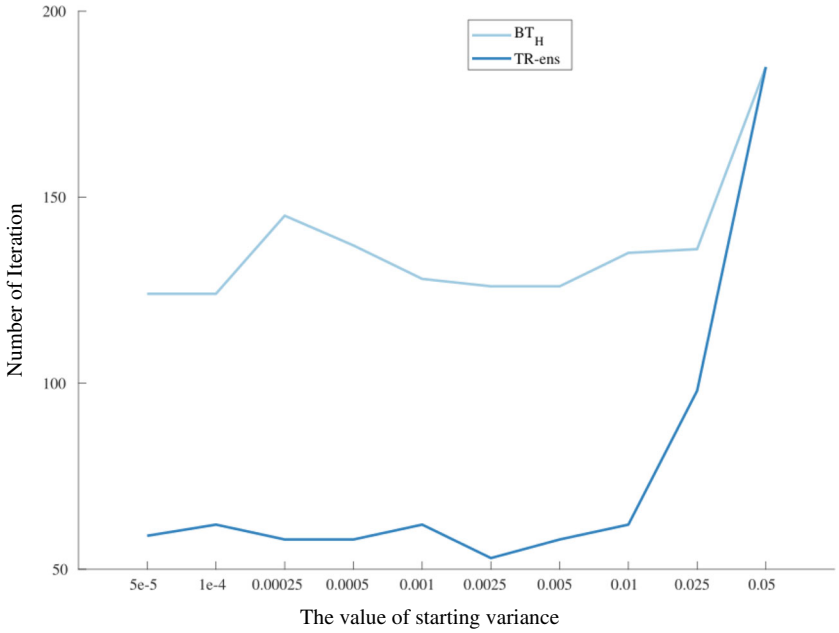

Fig. 3 Average number of iterations for 100 runs with different starting variance

control strategy, the NPVs of (a), (b), (c), and (d) scenarios are \$3.1873 $\times$ 10^9, \$3.0836 $\times$ 10^9, \$2.8567 $\times$ 10^9, and \$2.9081 $\times$ 10^9, respectively, an increase of 12.92%, 9.24%, 1.2%, and 3.03% compared to the initial NPV.

It is noted that all variables are scaled based on the maximum rate of producers $q_{p,max}$ and injectors $q_{i,max}$, respectively. In all four scenarios, the maximum rate is set to 3000 stb/day for producers and 4000 stb/day for injectors. Mathematically, scaled rates at time j of oil production rate, water production rate, and water injection rate are expressed as,

$$Q_{o,j} = \frac{q_{o,j}}{q_{p,max}}, \quad Q_{wp,j} = \frac{q_{wp,j}}{q_{p,max}}, \quad Q_{wi,j} = \frac{q_{wi,j}}{q_{i,max}}. \tag{18}$$

