



SPE 163657

Improving the Ensemble Optimization Method Through Covariance Matrix Adaptation (CMA-EnOpt)

R.M.Fonseca¹, O.Leeuwenburgh², P.M.J.Van den Hof^{1,3}, J.D.Jansen¹

¹ Delft University of Technology, ² TNO, ³ Eindhoven University of Technology

Copyright 2013, Society of Petroleum Engineers

This paper was prepared for presentation at the SPE Reservoir Simulation Symposium held in The Woodlands, Texas USA, 18–20 February 2013.

This paper was selected for presentation by an SPE program committee following review of information contained in an abstract submitted by the author(s). Contents of the paper have not been reviewed by the Society of Petroleum Engineers and are subject to correction by the author(s). The material does not necessarily reflect any position of the Society of Petroleum Engineers, its officers, or members. Electronic reproduction, distribution, or storage of any part of this paper without the written consent of the Society of Petroleum Engineers is prohibited. Permission to reproduce in print is restricted to an abstract of not more than 300 words; illustrations may not be copied. The abstract must contain conspicuous acknowledgment of SPE copyright.

Abstract

Ensemble Optimization (EnOpt) is a rapidly emerging method for reservoir model based production optimization. EnOpt uses an ensemble of controls to approximate the gradient of the objective function with respect to the controls. Current implementations of EnOpt use a Gaussian ensemble with a constant standard deviation, i.e. a diagonal covariance matrix with entries that remain constant during the optimization process. The Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) is a gradient-free optimization method, developed in the ‘machine learning’ community, which also uses an ensemble of controls but with a covariance matrix that is continually updated during the optimization process. It has shown to be an efficient method for several difficult small dimension optimization problems and has recently been applied in the petroleum industry for well location and production optimization. In this study we investigated the scope to improve the computational efficiency of EnOpt through the use of covariance adaptation (CMA-EnOpt). We optimized water flooding of a multi-layer sector model containing multiple sealing and non-sealing faults. The controls used were inflow control valve settings at pre-defined time intervals for injectors and producers with undiscounted net present value as the objective function. We compared EnOpt and CMA-EnOpt starting from identical covariance matrices. We achieved slightly higher (0.7%-1.8%) objective function values and modest speed-ups with CMA-EnOpt compared to EnOpt, depending on choice of user-defined parameters in both algorithms. However, the major benefit of CMA-EnOpt is its robustness with respect to the initial choice of the covariance matrix. A poor choice of the initial matrix can be detrimental to EnOpt, whereas the CMA-EnOpt performance is near-independent of the initial choice.

Introduction

Several studies have shown that there is considerable scope to improve the economic life-cycle performance of oil fields through the use of formal optimization methods in conjunction with reservoir simulation models. A very efficient way to perform such model-based life-cycle optimization is with the aid of gradient-based methods where the gradient is obtained through an adjoint technique. For an overview of this approach and a large number of references we refer to the review paper by Jansen (2011). The adjoint method is computationally very efficient, but, unfortunately, it is an intrusive method, requiring access to the simulator source code as well as extensive implementation efforts. Because it is practically impossible to access commercial simulator source codes for implementation of the adjoint there is a need for alternative methods for model-based production optimization in which the simulator is treated as a black-box.

One such method, Ensemble Optimization (EnOpt), has been shown to achieve good results for a variety of different reservoir models, a drawback being its lower computational efficiency and accuracy compared to the adjoint method. In EnOpt the gradient of the objective function with respect to the vector of control variables is approximated by evaluating the objective function values for an ensemble of control vectors, chosen from a multi-Gaussian random distribution with known mean and covariance matrix, and then using a least-squares approach. Predecessors to the EnOpt method were proposed by Lorentzen et al. (2006) and Nwazo (2006), whereafter Chen (2008) and Chen et al. (2009) gave systematic descriptions of the method as mostly used today. Thereafter, several publications addressed applications and computational aspects of the method; see e.g. Masoor et al. (2009), Chen and Oliver (2010), Su and Oliver (2010), Leeuwenburgh et al. (2010), and Chen and Oliver (2012).

As an alternative to exact or approximate gradient-based optimization methods one can revert to gradient-free methods such as genetic algorithms or evolutionary strategies as developed in the ‘machine-learning’ community. One of the latter, called the Covariance Matrix Adapted-Evolutionary Strategy (CMA-ES), has recently been used for well placement optimization by Ding (2008) and Bouzarkouna et al. (2011), a well control problem by Schulze-Riegert et al. (2011), and a smart well optimization problem by Pajonk et al. (2011). CMA-ES, developed by Hansen and co-workers starts from a random sample of control vectors with known statistics (mean and covariance matrix), computes the corresponding objective function values, selects the ‘best’ results, adapts the statistics for control vector sampling, and repeats the procedure until a converged objective function value is reached; see Hansen and Ostermeier (1996), Hansen and Ostermeier (2001), and Hansen (2006). The main idea is to systematically adapt the variance of the control vector sample in directions that have proven to be successful.

