A Reduced Random Sampling Strategy for Fast Robust Well Placement Optimization

Mansoureh Jesmani*[1], Behnam Jafarpour†[2], Mathias C. Bellout‡[3], and Bjarne Foss§[4]

[1] *Department of Engineering Cybernetics, NTNU, Tronheim, Norway*
[2] *University of Southern California, 925 Bloom Walk, HED 313, Los Angeles, CA 90089, USA*
[3] *Department of Petroleum Engineering and Applied Geophysics, NTNU*
[4] *Department of Engineering Cybernetics, NTNU, Tronheim, Norway*

Published: 29 August 2019, Journal of Petroleum Science and Engineering vol. 184 pp.106414
DOI: 10.1016/j.petrol.2019.106414

Abstract

Model-based decision-making in oilfield development often involves hundreds of computationally demanding reservoir simulation runs. In particular, well placement optimization under uncertainty in the geologic representation of the reservoir model is an overly time-consuming procedure as the performance of any proposed well configuration needs to be evaluated over multiple realizations, using computationally expensive flow simulations. To reduce computation, we propose an efficient robust optimization procedure in which at each iteration of the optimization procedure, instead of evaluating the well configuration over all available realizations, we approximate the expected performance using a small subset of *randomly* selected model realizations. Since the samples are selected randomly, all the realizations are expected to eventually be included in the performance evaluation after a certain number of iterations. However, using only a few random realizations to compute the expected cost function introduces noise in the estimated objective function, necessitating the use of a stochastic optimizer. In this paper, we use the Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm, which is known to be robust against noise in the objective function. We first evaluate the performance of different forms of the SPSA algorithm (including

*e-mail: jesmani.mansoureh@gmail.com
†e-mail: behnam.jafarpour@usc.edu
‡e-mail: mathias.bellout@ntnu.no
§e-mail: bjarne.foss@ntnu.no

discrete, continuous, and adaptive) using several numerical experiments, followed by a discussion of the properties of the proposed reduced random sampling approach and comparison with global optimization techniques. The method is applied to several numerical experiments, including case studies involving vertical, horizontal, and lateral wells, to evaluate its performance. The results from these experiments indicate that the reduced random sampling approach can provide significant computational gain with minimal impact on the attained optimization performance.

1 Introduction

Drilling wells is a major capital expense in hydrocarbon reservoir development. Well locations and their trajectories have a significant impact on the recovery performance. Optimizing the location and configuration of wells has therefore received significant attention in recent years. The optimization techniques which have been applied to the well placement problem can be classified into two groups of approaches: gradient-based and derivative-free approaches. In the following, the application of these two types of approaches to the well placement problem is briefly reviewed.

For gradient-based recovery optimization, adjoint formulations offer a very efficient gradient calculation approach, which often introduces a computation overhead less than the forward simulation time [21]. For well placement optimization, however, adjoint-based gradient calculation with respect to well locations is not straightforward. Indirect approaches have been devised to implement gradient-based algorithms for well placement optimization [38, 48, 52]. An important disadvantage of the adjoint method is the need to access and modify the reservoir simulator source code directly, which is not trivial for commercial simulation codes. A simple approach to obtain the required gradients is by using Finite Difference Method (FDM) approximation, which is computationally expensive when a large number of optimization variables is involved [3]. An efficient alternative to FDM is stochastic gradient approximation, including ensemble-based approximation or simultaneous perturbation algorithms [3, 24, 26].

In derivative-free methods, as the name suggests, the calculation of the objective function gradient is not required. Derivative-free methods can also be classified into local or global methods. Global derivative-free algorithms usually can deal with multiple local optima, while local derivative free algorithms only converge to local solutions. Three different deterministic derivative free methods (Hooke-Jeeves Direct Search (HDDS), Generalized Pattern Search (GPS), and a hybrid optimization parallel search package (HOPSPACK)) have been applied to the well placement problem in [6]. Further, stochastic derivative-free methods have been applied to the well placement optimization problem such as different types of Genetic Algorithm (GA)s [1, 3, 18, 51], covariance matrix adaptation evolution strategy (CMA-ES) [7], Particle Swarm Optimization (PSO) [23, 35], and Simulated Annealing (SA) [5].

In formulating reservoir management optimization problems, it is imperative to account for the uncertainty in the geological description of the reservoir model and the related input parameters, including the distribution of rock flow properties such as porosity, permeability, and fluid contacts. A commonly used approach to reflect the uncertainty in these properties is to assign an (often high-dimensional) Probability Density Function (PDF) to them. Unfortunately, even when such PDFs can be defined, it is not trivial to directly incorporate them into optimization problem formulations. In general, sampling methods and Monte-Carlo simulation techniques provide a practical (approximate) approach for incorporating model input uncertainty in optimization problems. To adequately approximate PDFs with samples, however, a large number of samples must be drawn, especially when a PDF function is complex. Thus, to take into account the geological uncertainty in the well placement problem, the performance of candidate well configurations has to be evaluated over a large number of geological realizations. In practice, however, because of the limited computational resources, a small number of model realizations is often selected. In realistic problems where computationally demanding forward reservoir simulation runs have to be performed, even with a small number of realizations (e.g., 100 realizations), the computational cost can become prohibitively expensive due to the large number of iterations involved in well placement optimization.

Several methods have been proposed in the literature to reduce the computational load of running multiple realizations [1, 18, 29, 40, 49]. In [1], the optimal placement of nonconventional wells under geological uncertainty is addressed by optimizing over multiple realizations subject to a prescribed risk attitude. The authors used a statistical proxy based on a cluster analysis approach to reduce the number of necessary simulation runs. Using the proxy, a systematic approach was used to select a representative subset of the initial models for a full simulation to approximate the output uncertainty. The authors showed that optimization results obtained by using up to 20% of the total number of scenarios (this subset was determined by the proxy), were very close to those achieved by including all the initial models in the optimization. In [49], Wang et al. propose a Retrospective Optimization (RO) method in which the original problem involving a large number of realizations is treated as a sequence of optimization subproblems with increasing number of realizations. At the beginning, the problem is started using a smaller number of realizations; the number of realizations is then gradually increased through the sequence of optimization subproblems. The initial solution for the current subproblem is simply the solution returned by the previous subproblem. Using small sample sizes at the beginning of the sequence leads to moderate computational load. Towards the end of the sequence, despite the large number of realizations, the initial solutions are typically close to the optimum, which may reduce the required number of iterations needed to converge to the optimum solution. One issue in adopting such a strategy with local optimization techniques is the possibility of biasing the final solution toward a local solution that has been obtained at early optimization subproblems with very few samples. Guyaguler and Horne [18] introduce utility theory in the well placement problem, in which the utility framework transforms the

stochastic problem into a deterministic one. They demonstrate how the utility framework can quantify both the influence of uncertainty in the reservoir description and the risk attitude of the decision maker.

In this paper, we propose a reduced sampling strategy whereby, at each optimization iteration, the expected performance of a given well configuration is approximated by a small subset of randomly selected model realizations. Because of the random sample selection, over a large number of iterations, it is expected that all samples are included at least once (and potentially multiple times, depending on the sample reduction ratio and number of iterations) in the performance evaluation. However, using a subset of random realizations at each iteration introduces noise when computing the objective function. To alleviate this problem, we implement the Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm, which is known to be robust against noisy objective functions. The robustness of SPSA is discussed in detail in the original publications by Spall [1992,1998], where the convergence proof of SPSA is based on the properties of the noise in the objective function.

In this work, we therefore optimize the trajectory of wells (including vertical, horizontal, and deviated wells) using a continuous variant of SPSA. The SPSA algorithm is a local optimization method that uses stochastic gradient approximation and is easy to implement.

SPSA was first applied to the well placement problem in [3]. For some cases, the authors observed that the SPSA algorithm outperformed the Nelder-Mead Simplex method, and GA. Discrete or integer versions of SPSA have been used for vertical well placement in [3, 29]. In [28], Li and Jafarpour use the SPSA algorithm for well placement in a joint well placement and production optimization problem to show the advantage of including variable controls in solving the well placement problem. In a follow-up paper [29], the authors combined the well placement and control problems into a joint optimization algorithm using the SPSA algorithm. In [29], Li et al. also investigate the behavior of their algorithm under geologic uncertainty and show preliminary results suggesting that, in robust optimization, using only a small random subset of model realizations could result in similar optimization performance as in the case where the entire ensemble of models is used. In this paper, we build on the preliminary work of [29] to investigate the performance of SPSA with small sample sizes in robust optimization.

Early petroleum applications of the SPSA algorithm were performed by [3, 17]. In [17], both SPSA and Adaptive SPSA (ASP) are applied to the history matching problem. The ASP algorithm uses the stochastic approximation of the gradient and the Hessian matrix. For some cases, the authors showed that the ASP algorithm achieved a superior rate of convergence compared to the SPSA algorithm. In [47], SPSA is implemented for production optimization, and compared with the steepest ascent method in which the gradient is calculated using the adjoint method. In some cases SPSA with an average stochastic gradient is shown to provide reasonable well control solutions. However, the convergence rate of the SPSA algorithm in those examples is slower than the steepest ascent method. A stochastic Gaussian search direction (SGSD) algo-

rithm is proposed in [27] to generate the stochastic search direction (instead of the Bernoulli distribution in the original SPSA algorithm) in history matching problems. Furthermore, a prior covariance matrix is proposed to approximate the inverse Hessian matrix. In [13], a modified version of SPSA (G-SPSA), where Gaussian perturbations are applied, is used to estimate the pre-conditioned gradient of an augmented Lagrangian in optimizing well controls, which is used to directly incorporate all inequality and equality constraints. To impose a degree of temporal smoothness on the optimization variables, in [13] a covariance matrix is used for pre-conditioning. Furthermore, in [12] the authors show that Ensemble-Based Optimization (EnOpt) [10] is a special case of the modified version of SPSA, i.e., smoothed G-SPSA. In the EnOpt method, the ensemble is used to determine the sensitivities of the objective function with respect to the optimization variables. The expectation of the smoothed G-SPSA gradient and that of the EnOpt gradient represent first-order approximation of a squared covariance matrix times the true gradient.

A particular property of the search space corresponding to the well placement problem is that reservoir heterogeneity is likely to result in a highly non-smooth objective function containing multiple optima [35]. Given this type of topology, gradient-based searches for optimizing well locations may converge to local optima within a few iterations, making the problem highly dependent on the initial well configuration. However, the local optima in these problems tend to have similar values, indicating that the objective function may contain a ridge-like topology. In this work, we compare the performance of SPSA with PSO, which is a global search algorithm, for several numerical experiments. The PSO algorithm [25] is inspired by the social behavior of animals such as bird flocking. PSO is initialized with a population of random solutions, called particles. Each particle is assigned a random velocity according to the experiences of the particle itself and its neighborhoods. In [35], PSO and binary Genetic Algorithm (bGA) are applied to several well placement problems, where it is observed that, on average, the PSO algorithm provides comparable or better results.

This paper is organized as follows: We first formulate the well placement problem under geological uncertainty. We then present the SPSA algorithm and a few of its variants, and discuss their application to the well placement problem. We continue this section with a discussion of different sampling methods. Results from several numerical experiments, including the placement of vertical, horizontal, and lateral wells, are presented and discussed before closing the paper with our conclusions.

2 Problem formulation

The well placement problem under uncertainty can be formulated as follows:

$$\max_{\boldsymbol{\zeta}} J_T(\boldsymbol{\zeta}, \mathbf{u}^n, \theta), \tag{1a}$$

subject to:

$$C_{wl}(\boldsymbol{\zeta}), C_{rb}(\boldsymbol{\zeta}, \theta), C_{wo}(\boldsymbol{\zeta}), C_{wd}(\boldsymbol{\zeta}) \leq 0, \tag{1b}$$

$$\mathbf{x}^0(\theta) = \mathbf{x}_0(\theta), \tag{1c}$$

$$\boldsymbol{g}^n(\mathbf{x}^{n+1}, \mathbf{x}^n, \boldsymbol{\zeta}, \mathbf{u}^n, \theta) = \mathbf{0}, \qquad n = 0, 1, \cdots, N, \tag{1d}$$

where $\boldsymbol{\zeta}$ represents well coordinates, $\mathbf{u}^n$ denotes well controls, θ stands for the uncertain parameters (e.g., permeability, porosity), $\mathbf{x}^n$ refers to the dynamic states of the simulation model g^n that describes the fluid flow governing equations at time step n; C_{wl}, C_{rb}, C_{wo}, and C_{wd} represent the well length, reservoir boundaries, well orientation, and well distance constraints (for more details see [22]); and N is the total number of time steps. Note that the well length, well orientation, and inter-well distance constraints in (1b) are functions of the well coordinate variables, while the reservoir boundary constraint is a function of both well coordinates and uncertain parameters, where the geometry of the reservoir is also uncertain. The probability distribution functions for the uncertain parameters θ are either generally unknown because of the complexity of real oil reservoirs, or they are known but do not lend themselves to nonlinear propagation using complex reservoir simulation equations. A practical way to represent and propagate the uncertainty in these parameters is to use their stochastic realizations (samples) in a Monte-Carlo simulation framework.