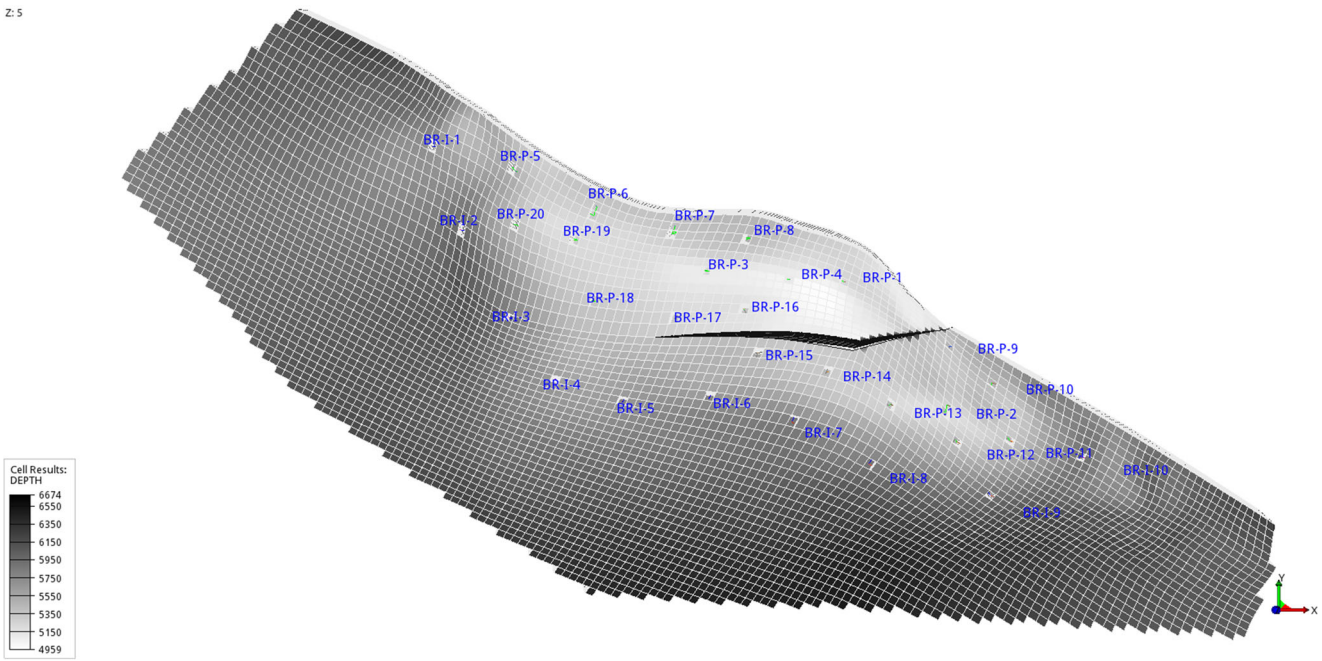

Fig. 4 Structure of the Brugge field from the top view showing the depth and the 30 wells

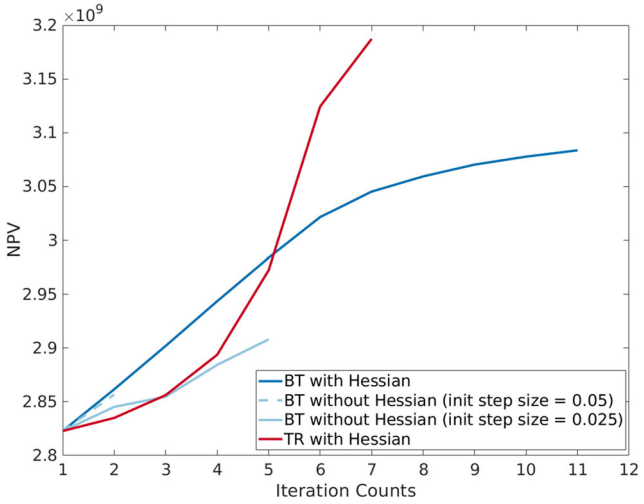

Fig. 5 The change of the NPV with iterations for four scenarios

Of particular note is the importance of the initial trust-region radius Δ_k in Eq. 13 and the initial step size β_0^1 in Eq. 3. We followed guidelines of [3] on choosing parameters for the trust-region method. In this example, Δ_k is set to 0.1. Two other parameters, γ_1 and γ_2, as described in Algorithm 2 are set to 0.9 and 2, respectively. Finally, η_1 and η_2 are set to 0.001 and 0.1, respectively.

For the backtracking method with Hessian approximations the initial step size β_0^1 is set to 0.05. This scenario was initially compared with the backtracking method without Hessian approximations scenario for which the initial step size was kept the same. However the latter scenario could not find further improvement after the second iteration as shown in Fig. 5. One possible explanation could be that the initial step size was set too large. To address this issue, we introduced the scenario where we reduced the initial step size of backtracking from 0.05 to 0.025. For all three cases, the step size for Σ was set to 0.01.

The results presented in Fig. 5 shows the same trend as for the Rosenbrock function, that including a Hessian approximation in EnOpt is beneficial. And again, the trust-region approach outperforms the backtracking.