In this paper we propose an improvement to the EnOpt introduced by Chen (2008), hereafter referred to as standard EnOpt. In standard EnOpt a constant diagonal covariance matrix is used throughout the optimization. We propose to use standard EnOpt in combination with covariance adaptation, a modification that we will refer to as Covariance Matrix Adapted EnOpt (CMA-EnOpt). In the remainder of this paper we will first provide an overview of CMA-EnOpt, followed by its application to a synthetic 3D reservoir model. A comparison of the results to those obtained with standard EnOpt will illustrate the computational advantages of CMA-EnOpt for model-based production optimization.

Theory

In this section we give a brief overview of the theoretical basis of CMA-EnOpt. We first define our objective function followed by an overview of standard EnOpt and the proposed modification. We apply the usual expression for Net Present Value (NPV) as objective function J :

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

where $q_{o,k}$ is the oil production rate in bbl/day, $q_{wp,k}$ is the water production rate in bbl/day, $q_{wi,k}$ is the water injection rate in bbl/day, r_o is the price of oil produced in \$/bbl, r_{wp} is the price of water produced in \$/bbl, r_{wi} is the price of water injected in \$/bbl, Δt_k is the difference between consecutive time steps in days, b is the discount factor expressed as a fraction per year, t_k is the cumulative time in days corresponding to time step k , and τ_r is the reference time for discounting (365 days).

Ensemble Optimization (EnOpt)

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

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

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

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

where

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

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

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

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

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

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

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

where in a practical implementation we would solve a linear system of equations for \mathbf{g} rather than compute the inverse. For the derivation of equation (3), see any introductory linear algebra textbook, e.g. Strang (2006). Equation (3) can also be expressed as

$$\mathbf{g} = \mathbf{C}_{uu}^{-1} \mathbf{C}_{uj}, \quad (4)$$

where

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

are ensemble (sample) covariance and cross-covariance matrices respectively (Chen, 2008, and Chen and Oliver, 2009). Normally, however, we have an underdetermined case, i.e. $M < N$. This implies that the matrix product $\bar{\mathbf{U}}^T \bar{\mathbf{U}}$ is rank deficient such that we cannot directly compute its inverse or solve the associated system of equations. Instead we can compute the Moore-Penrose pseudo inverse using a singular value decomposition (SVD); see e.g. Strang (2006). Alternatively, Chen (2008) and Chen and Oliver (2009) propose to simply use

$$\mathbf{g}' = \mathbf{C}_{uj} = \mathbf{C}_{uu} \mathbf{g} \quad (5)$$

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

$$\mathbf{g}'' = \mathbf{C}_{uu} \mathbf{C}_{uj} = \mathbf{C}_{uu} \mathbf{C}_{uu} \mathbf{g}. \quad (6)$$

Both expressions (5) and (6) can be interpreted as modified (approximate) gradients. In our study we applied equation (3) with an SVD.

Update rules

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

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

where the superscript ℓ is the iteration counter, and α^{ℓ} is a step length in the direction of the gradient. We determined α^{ℓ} with the aid of an inexact line search and the Arjimo conditions (Nocedal and Wright, 2006). In more sophisticated optimization algorithms an improved update direction (i.e. one different from \mathbf{g}) is determined by employing optimization methods that make use of the second derivatives of J with respect to \mathbf{u} , i.e. of the Hessian matrix, or, more commonly, of approximations to the Hessian. In particular, so called quasi-Newton methods use gradient information of subsequent iterates to construct an approximate Hessian \mathbf{H}^{ℓ} . The corresponding update rule then becomes

$$\tilde{\mathbf{u}}^{\ell+1} = \tilde{\mathbf{u}}^{\ell} + \alpha^{\ell} (\mathbf{H}^{\ell})^{-1} \mathbf{g}^{\ell}, \quad (8)$$

where the definition of \mathbf{H}^{ℓ} depends on the particular variety of quasi-Newton method applied; see e.g. Nocedal and Wright (2006) or Luenberger and Ye (2010) for further details. Note that, as usual, in an actual implementation computing the inverse is avoided. The gradient is the direction of a tangent (hyper) plane in a point touching the objective function, while the Hessian gives curvature information in that point, i.e. it defines a convex quadratic function. The basic idea underlying the various quasi-Newton methods is that the curvature information contained in the approximate Hessian is gradually increased by subsequent inclusion of gradient information from previous iterations. Although we do not use a quasi-Newton algorithm in the optimization examples in our study, the concept of using information from subsequent iterates to improve the estimate of the curvature of the objective function is an important aspect of CMA-ES, and thus also of CMA-EnOpt. Moreover, we note that the use of the preconditioner \mathbf{C}_{uu} in equation (5) plays a similar role as the preconditioner \mathbf{H}^{-1} in equation (8). Indeed it can be shown that for any point on a convex quadratic objective function the covariance matrix is equal (upto a scaling factor) to the inverse Hessian. For more general objective functions the equality is an approximation.

