A Theoretical look at Ensemble-Based Optimization in Reservoir Management

Andreas S. Stordal, Slawomir P. Szklarz, Olwijn Leeuwenburgh

Abstract Ensemble-based optimization has recently received great attention as a potentially powerful technique for life-cycle production optimization, which is a crucial element of reservoir management. Recent publications have increased both the number of applications and the theoretical understanding of the algorithm. However, there is still ample room for further development since most of the theory is based on strong assumptions. Here, the mathematics (or statistics) of Ensemble Optimization is studied, and it is shown that the algorithm is a special case of an already well-defined natural evolution strategy known as Gaussian Mutation. A natural description of uncertainty in reservoir management arises from the use of an ensemble of history-matched geological realizations. A logical step is therefore to incorporate this uncertainty description in robust life-cycle production optimization through the expected objective function value. The expected value is approximated with the mean over all geological realizations. It is shown that the frequently advocated strategy of applying a different control sample to each reservoir realization delivers an unbiased estimate of the gradient of the expected objective function. However, this procedure is more variance prone than the deterministic strategy of applying the entire ensemble of perturbed control samples to each reservoir model realization. In order to reduce the variance of the gradient estimate, an importance sampling algorithm is proposed and tested on a toy problem with increasing dimensionality.

Keywords Ensemble optimization · Production optimization · Robust optimization · Natural evolution · Gaussian mutation

1 Introduction

Reservoir management could be defined as the collection of activities aimed at making decisions on field development including operating strategies and placement of wells. In order to make these decisions with confidence, one has to be able to make reliable predictions of the consequences of such decisions, and to quantify the associated risks. The basis for this is a good description of the reservoir, as captured by a numerical flow model. This model should be able to reproduce the past production history with accept-able accuracy given the uncertainty in the data (which is achieved through the process of history matching). If this condition is met, the model is generally assumed to be capable of capturing the future behavior of the reservoir within reasonable uncertainty and can then be used as the basis for production optimization. The repeated exercise of history matching and optimizing production on a numerical reservoir model is often referred to as closed loop reservoir management (Brouwer et al. 2004; Jansen et al. 2004; Sarma et al. 2006). It has been demonstrated that significant scope exists for improved reservoir management through the use of numerical optimization methods in conjunction with reservoir simulation models (Peters 2011). While the focus so far has primarily been on water flooding optimization, polymer injection optimization (Raniolo et al. 2013) and optimization under gas coning conditions (Hasan et al. 2013) have also been explored.

The most efficient method for gradient-based life-cycle optimization is the so-called adjoint technique. An overview of applications to reservoir management can be found in the review paper by Jansen (2011). While the adjoint technique is computationally very efficient, it is unfortunately an intrusive method that requires access to the simulator source code. In addition to the complexity of implementation, this limits applicability to mostly academic research codes. The need therefore arises for alterna-tive optimization methods wherein the simulator is used as a black box. In optimization literature there exist numerous examples of such methods including evolutionary algorithms, genetic algorithms, and approximate gradient methods. The focus here is on the approximate gradient method known as the Ensemble Optimization technique (EnOpt) (Chen et al. 2009; Lorentzen et al. 2006; Nwaozo 2006). The gradient approximation in EnOpt is based on a linear regression between an ensemble of control samples and their corresponding objective function values. The control samples are drawn from a multivariate Gaussian distribution with a user-defined (constant) covariance matrix and a known mean. Several publications (Chen 2008; Chen and Oliver 2012; Leeuwenburgh et al. 2010; Su and Oliver 2010) have shown that EnOpt can achieve good results of practical value on a variety of different reservoir models and recovery techniques. A major drawback, however, is the significantly lower computational efficiency and accuracy compared to the adjoint method. Recently, Fonseca et al.(2013) showed that improved results can be obtained with EnOpt when the covariance matrix is allowed to adapt according to the best samples from the ensemble of con-trols. This modification was called CMA-EnOpt, where CMA stands for Covariance

Matrix Adaptation. The CMA-EnOpt technique is based on an evolutionary strategy, developed in the machine-learning community, called Covariance Matrix Adaptation-Evolutionary Strategy (CMA-ES) (Hansen and Ostermeier 1996, 2001; Lozano et al. 2006). The main principle of CMA-ES is to modify the covariance matrix in the directions that have proven to be successful. While the CMA-ES algorithm has recently been applied to a number of low-dimensional reservoir optimization applications such as well-placement optimization (Ding 2008; Bouzarkouna et al. 2011), and smart well optimization (Schulze-Riegert et al. 2011; Pajonk et al. 2011), it is not considered to be a practical approach for the more realistically complex and high-dimensional problems often encountered in reservoir optimization.

With increased focus on ensemble-based solutions in reservoir history matching over the last decade, a natural treatment of the uncertainty has been the use of multiple reservoir models arising from different geological realizations. It is natural to incorporate this uncertainty in the production optimization as well, as proposed by van Essen et al. (2006). The situation, where the objective function is defined as the mean objective function over all geological realizations, is also known as robust optimization. In Chen (2008), a method to achieve robust optimization with EnOpt was introduced in which the robust gradient is estimated by pairing each control sample to a different member of an ensemble of geological realizations. Using a new mathematical perspective on EnOpt, it is proven here that this strategy is an unbiased approach, albeit more variance prone than the strategy of applying the entire ensemble of control parameters to each of the geological realizations separately.

Although the number of applications has started to grow, the mathematical treatment and the understanding of EnOpt are based on strong assumptions of smoothness and bounded higher-order derivatives of the objective function and are therefore somewhat incomplete. This paper focuses on the mathematics (or statistics) of EnOpt and it is shown that it is a version of an already well-defined natural evolution strategy known as Gaussian Mutation (Amari 1998). Furthermore, the mathematical treatment is extended to include uncertainty. The importance of variance reduction techniques for improved algorithm performance is also studied. New and old concepts are illustrated on some simple toy models with increasing dimensionality. The paper is concluded with a summary and discussion.