Different measures of risk can be used for the objective function in (1a). The nominal profit approach is based on using, as objective function, the profit of one single realization, which can be different from that of the true reservoir model. The worst-case scenario approach is a conservative method in which the lowest profit within the range of possible outcomes is considered as the objective function. An alternative to the nominal and worst-case scenario approaches is the certainty equivalence approach, in which the profit is obtained from the expected value of the uncertain parameters. The expected profit approach takes the mean of the economic measure over model realizations. The main drawback of this approach is that it is possible to have very low profit realization, while the expected profit is high. To address this shortcoming, the mean variance approach uses a weighted sum of the mean and variance of the profit over all realizations, while the conditional value at risk approach maximizes the tail of the profit distribution beyond a certain quantile level. The worst-case scenario and conditional value at risk approaches are selected as appropriate risk measures for risk minimization in [8]. In this work, without loss of generality, we use the expected profit approach, even though other measures can also be used in the proposed random sampling strategy. The well placement problem under

uncertainty may now be formulated as follows:

$$\max_{\boldsymbol{\zeta}} \mathbf{E}(J(\boldsymbol{\zeta}, \mathbf{u}^n, \theta)) \approx \frac{1}{N_r} \sum_{i=1}^{N_r} J(\boldsymbol{\zeta}, \mathbf{u}^n, \theta_i), \tag{2a}$$

subject to:

$$C_{wl}(\boldsymbol{\zeta}), C_{rb}(\boldsymbol{\zeta}, \theta_i), C_{wo}(\boldsymbol{\zeta}), C_{wd}(\boldsymbol{\zeta}) \leq 0, \quad i = 0, 1, \cdots, N_r \tag{2b}$$

$$\mathbf{x}^0(\theta_i) = \mathbf{x}_0(\theta_i), \qquad i = 0, 1, \cdots, N_r \tag{2c}$$

$$\boldsymbol{g}^n(\mathbf{x}^{n+1}, \mathbf{x}^n, \boldsymbol{\zeta}, \mathbf{u}^n, \theta_i) = \mathbf{0}, \qquad i = 0, 1, \cdots, N_r, \quad n = 0, 1, \cdots, N, \tag{2d}$$

where the expectation $\mathbf{E}(J(\cdot))$ is approximated with averaging, and N_r is the number of model realizations. Usually, the Net Present Value (NPV) is used as the objective function, J, and is defined as:

$$J(\boldsymbol{\zeta}, \mathbf{u}^n, \theta_i) = \sum_{n=0}^{N-1} L^n(\mathbf{x}^{n+1}, \boldsymbol{\zeta}, \mathbf{u}^n, \theta_i), \tag{3}$$

$$\begin{aligned} L^n(\mathbf{x}^{n+1}, \boldsymbol{\zeta}, \mathbf{u}^n, \theta_i) = & \frac{\Delta t_n}{(1+\alpha)^{t^n}} \Big(\sum_{j=1}^{N_p} \big(r_o q_o^{j,n+1}(\boldsymbol{\zeta}, \mathbf{u}^n, \theta_i) + r_g q_g^{j,n+1}(\boldsymbol{\zeta}, \mathbf{u}^n, \theta_i) \big) \\ & - \sum_{j=1}^{N_p} r_{wp} q_{wp}^{j,n+1}(\boldsymbol{\zeta}, \mathbf{u}^n, \theta_i) - \sum_{j=1}^{N_i} r_{wi} q_{wi}^{j,n+1}(\boldsymbol{\zeta}, \mathbf{u}^n, \theta_i) \Big), \end{aligned}$$

where, for a waterflooding example, $q_o^{j,n+1}$, $q_g^{j,n+1}$, $q_{wp}^{j,n+1}$, and $q_{wi}^{j,n+1}$ are the flow rates of the oil, gas, water produced and water injected for well j at the output interval $n+1$, respectively, and Δt_n represents the length of each of the N time steps. Here, the number of injectors and producers are denoted by N_i and N_p, respectively; and the oil price, gas price, and the cost of water produced and injected are denoted by r_o, r_g, r_{wp}, and r_{wi}, respectively. The discount factor is represented by α.

In this work, we consider the placement of mono-bore wells (see e.g., [51] for multi-bore well placement optimization). There are different types of trajectories for well locations: vertical, horizontal, and deviated wells. Optimization of vertical wells has been widely studied in the past [20, 34, 38, 52]. In optimizing the location of vertical wells, it is common to parameterize the well locations using x and y coordinates of each cell that intersects a well as two optimization variables, assuming the wells are perforated in all of the layers. The location of deviated or horizontal well bores, however, is parametrized in different ways in the literature. In [16], the wells are parametrized such that they can be perforated only in predefined directions of i, j, k (i.e., along the Cartesian axis). The directions of the wells are determined in advance, and they are not included as optimization variables. Therefore, the location and extent of the wells are defined by the center point of the well trajectory (three variables) and length of each well. In [46], the trajectory of the wells is parametrized using multiple trajectory points. Thus, the number of decision variables is three times the number

of trajectory points. The authors also implement a smoothing procedure that keeps the curvature of the trajectory less than a maximum value. In [14], the Cartesian coordinates of the starting and end points of the deviated wells are used as decision variables. In [15], the trajectory of each well is parametrized in terms of six optimization variables in which the first three variables define the Cartesian coordinate of the center point of the well trajectory and the rest of the variables represent the length of the well, orientation of the well trajectory in the horizontal direction and the inclination angle of the well trajectory with respect to the vertical direction. Similarly in [51], the main bore is parametrized by six optimization variables; three variables define the Cartesian coordinate of the heel, while three variables define the length of the trajectory projected onto the $x-y$ plane, the orientation of the well in $x-y$ plane, and the depth to the trajectory end point.

In this work we use the Cartesian coordinates of the wells as optimization variables to avoid increasing the complexity of the overall constraint-handling procedure. Therefore, six optimization variables per well are used to define the toe and heel locations. For example, even though we could obtain a linear formulation of the well length constraint by using cylindrical or spherical coordinates, this would introduce highly nonlinear functions to describe the associated reservoir boundary constraint, thus increasing the overall complexity of the problem (see [23] for further details).

3 SPSA

The SPSA algorithm is well-suited for optimization problems where noisy measurements of the objective function are available, and/or when direct calculation of the objective function gradients is costly or infeasible. The basic unconstrained SPSA algorithm updates the optimization variables through the standard recursive form [45]:

$$\boldsymbol{\zeta}_{k+1} = \boldsymbol{\zeta}_k - a_k \hat{\mathbf{g}}_k(\boldsymbol{\zeta}_k), \tag{4}$$

where $\boldsymbol{\zeta}_k$ is the vector of decision variables at the kth iteration, $\hat{\mathbf{g}}_k(\boldsymbol{\zeta}_k)$ is the simultaneous perturbation estimate of gradient $\mathbf{g}(\boldsymbol{\zeta}) = \partial f/\partial \boldsymbol{\zeta}$ (f is the objective function), and a_k is a nonnegative scalar gain coefficient.

A key characteristic of the SPSA algorithm is how gradients are approximated. A central finite difference method approximates the gradient by perturbing the elements of the decision variable vector one at a time and evaluating the objective function for each perturbation. Thus, $2n_\zeta$ (where n_ζ is the number of decision variables) function evaluations are required for approximating the gradients using a central finite difference scheme. However, the key idea behind SPSA is to perturb all vector elements simultaneously. The SPSA algorithm therefore only needs two function evaluations; one for a forward perturbation and another for a backward perturbation of the variable vector, as follows:

$$\hat{\mathbf{g}}_k(\boldsymbol{\zeta}_k) = \frac{f(\boldsymbol{\zeta}_k + c_k\boldsymbol{\Delta}_k) - f(\boldsymbol{\zeta}_k - c_k\boldsymbol{\Delta}_k)}{2c_k}[\Delta_{k1}^{-1}, \Delta_{k2}^{-1}, \cdots, \Delta_{kn_\zeta}^{-1}], \tag{5}$$

where $\boldsymbol{\Delta}_k$ is a random perturbation vector, and c_k is a nonnegative scalar. The conditions specified for the perturbation vector $\boldsymbol{\Delta}_k$ are as follows [45]: (1) the entries should have independent and symmetrical distribution, and (2) they should have finite inverse moment expectation $E[|\Delta_{ki}|^{-1}]$. A common and simple distribution that fulfills the above conditions is the symmetric Bernoulli (± 1) distribution. It is worth mentioning that symmetric, uniform and normal distributions do not have finite inverse moment expectation, and thus they cannot be used with SPSA.

Although the gradient approximation in (5) requires fewer function evaluations, it is important to note that due to the approximate nature of the computed gradients the number of iterations required for the algorithm to converge is expected to increase. It is shown in [42, 45] that in comparison with the finite difference method, SPSA can achieve the same level of statistical accuracy under reasonably general conditions on the parameters even though it uses only $1/n_\zeta$ times the number of function evaluations used in the finite difference method. The SPSA algorithm converges to a local optimum point $\boldsymbol{\zeta}^*$ under certain conditions on the SPSA parameters (gain sequences), smoothness of the objective function close to the optimum, and when the required properties of the perturbation distribution $\boldsymbol{\Delta}$ (discussed above) are fulfilled. The conditions for the gain sequences are as follows [42, 45]:

$$a_k > 0, c_k > 0, \quad a_k \to 0, c_k \to 0, \quad \sum_{k=0}^{\infty} a_k = \infty, \quad \sum_{k=1}^{\infty} a_k^2/c_k^2 < \infty.$$

In [43], useful guidelines to select the gain sequence parameters (perturbation size c_k and step length a_k) are provided. To satisfy the above conditions, a_k and c_k are defined as follows:

$$a_k = \frac{a}{(A+k+1)^\alpha}, \tag{6a}$$

$$c_k = \frac{c}{(k+1)^\gamma}, \tag{6b}$$

where the recommended practical values for α and γ are 0.602 and 0.101, respectively. Moreover, it is suggested in [43] to set A at or around 10% of the maximum number of expected/allowed iterations. Parameter a can be chosen such that $\frac{\hat{\mathbf{g}}_0(\boldsymbol{\zeta}_0)a}{(A+1)^\alpha}$ is approximately equal to the minimum desired changes to $\boldsymbol{\zeta}$ in the early iterations. Finally, a suggested rule-of-thumb is to set c at a level approximately equal to the standard deviation of the noise in computing the objective function.

If the elements of $\boldsymbol{\zeta}$ have very different magnitudes, [43] recommends to use a matrix scaling of the gain α_k. To construct such a scaling matrix, prior information on the relative magnitude of the decision variables is required. However, if this information is not available, a second type of SPSA is suggested, namely, the ASP. Spall [44] introduces ASP as a stochastic analogous of the Newton-Raphson method, where scaling information is derived from the inverse of the approximated Hessian matrix. ASP automatically scales the solution updates

($\boldsymbol{\zeta}$) whenever there is significant difference between the magnitudes of the elements of $\boldsymbol{\zeta}$.

To approximate $\boldsymbol{\zeta}$ and the Hessian matrix of $f(\boldsymbol{\zeta})$, the ASP algorithm uses two parallel recursions:

$$\boldsymbol{\zeta}_{k+1} = \boldsymbol{\zeta}_k - a_k \bar{\bar{H}}_k^{-1} \hat{\mathbf{g}}_k(\boldsymbol{\zeta}_k), \tag{7a}$$

$$\bar{\bar{H}}_k = M_k(\bar{H}_k), \tag{7b}$$

$$\bar{H}_k = \frac{k}{k+1}\bar{H}_{k-1} + \frac{1}{k+1}\hat{H}_k,, \tag{7c}$$

where the definitions of α_k, and $\hat{\mathbf{g}}_k$ are similar to the definitions of the scalar gain coefficient and the approximation of the gradient in the basic SPSA algorithm in (4), respectively. M_k is a mapping function to cope with possible nonpositive definiteness of $\bar{H}_k$. $\hat{H}$ is a per-iteration estimate of the Hessian, and can be estimated as

$$\hat{H}_k = \frac{1}{2}\{\frac{\partial \mathbf{G}_k}{2c_k}[\Delta_{k1}^{-1}, \Delta_{k2}^{-1}, \cdots \Delta_{kn_\zeta}^{-1}] + (\frac{\partial \mathbf{G}_k}{2c_k}[\Delta_{k1}^{-1}, \Delta_{k2}^{-1}, \cdots \Delta_{kn_\zeta}^{-1}])^T\}, \tag{8}$$

where c_k and $\boldsymbol{\Delta}_k$ have the same definitions as in the original SPSA algorithm, and $\partial \mathbf{G}_k$ is given by

$$\partial \mathbf{G}_k = \mathbf{G}_k^{(1)}(\boldsymbol{\zeta}_k + c_k\boldsymbol{\Delta}_k) - \mathbf{G}_k^{(1)}(\boldsymbol{\zeta}_k - c_k\boldsymbol{\Delta}_k), \tag{9}$$

where $\mathbf{G}_k^{(1)}$ is a one-sided approximation of the gradient. The use of a one-sided approximation is suggested in [44] to keep the total number of function evaluations low compared with the standard two-sided form. The one-sided approximation is as follows:

$$\mathbf{G}_k^{(1)}(\boldsymbol{\zeta}_k \pm c_k\boldsymbol{\Delta}_k) = \frac{f(\boldsymbol{\zeta}_k \pm c_k\boldsymbol{\Delta}_k + \tilde{c}_k\tilde{\boldsymbol{\Delta}}_k) - f(\boldsymbol{\zeta}_k \pm c_k\boldsymbol{\Delta}_k)}{\tilde{c}_k} \begin{bmatrix} \tilde{\Delta}_{k1}^{-1} \\ \tilde{\Delta}_{k2}^{-1} \\ \vdots \\ \tilde{\Delta}_{kn_\zeta}^{-1} \end{bmatrix}, \tag{10}$$

where $\tilde{\boldsymbol{\Delta}}_k = [\tilde{\Delta}_{k1}^{-1}, \tilde{\Delta}_{k2}^{-1}, \cdots, \tilde{\Delta}_{kn_x}^{-1}]$ is chosen independently of $\boldsymbol{\Delta}_k$, but both $\tilde{\boldsymbol{\Delta}_k}$ and $\tilde{c}_k$ satisfy conditions similar to those of $\boldsymbol{\Delta}_k$ and c_k. Two function evaluations, $f(\boldsymbol{\zeta}_k \pm c_k\boldsymbol{\Delta}_k)$, are performed in (5) for gradient approximation. Two additional function evaluations, $f(\boldsymbol{\zeta}_k \pm c_k\boldsymbol{\Delta}_k + \tilde{c}_k\tilde{\boldsymbol{\Delta}}_k)$, are needed for the Hessian approximation. Thus, only four function evaluations are required at each iteration to approximate both the gradient and Hessian matrix.

The remaining aspect of the ASP implementation is to define the mapping function M_k in (7b). Recall that M_k is used to map $\bar{H}_k$ to a positive-definite matrix. To compute $\bar{\bar{H}}_k = M_k(\bar{H}_k)$, Zhu and Spall [53] propose the following steps:

- Perform eigenvalue decomposition of the Hessian matrix $\bar{H}_k$, and sort the eigenvalues in descending order:

$$\bar{H}_k = P_k \Lambda_k P_k^T, \tag{11}$$

$$\Lambda_k = \text{diag}\begin{bmatrix}\lambda_1 & \lambda_2 & \cdots & \lambda_{q-1} & \lambda_q & \lambda_{q+1} & \cdots & \lambda_{n_\zeta}\end{bmatrix} \tag{12}$$

  where P_k is the matrix containing the sorted eigenvectors in its columns and Λ_k is the diagonal matrix with the corresponding eigenvalues, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_q > 0$ and $\lambda_{n_\zeta} \leq \cdots \lambda_{q+1} \leq 0$.