Covariance Matrix Adaptation (CMA)

CMA-ES is a stochastic iterative optimization method in which the covariance matrix is updated at every iteration such that its largest principal direction, i.e. the eigenvector corresponding to its largest eigenvalue, is (approximately) re-aligned in the direction of the maximal increase of the objective function. CMA-ES uses two types of updates for the covariance matrix as briefly explained below. For a detailed overview we refer to Hansen (2006, 2011).

Rank-Mu Update

The motivation behind a rank-mu update is to use information within one single iteration (i.e. one ensemble of random control vectors $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_M\}^\ell$ and their corresponding objective function values $\{J_1, J_2, \dots, J_M\}^\ell$) through selecting the 'best' μ members (i.e. those corresponding to the μ highest objective function values) out of the M ensemble members:

$$\tilde{\mathbf{C}}_{uu}^{\ell+1} = (1 - c_\mu) \tilde{\mathbf{C}}_{uu}^\ell + c_\mu \frac{1}{\mu} \tilde{\mathbf{U}}^T \tilde{\mathbf{U}} \quad \text{with} \quad \tilde{\mathbf{U}} = [\mathbf{u}_1 - \tilde{\mathbf{u}}^\ell \quad \mathbf{u}_2 - \tilde{\mathbf{u}}^\ell \quad \dots \quad \mathbf{u}_\mu - \tilde{\mathbf{u}}^\ell]^T, \quad (9)$$

where $0 < c_\mu < 1$ is a learning rate, and where the control vectors $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_M$ have been ranked such that for their corresponding objective function values it holds that

$$J_1 \geq J_2 \geq \dots \geq J_\mu > J_{\mu+1} \geq \dots \geq J_M.$$

Equation (7) is called a rank-mu update because the matrix product $\tilde{\mathbf{U}}^T \tilde{\mathbf{U}}$ is, at most, of rank μ . Note that we use the distribution mean $\tilde{\mathbf{u}}^\ell$ instead of the ensemble mean $\bar{\mathbf{u}}^\ell$. It can be shown that, as a result, for $\mu = M$ and $c_\mu = 1$, $\tilde{\mathbf{C}}_{uu}^{\ell+1}$ is an unbiased estimator of the distribution covariance $\tilde{\mathbf{C}}^{\ell+1}$ (Hansen, 2006). The choice of the learning rate c_μ is crucial to the success of the optimization. Low values (close to zero) lead to slow learning whilst high values (close to one) may lead to degeneration of the covariance matrix. The choice of μ is user dependent; in this study we used $\mu = M/4$.

Rank-One Update

In equation (9) the covariance matrix update is determined using the best objective function values within one single iteration. It is also possible to update the covariance matrix by using information from subsequent iterates, in the same fashion as updating the Hessian in quasi-Newton methods. The necessary expression, derived in Hansen (2006), is given by

$$\tilde{\mathbf{C}}_{uu}^{\ell+1} = (1 - c_1) \tilde{\mathbf{C}}_{uu}^\ell + c_1 \mathbf{p}^{\ell+1} (\mathbf{p}^{\ell+1})^T, \quad (10)$$

where c_1 is the learning rate and \mathbf{p} is the 'evolution path', which is a function of iterates \mathbf{u} in earlier steps. For the exact definition of \mathbf{p} , see Hansen (2006). Because the outer product of two vectors results in a matrix of rank one, expression (10) is referred to as a rank-one update. The rank-one update has been shown to be particularly powerful when using small ensemble sizes (Hansen, 2006). Combining equations (9) and (10) one obtains the update rule

$$\tilde{\mathbf{C}}_{uu}^{\ell+1} = (1 - c_\mu - c_1) \tilde{\mathbf{C}}_{uu}^\ell + \underbrace{c_\mu \frac{1}{\mu} \tilde{\mathbf{U}}^T \tilde{\mathbf{U}}}_{\text{rank-}\mu \text{ update}} + \underbrace{c_1 \mathbf{p}^{\ell+1} (\mathbf{p}^{\ell+1})^T}_{\text{rank-one update}}. \quad (11)$$

Equation (11) utilizes information within one iteration as well as information from successive iterations. Hansen (2011) suggests that the former is more important when using a larger ensemble and the latter in smaller ensembles. Several variations to equation (11) have been proposed; see e.g. (Arnold and Hansen, 2010). In particular, in our study we employed a slightly simplified version where only the diagonal elements of the covariance matrix are updated.