2 Ensemble-Based Production Optimization

The aim of optimization in the context of reservoir management is to maximize the economic value, for example the net present value (NPV), which is usually expressed as

$$J(x) = \sum_{j=1}^{T} \frac{\left([q_{o,j} \cdot r_0 - q_{wp,j} \cdot r_{wp}] - [q_{wi,j} \cdot r_{wi}]\right) \Delta t_j}{(1+d)^{t_j/\tau_t}},$$

where the oil production, $q_{o,j}$, water production, $q_{wp,j}$, and water injection rates, $q_{wi,j}$, at time j change as a function of the control variables x (typically flow rates or bottom hole pressure). The costs of water injection and water production are denoted

r_{wp} and r_{wi} respectively, while r_o is the oil price. The discount factor is denoted by d, and τ_t is the reference period for discounting. Optimal values of the controls x can be found iteratively using an update scheme. In Chen et al. (2009), EnOpt was derived as an approximation of the preconditioned steepest ascend method (Tarantola 2005)

$$x_{k+1} = x_k + \beta_k \Sigma \mathcal{G}_k,$$

where k denotes the iteration number, x_k is the current control, Σ is a preconditioning matrix and $\mathcal{G}_k$ is the sensitivity of the objective function evaluated at x_k. The step size is denoted β_k. At iteration k of the EnOpt algorithm, an ensemble of N controls, $\{X_k^i\}_{i=1}^N$, is sampled from a multivariate Gaussian distribution with mean μ_k and covariance matrix Σ. The product $\Sigma\mathcal{G}_k$ is then approximated by the sample-based cross covariance

$$\mathbb{C}_{x_k,J} = (N-1)^{-1} \sum_{i=1}^{N} \left(X_k^i - \overline{X}_k\right) \left(J\left(X_k^i\right) - \overline{J}_k\right), \tag{2.1}$$

where $\bar{\cdot}$ denotes the arithmetic mean. In order to derive the approximation $\mathbb{C}_{x_k,J} \approx \Sigma\mathcal{G}_k$ a linearization is used together with the approximations $\overline{X}_k \approx \mu_k$ and $\overline{J}_k \approx J(X_k)$. The first approximation is always valid, as long as each ensemble member is not affected by bound constraints. The second approximation, however, depends on Σ and the behavior of $J(x)$ around μ_k. In general, the statement $\overline{J}_k \approx J(\overline{X}_k)$ is statistically equivalent to $\mathbf{E}[J(X)] \approx J(\mathbf{E}[X])$. This approximation is good if J is nearly linear or if diagonal elements of Σ are small. Hence, the quality of the EnOpt approximation of the preconditioned steepest ascend depends on both the objective function and the user-defined preconditioning matrix. Do and Reynolds (2013) used the same linearization to show the connection between EnOpt and Simultaneous Perturbation Stochastic Approximation (SPSA) of the gradient.

Chen et al. (2009) applied a second preconditioning to the gradient approximation by multiplying $\mathbb{C}_{x,J}$ by Σ from the left. The preconditioned gradient approximation is then given by

$$\Sigma^2\mathcal{G}_k \approx (N-1)^{-1} \sum_{i=1}^{N} \Sigma \left(X_k^i - \overline{X}_k\right) \left(J\left(X_k^i\right) - \overline{J}_k\right). \tag{2.2}$$

While the above approximation seems justified in many applications, the aim of this paper is to establish convergence properties of the EnOpt gradient in a probabilistic sense. A simple investigation of the asymptotic properties of Eq. (2.2) results in the following lemma.

Lemma 1 *Assume that $x \in \mathbb{R}^d$. The preconditioned EnOpt gradient formulation in Eq.* (2.2) *converges almost surely to*

$$\Sigma^2 \nabla_{\mu_k} \int_{\mathbb{R}^d} J(x) \Phi\left(x|\mu_k, \Sigma\right) \mathrm{d}x,$$

where $\Phi(\cdot|\mu_k, \Sigma)$ is a multivariate Gaussian density with mean μ_k and covariance matrix Σ.

Proof For notational convenience the integral limits are discarded. The sample $\{X_k^i\}_{i=1}^N$ is an i.i.d. sample from $\Phi(x|\mu_k, \Sigma)$. The strong law of large numbers and Slutsky's theorem (in vector form) can therefore be applied to Eq. (2.2) to get

$$\begin{aligned}
&(N-1)^{-1}\sum_{i=1}^{N}\Sigma\left(X_k^i-\overline{X}_k\right)\left(J\left(X_k^i\right)-\overline{J}_k\right)\\
&\xrightarrow{a.s}\Sigma\int(J(x)-\mathbf{E}[J])(x-\mu_k)\Phi\left(x|\mu_k,\Sigma\right)\,\mathrm{d}x\\
&=\Sigma\int J(x)(x-\mu_k)\Phi\left(x|\mu_k,\Sigma\right)\,\mathrm{d}x\\
&=\Sigma^2\int J(x)\nabla_{\mu_k}\log\Phi\left(x|\mu_k,\Sigma\right)\Phi\left(x|\mu_k,\Sigma\right)\,\mathrm{d}x\\
&=\Sigma^2\nabla_{\mu_k}\int J(x)\Phi(x|\mu_k,\Sigma)\,\mathrm{d}x,
\end{aligned}$$

where the following equalities are used

$$\begin{aligned}
\nabla_{\mu_k}\log\Phi(x|\mu,\Sigma)&=\Sigma^{-1}(x-\mu),\\
\left(\nabla_{\mu_k}\log\Phi(x|\mu,\Sigma)\right)\Phi(x|\mu,\Sigma)&=\nabla_{\mu_k}\Phi(x|\mu,\Sigma).
\end{aligned}$$

An alternative gradient estimate was presented in Fonseca et al. (2013). Let $\mathbb{S}_{x_k}$ denote the sample covariance matrix of $\{X_k^i\}_{i=1}^N$. The alternative ensemble gradient formulation is given by

$$\mathbb{S}_{x_k}^{-1}\mathbb{C}_{x,J}.$$

Since $\mathbb{S}_{x_k}^{-1}$ converges almost surely to Σ^{-1}, Lemma 1 can be combined with Slutsky's theorem to show that this implementation of EnOpt approximates the gradient

$$\nabla_{\mu_k}\int J(x)\Phi(x|\mu_k,\Sigma)\,\mathrm{d}x.$$

The EnOpt algorithm can therefore be summarized in the following corollary.

Corollary 1 *The EnOpt algorithm searches for the optimum of the objective function*

$$L(\mu)=\int J(x)\Phi(x|\mu,\Sigma)\,\mathrm{d}x, \tag{2.3}$$

with respect to μ.

Let $\hat{x}$ be the maximizer of $J(x)$, that is, $\hat{x}=\arg\max J(x)$. It then follows that

$$J(\hat{x})=\int J(x)\delta_{\hat{x}}(x)\,\mathrm{d}x,$$

where δ is the Dirac delta function. Hence, optimizing $J(x)$ is equivalent to optimizing

$$\widetilde{J}(\lambda) = \int J(x)\delta_\lambda(x)\,\mathrm{d}x,$$

with respect to λ. Since $\widetilde{J}(\lambda) \geq L(\lambda)$ and $L(\lambda) \to \widetilde{J}(\lambda)$ as $\Sigma \to 0$, it seems natural to also optimize the right hand side of Eq. (2.3) with respect to Σ in order to obtain improved results with EnOpt. The new objective function is defined as

$$L(\lambda) = \int J(x)\Phi(x|\lambda)\,\mathrm{d}x, \tag{2.4}$$

where $\lambda = [\mu \quad \Sigma]$. This idea has already been proved useful in experiments by Fonseca et al. (2013), where covariance matrix adaptation (CMA) was incorporated into EnOpt. However, no mathematical foundation was provided for using CMA in conjunction with EnOpt. In order to minimize Eq. (2.4), the gradients with respect to both μ and Σ are required. These gradients are already provided in the literature in the development of a Gaussian Mutation Optimization (GMO) algorithm (Amari 1998; Sun et al. 2009). The gradient of Eq. (2.4) with respect to λ is computed in Amari (1998) as