- Replace the negative eigenvalues as follows:

$$\hat{\lambda}_q = \delta\lambda_{q-1},\ \hat{\lambda}_{q-1} = \delta\hat{\lambda}_q,\ \cdots,\ \hat{\lambda}_{n_\zeta} = \delta\hat{\lambda}_{n_\zeta - 1}, \tag{13}$$

  where δ is a small value in the range of $(0, 1)$. Thus, the new eigenvalue matrix $\hat{\Lambda}_k$ has the following form:

$$\hat{\Lambda}_k = \text{diag}\begin{bmatrix}\lambda_1 & \lambda_2 & \cdots & \lambda_{q-1} & \hat{\lambda}_q & \hat{\lambda}_{q+1} & \cdots & \hat{\lambda}_{n_\zeta}\end{bmatrix} \tag{14}$$

- The mapping M_k can then be expressed as follows:

$$M_k(\bar{H}_k) = P_k \hat{\Lambda}_k P_k^T \tag{15}$$

  The inversion of $\bar{\bar{H}}_k$ in (7c) can be computed easily using its eigenvalue decomposition:

$$(\bar{\bar{H}}_k)^{-1} = P_k \hat{\Lambda}_k^{-1} P_k^T \tag{16}$$

The SPSA and ASP algorithms for performing n iterations are summarized in Algorithm 1. In our implementation, in addition to the basic SPSA algorithm, we also add a blocking step and a line search method. In the blocking step, the updated solution $\boldsymbol{\zeta}_{k+1}$ is blocked, if the objective function at the intended value $\boldsymbol{\zeta}_{k+1}$ does not show improvement relative to $\boldsymbol{\zeta}_k$ [45]. We also suggest implementing a line search method to improve the objective function before blocking the step. However, only a few iterations of the line search method are performed, as the gradient (and Hessian matrix) are approximated and the descent direction might not have sufficient accuracy.

4 Solution approach

4.1 Well placement optimization

The performance of a candidate solution for the well placement problem is typically evaluated over multiple realizations (N_r in (1)). To reduce computation

Algorithm 1: SPSA [ASP] algorithm including a line search method.

Select $a, A, c, \alpha, \gamma, \rho, n_l$;
Set iteration number $k = 0$;
Initialize the optimization variable $\boldsymbol{\zeta}_0$;
while $k \leq n$ **do**
  Update a_k and c_k using (6); Generate $\boldsymbol{\Delta}_k$ [and also $\tilde{\boldsymbol{\Delta}}_k$];
  Calculate the gradient using (5); [Calculate and modify the Hessian using (7b), (7c), and (8)-(10)];
  Calculate $\boldsymbol{\zeta}_{k+1}$ using (4) [(7a)];
  Set $l = 0$; **while** $f(\boldsymbol{\zeta}_{k+1}) \geq f(\boldsymbol{\zeta}_k)$ ***and*** $l \leq n_l$ **do**
    Calculate $\boldsymbol{\zeta}_{k+1}$ using (4) [(7a)];
    $l = l + 1$;
    $a_k = \rho a_k$;
  end
  if $f(\boldsymbol{\zeta}_{k+1}) \geq f(\boldsymbol{\zeta}_k)$ **then**
    $\boldsymbol{\zeta}_{k+1} = \boldsymbol{\zeta}_k$;
  end
  $k = k + 1$;
end

load, we propose approximating the expected performance (J_T in (1)) using only a small subset of randomly selected model realizations as follows:

$$J_T(\boldsymbol{\zeta}) = \hat{J}_T(\boldsymbol{\zeta}) + \epsilon, \tag{17}$$

$$\hat{J}_T = \frac{1}{N_s} \sum_{i=1}^{N_s} J(\boldsymbol{\zeta}, \mathbf{u}^n, R(m_i)), \tag{18}$$

where N_s is the size of the subset ($N_s \ll N_r$), R is the set of randomly selected realizations of uncertain parameters of the reservoir model (θ in (2)), while $R(m_i)$ is the ith member of R, and ϵ denotes the approximation error. It is important to note that the original number of realizations N_r approximates the expected value of the objective function, and by using $N_s < N_r$, we increase the approximation level. In fact, if the original N_r realizations are sampled from the probability distribution of the uncertain reservoir parameters to approximate the cost function, it is reasonable to expect that another set of N_s could also be used for the same purpose. That is, it is not necessary to approximate the expected value of the cost function using the same set of realizations (although, in practice, for convenience, the same set of realizations are used). The idea behind reduced random sampling is to use fewer samples to compute a noisier version of the expected cost function, but randomly select those samples at each iteration. In the extreme case where $N_s = 1$, this approach is equivalent to using a randomized nominal profit (cost function). If the random sampling strategy is performed on a fixed set of realizations, after a certain number of iterations, all realizations are expected to have been visited by the random selection. Since

the SPSA algorithm uses approximate gradients, it is robust against moderate levels of noise in the objective function [45]. The performance of the SPSA algorithm when optimizing for a noisy objective function is investigated in our case studies.

For a well placement optimization procedure, where the expected performance is computed using all of the realizations, a simple convergence criterion is sufficient to assess the performance. In this case, the optimization procedure terminates if the performance improvement is less than a user defined value. However, in the case where the performance is approximated using (18), a new convergence criterion should be defined. In this paper, we propose to stop the SPSA algorithm when the relative standard deviation of the objective function improvement over a given number of iterations l_s falls below a user defined value d_s. For example, we terminate the algorithm if the relative standard deviation of the NPV progression for 20 optimization iterations is less than 10%.

The well placement problem formulation (1) is a constrained optimization problem. However, since SPSA is an unconstrained optimization algorithm, we need to incorporate a constraint-handling procedure.

In this work, we implement a projection method to deal with the given well placement constraints. For more details about different constraint handling techniques for well placement problems see [23] and [22]. In [37], a projection algorithm is proposed to handle inequality constraints given as explicit functions of the parameters. The recursive form (4) is replaced by

$$\boldsymbol{\zeta}_{k+1} = P(\boldsymbol{\zeta}_k - a_k \hat{\mathbf{g}}_k(\boldsymbol{\zeta}_k)), \tag{19}$$

where P is the projection map that projects the candidate solution $\boldsymbol{\zeta}_{k+1}$ to the nearest feasible point. It is shown in [37] that the solution converges almost surely to a Kuhn-Tucker point.

The SPSA implementation for the well placement problem under uncertainty is summarized in Algorithm 2. Note that the ASP algorithm for the well placement problem can be implemented by introducing the Hessian approximation as shown in Algorithm 1. Since the subset of realizations is randomly chosen at each iteration, we need to re-evaluate the current solution for each new subset. Otherwise, the blocking step is not valid, as the current and updated solutions are evaluated differently. The implementation of the projection method is shown in Algorithm 2.

As additional comparison, we also implement a discrete version of the SPSA algorithm that is proposed in [29]. In this type of SPSA, the step size of the optimization line search is limited to a single grid block. At each iteration, a random neighbor of the grid block is chosen. The NPV is evaluated for locating a vertical well in this grid block as well as in the grid block in the opposite direction. The well is then moved to the grid block with the higher NPV.

It is important to note that the random selection may result in sampling bias as some realizations may not be selected in a finite number of iterations, while these realizations may contain significant features. Therefore, the solution may not incorporate the effect of some realizations (due to randomness). One way

Algorithm 2: The SPSA algorithm for the well placement problem under uncertainty using the projection algorithm.

Select $a, A, c, \alpha, \gamma, \rho, n_l, l_s, d_s$;
Set iteration number $k = 0$;
Initialize the optimization variable $\boldsymbol{\zeta}_0$; Set subset size N_s;
while $k < l_s$ *or* $d_k \geq d_s$ **do**
  Select N_s random realizations; Evaluate $\hat{J}_T(\boldsymbol{\zeta}_k)$; Update a_k and c_k using (6); Generate $\boldsymbol{\Delta}_k$; Evaluate $\hat{J}_T(\boldsymbol{\zeta}_k \pm c_k\boldsymbol{\Delta}_k)$ using (18); Approximate the gradient $\hat{\mathbf{g}}_k$ using (5) where $f(\boldsymbol{\zeta}_k \pm c_k\boldsymbol{\Delta}_k) = \hat{J}_T(\boldsymbol{\zeta}_k \pm c_k\boldsymbol{\Delta}_k)$; Calculate $\boldsymbol{\zeta}_{k+1}$ using (19); Set $l = 0$; **while** $\hat{J}_T(\boldsymbol{\zeta}_{k+1}) \geq \hat{J}_T(\boldsymbol{\zeta}_k)$ ***and*** $l \leq n_l$ **do**
    Calculate $\boldsymbol{\zeta}_{k+1}$ using (4); $l = l + 1$;
    $a_k = \rho a_k$;
  end
  if $\hat{J}_T(\boldsymbol{\zeta}_{k+1}) \geq \hat{J}_T(\boldsymbol{\zeta}_k)$ **then**
    $\boldsymbol{\zeta}_{k+1} = \boldsymbol{\zeta}_k$; $\hat{J}_T(\boldsymbol{\zeta}_{k+1}) = \hat{J}_T(\boldsymbol{\zeta}_k)$;
  end
  if $k \geq l_s$ **then**
    Calculate absolute value of the relative standard deviation of $[\hat{J}_T(\boldsymbol{\zeta}_{k+1}), \hat{J}_T(\boldsymbol{\zeta}_k), \cdots, \hat{J}_T(\boldsymbol{\zeta}_{k-l_s+2})]$, and store it in d_{k+1};
  end
  $k = k + 1$;
end

to avoid sampling bias is by clustering the realizations based on a predefined measure and ensure that at least one realization is selected from each cluster at every iteration. There are different measures and algorithms for clustering geologic model realizations based on their flow response predictions. In [2], a ranking technique is suggested to cluster realizations based on the P10, P50, and P90 quantiles of the responses of interest. Ranking can be based on a statistical measure defined for parameter realizations or the performance criterion (e.g., NPV). The ranking based on NPV may not correctly capture the performance measure of the realizations for all possible well configuration scenarios. The choice of a statistical measure that may be used for ranking should be considered carefully as some measures (e.g., oil in place) may have a poor correlation with NPV or any other performance metric [39]. To find the similarity between two realizations, a dissimilarity distance is defined in [39] based on streamline simulation, which requires less computational time compared to full multi-phase flow simulations. A kernel k-mean clustering approach is then used to cluster the realizations based on the dissimilarity distance. In [41], a weighted combination of flow-based quantities and geological features of the realizations has been used to represent the realizations and the k-mean algorithm is applied for clustering.

5 Case Studies

We present several numerical examples to evaluate the performance of the reduced random sampling strategy and to compare different variants of the SPSA algorithm and the PSO algorithm using projection constraint handling technique.

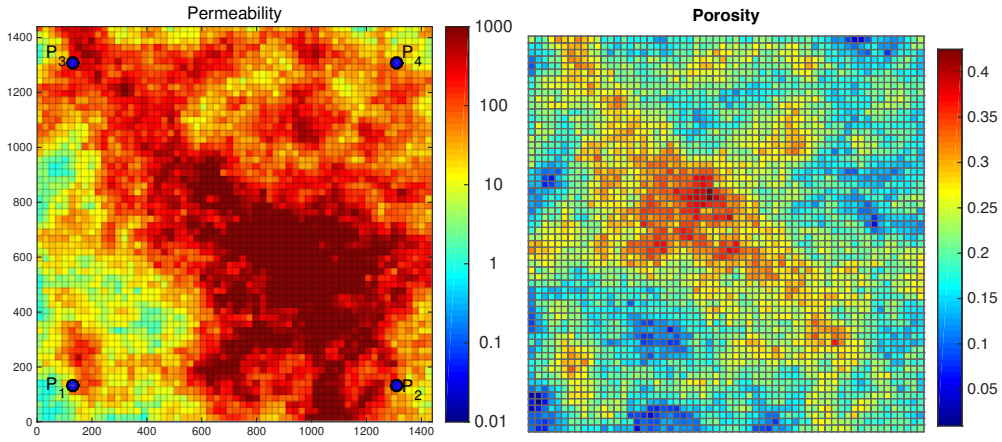
While PSO is a global method with differenct properties and SPSA is a local search technique, our motivation for the comparisons is mainly regarding the computational efficiency. PSO has shown good performance for well placement optimization. However, given its global nature, it requires hundreds of simulation runs. Although SPSA is a local search approach in which gradients are approximated, a reduced step-size approach can result in a large number of iterations. Therefore, a comparison with PSO can provide a relative measure of computational demand of the proposed SPSA implementation. Moreover, the well placement objective function is generally known to include several local solutions with similar values. Therefore, it is possible for local methods to find solutions that have objective function values that are close to the global solution.

The reservoir models used in our example include Model A, a deterministic 2-dimensional two-phase (oil-water) reservoir model; Model B, a 2-dimensional two-phase model with uncertain permeability and porosity distributions; and Model C, a 3-dimensional synthetic field model with realistic grid geometry, porosity and permeability data.

Model A is a reservoir discretized into a 60×60 2-dimensional grid. The permeability and porosity fields for this case is a cut-off of layer 21 of the SPE 10 model [11]. Figure 1a depicts the resulting reservoir, which has a high-permeability region that extends diagonally from the top left to bottom right corner. The porosity distribution is shown in Figure 1b. Initial pressure and saturation distributions are calculated by establishing hydrostatic equilibrium. The datum depth is equal to 1700m and the pressure at the datum depth is set to 170barsa. The depth of the water-oil contact is 2200m. The simulation model involves a two-phase oil-water flow (without including the capillary pressure effects), running for 8 years. The remaining simulation parameters are summarized in Table 1. The oil and water relative permeability curves are presented in Figure 2.

The parameters of the SPSA algorithm are initialized based on the recommendation in [43] with α,γ, and A specified as 0.602, 0.101, and 100, respectively. The line search coefficient ρ in Algorithm 2 is set to 2. We iterate on the line search loop three times ($n_l = 3$). The rest of the parameters are initialized differently in each case study.

There are extensive discussions about the tuning of the parameters of the PSO algorithm in the literature. In this work, we follow the recommendations in [9, 36], which are based on various trials for different benchmark problems (For more details see [22, 23]). The PSO algorithm is implemented using the Generic Optimization Program (GenOpt), a software package developed by the Lawrence Berkeley National Laboratory [50].



(a) (b)

Figure 1: (a) The permeability field of Model A (logarithm of permeability is displayed). Production wells are represented as blue circles. (b) The porosity field of Model A.

Table 1: Simulation parameters of Models A and B.

Grid block size	24m×24m×24m
Reference pressure (p_i)	273barsa
Water formation volume factor at p_i	1.03 rm^3/sm^3
Water viscosity at p_i	0.31cP
Water density	1037 kg/m^3
Oil formation volume factor at p_i	1.74rm^3/sm^3
Oil viscosity at p_i	0.8098cP
Oil density	786 kg/m^3
Rock compressibility at 1 barsa	4.4×10^{-5} $(barsa)^{-1}$

Table 2: Economic parameters.