Covariance Matrix Adapted-EnOpt (CMA-EnOpt)

In standard EnOpt a constant distribution covariance matrix $\tilde{\mathbf{C}}$ is used which is typically chosen as a diagonal covariance matrix with equal diagonal elements σ^2 . The value of the standard deviation σ can have a major influence on the quality of the approximate gradient and therefore on the performance of the optimization algorithm. However there is no well defined method to choose this value. Thus we propose to use the covariance adaption strategy described above to gradually improve the distribution covariance matrix in the the EnOpt method, leading to a Covariance Matrix Adapted-EnOpt method (CMA-EnOpt).

Algorithm

1. Set $\ell = 0$. Choose an initial control vector $\tilde{\mathbf{u}}^0$.
2. Stochastically generate an ensemble of controls around $\tilde{\mathbf{u}}^0$ with a user defined diagonal covariance matrix $\tilde{\mathbf{C}} = \sigma^2 \mathbf{I}$.
3. Run a reservoir simulation for every member of the ensemble and calculate the corresponding objective function values using equation (1).
4. If converged or if maximum allowed number of iterations reached, stop. Else, continue.

5. Compute the EnOpt gradient \mathbf{g}^ℓ using equation (3).
6. Perform an incomplete line search to determine α^ℓ and compute the updated control vector $\tilde{\mathbf{u}}^{\ell+1}$ using equation (7).
7. Compute the updated (diagonal elements of the) covariance matrix $\tilde{\mathbf{C}}_{uu}^{\ell+1}$ using equation (11).
8. Regenerate an ensemble of controls around $\tilde{\mathbf{u}}^{\ell+1}$ with the updated covariance matrix $\tilde{\mathbf{C}}_{uu}^{\ell+1}$.
9. Set $\ell = \ell + 1$. Return to step 3.

3D Synthetic Reservoir Model

Advances in technology have led to an increase in the application of downhole chokes or 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 are manipulated to optimize waterflooding over the producing life of the reservoir which is 15 years, or 5470 days. The model, illustrated in **Fig. 1**, consists of $25 \times 32 \times 5 = 4000$ grid blocks. The approximate size of the grid blocks is $110 \times 90 \times 20$ m, representing an area of $2.5 \times 3.5 \times 0.1$ km. The geological structure consists of connected uplifted/offset blocks. The reservoir is produced using an inverted five-spot well pattern, i.e. four producers at the edges of the grid with an injector in the centre of the grid. The reservoir is divided into five layers having different horizontal permeabilities, but the permeability in each individual layer is constant. There is a sealing fault on the North-Western side of the block, close to producer 1. The initial average reservoir pressure is 200 bars. **Table 1** lists the geological and fluid properties used to describe the model.

TABLE 1: GEOLOGICAL AND FLOW PROPERTIES OF THE SYNTHETIC RESERVOIR MODEL		
Property	Values	Units
Porosity	0.2	--
Permeability- (layer 1 – layer 5)	100-300-50-600-100	mD
Reservoir pressure	200	bar at 1950 m
Density of oil	800	kg/m ³
Density of water	1000	kg/m ³
Temperature	77	°C
Oil compressibility	4e-5	1/bar
Water compressibility	4e-5	1/bar
Viscosity of oil	2	cP at 1 bar
Viscosity of water	0.5	cP at 1 bar

The wells penetrate all five layers with an ICV in every layer resulting in a total of 25 controls per time step. The producing life of the reservoir is divided into 15 time intervals of one year (365 days) each, which results in a total of 375 controls which are 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. 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. No capillary pressures are included. The reservoir rock is incompressible. We used a commercial fully implicit finite difference black oil simulator (Eclipse, 2011).

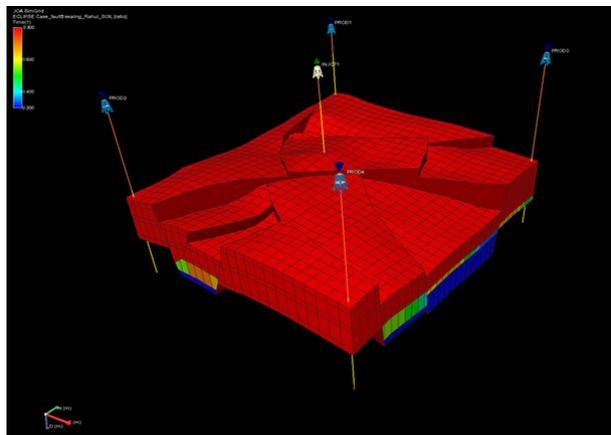


Fig. 1: Reservoir model. The colors indicate the initial water saturation.