where $$\nabla_\lambda L(\lambda) = \int \Phi(x|\lambda) J(x) \nabla_\lambda \log \Phi(x|\lambda)\mathrm{d}x,$$

$$\nabla_\lambda \log \Phi(x|\lambda) = \left[\Sigma^{-1}(x-\mu) \quad \frac{1}{2}\mathrm{vec}\left(\Sigma^{-1}(x-\mu)(x-\mu)^T\Sigma^{-1} - \Sigma^{-1}\right)\right],$$

and where vec denotes vectorization of a matrix. Since the Gaussian parameter space is not Euclidean, but has a Riemannian metric structure, the steepest ascend direction is given by the natural gradient (Amari 1998; Sun et al. 2009)

$$\overline{\nabla}_\lambda L(\lambda) = \mathbb{I}(\lambda)\nabla_\lambda L(\lambda),$$

where $\mathbb{I}(\lambda)$ is the inverse of the Fisher information matrix. For Gaussian densities, $\mathbb{I}(\lambda)$ is a diagonal block matrix with diagonal elements Σ and $\Sigma \otimes \Sigma$, where $\otimes$ denotes the Kronecker product. The natural gradient is then given by

$$\begin{aligned}\overline{\nabla}_\lambda L(\lambda) &= \int \Phi(x|\lambda) J(x) \nabla_\lambda \mathbb{I}(\lambda) \log \Phi(x|\lambda)\mathrm{d}x \\ &= \left[\mathbf{E}[J(X)(X-\mu)] \quad \mathbf{E}\left[\mathrm{vec}\left(J(X)\left((X-\mu)(X-\mu)^T - \Sigma\right)\right)\right]\right],\end{aligned}$$

where the expectation is with respect to $\Phi(x|\lambda)$. A Monte Carlo approximation of this expectation at iteration k, with $\lambda_k = [\mu_k \quad \Sigma_k]$ is defined by the mean gradient

$$N^{-1}\left[\sum_{i=1}^{N} J\left(X_k^i\right)\left(X_k^i - \mu_k\right) \quad \mathrm{vec}\left(\sum_{i=1}^{N} J(X_k^i)\left(\left(X_k^i-\mu_k\right)\left(X_k^i-\mu_k\right)^T - \Sigma_k\right)\right)\right], \tag{2.5}$$

where $\{X_k^i\}_{i=1}^N$ is a sample from $\Phi(x|\lambda_k)$. A more general formulation is given by

$$\left[\sum_{i=1}^{N} W_k^i \left(X_k^i - \mu_k\right) \quad \text{vec}\left(\sum_{i=1}^{N} W_k^i \left(\left(X_k^i - \mu_k\right)\left(X_k^i - \mu_k\right)^T - \Sigma_k\right)\right)\right],$$

where $W_k^i = N^{-1} J(X_k^i)$. Other choices for the weights $\{W_k^i\}_{i=1}^N$ are possible as well. For example, it is possible to give more weight to the best samples by simply using ranks or logarithmic ranks. However, these approaches are ad hoc and more of practical interest. They will not be discussed further since the focus here is on the theoretical aspects. It is also possible to impose a certain structure on Σ. For example, if Σ is a diagonal matrix, it is straight forward to compute $\nabla_\lambda \log \Phi(x|\lambda)$. The sample approximation of the gradient in Eq. (2.5) then becomes

$$N^{-1}\left[\sum_{i=1}^{N} J\left(X_k^i\right)\left(X_k^i - \mu_k\right) \quad \text{vec}\left(\sum_{i=1}^{N} J\left(X_k^i\right)\left(\text{diag}\left(\left(X_k^i - \mu_k\right)\left(X_k^i - \mu_k\right)^T\right) - \Sigma_k\right)\right)\right].$$

The Gaussian Mutation algorithm evolves according to

$$\begin{aligned}
\mu_{k+1} &= \mu_k + \beta_k^1 \textstyle\sum_{i=1}^{N} W_i \left(X_k^i - \mu_k\right), \\
\Sigma_{k+1} &= \Sigma_k + \beta_k^2 \textstyle\sum_{i=1}^{N} W_i \left(\left(X_k^i - \mu_k\right)\left(X_k^i - \mu_k\right)^T - \Sigma_k\right),
\end{aligned}$$

where $\{X_k^i\}_{i=1}^N$ is a sample from $\Phi(x|\lambda_k)$ and β_k^1 and β_k^2 are step sizes. Hence, with the choice $W_k^i = N^{-1}(J(X_k^i) - \overline{J})$ we see from Eq. (2.1) that EnOpt is a special case of Gaussian Mutation without evolution of Σ. It is worth noting that subtracting $\overline{J}$ from $J(X_k^i)$ does not change the expected value of the gradient estimate. In other words, the gradient estimate of EnOpt is statistically equivalent to that of GMO. In addition, GMO is very similar to the governing equations of CMA-EnOpt as presented in Fonseca et al. (2013). The differences between GMO and CMA-EnOpt are

1. In CMA-EnOpt there is an additional rank one update, which does not show up in the mathematics of GMO.
2. A different weighting scheme is used for the covariance matrix update in CMA-EnOpt compared to GMO.
3. A slightly different gradient formulation is used in CMA-EnOpt that corresponds to selecting $W_i = N^{-1} S_x^{-1}(J(X_i) - \overline{J})$ in the GMO. Details can be found in Fonseca et al. (2013)
4. The entire ensemble of controls is used to adapt the covariance matrix in GMO whereas in CMA-EnOpt the number of ensemble members used to update the covariance matrix is a user-defined choice.

It was shown in Akimoto et al. (2010) that CMA-ES with global weighted recombina-tion and rank-μ update (not to be confused with the mean vector μ) can be formulated as a GMO algorithm. Further, it was shown that the performance of the different algo-rithms is problem dependent. Sometimes CMA-ES performs better than GMO and vice versa. Since the focus here is on the mathematical properties of GMO and EnOpt, these performance differences will not be further discussed.