Oil price	r_o	$315\,\$/Sm^3$
gas price	r_g	$3\,\$/Mscf$
Water injection cost	r_{wi}	$31.5\,\$/Sm^3$
Water production cost	r_{wp}	$31.5\,\$/Sm^3$
Discount factor	α	10%

The economic parameters for the NPV are given in Table 2. The cost functions for Models A and B are computed by running reservoir simulations using the MATLAB Reservoir Simulation Toolbox (MRST), see [31]. The Eclipse reservoir simulation software is used for calculating the cost function in Model C.

Note that the numerical experiments are performed without including parallelization. This will, however, be commented towards the end of the paper. The first example considers the Rosenbrock function, while the remaining examples involve case studies related to the well placement optimization problem. The Rosenbrock function is included as an initial testbench to provide easily accessible comparisons between the three algorithms with different noise levels.

5.1 Case study 1: Rosenbrock function

In this case study, we optimize a 20-dimensional Rosenbrock function of the form:

$$f(x) = \sum_{i=1}^{5} 100(x(2i) - x(2i-1)^2)^2 + (1 - x(2i-1))^2. \tag{20}$$

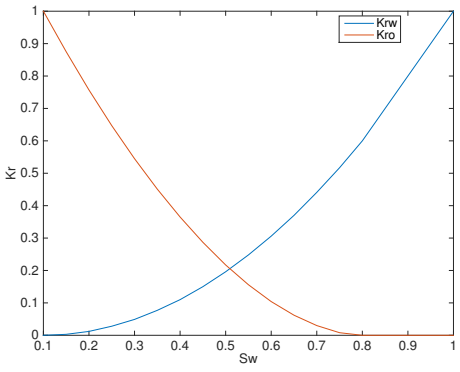


Figure 2: Relative permeability curves for the oil and water phases for Models A and B.

The Rosenbrock function has exactly one minimum inside a long, narrow, parabolic shaped flat valley at $\mathbf{x} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \end{bmatrix}$. Two different starting points are considered, the first one is close to the optimum ($\mathbf{x_{0,c}} = [0.9 \quad 0.9 \quad 0.9 \quad 1 \quad \cdots \quad 1]$) and the other one is fairly far away from the minimum ($\mathbf{x_{0,f}} = [10 \quad 0.9 \quad 0.9 \quad 1 \quad \cdots \quad 1]$). Moreover, we also optimize a noisy version of the Rosenbrock function to illustrate the robustness of the SPSA algorithm to noisy objective function calculations. A normal distributed noise with zero mean and standard deviation equal to 5% of the magnitude of the mean function value is added to the Rosenbrock function.

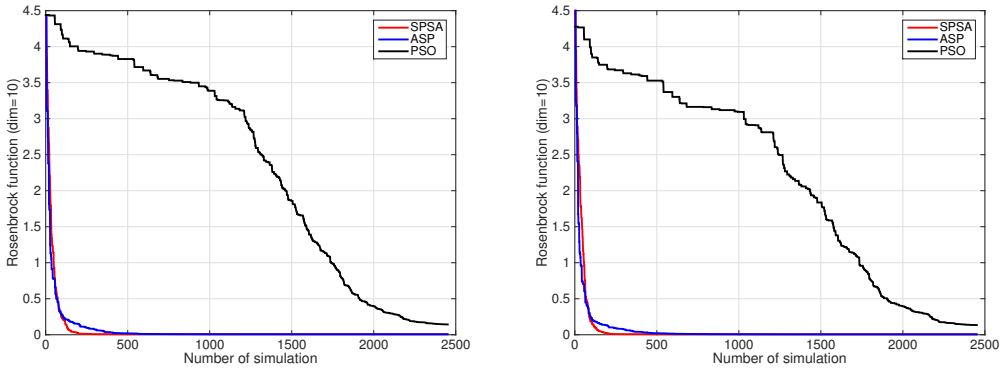
Given the stochastic nature of the algorithms, to draw reliable conclusions, for each case we perform 50 optimization experiments. Figures 3 and 4 show the progress of the objective function versus number of function evaluations from the starting point $\mathbf{x_{0,c}}$ and $\mathbf{x_{0,f}}$, respectively. In Figure 3, when the starting point is chosen close to the optimum, the progression of the SPSA and ASP algorithms is faster than the PSO algorithm. However, when the starting point is chosen relatively far from the optimum (Figure 4), the mean curve corresponding to the PSO algorithm exhibits a faster rate of convergence in the beginning of the optimization. However, at later iterations PSO takes many more iterations to pin down the exact solution. This behavior is consistent with the general properties of global and local search methods; that is, global methods are more efficient in approaching the solution from distant point while they tend to be slow in converging to the exact solution from nearby points. To mitigate the slow progression of the PSO near the optimum, hybrid approaches that use the PSO algorithm initially (during the exploration stage) and switch to a local search algorithm (exploitation stage) near the solution have been suggested in the literature [19, 30, 32] (for more details, see [4]).

One observation from Figure 4 is that ASP converges faster than SPSA, mainly because the elements of the solution have very different magnitudes and ASP automatically scales them. In Figure 3, ASP converges more slowly than SPSA. In this experiment, the optimization variables have the same magnitude and there is no need for a scaling matrix. However, ASP requires two-times the number of simulation runs at each iteration to approximate the Hessian matrix.

Overall, we observe that the SPSA, ASP, and PSO algorithms in Figures 3 and 4 have similar performance in the noise-free compared to the noisy test cases. Interestingly, in Figure (b), SPSA converges faster in the presence of noise.

Note that in this specific problem for which we know the global solution and there are no local solutions in the vicinity of the global solution, initializing the optimization closer to the global solution increases the chance of converging to the global solution. In this example, the iterations are not converging to local solutions. Therefore, the statements that we make about this specific example are indeed correct. In general, starting the optimization solution in the neighborhood of the global solution will not guarantee convergence to the global solution.

In Figure 5, the performance of the SPSA and ASP is shown when the level of noise is increased. When the level of noise is increased to 10% of the magnitude



(a) Noise-free cost function.

(b) Cost function with added noise.

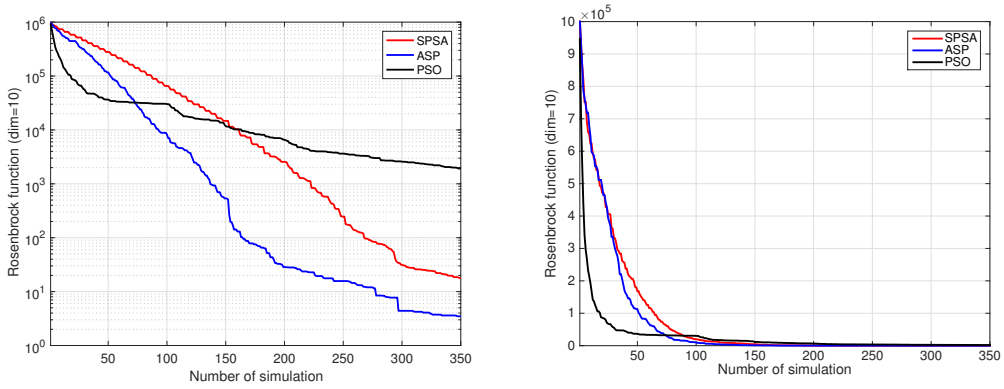
Figure 3: Minimization of the Rosenbrock function (20) starting close to the optimum ($\mathbf{x_{0,c}}$).

of the objective function value, ASP converges with the same rate as in the case with a noise increase of 5%, while SPSA converges to the optimum more slowly. In the case of increasing the noise to 20%, both the SPSA and ASP algorithms do not converge to the optimum, while the solution is still close to the optimum.

5.2 Case study 2: Placement of a vertical well

Before presenting the results for the robust optimization, the convergence behavior of PSO amd SPSA in a deterministic optimization problem that does not involve geologic uncrtainty is explored. This case study considers well placement optimization for Model A with the well configuration consisting of five wells. Four of these wells are vertical producers with fixed locations close to the corners of the reservoir (these wells are shown as blue circles in Figure 1a). The producers in this case are set to maintain a constant bottom-hole pressure (BHP) of 90 barsa with an upper limit liquid rate of 2000 Sm^3/day, while the injector is controlled with a constant BHP of 230 barsa without any rate limit. The fifth well is a vertical injector. In this example, we determine the optimal location of a single vertical injector. The number of optimization variables is equal to 2 (that is, the x and y coordinates of the well location). In the PSO implementation, we use 16 particles and 100 generations. In the SPSA application, we set $a = 20$, $c = 10$, and $n_l = 3$. In this section, only a summary of observations is presented and details are deferred to Appendix A1.

In the following, a summary of our main observations for Case study 2 is presented (see Appendix A1 for details). The step length in the continuous SPSA is user-defined and gradually decreases through iterations, while the step length in the discrete SPSA is set to a single grid block. Therefore, we observe that the continuous SPSA outperforms the discrete SPSA algorithm. As noted be-



(a) Noise-free cost function. (b) Cost function with added noise.

Figure 4: Minimization of the Rosenbrock function (20) starting far from the optimum ($\mathbf{x_{0,f}}$).

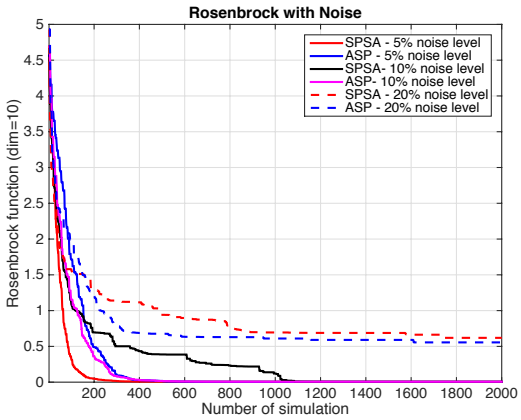


Figure 5: Minimization of the Rosenbrock function (20) starting close to the optimum ($\mathbf{x_{0,c}}$) for different levels of noise.

fore, the SPSA algorithm is a local search method. Therefore, to explore the search space globally, when using SPSA, it is recommended to either start the optimization from different initial guesses or implement a hybrid approach. In this case study, ASP provides slightly better performance than the other types of SPSA algorithms. This means that the Hessian information helps to improve the approximation of the descent direction. Although, ASP requires two more function evaluations per iteration compared to the SPSA algorithm. Finally, the ASP method converges faster in terms of simulation runs than the PSO algorithm, though parallelization can be used to improve the computational time of the PSO algorithm.

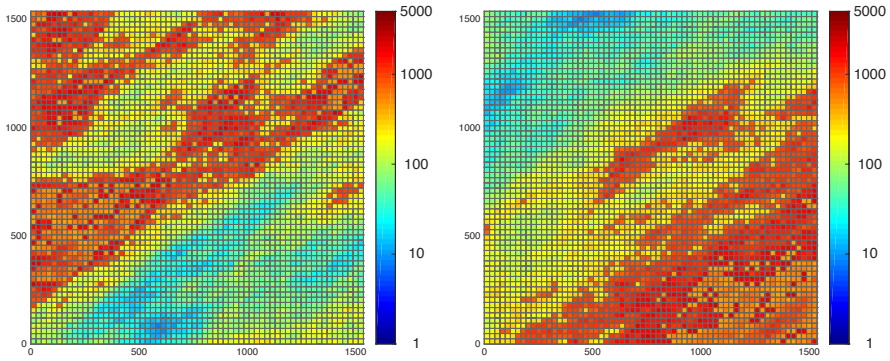
5.3 Case study 3: Placement of a horizontal well under uncertainty

This case study considers well placement optimization of Model B. Model B is a reservoir discretized into a 64×64 2-dimensional grid with $N_r = 100$ permeability and porosity models that represent the uncertainty in the geological model. The porosity distributions are selected to be proportional to the log-permeability distributions. Figure 6 shows four sample log-permeability models that exhibit distinct spatial variability. The initial water saturation and pressure are equal to 0.1 and 170 barsa, respectively. The production time frame is 18 years. The other simulation parameters are equal to those used in Model A in Table 1.

The uncertainty in the model is represented with 100 realizations. The well configuration consists of five wells. Four of these wells are vertical injectors with fixed locations close to each corner of the reservoir. The fifth well is a horizontal producer with variable heel/toe location. Injectors I_1, I_2, I_3, and I_4 are set to have a constant rate of $100, 200, 1300$, and $600\ m^3/day$, respectively. An upper limit of 400 barsa is also defined for the BHP in the injectors. The producer is controlled using reactive control with a constant liquid rate of $2200\ m^3/day$, a lower limit BHP of 90barsa and a maximum watercut limit of 80%. The latter constraint means that the producer is shut-in if its watercut exceeds the limit.

Both well length and inter-well distance constraints are implemented in this case. The minimum and maximum well lengths are set to 200m and 600m, respectively, while the minimum inter-well distance is set to 200m. The projection method is implemented to handle the constraints. The projection method for the well length constraints works by moving the toe and heel apart an equal distance along the direction of the well if the well is shorter than the minimum length. In the case where the well is longer than the maximum well length, the toe and heel are moved closer an equal distance along the direction of the well [for more details see 33].

In this example, we optimize the NPV by moving a single horizontal producer. The number of optimization variables is equal to 4 (there are two pairs of two variables to locate the toe and heel of the injector). We implement the proposed Reduced Random Sampling Strategy (RRSS) in the SPSA and PSO algorithms, using different sizes of subset N_s. Moreover, the effectiveness of the



(a) Sample 1.

(b) Sample 2.

(c) Sample 3.

(d) Sample 4.

Figure 6: Four sample log-permeability realizations out of 100 realizations of Model B.

decoder and the projection constraint-handling methods are compared for both the PSO and SPSA algorithms.

In the implementation of the PSO algorithm, we use 25 particles and 50 generations. When the SPSA algorithm is used with the projection method, we set $a = 1467$, $c = 36$, $n_l = 3$, $A = 100$. When the objective function is noisy, the convergence parameters in Algorithm 2 are set to $l_s = 10$ and $d_s = 0.5\%$. However, in the case where all realizations are used to evaluate a given well configuration in the SPSA algorithm, the objective function is assumed to have no noise. We reiterate that 100 or even more realizations are not sufficient to capture the range of variability in model parameters. However, for our discussion, we assume that the case with 100 realization represents a noise-free reference case. We stop the SPSA algorithm if the improvement of the output for a window of 20 iterations is less than 1%.

Table 3 compares the results of the PSO and SPSA algorithms. The table shows two different strategies for selecting the subset of realization at each iteration, fixed realizations and randomly selected samples. In the reference case, all realizations are used to evaluate the NPV objective function for a given well configuration. In the RRSS, 5 and 16 realizations are selected randomly at each iteration. We note that the results are averaged over 5 optimization experiments, each with a different initialization. Therefore, we present the averaged number of simulation runs and iteration numbers.