Comparison between standard EnOpt and CMA-EnOpt

We performed several comparisons between standard EnOpt and CMA-EnOpt to optimize the ICV settings with the aim to maximize NPV as defined in equation (1). We used an oil price $r_o = 130$ $\$/\text{m}^3$, water production costs $r_{wp} = 25$ $\$/\text{m}^3$, and water injection costs $r_{wi} = 6$ $\$/\text{m}^3$. The discount rate b was set to 0 i.e. we used undiscounted NPV. Well productivity index (PI) multipliers were used to model the ICVs in the simulator with bounds of 1×10^{-4} and 1. The starting point for the optimization was an initial control vector having values equal to 1. Thus all the ICVs/chokes were fully open as a starting strategy. The initial value of σ was equal to 0.1 and we used a fixed ensemble size of 50 samples. Random control values outside the range $1 \times 10^{-4} - 1$ were simply reset to their bounds. The optimization was allowed to run for 80 iterations which usually resulted in a near horizontal (i.e. nearly converged) objective function graph; see Fig. 2. We used various settings of the initial distribution covariance matrix (i.e. of the standard deviation σ) required to generate the initial ensemble, leading to different optimization results, see Fig. 2. The best standard EnOpt run resulted in an objective function value of 8.95×10^9 $\$$ while the best CMA-EnOpt run achieved a slightly higher value of 9.01 $\$$, i.e. 0.7% higher. An illustration of the corresponding ICV settings for one of the wells is presented in Fig.3.

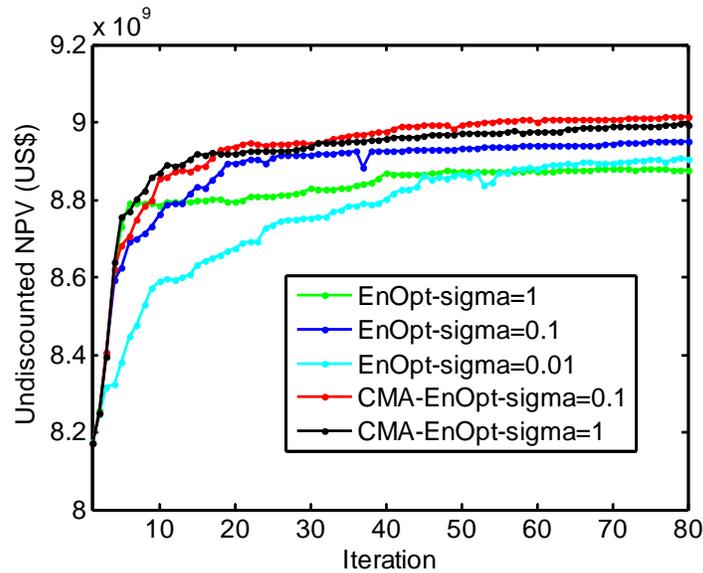


Fig. 2: Comparison of optimization performance for different initial covariance matrices.