Of particular interest are the oil field cumulative production, water field cumulative production, and water field cumulative injection as shown in Figs. 6, 7, and 8, respectively. Trust-region method with Hessian approximations has the highest cumulative oil production among four scenarios as shown in Fig. 6. The ideal result would be that the strategy ends up with more oil production, and less water production and water injection. However, as shown in Figs. 7 and 8 trust-region method with Hessian approximations results in the highest water injection and production among the four scenarios. These unwanted higher water injection and production values are compensated by more oil production for a better economic value.

5 Conclusions

In this article we have introduced a natural Hessian approximation for ensemble-based optimization and its

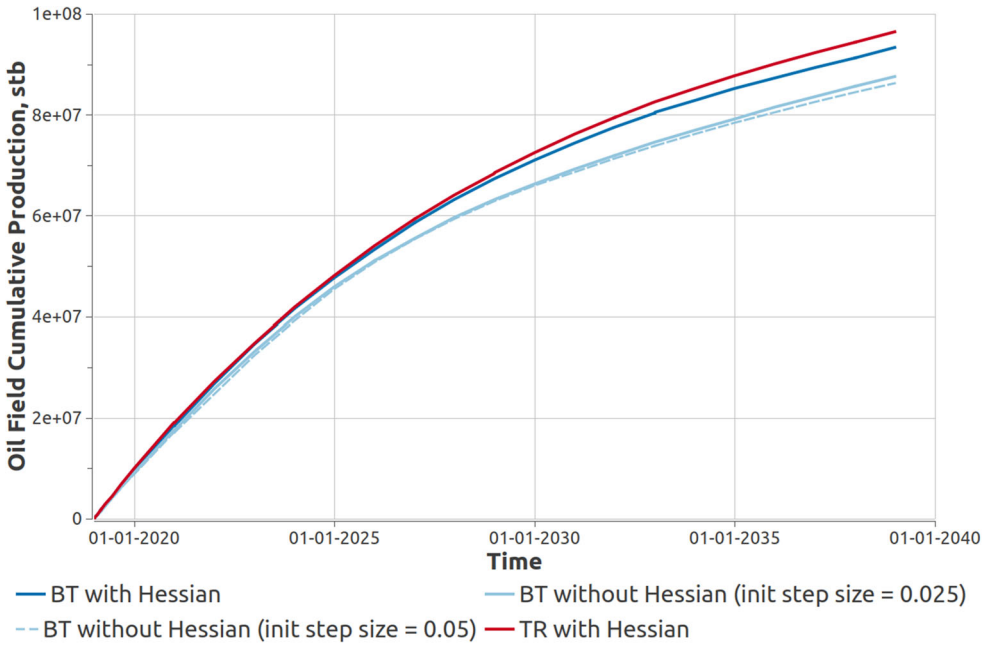

Fig. 6 Field cumulative oil production for four scenarios

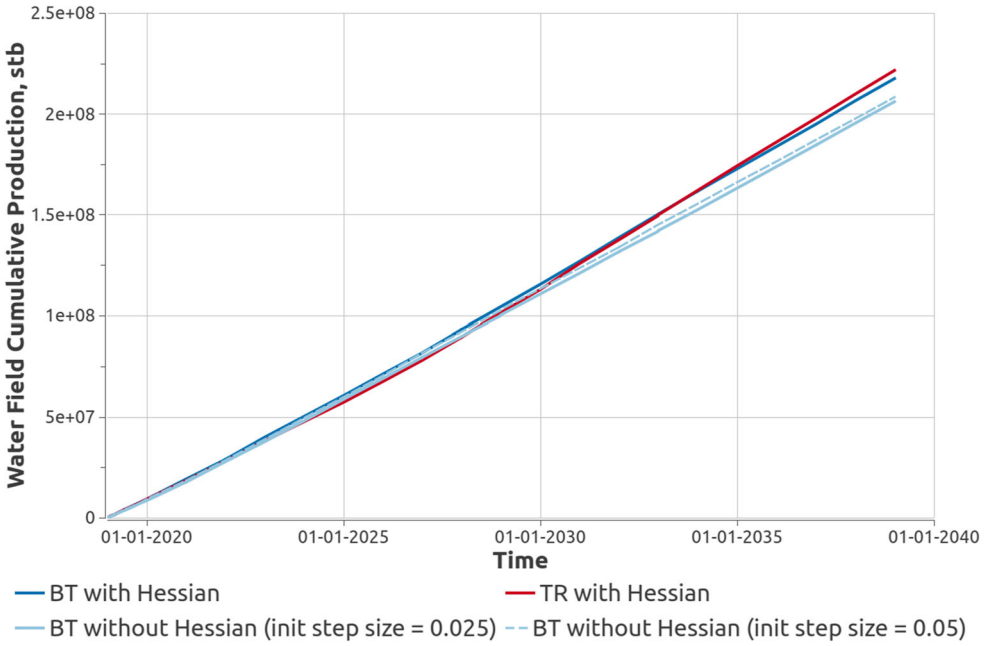

Fig. 7 Field cumulative water production for four scenarios