For the reference case, the averaged expected NPV over 100 realizations obtained by the PSO algorithm is approximately 2% greater than the result of the SPSA algorithm. However, the SPSA algorithm converges to the optimum after approximately 900 simulation runs per realizations, which is 37% less than the number of simulation runs per realization required by the PSO method. We also observe from Table 3 that the standard deviation in all optimization experiments is relatively small, which indicates that our results should be repeatable.

In Table 3, we note that, the required number of simulation runs for the SPSA algorithm with the RRSS and $N_s = 5$ random realizations is 33 times less than the number of simulation runs used when applying the SPSA algorithm with all 100 realizations. The SPSA algorithm with the RRSS converges to an optimum with the corresponding $E(NPV)$ slightly higher than that obtained for the SPSA with 100 realizations (note that in all cases the attained $E(NPV)$ values are based on 100 realizations). For example, the RRSS with $N_s = 16$ requires 17 times less function evaluations than the required number of simulation runs for the corresponding reference case. Comparison between the performance of the SPSA using the RRSS and the reference case is indicative of the robustness of the SPSA method. Using the RRSS yields significant computational saving, while the optimum solution remains almost similar.

Figure 7 shows the progression of the objective function for the SPSA algorithm using the RRSS. Moreover, the progression of the corresponding averaged NPV over 100 realizations is illustrated in the same figure. Note that for the RRSS, the NPV evaluation for all the realizations is not required. However, we calculate the $E(NPV)$ for illustration purposes. In Figure 7, we note that the fluctuations of the objective functions decrease as the iterations proceed

Table 3: Comparison between the PSO and SPSA algorithms using the projection constraint-handling method and two different realization selections over 5 optimization experiments for Case study 3. $E(NPV)$ is the averaged of the NPV over 100 realizations. $\sigma(E(NPV))$ is the standard deviation of the obtained objective function over 5 optimization experiments.

Algorithm	Selection method	# Rel.	#Sim.	#Itr.	$E(NPV)$ $(\times 10^8)$	$\sigma(E(NPV))$ $(\times 10^7)$
PSO	Reference	100	125000	50	7.71	0.13
PSO	RRSS	5	6250	50	7.64	0.27
SPSA	Reference	100	90700	235	7.55	0.86
SPSA	RRSS	5	2680	118	7.59	0.31
SPSA	RRSS	16	5056	71	7.40	1.31
SPSA	Fixed	16	2080	39	7.01	6.91

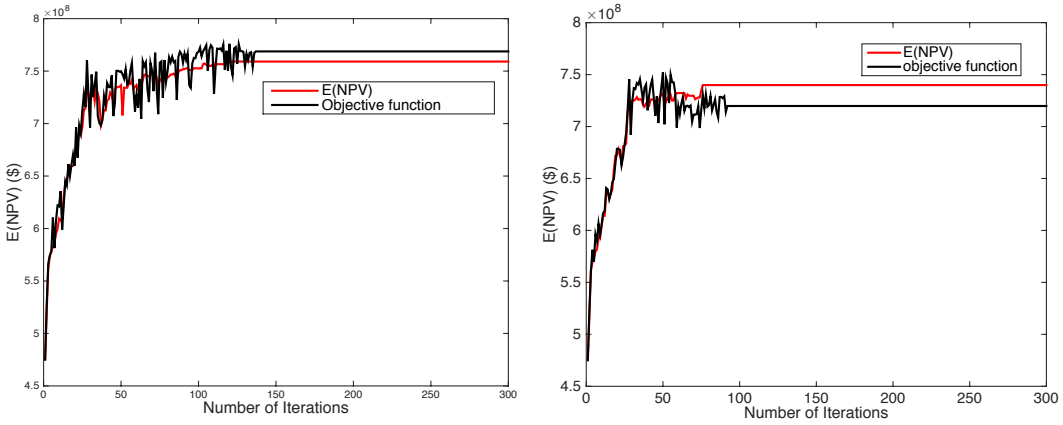
until the stopping criteria is satisfied. We also note from the same figure that $E(NPV)$ may converge to different values than the objective function (e.g., Figures 7a-7b). The reason is that the objective function is defined as the mean of NPV over a set of random realizations.

Implementation of the RRSS in the PSO algorithm gives similar results as in the SPSA algorithm. With the subset size of $N_s = 5$, the number of required simulation runs reduced 20 times, while the obtained averaged NPV over 100 realizations is decreased less than 1%.

In the last rows of Table 3, 16 model realizations are selected as representative of the parameter uncertainty by using the k-mean clustering method. Therefore, the subset of 16 model realizations is fixed through the whole of the optimization procedure. Comparing the results with the RRSS shows that using the fixed subset leads to early convergence.

Remark 1: As for a rule of thumb for sample size selection, the main difficulty is that the minimum sample size is generally problem dependent. One approach to determine the adequacy of the sample size is based on the magnitude of forecast errors. For instance, one can consider a number of alternative well configurations and study the approximate quality of the forecasts for various sample sizes to determine an appropriate sample size. In addition, from Table 3, we observe that different sample sizes may result in different required iterations for convergence, suggesting that the number of iterations may be adjusted to account for the selected sample size.

Field oil production, field water production, and field water injection rates for the initial and optimum well configurations are shown in Figure 8. On average, the reservoir with the initial well configuration stops operation after 30 time steps, while the reservoir with the optimized well configuration continues



(a) $N_s = 5$ using the projection method.

(b) $N_s = 16$ using the projection method.

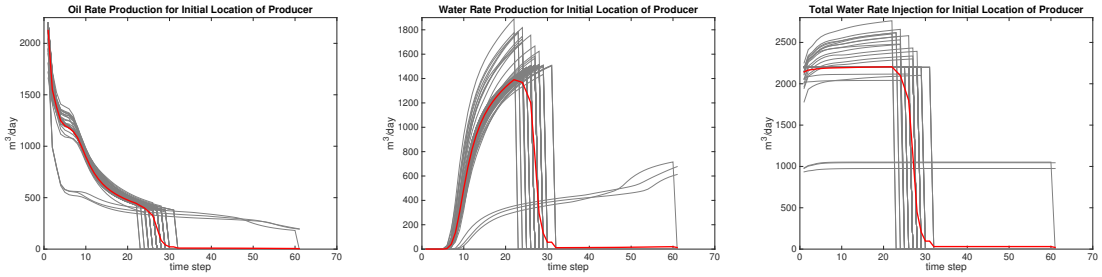
Figure 7: The progression of the objective function of the SPSA method using the RRSS (black lines) and the corresponding $E(NPV)$ over 100 realizations (red lines) for both the decoder and projection constraint-handling techniques.

Table 4: Comparison of field fluid productions and water injections for the initial and optimum well configurations in case study 3.

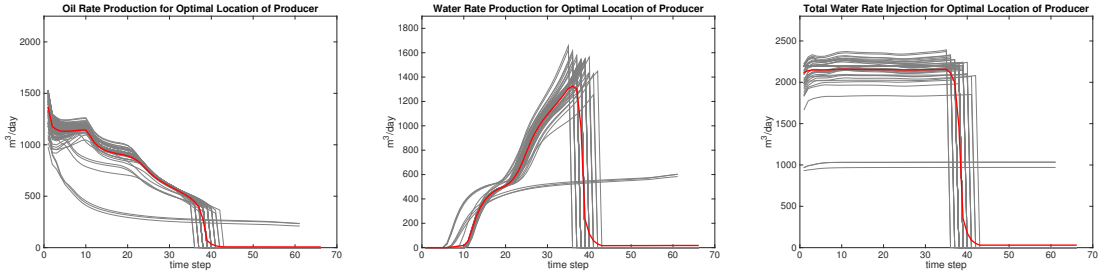
	FOPT (m^3)	FWPT (m^3)	FWIT (m^3)	E(NPV) ($)
INT	2.3972e+06	2.3587e+06	6.4730e+06	4.7514e+08
OPT	3.5669e+06	2.4436e+06	9.012500e+6	7.6038e+08

production until 40 time steps. In Table 4, the total fluid productions and water injections for the initial and optimum well configurations are compared. The total oil production for the optimized case is approximately 1.5 times the cumulative production for the initial case, while the total water production is in the same range in both cases. With the initial well configuration water injection from the southwest and northeast wells leads to early water breakthroughs. After optimization the water breakthrough and water production (from the southwest and the northeast injectors) are reduced. Moreover, after optimizaiton the producer is placed closer to the northwest region to improve the sweep in that area.

Figure 9 compares the location of the well through iterations of the RRSS and the reference using SPSA. We can observe from this figure that the RRSS converge to the same region as the reference after some iterations.



(a) Field oil rate production. (b) Field water rate production. (c) Field water rate injection.



(d) Field oil rate production. (e) Field water rate production. (f) Field water rate injection.

Figure 8: Comparison between field rates of the initial and optimum well configurations for Case study 3. The first and second rows show field rates of the initial and the optimum well configuration, respectively. Rates of the realizations are shown by gray, and the mean curves are illustrated by red.

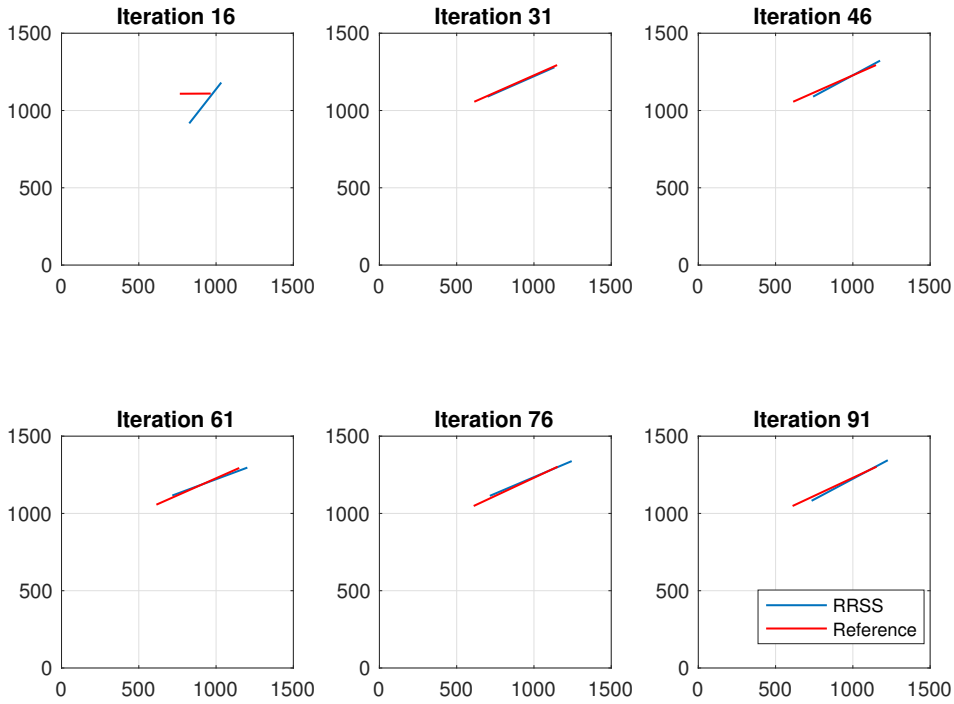


Figure 9: Comparison of well locations through iterations in Case study 3 when using the SPSA with the reference and RRSS. The blue shows the RRSS solutions and the red lines illustrates the reference candidate solution.

5.4 Case study 4: Placement of a lateral well under uncertainty

This case study considers well placement optimization for Model C.

Model C is a three phase synthetic reservoir model in which grid geometry, and permeability and porosity distributions are taken from a North Sea field. The permeability and porosity distributions are modeled as uncertain parameters. The uncertainty in Model C is represented by 100 realizations of these properties. Figures 10 and 11 show the permeability and porosity distributions for one sample. The model has 14 layers, each containing 40×64 grid blocks of which 27755 are active. The height of the cells range from 2 to 3m and the horizontal dimension of the grids is between 100m and 200m. Capillary pressure effects are also neglected in this model. The relative permeability curves for this model are shown in Figure 12. Figure 13 presents the gas formation volume factor and the gas viscosity as functions of the gas pressure. In this figure, the vaporized gas-oil ratios for saturated gas are equal to zero. Figure 14 presents the oil formation volume factor and the oil viscosity as functions of the bubble point pressure for different dissolved gas-oil ratios. The rest of the simulation parameters are represented in Table 5. The depth of the water-oil contact and gas-oil contact are 1705m and 500m, respectively. To calculate the initial saturation and pressure using the equilibration facility in the Eclipse reservoir simulator, the pressure at datum depth of 2469m is set to 382barsa. The initial oil saturation of the reservoir is shown in Figure 15. There is no free gas initially.

The location of 7 vertical wells, 4 producers and 3 injectors, are fixed, while the optimization problem is to find the optimal location of a single lateral pro-

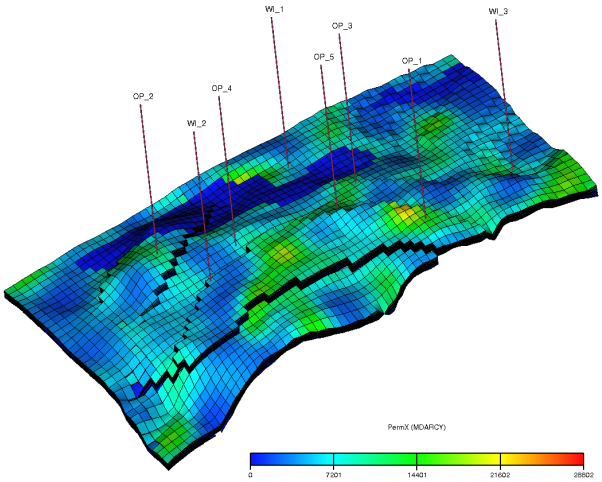


Figure 10: The permeability field of a sample of Model C with seven vertical wells and one lateral producer well (OP-1).

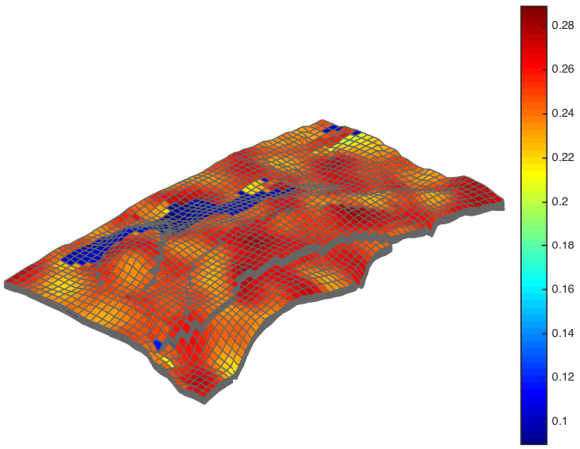


Figure 11: The porosity distribution of a sample of Model C.