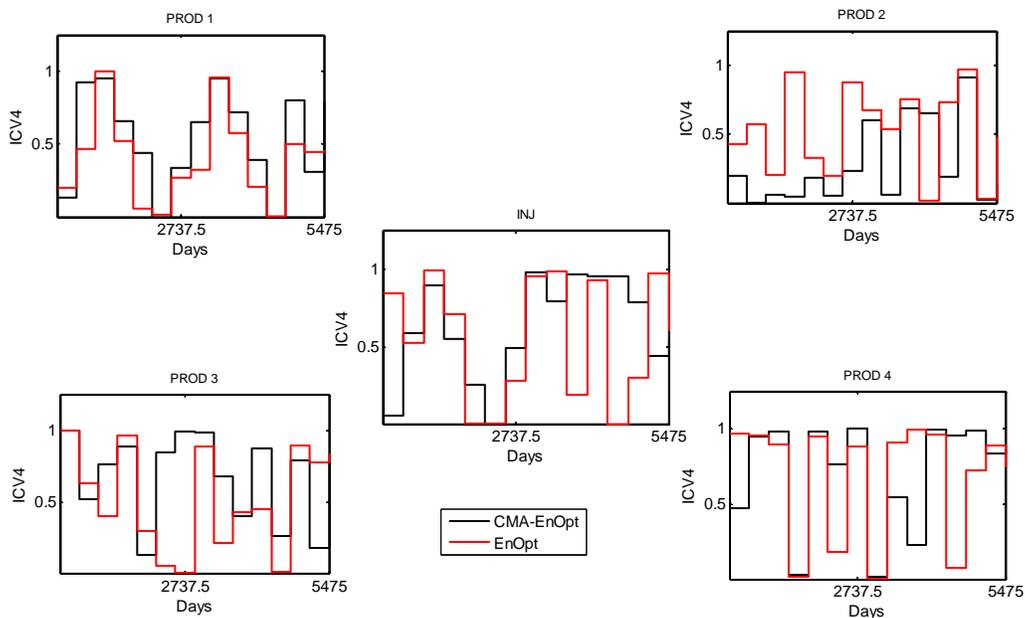


Fig. 3: Optimal control settings for ICV 4 as computed by CMA-EnOpt (black) and EnOpt (red).

Improved Robustness

Exact gradient-based methods are inherently local methods (unless forced to take non-optimal steps) while most gradient-free methods have been shown to possess more global search characteristics. In the EnOpt method the choice of the initial distribution covariance matrix is very important: large variances results in a more global search strategy and small variances a more local one. We tested CMA-EnOpt and standard EnOpt with different initial starting values for the standard deviation σ . Fig. 2 illustrates that the choice of the initial covariance matrix has a significant impact on the performance of standard EnOpt, because the matrix remains constant throughout the optimization. When using a large value of σ for EnOpt (green line), the algorithm achieves poor results, but also for a very small value of σ (light blue line) EnOpt does not perform well, as opposed to its performance for an intermediate value of σ (blue line). For CMA-EnOpt (red and black lines) the initial starting covariance matrices have little impact on the final objective function values and both runs achieve better results than EnOpt. When starting with a smaller value of σ , CMA-EnOpt achieves a slightly higher objective function value than when starting with an identity covariance matrix ($\sigma=1$). This examples shows that CMA-EnOpt performs slightly better than standard EnOpt with an an optimized distribution covariance matrix. However CMA-EnOpt performs much better than standard EnOpt with a poor initial guess for the covariance. More than computational gains, the strength of CMA-EnOpt therefore appears to be its robustness with respect to initial choices of the covariance matrix.

Learning Rates

CMA-EnOpt contains several parameters that require user defined values, in particular the learning rates c_μ and c_1 . Hansen (2011) discusses strategies to determine these rates based on the dimension of the problem. However his test cases are of a relatively small dimension, whereas production optimization problems typically have hundreds to thousands of control variables. Fig. 4 illustrates that if we choose the learning rates too small, the advantage of CMA-EnOpt over standard EnOpt is negligible if any. Higher learning rates have shown to achieve significantly better results as shown in Fig. 4. These results were obtained when we only updated the diagonal elements of the covariance matrix. Hansen (2011) reports that if the full covariance matrix is updated, high learning rates can have a detrimental impact on the optimization because they may lead to covariance matrix degeneration.

Fig. 5 illustrates standard deviations σ for two control variables (i.e. the corresponding square root diagonal values of the distribution covariance matrix) for different learning rates. The high learning rate case (red lines) reflects an 80-20 update rule with $c_\mu = 0.12$ and $c_1 = 0.08$. In this case 20% new information is incorporated into the covariance matrix every iteration. We observe from Fig. 5 that different control variables have different optimal standard deviations. The slow learning rates, based on the recommendations for small-size problems as described in Hansen (2011) and Hansen (2006), reflect a 99.5-0.5 update rule i.e. only 0.5% new information is incorporated into the updated covariance matrix at each iteration. This leads to negligible changes in the standard deviation (black line, overlapped by the blue line). In this case CMA-EnOpt behaves just like standard EnOpt with a fixed standard deviation (blue line). Apparently, learning rates have a significant impact on the performance of the optimization algorithm.

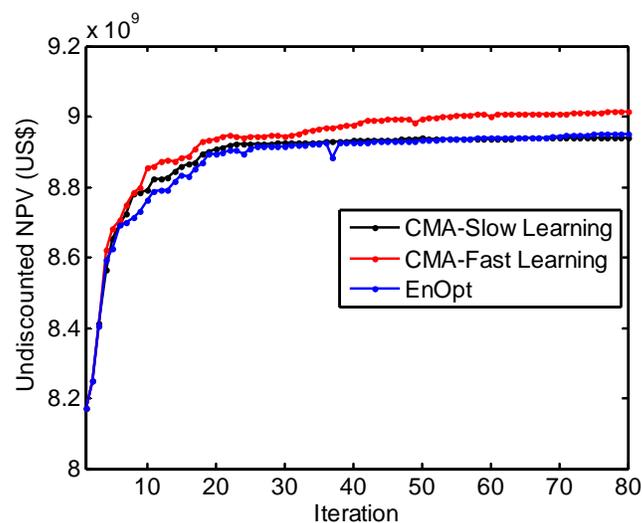


Fig. 4: Comparison of the effect of different learning rates on the optimization.