3 Geological Uncertainty

In closed loop reservoir management, multiple geological realizations, $\{Y_i\}_{i=1}^N$, are often available (e.g., from an ensemble of history-matched reservoir models). A robust optimization procedure is typically implemented in order to find one optimal production strategy. The objective function in robust optimization is given by the mean objective function (van Essen et al. 2006) over all geological realizations

$$J(x) = N^{-1} \sum_{i=1}^{N} J(x, Y_i). \tag{3.1}$$

For simplicity, the covariance matrix is ignored for a moment and the focus is only on the mean control vector μ. The GMO gradient approximation with respect to μ for the robust objective function in Eq. (3.1) is given by

$$N^{-1} \sum_{i=1}^{N} J(X_i)(X_i - \mu) = N^{-2} \sum_i \sum_j J(X_i, Y_j)(X_i - \mu).$$

Unfortunately, this formulation requires $N \times N$ simulations at each iteration, since each of the N control strategies is implemented for every single geological realization. To overcome this computational bottle neck, Chen et al. (2009) proposed approximat-ing the gradient of $J(x)$ using the cross covariance between the objective function and the controls, but with each control sample applied to only one, and not all, of the individual geological realizations. This approach requires only N simulations. Other approaches are possible as well. For example, one could apply multiple (but less than N) control samples to each single reservoir model realization. There is no clear math-ematical justification for this approach in the literature, but with the new mathematical insights described in the previous section, this approach can be justified. In order to unify this modified approach with the theory of natural evolution, it is better to inter-pret Eq. (3.1) as a Monte Carlo approximation of $\mathbf{E}_Y[J(x, Y)]$. The new objective function, J, is then given by

$$\widetilde{J}(x) = \mathbf{E}_Y[J(x, Y)].$$

A straightforward generalization of Corollary 1 shows that the robust EnOpt algorithm as presented in Chen et al. (2009) searches for the optimum of

$$L(\mu) = \int \int \mathbf{E}_Y[J(X, Y)]\Phi(x|\mu, \Sigma)\, \mathrm{d}x = \mathbf{E}_{X,Y}[J(X, Y)].$$

The robust formulation of EnOpt can now be extended to GMO by again defining $\lambda = [\mu\ \Sigma]$. Let $f(y)$ be the density of Y representing the geological uncertainty. Since X and Y are independent

$$\overline{\nabla}_\lambda L(\lambda) = \overline{\nabla}_\lambda \int J(x, y) f(y) \Phi(x|\lambda) \, \mathrm{d}x \, \mathrm{d}y,$$

with Monte Carlo estimate

$$N^{-1} \sum_{i=1}^{N} J(X_i, Y_i) \overline{\nabla}_\lambda \log \Phi(X_i|\lambda),$$

where $X_i \sim \Phi(x, |\lambda)$ and $Y_i \sim f(y)$. This is a simple consequence of the fact that the gradient approximated in EnOpt with geological uncertainty is the expected value of a random variable $Z = (X, Y)$ for which the joint density $f(z) = f(y)\Phi(x|\lambda)$ is known. Note that applying all controls to each geological realization is only valid if the control samples and geological samples are statistically independent, which of course is the case in reality. Hence, using the same argument as above, the gradient under geological uncertainty can be approximated using a different control variable for each geological realization. In fact, due to the statistical independence of the geological variables and the control variables, a statistically unbiased estimate of the gradient can be obtained by applying any number of control variables to each geological realization. Ideally, in addition to sampling new controls at each iteration one should also sample new geological realizations from $f(y)$. However, in practice, with only a finite number of samples from $f(y)$ available due to the vast computational time required for history matching, the same set of realizations is used throughout the optimization. It is therefore natural to believe that premature convergence is difficult to avoid in practice.

3.1 Variance Reduction

In addition to the (possible) high dimension of the control vector, the geological uncertainty makes GMO more prone to Monte Carlo sampling errors. A large variance in the gradient estimate typically leads to premature convergence. The gradient estimate may be improved through variance reduction techniques that can be applied at each iteration of the algorithm. For simplicity and ease of notation, the focus in the following is on the variance of the gradient estimate with respect to μ. The theory is the same for the variance of the gradient with respect to Σ. For notational convenience, the geological uncertainty is ignored in the following without loss of generality.

At iteration k, the Monte Carlo estimate of the gradient in GMO is given by $\sum_{i=1}^{N} W_i(X_k^i - \mu_k)$. In the GMO formulation $W_k^i = N^{-1} J(X_k^i)$ whereas in the original formulation of EnOpt $W_k^i = N^{-1}\left(J(X_k^i) - \overline{J}\right)$. Sun et al. (2009) showed that for any scalar b, $\mathbf{E}[\nabla(J(X) - b)] = \mathbf{E}[\nabla J(X)]$, whereas $\mathrm{Var}[\nabla(J(X) - b)]$ is a function of b. It is therefore possible to minimize the variance with respect to b. The optimal variance reduction coefficient (Sun et al. 2009) satisfies

$$b = \frac{\mathbf{E}\left[\left(\mathbb{I}(\lambda)\nabla_\lambda \log \Phi(X|\lambda) J(X)\right)^T \left(\mathbb{I}(\lambda)\nabla_\lambda \log \Phi(X|\lambda)\right)\right]}{\mathbf{E}\left[\left(\mathbb{I}(\lambda)\nabla_\lambda \log \Phi(X|\lambda)\right)^T \left(\mathbb{I}(\lambda)\nabla_\lambda \log \Phi(X|\lambda)\right)\right]},$$

with Monte Carlo estimate at iteration k given by

$$\hat{b} = \frac{\sum_{i=1}^{N} J\left(X_k^i\right)\left(\mathbb{I}(\lambda)\nabla_\lambda \log(\Phi\left(X_k^i|\lambda\right)\right)^T\left(\mathbb{I}(\lambda)\nabla_\lambda \log(\Phi\left(X_k^i|\lambda\right)\right)}{\sum_{i=1}^{N}\left(\mathbb{I}(\lambda)\nabla_\lambda \log(\Phi\left(X_k^i|\lambda\right)\right)^T\left(\mathbb{I}(\lambda)\nabla_\lambda \log(\Phi\left(X_k^i|\lambda\right)\right)}. \tag{3.2}$$

It will be shown later that the Monte Carlo estimate of b is not a good choice for the background term, especially when the sample size is small compared to the dimension of the search space. The reason for this is that the ratio of the two estimators in Eq. (3.2) introduces bias in the gradient estimate. For the value of b in Eq. (3.2) the equality $\mathbf{E}\left[\nabla(J(X) - b)\right] = \mathbf{E}\left[\nabla J(X)\right]$ is no longer satisfied. It is straight forward to show that $b = J(\mu_k)$ also reduces the variance of the gradient estimate without changing the expected value. Hence a suggestion is to use $W_t = N^{-1}\sum_{i=1}^{N}\left(J(X_k^i) - J(\mu_k)\right)$ in the gradient estimate. The results reported in Do and Reynolds (2013) also suggest that using $J(\mu_k)$ instead of $\overline{J}$ leads to improved results.