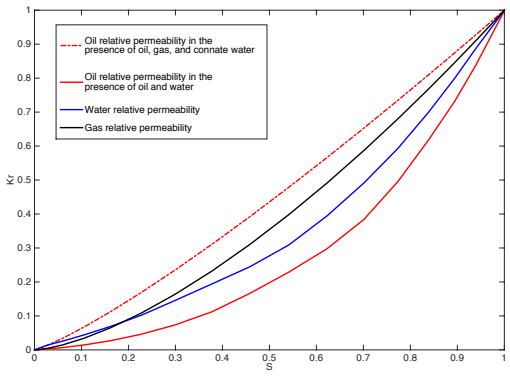
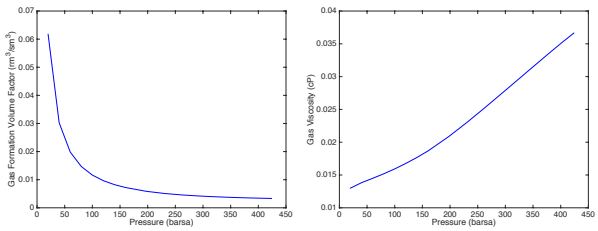


Figure 12: Relative permeability curves of Model C. The red curves are relative permeability of oil versus oil saturation. Relative permeability of gas versus gas saturation is illustrated by black. Relative permeability of water versus water saturation is shown by blue.

Table 5: Simulation parameters for Model C.

Reference pressure (p_i)	344.83barsa
Water formation volume factor at p_i	$1.0292\,rm^3/sm^3$
Water viscosity at p_i	0.36cP
Water density	1001.1 kg/m^3
Oil density	842.3 kg/m^3
Gas density	0.9 kg/m^3
Rock compressibility at 383 barsa	4.12×10^{-5} $(barsa)^{-1}$



(a) (b)

Figure 13: Model C: (a) The gas formation volume factor, (b) the gas viscosity.

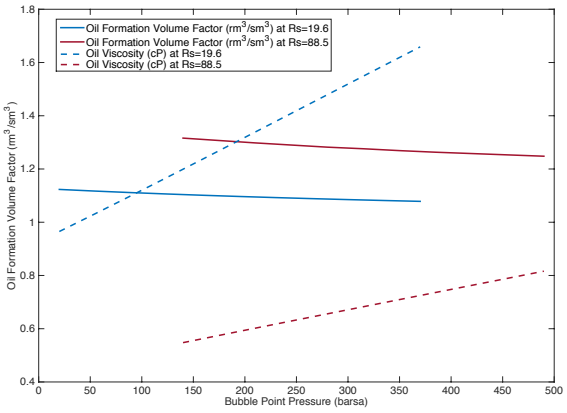


Figure 14: Model C: The oil formation volume factor (solid lines) and the oil viscosity (dashed lines) as function of the bubble point pressure for two different gas-oil ratios R_s.

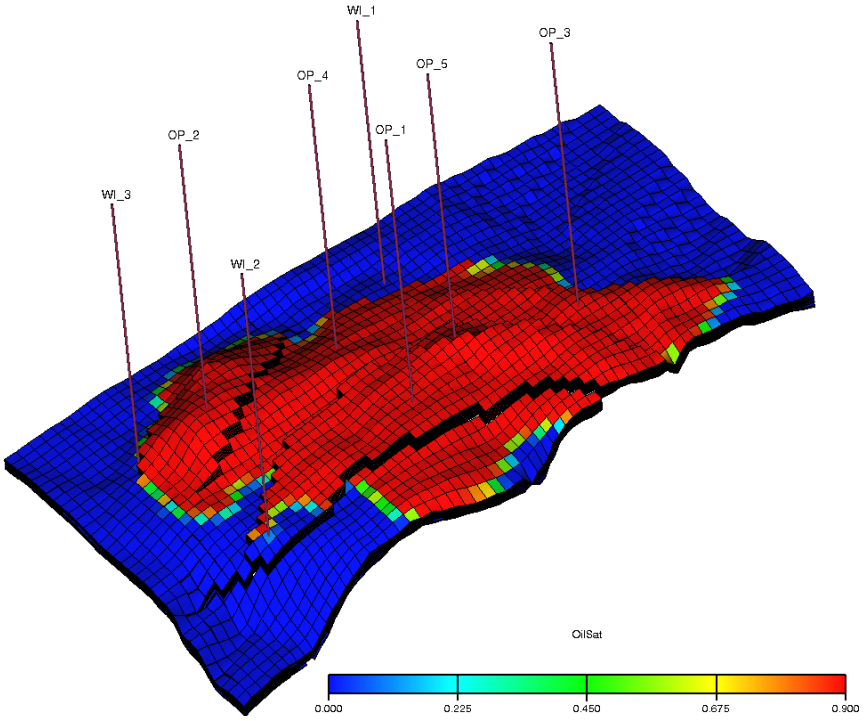


Figure 15: The initial oil saturation of Model C. The producers are located in the oil zone and the injectors are located on the border between the oil and water zones.

Table 6: Comparison between the PSO using 100 realizations and SPSA algorithms using two different realization selections over 5 optimization experiments for Case study 4. $E(NPV)$ is the averaged of the NPV over 100 realizations. $\sigma(E(NPV))$ is the standard deviation of the obtained objective function over 4 optimization experiments.

Algorithm	Selection method	# Rel.	#Sim.	#Itr.	$E(NPV)$ $(\times 10^9)$	$\sigma(E(NPV))$ $(\times 10^7)$
PSO	Reference	100	198000	55	5.86	0.5
SPSA	Reference	100	22100	59	5.76	10.7
SPSA	RRSS	5	1895	89	5.77	11.0

ducer (a lateral well instead of the vertical producer OP_1 in Figure 15). Therefore, the number of optimization variables is equal to 6 (there are two pairs of three variables to locate the toe and heel of the injector). As shown in Figure 15, the layers of the reservoir are undulated. This means each grid block in a given layer has a different height. Therefore, for a specific planar coordinate a maximum and minimum height is defined and vertical coordinates of any candidate are mapped into this range. For more details about this map, see [23].

The producer is placed within the region defined by closest faults to the initial guess. Moreover, we also include a well length constraint with a minimum well length of 500m and a maximum of 2000m. In applying the PSO algorithm, we use 36 particles and 55 generations. In the implementation of the SPSA algorithm, we set $a = 6$, $c = 0.2$, $n_l = 3$. The stopping criteria are chosen using the same approach to those in Case study 3. Note that results are averaged over 4 optimization experiments.

Table 6 compares the results of the PSO and SPSA algorithms using RRSS and the reference case. For RRSS, the size of the subset is set to 5. For the reference case, the averaged expected NPV over 100 realizations obtained by the PSO algorithm is slightly better than the results of the SPSA algorithm. However, PSO requires 8 times more objective function evaluations compared to the required simulation runs for the SPSA algorithm. In Table 6, we note that the required number of simulation runs for the SPSA algorithm with the RRSS is 11 times less than the number of simulation runs when applying the SPSA algorithm with the reference case. Moreover, the SPSA algorithm with the RRSS converges to an optimum with the corresponding expected NPV slightly higher than that obtained for the SPSA with the reference case.

Figure 16 shows the optimal well configuration using different optimization algorithms and different sampling strategies. The location of producer OP-1 converged to the same region for both algorithms. However, there is no guarantee that the solution of the approximate method would be the same as the solution with large number of sample. Note that because the region corresponding to the producer OP-1 varies in the vertical direction, in Figure 16 part of producer OP-1 appears to be located outside of the region.

Remark 2: Note that the results may not hold, if other higher order statistics of NPV are used. In general, sample mean converges to the population mean much faster (with increasing sample sizes). However, it takes many more samples to have a good approximation of variance or standard deviation.

Remark 3: Regarding convergence analysis, we have indicated that the requirements set forward in the original SPSA algorithm are not likely to be met in complex problems where the objective functions involve complex nonlinear mapping. Therefore, regardless of sample size, the convergence of the SPSA cannot be established. Hence, the main issues is whether a reduced (random) sample size will lead to significant performance reduction. In this context, the method should be viewed as an approximate approach to reduce the computational complexity of stochastic optimization. While the convergence analysis is not applicable to our problem, the robustness of SPSA and the non-exact form of the calculated gradients offer an opportunity to exploit the algorithm and develop computationally efficient (yet heuristic) methods for robust optimization. As in other approximate methods, the solution should not be expected to be the same. However, our results show that the solutions obtained from the proposed approach are quite promising.

Remark 4: In order to show the improvement related to randomizing the sampling, we consider a simple 2-D reservoir model with 4 producers on the corners and a single injector in the center. The reservoir has 21×21 cells. The NPV for all possible injector locations is calculated for 1000 different realizations

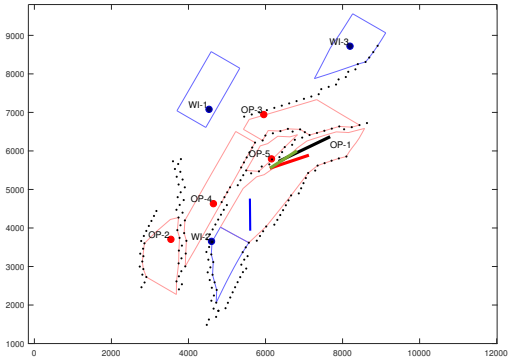


Figure 16: Top view of the reservoir of Case study 4. The producers and injectors are illustrated by red and blue circles, respectively. The initial guess of the location of producer OP-1 is illustrated by blue line. The optimized locations of producer OP-1 using the PSO and SPSA algorithms with the reference case are illustrated by red and black lines, respectively. The optimized locations of producer OP-1 using the SPSA algorithms with the RRSS method is illustrated by green line.

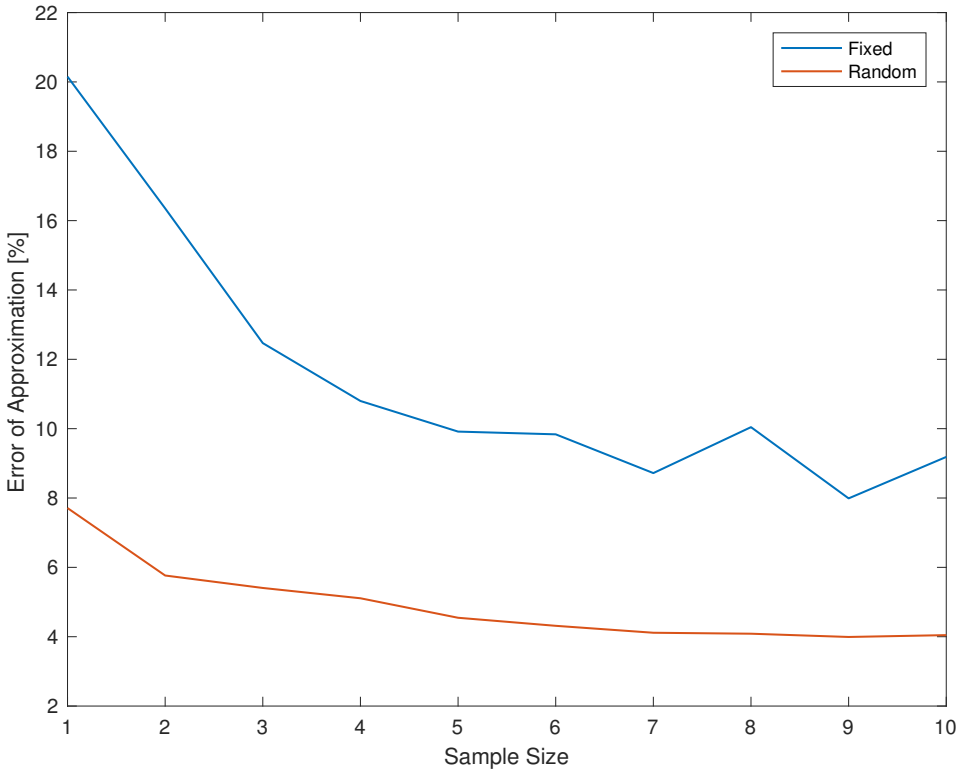


Figure 17: Mean of absolute value of error over all possible well configuration as a function of sample size. The blue and red lines show the errors of mean of NPV approximation using fixed and random sampling.

of permeability distribution. We cluster the realizations using k-mean clustering for different sample size. Then, for the same sample size, we compute the mean of NPV once with fixed samples (using k-mean clustering of permeability distributions) and once with random samples. The average relative error of the expected NPV over all possible well configurations is compared using fixed and random samples. Note that the results shown in Figure 17 are the average over 10 experiments for the random sampling strategy. The random sampling has a lower error of approximation than the fixed sample strategy does. Moreover, one can see that the error is decreased by increasing the sample size up to a certain size, where the reduction becomes less noticeable.

6 Discussion and Conclusion

In this paper, we proposed a reduced random sampling strategy for improving the computational complexity of robust well placement optimization under geologic uncertainty. In the proposed method, at each iteration of the robust optimization with the SPSA algorithm, a small subset of model realizations are random selected from the full set to approximate the expected value of the objective function. This strategy leads to significant reduction in computation while introducing approximation in computing the objective function. The random selection strategy is used to ensure that, after a certain number of iterations, the full set of model realizations (i.e., geologic variability) is incorporated in the optimization algorithm. Our experience suggests that when SPSA is adopted all the realizations are used multiple times throughout the iterations. Our results also show that the use of reduced sampling with the proposed strategy tends to have minimal impact on the performance of the optimization problem. An important property of the SPSA algorithm that makes it a suitable candidate for this approach is its robustness against noise in the objective function.

We examined the effectiveness of the proposed strategy by implementing two variants of the SPSA algorithm and comparing them with the PSO algorithm. The motivation for comparison with PSO was primarily to provide a relative measure of computational demand to state-of-the-art methods. to check the computational Several numerical examples were used to evaluate the performance of the proposed approach, including the Rosenbrock function, a single well placement problem for which the global solution was known through an exhaustive search, as well as a field case study. The results from these examples show that the proposed random selection method can lead to significant computational saving while providing solutions (expected NPV values) that remain almost the same as in the case where all realizations are used. Comparing the performance of different variants of the SPSA method with the PSO algorithm indicates that the required number of simulation runs for the continuous and adaptive variants of the SPSA algorithm are significantly less than the required number of simulation runs for the PSO method. This is primarily because PSO is a global optimization algorithm that is known to be computationally very demanding whereas the SPSA algorithm is a local search method in which the performance is highly dependent on the initial solution. The robustness of the SPSA algorithm against noise in the objective function is an important property that can also contribute to the efficiency of SPSA in the reduced random sampling approach. Furthermore, we also observed that when the elements of the solution have very different magnitude the rescaling effect of the ASP algorithm results in a faster convergence rate than the standard SPSA algorithm.