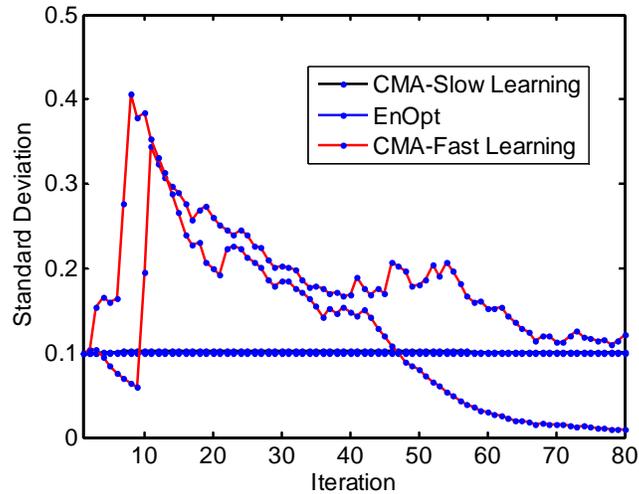


Fig. 5: Illustration of impact of learning rate on the success of the optimization.

Discussion

We performed an initial investigation into the potential benefits of CMA-EnOpt over standard EnOpt. An obvious next step would be the comparison of both methods with a CMA-ES, i.e. with the evolutionary strategy that formed the basis for the covariance matrix adaptation strategy in CMA-EnOpt. Initial comparisons, not reported in this paper, indicate that the CMA-ES results are inferior (in terms of computational efficiency) to those of the two EnOpt varieties. I.e. explicitly using (approximate) gradient information seems to pay off. However, further numerical comparisons are required to evaluate the strength and weaknesses of the three methods.

Conclusions

- A comparison between CMA-EnOpt and standard EnOpt revealed slightly higher (0.7%-1.8%) objective function values and modest speed-ups, depending on choice of user-defined parameters in both algorithms.
- The major benefit of CMA-EnOpt is its robustness with respect to the initial choice of the covariance matrix. A poor choice of the initial matrix can be detrimental to EnOpt, whereas the CMA-EnOpt performance is near-independent of the initial choice.
- Learning rates are crucial for the success of CMA-EnOpt. For the high dimensional problem used in this study (375 control variables), an 80%-20% update rule proved to be successful.

Nomenclature

- b = discount rate
- c = learning rate
- $\tilde{\mathbf{C}}$ = distribution covariance matrix
- \mathbf{C}_{uw} = ensemble cross-covariance matrix
- \mathbf{C}_{uu} = ensemble covariance matrix
- \mathbf{g} = gradient vector
- ℓ = iteration counter
- \mathbf{H} = Hessian matrix
- \mathbf{j} = vector of mean-shifted objective function values
- J = objective function value
- \bar{J} = mean objective function value
- k = time step counter
- K = total number of time steps
- M = number of ensemble members
- N = number of control variables
- \mathbf{p} = evolution path
- q = flow rates

r = price per unit volume
 t = time
 u = control variable
 \mathbf{u} = vector of control variables
 $\bar{\mathbf{u}}$ = ensemble mean
 $\bar{\mathbf{u}}$ = distribution mean
 $\bar{\mathbf{U}}$ = matrix of ensemble mean-shifted control vectors
 $\tilde{\mathbf{U}}$ = matrix of distribution mean-shifted control vectors
 α = step size
 μ = number of 'best' ensemble members
 τ = reference time for discounting

Subscripts

o = oil
 w = water
 wp = produced water
 wi = injected water

Acknowledgements

This research was carried out within the context of the ISAPP Knowledge Centre. ISAPP (Integrated Systems Approach to Petroleum Production) is a joint project of TNO, Delft University of Technology, ENI, Statoil and Petrobras.