An expensive way to reduce the variance of the gradient estimate is simply to increase the number of control samples. This may not be a practical option in all cases, especially when parallel computing capacity is limited. Instead, at iteration k, one may take advantage of all the samples from the $k-1$ previous iterations. These samples can be used to construct a weighted gradient estimate in an importance sampling framework. The idea is similar to the conjugate gradient method and the rank one update in CMA-ES in the sense that the goal is to improve the gradient using information from past iterations. The gradient at iteration k may be re-written using the identity

$$\int (J(x) - J(\mu_k))\Phi(x|\mu_k, \Sigma_k)\,\mathrm{d}x = \int (J(x) - J(\mu_k))\, w_k^j(x)\Phi\left(x|\mu_j, \Sigma_j\right)\,\mathrm{d}x, \tag{3.3}$$

where

$$w_k^j(x) = \frac{\Phi(x; \mu_k, \Sigma_k)}{\Phi\left(x; \mu_j, \Sigma_j\right)},$$

for all j from 1 to k. Since the samples from $\Phi(\cdot|\mu_j)$ for all j up to $k-1$ are already available, an unbiased importance sampling estimate of Eq. (3.3) can be computed, without additional numerical simulation, as

$$(Nk)^{-1}\sum_{j=1}^{k}\sum_{i=1}^{N}\left(\left(J\left(X_j^i\right) - J\left(\mu_k\right)\right)\left(X_j^i - \mu_k\right) w_k^j\left(X_j^i\right)\right).$$

However, it is not necessarily a good idea to apply the importance sampling using the samples from all the previous iterations since the variance might actually increase. The following theory is presented in one dimension, but it is easily generalized to higher dimensions.

Assume that all samples from iteration ℓ to k are used in the importance sampling algorithm. The quantities of interest are

$$\mathrm{Var}\left[(N(k-\ell+1))^{-1}\sum_{j=\ell}^{k}\sum_{i=1}^{N}\left(\left(J\left(X_j^i\right) - J\left(\mu_k\right)\right)\left(X_j^i - \mu_k\right) w_k^j\left(X_j^i\right)\right)\right],$$

and

$$\mathrm{Var}\left[N^{-1}\sum_{i=1}^{N}\left(\left(J\left(X_k^i\right)-J(\mu_k)\right)\left(X_k^i-\mu_k\right)\right)\right].$$

For the former to be smaller than the latter (a smaller variance of the gradient estimate) the following strict inequality must be satisfied

$$\begin{aligned}&\textstyle\sum_{j=\ell}^{k-1}\mathrm{Var}\left(\left(J(X_j)-J(\mu_k)\right)\left(X_j-\mu_k\right)w_k^j\left(X_j\right)\right)\\&\quad<\left[(k-\ell)^2+2(k-\ell)\right]\mathrm{Var}\left(\left(J(X_k)-J(\mu_k)\right)\left(X_k-\mu_k\right)\right).\end{aligned}\tag{3.4}$$

Let $\ell = k-1$ in Eq. (3.4). The variance of the importance sampling gradient is reduced using the sample from the previous iteration if

$$\begin{aligned}&\mathrm{Var}\left((J(X_{k-1})-J(\mu_k))(X_{k-1}-\mu_k)w_k^{k-1}(X_{k-1})\right)\\&\quad<3\mathrm{Var}\left((J(X_k)-J(\mu_k))(X_k-\mu_k)\right).\end{aligned}$$

Since the variance is not known analytically, the above result is not very useful. It is possible, however, to provide a conservative alternative as follows. Define $\gamma_k = \int(J(x)-J(\mu_k))(x-\mu_k)\Phi(x;\mu_k,\Sigma_k)\,\mathrm{d}x$ and $\Gamma_k(x) = ((J(x)-J(\mu_k))(x-\mu_k))^2$. Then

$$\mathrm{Var}\left((J(X_j)-J(\mu_k))(X_j-\mu_k)w_k(X_j)\right)=\int\Gamma_k(x)w_k^j(x)^2\Phi(x;\mu_k,\Sigma_k)\,\mathrm{d}x-\gamma_k^2.$$

The variance of the importance sampling estimate can then be expressed as

$$\sum_{j=\ell}^{k-1}\mathbf{E}\left(\Gamma_k(X_j)w_k^j(X_j)^2\right)-\gamma_k^2\le\sum_{j=\ell}^{k-1}\left|\left|w_k^j\right|\right|_\infty\mathbf{E}\left(\Gamma_k(X_k)\right)-\gamma_k^2,$$

and from Eq. (3.4) the variance reduction criterion is given by

$$\sum_{j=\ell}^{k-1}\left|\left|w_k^j\right|\right|_\infty<(k-l)^2+2(k-l).\tag{3.5}$$

The expression in Eq. (3.5) is also valid for control vectors in higher dimensions. If $\ell = k-1$ the sample from the previous iteration can be used if $||w_k^{k-1}||_\infty < 3$. This approach may also be combined with the importance mixing approach (Sun et al. 2009) to reduce the computational cost. It must be noted, however, that the random variables in the importance mixing samples are not independent and the algorithm may therefore require a larger sample size. A summary of the proposed changes to the EnOpt algorithm is presented in Algorithm 1.

Algorithm 1: Modified gradient computation in the EnOpt algorithm

At iteration k:

Sample N i.i.d. $\{X_k^i\}$ random variables from $\Phi(x|\mu_k, \mathbb{\Sigma}_k)$

Select $\ell \in (1, 2, \ldots, k-1)$ that maximizes $\left|(k-l)^2 + 2(k-l) - \sum_{j=\ell}^{k-1} ||w_k^j||_\infty^2\right|$

if $\sum_{j=\ell}^{k-1} ||w_k^j||_\infty^2 > (k-l)^2 + 2(k-l)$ **then**

  Set $\ell = k$

end

Calculate the gradients

$$\mu_{k+1} = \mu_k + \beta_k^1 \sum_{j=\ell}^{k} \sum_{i=1}^{N} W_j^i (X_j^i - \mu_k),$$

$$\mathbb{\Sigma}_{k+1} = \mathbb{\Sigma}_k + \beta_k^2 \sum_{j=\ell}^{k} \sum_{i=1}^{N} W_j^i \left((X_j^i - \mu_k)(X_j^i - \mu_k)^T - \mathbb{\Sigma}_k \right),$$

where

$$W_j^i = (N(k-\ell))^{-1} \left(J(X_j^i) - J(\mu_k) \right) \frac{\Phi(X_j^i; \mu_k, \mathbb{\Sigma}_k)}{\Phi(X_j^i; \mu_j, \mathbb{\Sigma}_j)}.$$

Optimize step sizes and update control parameter μ_k and covariance matrix $\mathbb{\Sigma}_k$.

Set $k = k+1$

4 Numerical Examples

4.1 The Rosenbrock Function

The first example is included to show the potential improvement of GMO with respect to the original EnOpt algorithm. The two algorithms are implemented and compared on the well-known Rosenbrock (1960) function. The Rosenbrock function is a non-convex function given by

$$f(x) = (1 - x_1)^2 + 100 \left(x_2 - x_1^2 \right)^2.$$

The function has a global minimum at $x_{\min} = (1,\ 1)$ inside a flat and narrow valley as shown in Fig. 1.

The starting point for the optimization is $x_0 = (-1.5, 0.5)$, and the same random seed is used for both the GMO and the EnOpt algorithm. The step sizes are selected as $\beta_k^1 = 1$, and $\beta_k^2 = 0.1$ for the mean and covariance matrix respectively. Simple backtracking is implemented, where the step size is chopped in half if the objective function increases after the update. The initial covariance diagonal has a value of 0.1 for both diagonal elements. The algorithms are run until $||x_k - x_{\min}|| < 10^{-3}$.