The well placement examples that are presented in this paper consider an infill drilling scenario for vertical and horizontal wells. If multiple wells are used, the dimension of the decision variable will increase proportionally. Another important aspect is that the well controls are fixed in our example. Control variables can significanlty increase the size of decision variables, especially when time-depenent controls in life-cycle optimization are used. Our ongoing work is

focused on implementing the proposed approach in high-dimensional problems that involve well controls and locations. It is also important to note that while some of the realistic aspects can complicate the formulation of the original optimization problem and result in more complex objective functions, a main consideration in the proposed formulation (and in general robust optimization) is the complexity of the geologic model as it controls the number of samples that must be used.

The use of SPSA was motivated by the robustness of this method to the noise in the objective function, which is the case when very few samples are used. The premise of a reduced sampling strategy, however, is that the number of iterations should be large enough so that on average each sample is used a few (2-3) times in computing the objective function. In [**?**], the authors observed that a reduced sampling strategy did not produce good results in their implementation. However, the implementation details, including the algorithm used, number of iterations compared to the sample size used, geologic complexity as it relates to sample size, and use of approximate covariances in computing the gradient in EnOpt, can affect the performance of the method, and the outcomes reported in their study.

The results in this paper suggest that effective reduced sampling strategies can reduce the computational burden of robust field development optimization when geologic uncertainty is incorporated through several realizations of model parameters. In the examples of this paper, the objective function was the expected value of NPV over model realizations, without including additional statistical information (e.g., variance) in the objective function. In general, the accuracy of the (sample) mean of a distribution is known to be less affected by reduction in the number of realizations. Additional studies are needed to explore the performance of the reduced sampling method when the objective function includes additional terms representing higher-order statistical information. Furthermore, additional investigation is needed to examine the performance of the developed approach in real applications where more complex wells, including multiple horizontal/deviated wells, and various linear and nonlinear constraints are involved.

Appendix 1: Results from Case Study 2

The search space is the square defined by the four producers. We perform an exhaustive search of all grid blocks inside this square to obtain the global maximum of the NPV function. Figure A1 shows the resulting NPV surface for placing the single injector. The surface has low NPV values that extend from north-west to south-east, corresponding to the high-perm zone in that region. There are two zones with high NPV values in Figure A1; one to the south and the other to the east of the reservoir. The highest NPV is equal to 5.77×10^8USD which corresponds to the injector being located in grid block (12,32) in the eastern region of the model. In the southern region we have a local optimum located in grid block (33,17) with NPV equal to 5.09×10^8USD.

Table A1: Case study 2: Comparison between the required number of iterations (Itr.) and simulation runs (Sim.) for convergence of the different optimization algorithm.

Algorithm	Starting point 1		Starting point 2		Starting point 3		Starting point 4	
	Itr.	Sim.	Itr.	Sim.	Itr.	Sim.	Itr.	Sim.
SPSA (Continuous)	52	260	160	800	-	-	167	835
SPSA (Discrete)	130	390	422	1266	-	-	207	621
ASP	25	175	78	546	109	763	99	693
PSO	96	1536	96	1536	97	1552	109	1744

Since the performance of local methods is dependent on the initial guess, we start the optimization problem from several initial locations to have a valid comparison between the SPSA/ASP and PSO algorithms. The different starting points are depicted by black dots in Figure A1 . We perform 50 optimization experiments for each initial guess and optimization algorithms.

Figure A2 shows the progress of the mean NPV as a function of the number of simulations over 50 optimization runs. Table A1 presents the average number of iterations and simulation runs required by different algorithms.

In Figure A2a, the search methods are initialized using starting point 1. Figure A3 compares the location of the well through iterations of different methods.In this case, the PSO algorithm and all types of the SPSA method converge to the global optimum. The progression of the PSO algorithm is faster in the beginning than those of the SPSA methods. However, the progression of the PSO

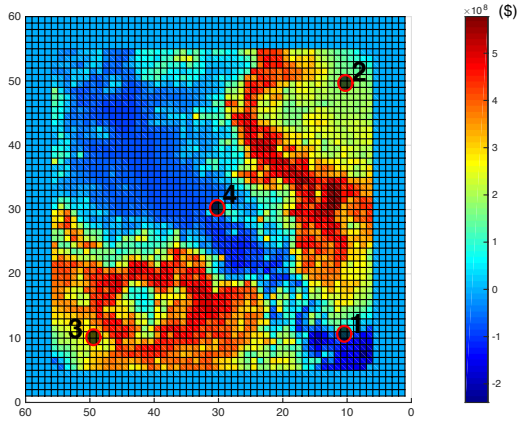
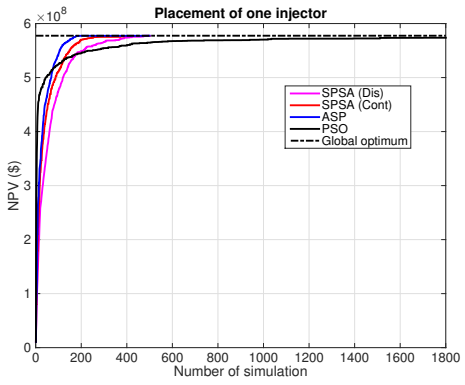
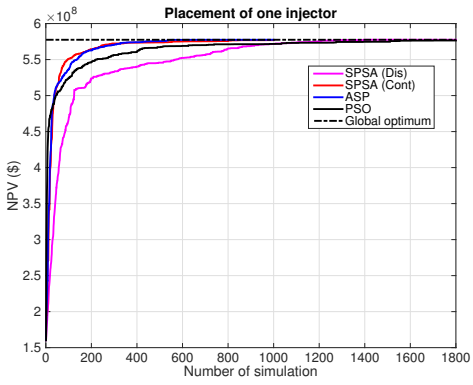


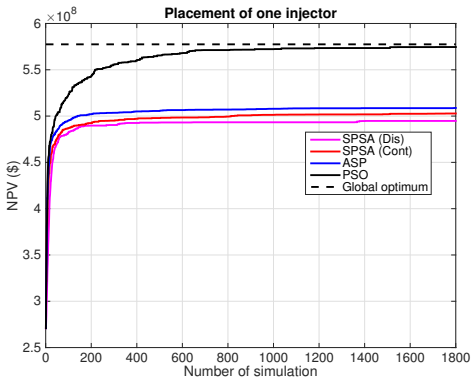
Figure A1: Case study 2: Exhaustive search result of the NPV. Different starting points for the optimization procedure are shown as black circles.



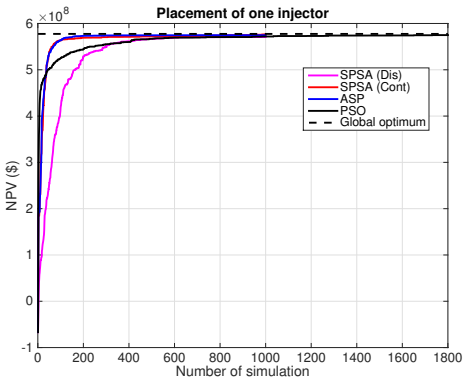
(a) Starting point 1.



(b) Starting point 2.



(c) Starting point 3.



(d) Starting point 4.

Figure A2: Case 2: Comparison of mean NPV over 50 optimization runs between discrete SPSA, continuous SPSA, ASP, and PSO algorithms starting from different initial guess location.

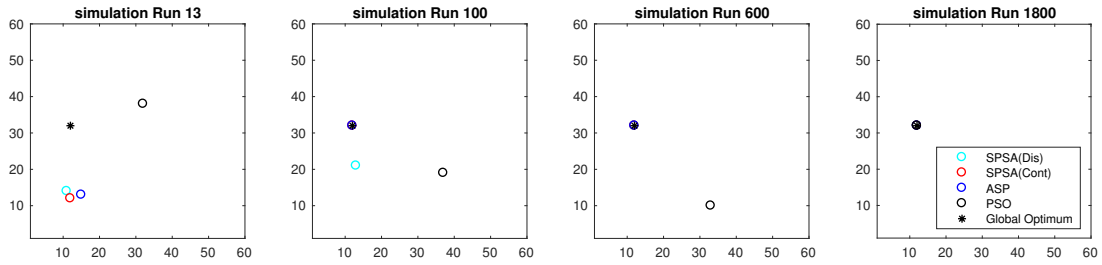


Figure A3: Comparison of well locations through iterations in Case study 2 with initial guess at Starting point 1.

slows down when closer to the optimum solution. It is also observed from Table A1 that SPSA algorithms require fewer function evaluations compared with the PSO algorithm. This can be explained by the different nature of the algorithms. The PSO is a population-based method requiring many function evaluations to update the positions of the particles towards the optimum, while SPSA requires only two objective functions to approximate the descent direction. Among the different variants of the SPSA algorithm, the ASP algorithm requires the fewest number of iterations and simulation runs. We also notice that the discrete version of the SPSA algorithm converges at a slower rate than the continuous SPSA and ASP, because it moves one grid block at each iteration.

We can observe from Figure A3 that after 13 simulation runs candidate solutions of different SPSA algorithms are located in the same area, while those of the PSO is in another area. After 100 simulation runs, the continuous SPSA and ASP converges to the global point, while the solution of the discrete SPSA is also closer to the global optimum than the solution of PSO. After 600 simulation runs, all types of SPSA algorithm have converged to the global point, while the PSO algorithm still searches. After 1800 simulation runs, PSO also converges. Note that these locations are all from one of the optimization run, while the NPV presented in A2 is the mean value over 50 optimization runs.

In Figures A2b, the optimization methods are initialized at starting point 2. All of the optimization methods converge successfully to the global optimum. The convergence rates of the ASP, and PSO are almost similar in terms of the number of iterations, see Table A1. However, the PSO algorithm requires 3 times more function evaluations than the ASP algorithm does. The continuous SPSA method requires around 2 times less simulation runs than the PSO algorithm does. The discrete SPSA algorithm requires approximately 1.5 times more function evaluations than is needed by the continuous SPSA algorithm.

We can observe from Figure A4 that after 13 simulation runs candidate solutions of continuous SPSA and ASP are closer to the global optimum than the solutions of discrete SPSA and PSO. After 100 simulation runs, ASP is only one grid block far from the optimum, the continuous SPSA is also close to the global point. The discrete SPSA and PSO still search the search space.

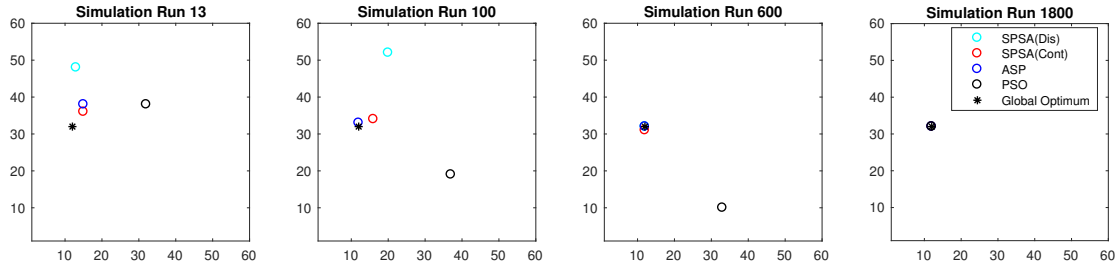


Figure A4: Comparison of well locations through iterations in Case study 2 with initial guess at Starting point 2.

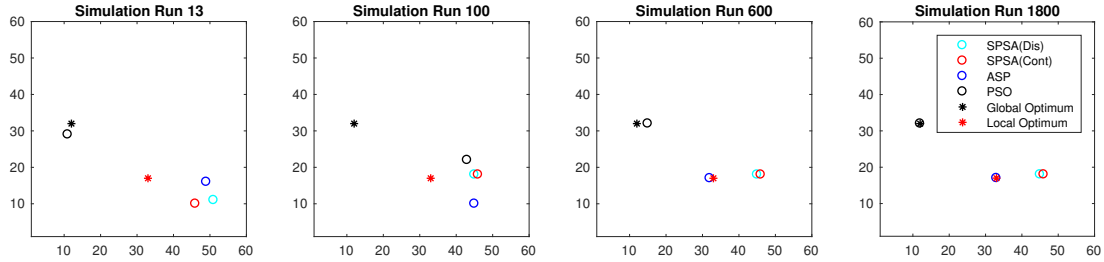


Figure A5: Comparison of well locations through iterations in Case study 2 with initial guess at Starting point 3.

After 600 simulation runs, all type of SPSA almost converge to the global point, while PSO still explore the search space. After 1800 simulation runs, PSO also converges.

In Figure A2c, we use starting point 3, located in the southern region. We can observe from Figure A5 that only PSO converges to the global optimum. The ASP algorithm converges to the local optimum, while the two other types of the SPSA algorithms do not converge to the local optimum in some of the experiments. The required number of simulation runs needed by ASP to converge to the local optimum is about half the number needed by PSO. We can observe from Figure A5 that after 13 simulation runs, PSO is close to the global solution, while all types of the SPSA are close to the local optimum. After 100 simulation runs, all algorithms explore close to the local optimum. After 600 simulation runs, the PSO algorithm is again close to the global optimum, and ASP is close to the local optimum, while the two types of SPSA algorithms are still exploring the area around the local optimum. Finally, after 1800 simulation runs the PSO algorithm converges to the global solution and ASP converges to the local solution.

In Figure A2d, we start the search procedure from initial point 4. In this case, all of the algorithms converge to the global optimum. The convergence

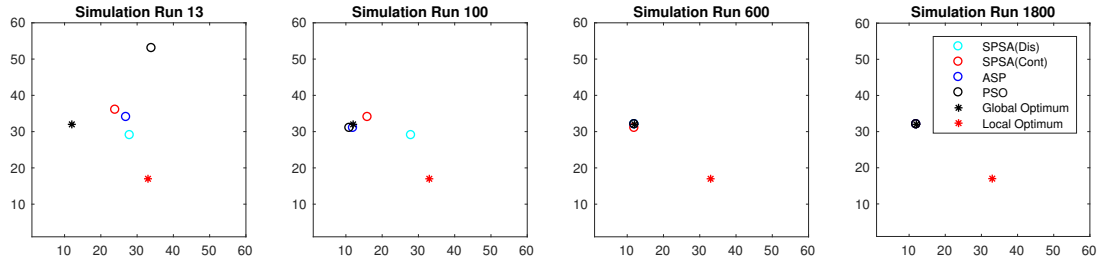


Figure A6: Comparison of well locations through iterations in Case study 2 with initial guess at Starting point 4.

rates of the SPSA-based methods are approximately two times faster than the progression of PSO in terms of the total number of simulation runs. We can observe from Figure A6 that after 13 simulation run, all types of SPSA algorithm are in a same area close to the initial guess while PSO explores the search space. After 100 simulation runs, PSO and ASP almost converge to the global optimum, while the continuous SPSA is closer to the global optimum than the discrete SPSA. After 600 simulation runs, all algorithms almost converge to the global optimum. At simulation run 1800, all of the algorithms find the global optimum.