derivation in the context of trust-region methods. In particular, the case using Steihaug's approach for solving the trust-region subproblem. The natural Hessian can be approximated with a Monte Carlo approach which makes it suitable for the ensemble-based optimization in reservoir management, and within the natural evolution in other fields of engineering. The presented methodology showed improved convergence rate for the Rosenbrock function when compared to more standard EnOpt algorithms. Furthermore, it achieved higher Net-Present-Value than other EnOpt algorithms in the Brugge reservoir test case.

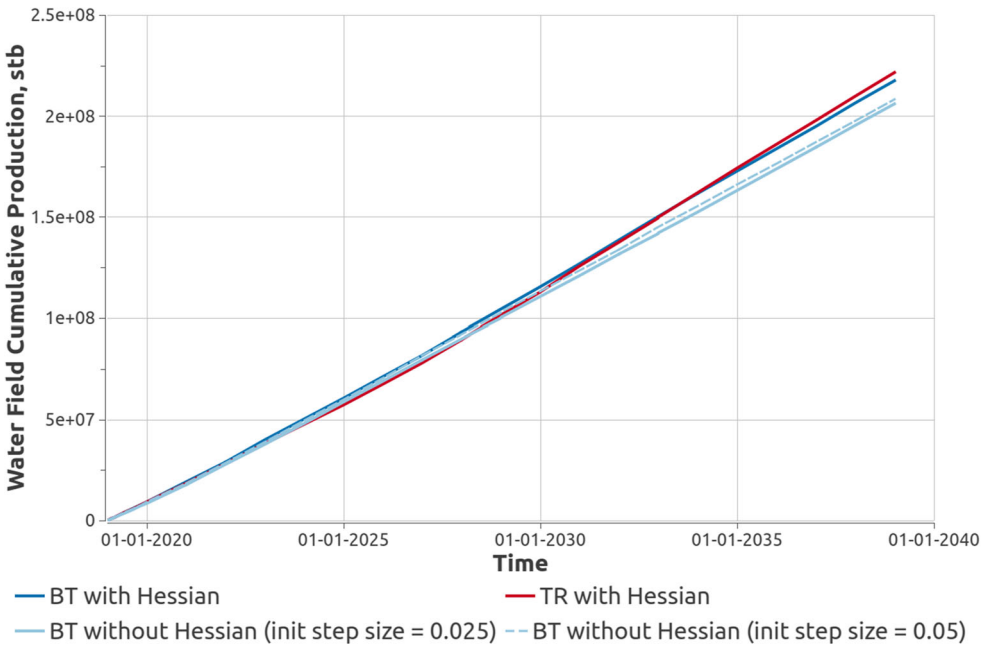

Fig. 8 Field cumulative water injection for four scenarios

Acknowledgements The authors acknowledge the Research Council of Norway and the industry partners, ConocoPhillips Skandinavia AS, Aker BP ASA, Vår Energi AS, Equinor ASA, Neptune Energy Norge AS, Lundin Norway AS, Halliburton AS, Schlumberger Norge AS, Wintershall Norge AS, and DEA Norge AS, of The National IOR Centre of Norway for support. The authors are also immensely grateful to Richard J. Robertson and Patrick N. Raanes for the comments on an earlier version of the manuscript.