It is clear from Fig. 2 that the result from GMO is a significant improvement over EnOpt with almost 20 times fewer iterations required to find the global minimum. This is due to the fact that GMO adapts the covariance to the shape of the objective function and hence samples more strategically than the EnOpt algorithm (i.e., avoids

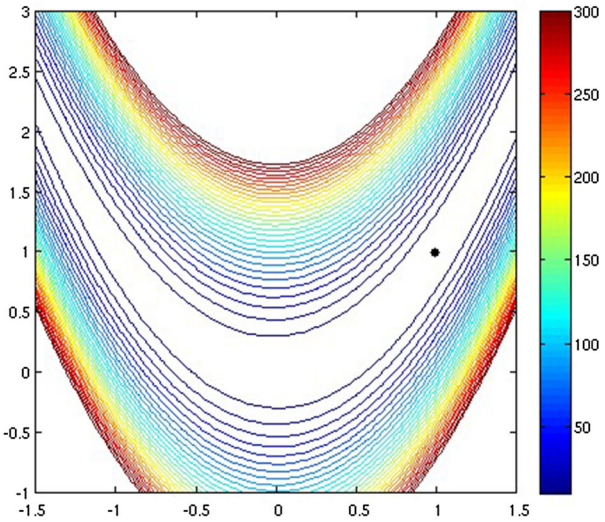


Fig. 1 The two-dimensional Rosenbrock function with global minimum at (1, 1)

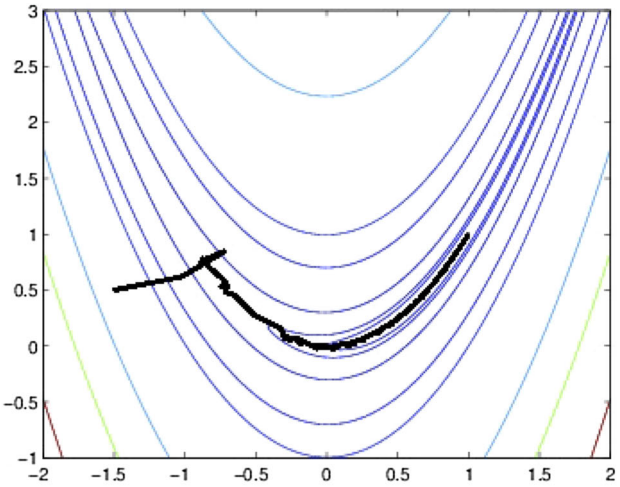
regions with high objective function values). Fonseca et al. (2013) obtained similar results with the CMA-EnOpt algorithm.

4.2 Variance Reduction on a Modified Rosenbrock Function

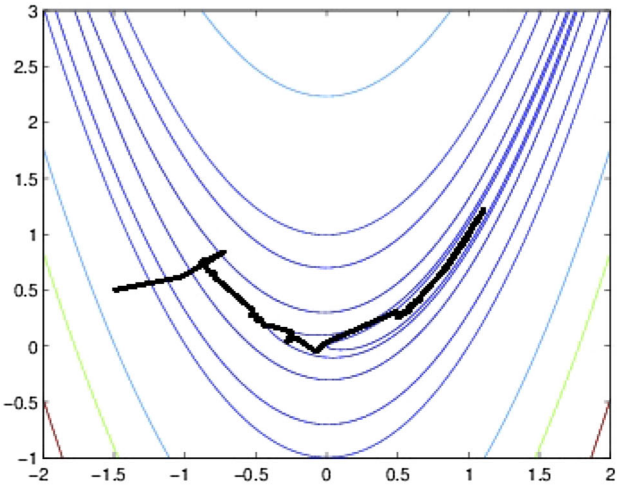
The second example has a simpler objective function than the first example. The Rosenbrock function is modified so that it is possible to analytically compute the mean and variance of gradients along with the optimal background coefficient. The new objective function is given by

$$J(x, y) = (1 - x)^2 + (y - x)^2.$$

The EnOpt gradient is evaluated both with and without variance reduction for two different scenarios. First, $y = 0$ is assumed to be a known parameter. In the second scenario, y is a standard normal random variable (mimicking geological uncertainty of the model). Initially, the mutation density is the standard normal density and the natural gradient for μ is estimated using samples of size $N = 10,\ 10^2,\ 10^3$. For the case where y is random, the gradient is first estimated with all x samples applied to each y, and then with the more standard pairing of samples (one to one). The true natural gradient with respect to μ in all cases is -2. For the optimal background term, the values are $b = 7$ when $y = 0$, and $b = 8$ when y is random. These values for b are used for all the runs in Table 3. For the runs in Table 2, the optimal background term is estimated [Eq. (3.2)], while Table 1 shows the results for the runs without any variance reduction. When y is uncertain, the gradient estimate is said to be permuted if all x samples are evaluated for each y sample. That is for each i we evaluate $J(x_j, y_i)$



(a) No covariance update, 2824 iterations



(b) Covariance update, 142 iterations

Fig. 2 Minimizing the Rosenbrock function with and without covariance update

for all j from 1 to N. The experiments are repeated 1000 times in order to get a good estimate of the mean and variance of the different gradient methods.

It is clear that, for this model, estimating the optimal variance reduction term leads to severe bias in the gradient for small sample sizes (Table 2). The strategy of applying one control vector to one model leads to an unbiased estimate of the gradient, although a reduction in the variance is observed if all control samples are applied to each of the model samples (Table 3). The difference in variance is relatively small here, but it is reasonable to believe that the difference will increase with the dimension of the

Table 1 Natural gradient estimation without variance reduction, 1000 repetitions

Sample size	y input	Permuted	Mean	Variance
10	$y = 0$	NA	−2.025	8.050
10	$y = N(0, 1)$	No	−1.981	10.995
10	$y = N(0, 1)$	Yes	−1.965	9.964
100	$y = 0$	NA	−1.993	0.314
100	$y = N(0, 1)$	No	−2.007	1.118
100	$y = N(0, 1)$	Yes	−2.005	1.016
1,000	$y = 0$	NA	−1.998	0.031
1,000	$y = N(0, 1)$	No	−1.997	0.110
1,000	$y = N(0, 1)$	Yes	−1.997	0.100

Table 2 Natural gradient estimation with estimated background term, 1000 repetitions

Sample size	y input	Permuted	Mean	Variance
10	$y = 0$	NA	−1.486	1.347
10	$y = N(0, 1)$	No	−1.468	3.783
10	$y = N(0, 1)$	Yes	−1.468	3.595
100	$y = 0$	NA	−1.947	0.176
100	$y = N(0, 1)$	No	−1.943	0.592
100	$y = N(0, 1)$	Yes	−1.941	0.504
1,000	$y = 0$	NA	−1.992	0.017
1,000	$y = N(0, 1)$	No	−1.994	0.064
1,000	$y = N(0, 1)$	Yes	−1.993	0.053

Table 3 Natural gradient estimation with optimal background term, 1000 repetitions

Sample size	y input	Permuted	Mean	Variance
10	$y = 0$	NA	−2.011	3.279
10	$y = N(0, 1)$	No	−2.035	4.606
10	$y = N(0, 1)$	Yes	−2.034	3.762
100	$y = 0$	NA	−2.004	0.317
100	$y = N(0, 1)$	No	−1.993	0.046
100	$y = N(0, 1)$	Yes	−1.998	0.361
1,000	$y = 0$	NA	−2.000	0.032
1,000	$y = N(0, 1)$	No	−2.000	0.045
1,000	$y = N(0, 1)$	Yes	−2.005	0.035

problem. From the results in Tables 1 and 2, it seems that for small sample sizes it is better to use the original gradient than to estimate the optimal background term.