Finally, it is important to note that all methods include tuning parameters that may affect the performance of the algorithms. In our examples, the tuning parameters are either set based on trial and error or using suggested values from the literature. The summary of our observations for Case study 2 are presented in the text.

References

[1] Vincent Artus, Louis J. Durlofsky, Jérôme E. Onwunalu, and Khalid Aziz. Optimization of nonconventional wells under uncertainty using statistical proxies. *Computational Geosciences*, 10(4):389–404, 2006. ISSN 1420-0597. doi: 10.1007/s10596-006-9031-9. URL `http://dx.doi.org/10.1007/s10596-006-9031-9`.

[2] P. R. Ballin, A. G. Journel, and K. Aziz. Prediction of uncertainty in reservoir performance forecast. *Journal of Canadian Petroleum Technology*, 31(4), 1992. doi: 10.2118/92-04-05.

[3] W. Bangerth, H. Klie, M. F. Wheeler, P. L. Stoffa, and M. K. Sen. On optimization algorithms for the reservoir oil well placement problem. *Computational Geosciences*, 10(3):303–319, 2004. doi: 10.1007/s10596-006-9025-7.

[4] Alec Banks, Jonathan Vincent, and Chukwudi Anyakoha. A review of particle swarm optimization. part ii: hybridisation, combinatorial, multicriteria

and constrained optimization, and indicative applications. *Natural Computing*, 7(1):109–124, 2007. ISSN 1572-9796. doi: 10.1007/s11047-007-9050-z. URL http://dx.doi.org/10.1007/s11047-007-9050-z.

[5] B. L. Beckner and X. Song. Field development planning using simulated annealing - optimal economic well scheduling and placement. In *SPE Annual Technical Conference and Exhibition*, Dallas, Texas, 1995. Society of Petroleum Engineers. ISBN 978-1-55563-441-4. doi: 10.2118/30650-MS.

[6] Mathias C. Bellout, David Echeverría Ciaurri, Louis J. Durlofsky, Bjarne Foss, and Jon Kleppe. Joint optimization of oil well placement and controls. *Computational Geosciences*, 16(4):1061–1079, 2012. ISSN 1420-0597. doi: 10.1007/s10596-012-9303-5. URL http://dx.doi.org/10.1007/s10596-012-9303-5.

[7] Zyed Bouzarkouna, Didier Yu Ding, and Anne Auger. Well placement optimization with the covariance matrix adaptation evolution strategy and meta-models. *Computational Geosciences*, 16(1):75–92, 2012. doi: 10.1007/s10596-011-9254-2.

[8] Andrea Capolei, Bjarne Foss, and John Bagterp Jørgensen. Profit and risk measures in oil production optimization. *IFAC-PapersOnLine*, 48(6): 214 – 220, 2015. ISSN 2405-8963. doi: http://dx.doi.org/10.1016/j.ifacol.2015.08.034. URL http://www.sciencedirect.com/science/article/pii/S2405896315009015.

[9] A. Carlisle and G. Dozier. An off-the-shelf pso. In *In Proceedings of the Workshop on Particle Swarm Optimization*, Indianapolis, IN, 2001. doi: 10.1.1.589.485.

[10] Y. Chen and D. S. Oliver. Ensemble-based closed-loop optimization applied to brugge field. *SPE Reservoir Evaluation & Engineering*, 13(1):56–71, 2010. doi: 10.2118/118926-PA.

[11] M.A. Christie and M.J.: Blunt. Tenth spe comparative solution project: a comparison of upscaling techniques. *SPE Reserv. Evalu. Eng*, 4, 2011.

[12] Sy T. Do and Albert C. Reynolds. Theoretical connections between optimization algorithms based on an approximate gradient. *Computational Geosciences*, 17(6):959–973, 2013. doi: 10.1007/s10596-013-9368-9.

[13] Sy T. Do, Fahim Forouzanfar, and Albert C. Reynolds. Estimation of optimal well controls using the augmented lagrangian function with approximate derivatives. In *IFAC Workshop on Automatic Control in Offshore Oil and Gas Production*, Trondheim, Norway, 2012. doi: 10.3182/20120531-2-NO-4020.00021.

[14] Alexandre Anoze Emerick, Eugenio Silva, Bruno Messer, Luciana Faletti Almeida, Dilza Szwarcman, Marco Aurelio Cavalcanti Pacheco, and Marley

Maria Bernardes Rebuzzi Vellasco. Well placement optimization using a genetic algorithm with nonlinear constraints. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 2009. ISBN 978-1-55563-209-0. doi: 10.2118/118808-MS.

[15] F. Forouzanfar, W. E. Poquioma, and A. C. Reynolds. A covariance matrix adaptation algorithm for simultaneous estimation of optimal placement and control of production and water injection wells. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 2015. doi: 10.2118/173256-MS.

[16] Fahim Forouzanfar, Albert C. Reynolds, and Gaoming Li. Optimization of the well locations and completions for vertical and horizontal wells using a derivative-free optimization algorithm. *Journal of Petroleum Science and Engineering*, 86–87:272–288, 5 2012. doi: http://dx.doi.org/10.1016/j.petrol.2012.03.014. URL `http://www.sciencedirect.com/science/article/pii/S0920410512000678`.

[17] G. Gao, G. Li, and A. C. Reynolds. A stochastic optimization algorithm for automatic history matching. In *SPE Annual Technical Conference and Exhibition*, Houston, Texas, September 2004. doi: 10.2118/90065-PA.

[18] Bari Güyagüler and Roland N. Horne. Uncertainty assessment of well-placement optimization. *SPE Reservoir Evaluation and Engineering*, 7(1): 24–32, 2004. doi: 10.2118/87663-PA.

[19] J Habibi, SA Zonouz, and M Saneei. A hybrid ps-based optimization algorithm for solving traveling salesman problem. In *IEEE symposium on frontiers in networking with applications (FINA 2006)*, Vienna, Austria, April 2006.

[20] Obiajulu J. Isebor, Louis J. Durlofsky, and David Echeverría Ciaurri. A derivative-free methodology with local and global search for the constrained joint optimization of well locations and controls. *Computational Geosciences*, 18(3-4):463–482, 2014. doi: 10.1007/s10596-013-9383-x.

[21] J.D. Jansen. Adjoint-based optimization of multi-phase flow through porous media – a review. *Computers & Fluids*, 46(1):40 – 51, 2011. ISSN 0045-7930. doi: http://dx.doi.org/10.1016/j.compfluid.2010.09.039. URL `http://www.sciencedirect.com/science/article/pii/S0045793010002677`. 10th {ICFD} Conference Series on Numerical Methods for Fluid Dynamics (ICFD 2010).

[22] Mansoureh Jesmani, Mathias C. Bellout, Remus Hanea, and Bjarne Foss. Particle swarm optimization algorithm for optimum well placement subject to realistic field development constraints. In *SPE Reservoir Characterisation and Simulation Conference and Exhibition*. Society of Petroleum Engineers, 2015. doi: 10.2118/175590-MS. URL `http://dx.doi.org/10.2118/175590-MS`.

[23] Mansoureh Jesmani, Mathias C. Bellout, Remus Hanea, and Bjarne Foss. Well placement optimization subject to realistic field development constraints. *Computational Geosciences*, 20(6):1185–1209, 2016. ISSN 1573-1499. doi: 10.1007/s10596-016-9584-1. URL `http://dx.doi.org/10.1007/s10596-016-9584-1`.

[24] Mansoureh Jesmani, Behnam Jafarpour, Mathias C. Bellout, Remus Hanea, and Bjarne Foss. Application of simultaneous perturbation stochastic approximation into well placement optimization under uncertainty. In *EUROPEC/EAGE Conference and Exhibition*, Amsterdam, The Netherlands, 2016. URL `http://dx.doi.org/10.3997/2214-4609.201601873`.

[25] James Kennedy and Russell C. Eberhart. Partical swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks*, pages 1942–1948, 1995.

[26] O. Leeuwenburgh, P. J. P. Egberts, and O. A. Abbink. Ensemble methods for reservoir life-cycle optimization and well placement. In *SPE/DGS Saudi Arabia Section Technical Symposium and Exhibition*. Society of Petroleum Engineers, 2010. doi: 10.2118/136916-MS.

[27] Gaoming Li and Albert C. Reynolds. Uncertainty quantification of reservoir performance predictions using a stochastic optimization algorithm. *Computational Geosciences*, 15(3):451–462, 2011. doi: 10.1007/s10596-010-9214-2.

[28] Lianlin Li and Behnam Jafarpour. A variable-control well placement optimization for improved reservoir development. *Computational Geosciences*, 16(4):871–889, September 2012. doi: 10.1007/s10596-012-9292-4.

[29] Lianlin Li, Behnam Jafarpour, and M. Reza Mohammad-Khaninezhad. A simultaneous perturbation stochastic approximation algorithm for coupled well placement and control optimization under geologic uncertainty. *Computational Geosciences*, 17(1):167–188, 2013. doi: 10.1007/s10596-012-9323-1.

[30] Jane-Jing Liang and Ponnuthurai Nagaratnam Suganthan. Dynamic multi-swarm particle swarm optimizer with local search. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on.*, volume 1, 2005. doi: 10.1109/CEC.2005.1554727.

[31] K.-A. Lie, S. Krogstad, I. S. Ligaarden, J. R. Natvig, H. M. Nilsen, and B. Skaflestad. Open source Matlab implementation of consistent discretisations on complex grids. *Computational Geosciences*, 16(2):297–322, 2012. doi: 10.1007/s10596-011-9244-4.

[32] H. S. Lope and L. S. Coelho. Particle swarn optimization with fast local search for the blind traveling salesman problem. In *Fifth International Conference on Hybrid Intelligent Systems (HIS'05)*, pages 245–250, 2005. doi: 10.1109/ICHIS.2005.86.

[33] Hilmar Magnusson. Development of constraint handling techniques for well placement optimization in petroleum field development. Master's thesis, NTNU, 2016.

[34] Shiva Navabi, Reza Khaninezhad, and Behnam Jafarpour. A unified formulation for generalized oilfield development optimization. *Computational Geosciences*, 2016. doi: 10.1007/s10596-016-9594-z.

[35] Jérôme E. Onwunalu and Louis J. Durlofsky. Application of a particle swarm optimization algorithm for determining optimum well location and type. *Computational Geosciences*, 14(1):183–198, 2010. doi: 10.1007/s10596-009-9142-1.

[36] K.E. Parsopoulos and M.N. Vrahatis. Recent approaches to global optimization problems through particle swarm optimization. *Natural Computing*, 1(2-3):235–306, 2002.

[37] Payman Sadegh. Constrained optimization via stochastic approximation with a simultaneous perturbation gradient approximation. *Automatica*, 33(5):889–892, 5 1997. doi: http://dx.doi.org/10.1016/S0005-1098(96)00230-0. URL `http://www.sciencedirect.com/science/article/pii/S0005109896002300`.

[38] Pallav Sarma and Wen H. Chen. Efficient well placement optimization with gradient-based algorithms and adjoint models. In *Intelligent Energy Conference and Exhibition*, Amsterdam, The Netherlands, 2008. Society of Petroleum Engineers. doi: 10.2118/112257-MS.

[39] Céline Scheidt and Jef Caers. Representing spatial uncertainty using distances and kernels. *Mathematical Geosciences*, 41(4):397, 2008. ISSN 1874-8953. doi: 10.1007/s11004-008-9186-0. URL `http://dx.doi.org/10.1007/s11004-008-9186-0`.

[40] Mehrdad G. Shirangi and Louis J. Durlofsky. Closed-loop field development under uncertainty by use of optimization with sample validation. *SPE Journal*, 20(5):908–922, 2015. doi: 10.2118/173219-PA.

[41] Mehrdad G. Shirangi and Louis J. Durlofsky. A general method to select representative models for decision making and optimization under uncertainty. *Computational Geosciences*, 96:109–123, 2016. doi: http://dx.doi.org/10.1016/j.cageo.2016.08.002.

[42] James C Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *Automatic Control, IEEE Transactions on*, 37(3):332–341, 1992.

[43] James C Spall. Implementation of the simultaneous perturbation algorithm for stochastic optimization. *Aerospace and Electronic Systems, IEEE Transactions on*, 34(3):817–823, 1998.

[44] James C. Spall. Adaptive stochastic approximation by the simultaneous perturbation method. *Automatic Control, IEEE Transactions on*, 45(10): 1839–1853, 2000. doi: 10.1109/TAC.2000.880982.

[45] James C. Spall. *Introduction to stochastic search and optimization: estimation, simulation, and control.*, volume 65. John Wiley & Sons, 2005.

[46] S. Vlemmix, G. J. P. Joosten, R. Brouwer, and J.-D Jansen. Adjoint-based well trajectory optimization. In *EUROPEC/EAGE Conference and Exhibition*, Amsterdam, The Netherlands, 2009. Society of Petroleum Engineers. doi: 10.2118/121891-MS.

[47] C. Wang, G. Li, and A. C. Reynolds. Production optimization in closed-loop reservoir management. *SPE Journal*, 14(3):506 –523, 2009. doi: 10.2118/109805-PA.

[48] Chunhong Wang, Gaoming Li, and Albert Coburn Reynolds. Optimal well placement for production optimization. In *Eastern Regional Meeting*, Lexington, Kentucky USA, 2007. Society of Petroleum Engineers. doi: 10.2118/111154-MS.

[49] Honggang Wang, David Echeverría-Ciaurri, Louis J. Durlofsky, and Alberto Cominelli. Optimal well placement under uncertainty using a retrospective optimization framework. *SPE Journal*, 17(1):112–121, 2012. doi: 10.2118/141950-PA.

[50] Michael Wetter. *GenOpt, Generic Optimization Program.* Simulation Research Group, Building Technologies Department, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, 3.1.0 edition, 2011.

[51] Burak Yeten, Louis J. Durlofsky, and Khalid Aziz. Optimization of nonconventional well type, location, and trajectory. *SPE Journal*, 8(3):200–210, 2003. doi: 10.2118/86880-PA.

[52] M. Zandvliet, M. Handels, G. van Essen, R. Brouwer, and J.D. Jansen. Adjoint-based well-placement optimization under production constraints. *SPE Journal*, 13(4):392–399, 2008. doi: 10.2118/105797-PA.

[53] Xun Zhu and James C. Spall. A modified second-order spsa optimization algorithm for finite samples. *International Journal of Adaptive Control and Signal Processing*, 16(5):397–409, 2002. doi: 10.1002/acs.715.