References

- Arnold, D.V. and Hansen, N. 2010. Active Covariance Matrix Adaptation for the (1+ 1)-CMA-ES. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation. GECCO'10*, Portland, Oregon, USA, 7-11 July.
- Bouzarkouna, Z., Ding, Y.D. and Auger, A. 2011. Partially Separated Meta-models with Evolution Strategies for Well Placement Optimization. Paper SPE-143292-MS presented at the SPE EUROPEC/EAGE Annual Conference and Exhibition, Vienna, Austria, 23-26 May. DOI: 10.2118/143292-MS
- Chen, Y. 2008. Efficient Ensemble based Reservoir Management. PhD Thesis, University of Oklahoma, USA.
- Chen, Y. and Oliver, D.S. 2010. Ensemble-Based Closed-Loop Optimization Applied to Brugge Field. *SPEEE* **13** (1) 56-71. DOI: 10.2118/118926-PA
- Chen, Y. and Oliver, D.S. 2012. Localization of Ensemble-Based Control-Setting Updates for Production Optimization. *SPEJ* **17** (1) 122-136. DOI: 10.2118/125042-PA
- Chen, Y., Oliver, D.S. and Zhang, D. 2009. Efficient Ensemble-Based Closed-Loop Production Optimization. *SPEJ* **14** (4) 634-645. DOI: 10.2118/112873-PA
- Ding, Y. 2008. Optimization of Well Placement Using Evolutionary Algorithms. Paper SPE-113525-MS presented at the Europec/EAGE Conference and Exhibition, Rome, Italy, 9-12 June. DOI: 10.2118/113525-MS
- Eclipse, 2011. <http://www.slb.com/services/software/reseng/eclipse>
- Hansen N. 2006. The CMA evolution strategy: a comparing review. In Lozano, J.A., Larranaga, P, Inza, I., and Bengoetxea, E., eds., *Towards a new evolutionary computation. Advances on estimation of distribution algorithms*, 75–102. Springer
- Hansen, N. 2011. The CMA Evolution Strategy: A Tutorial. <http://www.lri.fr/~hansen/cmatutorial110628.pdf>
- Hansen, N. and Ostermeier, A. 1996. Adapting Arbitrary Normal Mutation Distributions in Evolution Strategies: The Covariance Matrix Adaptation. In *Proc. 1996 IEEE International Conference on Evolutionary Computation*, Picasataway, New Jersey, USA.
- Hansen, N. and Ostermeier, A. 2001. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary computation* **9** (2) 159-195.
- Jansen, J.D. 2011. Adjoint-based Optimization of Multi-Phase Flow Through Porous Media—A review. *Computers & Fluids* **46** (1) 40-51. DOI: 10.1016/j.compfluid.2010.09.039.
- Leeuwenburgh, O., Egberts, P.J.P. and Abbink, O.A. 2010. Ensemble Methods for Reservoir Life-Cycle Optimization and Well Placement. Paper SPE-136916-MS presented at the SPE/DGS Saudi Arabia Section Technical Symposium and Exhibition, Al-Khobar, Saudi Arabia, 4-7 April. DOI: 10.2118/136916-MS
- Lorentzen, R.J., Berg, A., Naevdal, G. and 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. DOI:

10.2118/99690-MS

Luenberger, D.G. and Ye, Y., 2010: *Linear and nonlinear programming*, 3rd ed. Springer, New York

Masroor, M.C., Phale, H.A., Liu, N. and Oliver, D.S. 2009. An Improved Approach for Ensemble-Based Production Optimization. Paper SPE 121305 presented at the SPE Western Regional Meeting, San Jose, USA, 24-26 March. DOI: 10.2118/121305-MS.

Nocedal, J. and Wright, S.J., 2006: *Numerical optimization*, 2nd ed., Springer, New York

Pajonk, O., Schulze-Riegert, R., Krosche, M., Hassan, M. and Nwakile, M.M. 2011. Ensemble-Based Water Flooding Optimization Applied to Mature Fields. Paper SPE-142621-MS presented at the SPE Middle East Oil and Gas Show and Conference, Manama, Bahrain, 25-28 September 2011. DOI: 10.2118/142621-MS

Ros, R. and Hansen, N. 2008. A simple modification in CMA-ES achieving linear time and space complexity. *Parallel Problem Solving from Nature-PPSN X*, 296-305.

Schulze-Riegert, R., Bagheri, M., Krosche, M., Kueck, N. and Ma, D. 2011. Multiple-Objective Optimization Applied to Well Path Design under Geological Uncertainty. Paper SPE-141712-MS presented at the SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, 21-23 February 2011. DOI: 10.2118/141712-MS

Strang, G., 2006: *Linear algebra and its applications*, 4th ed., Thomson Brooks/Cole, Pacific Grove

Su, H.-J. and Oliver, D.S. 2010. Smart Well Production Optimization Using An Ensemble-Based Method. *SPEE* **13** (6) 884-892. DOI: 10.2118/126072-PA