Next, the same model with $y = 0$ is studied. This time, however, the mutation distribution is changed a few times in order to mimic iterations. It is then possible to

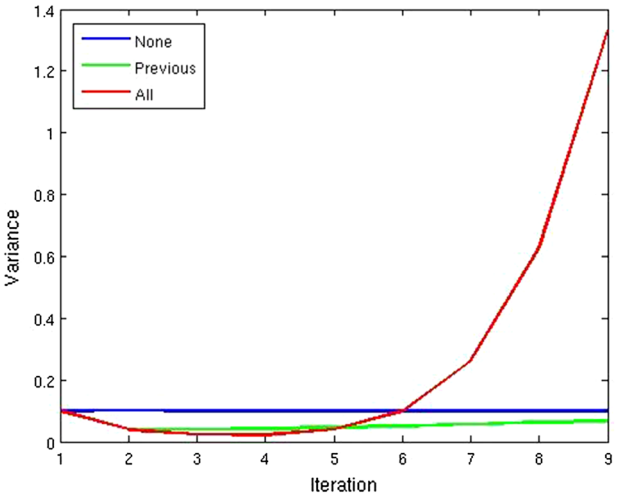


Fig. 3 Variance of gradient estimate using importance sampling

implement the importance sampling strategy. At iteration k, the mutation distribution has mean μ_k where $\mu_k = \mu_{k-1} + 0.25$, starting with $\mu_1 = -1$ and finishing at $\mu_9 = 1$. The variance of a single realization of the natural gradient estimate at iteration k is

$$\mathrm{Var}[J(X_k)(X_k - \mu_k)] = 81 - 60\mu_k + 64\mu_k^2 - 8\mu_k^3 + 4\mu_k^4.$$

The variance of the gradient estimates is therefore divided by this number at each iteration in order for them to be compared. To isolate the effect of importance sampling on the variance, the background term is set to zero. In other words, $J(X_j^i)$ is used instead of $J(X_j^i) - J(\mu_k)$ in the gradient estimate. From Fig. 3, it is clear that there is both a potential advantage and a disadvantage of using importance sampling. When all the previous samples are used, the gradient esti-mate deteriorates after six iterations. However, it is also seen that only using the samples from the previous iteration consistently improves the gradient estimate in terms of variance reduction. Next, the adaptive importance sampling scheme (Algo-rithm 1) is implemented on a generalized Rosenbrock function with increasing dimensionality.

4.3 Increasing Dimensionality

In this section, the GMO algorithm is implemented with and without the adaptive importance sampling scheme (Algorithm 1) on a generalized Rosenbrock function with increasing dimensionality. The objective function is given by

$$J(x) = J(x_1, \ldots, x_d) = \sum_{i=1}^{d/2} \left[100 \left(x_{2i-1}^2 - x_{2i} \right)^2 + (x_{2i-1} - 1)^2 \right],$$

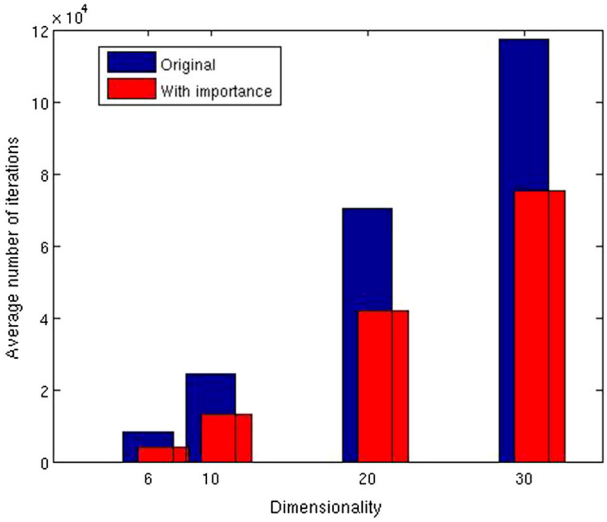


Fig. 4 Average number of iterations for the generalized Rosenbrock

for an even integer d. The global optimum is located at $x_o = (1, 1, \ldots, 1)$. The gradients with respect to both μ and Σ are calculated according to Algorithm 1 with $W(x) = J(x) - J(\mu)$. The gradient with respect to Σ is normalized in order to avoid tuning of the step size when the dimension increases. The step sizes are fixed with $\beta^1 = 1$ and $\beta^2 = 0.15$ for all runs. For each dimension, $d = (6, 10, 20, 30, 50)$, a sample size of $N = 10$ is used to estimate the gradient. The algorithm stops when $\|\mu - x_o\| < 10^{-2}(d - 1)$. The starting point is the d-dimensional vector, $(-1.5, -1, 5, \ldots, 0.5, 0.5)$, with $d/2$ entries equal to -1.5 and $d/2$ entries equal to 0.5. Each run is repeated with 50 different random seeds and the average number of iterations for the first four dimensions is reported in Fig. 4. The results clearly show the benefit of using importance sampling for this particular objective function. Note that the ratio of the average number of iterations required with and without importance sampling seems to be independent of the dimension. For $d = 50$, the results of each individual iteration are also shown (Fig. 5) in order to demonstrate that also the variance of the number of iterations required is smaller when adaptive importance sampling is used. The average number of iterations required with importance sampling is 73.5 % of that without importance sampling for this particular objective function.

5 Conclusions

In this paper, it was shown that the ensemble-based optimization algorithm is equivalent to a natural evolution strategy with a fixed covariance matrix and with a Gaussian sampling density. Furthermore, it was also shown that including the gradient of the covariance matrix of the distribution, the natural evolution strategy known as Gaussian Mutation is an ensemble-based optimization technique with a link to evolutionary strategies with covariance matrix adaptation as discussed by Akimoto et al. (2010).