Funding Open access funding provided by NORCE Norwegian Research Centre AS

Data Availability The datasets generated during and/or analyzed during the current study are available from the corresponding authors on reasonable request.

References

1. Amari, S.I.: Natural gradient works efficiently in learning. Neural Comput. **10**(2), 251–276 (1998). https://doi.org/10.1162/089976698300017746
2. Chen, Y., Oliver, D.S., Zhang, D.: Efficient ensemble-based closed-loop production optimization. SPE J. **14**(4), 634–645 (2009). https://doi.org/10.2118/112873-PA
3. Conn, A.R., Gould, M.I.M., Toint, P.L.: Trust-region methods MPS-SIAM series on optimization. https://doi.org/10.1137/1.9780898719857 (2000)
4. Do, S.T., Reynolds, A.C.: Theoretical connections between optimization algorithms based on an approximate gradient. Comput. Geosci. **17**(6), 959–973 (2013). https://doi.org/10.1007/s10596-013-9368-9
5. van Essen, G., den Hof, P.V., Jansen, J.D.: Hierarchical long-term and short-term production optimization. SPE J. **16**(1), 191–199 (2011). https://doi.org/10.2118/124332-PA
6. Evensen, G.: Data Assimilation. Springer-Verlag, Berlin (2009)
7. Fonseca, R., Leeuwenburgh, O., den Hof, P.V., Jansen, J.D.: Improving the ensemble-optimization method through covariance-matrix adaptation. SPE J. **20**(1), 155–168 (2014). https://doi.org/10.2118/163657-PA
8. Forouzanfar, F., Li, G., Reynolds, A.C.: A two-stage well placement optimization method based on adjoint gradient. In: SPE Annual Technical Conference and Exhibition, 19-22 September. Florence Italy, SPE (2009). https://doi.org/10.2118/135304-MS
9. Jansen, J.D.: Adjoint-based optimization of multi-phase flow through porous media – a review. Comput. Fluids **46**(01), 40–51 (2011). https://doi.org/10.1016/j.compfluid.2010.09.039
10. Lorentzen, R.J., Berg, A., Nævdal, G., Vefring, E.H.: A new approach for dynamic optimization of water flooding problems. Intelligent Energy Conference anExhibition. Society of Petroleum Engineers, Amsterdam, The Netherlands (2006). https://doi.org/10.2118/99690-MS
11. Lozano, J., Larranaga, P., Inza, I., Bengoetxea, E. (eds.): The CMA Evolution Strategy: A Comparing Review. Springer, Berlin (2006)
12. Peters, E., Arts, R.J., Brouwer, G.K., Geel, C.R., Cullick, S., Lorentzen, R.J., Chen, Y., Dunlop, K.N., Vossepoel, F.C., Sarma, R.X.P., Alhutali, A.H., Reynolds, A.C.: Results of the Brugge benchmark study for flooding optimization and history matching. SPE Res. Eval. Eng, **13**(3), 391–405 (2010). https://doi.org/10.2118/119094-PA
13. Sarma, P., Chen, W.H., Durlofsky, L.J., Aziz, K.: Production optimization with adjoint models under nonlinear control-state path inequality constraints. SPE J. **11**(2), 326–339 (2008). https://doi.org/10.2118/99959-PA
14. Steihaug, T.: The conjugate gradient method and trust regions in large scale optimization. SIAM J. Numer. Anal. **20**(3), 626–637 (1983). https://doi.org/10.1137/0720042
15. Stordal, A.S., Szklarz, S.P., Leeuwenburgh, O.: A theoretical look at ensemble-based optimization in reservoir management. Math. Geosci. **48**(4), 399–417 (2016). https://doi.org/10.1007/s11004-015-9598-6
16. Wierstra, D., Schaul, T., Glasmachers, T., Sun, Y., Peters, J., Schmidhuber, J.: Natural evolution strategies. J. Mach. Learn. Res. **15**(1), 949–980 (2014)
17. Zandvliet, M., Handels, M., van Essen, G., Brouwer, R., Jansen, J.D.: Adjoint-based well-placement optimization under production constraints. SPE J. **13**(4), 392–399 (2008). https://doi.org/10.2118/105797-PA