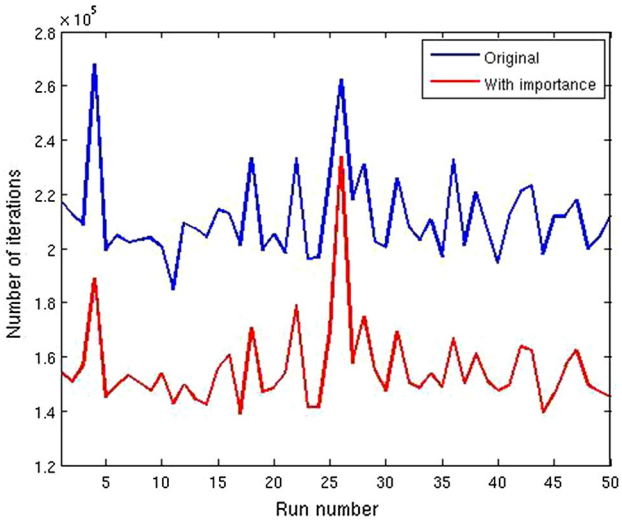


Fig. 5 Number of iterations for 50 runs with the 50-dimensional Rosenbrock function

The relationship with ensemble optimization with covariance matrix adaptation (Fonseca et al. 2013) was clarified as well. Furthermore, the mathematical framework of Gaussian Mutation was used to justify the application of EnOpt in robust optimization under geological uncertainty. Some variance reduction techniques were discussed and it was demonstrated on a simple example that estimating an optimal variance reduction term may lead to severe bias in the gradient estimate. An alternative, unbiased, variance reduction technique in terms of an adaptive importance sampling strategy was suggested and successfully implemented in a simple example for dimension up to 50. The resulting strategy reduced the variance of the gradient estimate with approximately 30 %.

Acknowledgments The first author acknowledges the Research Council of Norway and the industrial participants, ConocoPhillips Skandinavia AS, BP Norge AS, Det Norske Oljeselskap AS, Eni Norge AS, Maersk Oil Norway AS, DONG Energy AS, Denmark, Statoil Petroleum AS, GDF SUEZ E&P NORGE AS, Lundin Norway AS, Halliburton AS, Schlumberger Norge AS, Wintershall Norge AS, of The National IOR Centre of Norway for financial support.

References

Akimoto Y, Nagata Y, Ono I, Kobayashi S (2010) Biderectional relation between CMS evolution strategies and natural evolution strategies. PPSN XI Part I LNCS 6238:154–163

Amari SI (1998) Natural gradients works efficiently in learning. Neural Comput 10(2):333–339

Bouzarkouna Z, Ding D, Auger A (2011) Well placement optimization with the covariance matrix adaptation evolution strategy and meta-models. Comput Geosci:1–18

Brouwer DR, Naevdal G, Jansen JD, Vefring EH, van Kruijsdijk CPJW (2004) Improved reservoir management through optimal control and continuous model updating. In: SPE Annual Technical Conference and Exhibition, Houston. Texas, pp 26–29 **(SPE90149)**

Chen Y (2008) Efficient ensemble based reservoir management. Ph.d. thesis, University of Oklahoma

Chen Y, Oliver DS (2012) Localization of ensemble-based control-setting updates for production optimization. SPE J 17(1):122–136

Chen Y, Oliver DS, Zhang D (2009) Efficient ensemble-based closed-loop production optimization. SPE J 14(4):634–645
Ding YD (2008) Optimization of well placement using evolutionary algorithms. In: Europec/EAGE Conference and Exhibition
Do ST, Reynolds AC (2013) Theoretical connections between optimization algorithms based on an approximate gradient. Comput Geosci 17(6):959–973
Fonseca RM, Leeuwenburgh O, Hof PVD, Jansen JD (2013) Improving the ensemble optimization method through covariance matrix adaptation (cma-enopt). In: SPE Reservoir Simulation Symposium
Hansen N, Ostermeier A (1996) Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In: 1996 IEEE International Conference on Evolutionary Computation. IEEE, pp 312–317
Hansen N, Ostermeier A (2001) Completely derandomized self-adaptation in evolution strategies. Evol Comput 9(2):159–195
Hasan A, Foss B, Sagatun S (2013) Optimization of oil production under gas coning conditions. J Pet Sci Eng 105:26–33
Jansen J (2011) Adjoint-based optimization of multi-phase flow through porous media—a review. Comput Fluids 46:40–51
Jansen JD, Brouwer DR, Naevdal G, van Kruijsdijk CPJW (2004) Closed-loop reservoir management. In: EAGE 66th Conference & Exhibition. Paris, pp. 7–10 **(Presented at Workshop "Uncertainties in production forecasts and history matching")**
Leeuwenburgh O, Egberts PJ, Abbink OA (2010) Ensemble methods for reservoir life-cycle optimization and well placement. SPE/DGS Saudi Arabia Sect Tech Symp Exhib 4–7:2010
Lorentzen RJ, Berg AM, Nævdal G, Vefring EH (2006) A new approach for dynamic optimization of water flooding problems. In: SPE Intelligent Energy Conference and Exhibition. Amsterdam, pp 11–13 **(SPE99690)**
Lozano J, Larranaga P, Inza I, Bengoetxea E (2006) (eds) The CMA Evolution Strategy: a comparing review. Springer, pp 75–102
Nwaozo J (2006) Dynamic optimization of a water flood reservoir. Ph.D. thesis, University of Oklahoma
Peters E et al (2011) Brugge paper SPE REE
Pajonk O, Schulze-Riegert R, Krosche M, Hassan M, Nwakile MM (2011) Ensemble-based water flooding optimization applied to mature fields. SPE Middle East Oil Gas Show Conf 25–28:2011
Raniolo S, Dovera L, Cominelli A, Callegaro C, Masserano F (2013) History match and polymer injection optimization in a mature field using the ensemble kalman filter. In: 17th European Symposium on Improved Oil Recovery, St. Petersburg, Russia. pp 16–18
Rosenbrock H (1960) An automatic method for finding the greatest or least value of a function. Comput J 3(3):175–184
Sarma P, Durlofsky LJ, Aziz K, Chen WH (2006) Efficient real-time reservoir management using adjoint-based optimal control and model based updating. Comput Geosci 10:3–36
Schulze-Riegert R, Bagheri M, Krosche M, Kueck N, Ma D (2011) Multiple-objective optimization applied to well path design under geological uncertainty. SPE Reserv Simul Symp 21–23:2011
Su H-J, Oliver DS (2010) Smart well production optimization using an ensemble-based method. SPE Reserv Eval Eng 13(6):884–892
Sun Y, Wierstra D, Schaul T, Schmidhuber J (2009) Efficient natural evolution strategies. In: Proceedings of GECCO. pp 539–545
Tarantola A (2005) Inverse problem theory: methods for data fitting and model parameter estimation van
Essen GM, Zandvliet MJ, van den Hof PMJ, Bosgra OH, Jansen JD (2006) Robust waterflooding optimization of multiple geological scenarios. In: SPE Annual Technical Conference and Exhibition, San Antonio. Society of Petroleum Engineers, Texas **(SPE102913)**