Optimization of high-reliability-based hydrological design problems by robust automatic sampling of critical model realizations

Peter Bayer,[1] Michael de Paly,[2] and Claudius M. Bürger[3]

Received 6 April 2009; revised 25 November 2009; accepted 14 December 2009; published 1 May 2010.

[1] This study demonstrates the high efficiency of the so-called stack-ordering technique for optimizing a groundwater management problem under uncertain conditions. The uncertainty is expressed by multiple equally probable model representations, such as realizations of hydraulic conductivity. During optimization of a well-layout problem for contaminant control, a ranking mechanism is applied that extracts those realizations that appear most critical for the optimization problem. It is shown that this procedure works well for evolutionary optimization algorithms, which are to some extent robust against noisy objective functions. More precisely, differential evolution (DE) and the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) are applied. Stack ordering is comprehensively investigated for a plume management problem at a hypothetical template site based on parameter values measured at and on a geostatistical model developed for the Lauswiesen study site near Tübingen, Germany. The straightforward procedure yields computational savings above 90% in comparison to always evaluating the full set of realizations. This is confirmed by cross testing with four additional validation cases. The results show that both evolutionary algorithms obtain highly reliable near-optimal solutions. DE appears to be the better choice for cases with significant noise caused by small stack sizes. On the other hand, there seems to be a problem-specific threshold for the evaluation stack size above which the CMA-ES achieves solutions with both better fitness and higher reliability.

Citation: Bayer, P., M. de Paly, and C. M. Bürger (2010), Optimization of high-reliability-based hydrological design problems by robust automatic sampling of critical model realizations, *Water Resour. Res.*, *46*, W05504, doi:10.1029/2009WR008081.

1. Introduction

[2] Hydrological modeling always involves dealing with uncertainties. Well-known in groundwater modeling is the uncertainty of spatially distributed parameters such as hydraulic conductivity. In most cases the values of such a parameter field have to be determined from a few point measurements that are transformed into a spatial distribution by expert judgment or pattern simulation (e.g., geostatistics).

[3] In the popular Monte Carlo framework, numerous equally probable realizations of a parameter field are generated instead of only one deterministic model configuration. The uncertainty of the model output or its prediction is then estimated by constructing a probability distribution from the total number of model predictions. Obviously, the computational effort for obtaining such a model prediction with uncertainty quantification then increases proportional to the number of realizations: a fact that can be prohibitive when repeated model runs are foreseen [*Mantoglou and Kourakos*, 2007; *Kourakos and Mantoglou*, 2008; *Franssen and Kinzelbach*, 2009; *Ascough et al.*, 2008]. This is particularly important for simulations intended for an optimization procedure, a so-called combined simulation-optimization. Here the model is used to compute the value of an objective function or cost function that has to be iteratively evaluated within the optimization procedure in order to find the optimal solution. The number of model iterations can be easily in the range of thousands for a practical optimization problem. Assuming a set of thousand realizations, this would result in millions of model simulations for the entire optimization procedure. Examples for such applications are the optimization of pumping wells [*Bayer and Finkel*, 2007; *Ng and Eheart*, 2008], the optimization of remediation technologies [*Wagner and Gorelick*, 1989; *Smalley et al.*, 2000; *Zheng and Wang*, 2002; *Hu et al.*, 2007; *Ricciardi et al.*, 2007], water supply [*Cui and Kuczera*, 2005; *Kirsch et al.*, 2009], and monitoring or sampling networks [*Meyer and Brill*, 1988; *Zhang et al.*, 2005; *Chadalavada and Datta*, 2008].

[4] This study presents a strategy that significantly reduces the computational effort of combined simulation-optimization with multiple realizations. It is based on automated sampling of a small number of critical realizations. By critical we refer to those model variants that are most influential for or constrain the optimal solution. As an illustrative example, let us consider a well capture

[1] Ecological System Design, Institute for Environmental Engineering, ETH Zurich, Zurich, Switzerland.
[2] Center for Bioinformatics Tübingen, University of Tübingen, Tübingen, Germany.
[3] Center for Applied Geosciences, University of Tübingen, Tübingen, Germany.

0043-1397/10/2009WR008081

problem: a model simulates the movement of a contaminant plume that has to be hydraulically captured by a down-gradient well with a fixed position. The objective is to minimize the pumping rate of this well and still to guarantee capture for a given set of hydraulic conductivity realizations. If this set includes all realizations then the solution is believed to be 100% reliable [e.g., *Wagner and Gorelick*, 1989; *Vasquez et al.*, 2000; *Ko and Lee*, 2008]. For this problem setup (single well with fixed position, capture for all realizations) there exists one most critical realization. If the pumping rate is continuously lowered (starting from a high extraction rate that ensures capture for all realizations), this one is the first of all realizations which shows a loss of full capture. However, as far as no additional (e.g., hydrogeological) knowledge can be utilized, all realizations have to be modeled one by one in order to recognize this most critical one [e.g., *Morgan et al.*, 1993].

[5] Hydrological management problems tend to be more complex than this example. It may be desirable to optimize well layouts of well galleries, where both pumping rates and well positions are to be adjusted [*Huang and Mayer*, 1997; *Hsiao and Chang*, 2005; *Bayer et al.*, 2009]. Time-dependent solutions may be foreseen with management periods [*Chang et al.*, 2007; *Tilmant and Kelman*, 2007; *Schütze and Schmitz*, 2009], and monetary objectives may be preferred to physical design criteria [*Mayer et al.*, 2002; *Nunes et al.*, 2004; *Papadopoulou et al.*, 2003; *Becker et al.*, 2006]. As has been demonstrated in several of these studies, solving such problems, which are often nonlinear and nonconvex, requires suitable optimization algorithms. Heuristics such as evolutionary algorithms have evolved as a method of choice. If properly applied, these methods can be very robust, reliably find (close to) optimal solutions, and cope with noise (e.g., resulting from the numerical model [*Ahlfeld and Hoque*, 2008]). All of that, however, comes at the expense of a considerable number of function calls compared to classic optimization algorithms (e.g., Gauss-Newton search) on sufficiently smooth objective functions. Despite that, their more general applicability and exclusive features have created a broad application field and a huge number of evolutionary algorithm variants have been developed [*De Jong*, 2006]. Also, problem-oriented modifications are evolving, like the ones adapted to the specialties of model-based hydrological management [e.g., *Smalley et al.*, 2000; *Espinoza and Minsker*, 2006; *Guo et al.*, 2007].

[6] In our previous work we discussed different strategies to capitalize on evolutionary search to solve multiple realization-based problems [*Bayer et al.*, 2008]. The main idea was to sample not the entire set of potential realizations but a representative subset and consequently to keep the computational effort low. One simple way is to sequentially use each realization for the objective function evaluation until one is found that indicates a constraint violation or a nonfeasible solution. If that is the case, the tested candidate solution is not desirable and no further analysis regarding the remaining realizations is necessary. In a hypothetical freshwater extraction problem, this nearly halved the number of model runs needed for optimization in comparison to using the complete realization set.

[7] In other applications, *Smalley et al.* [2000] and *Wu et al.* [2006] demonstrated the efficiency of limiting the realizations per objective function evaluation to a very small number (e.g., 4–10). By randomly replacing the realizations within this evaluation set during each objective function call, that is, when a new candidate solution is tested, many more than 4–10 realizations influence the course of the optimization. Within this procedure the goal remains to find both highly reliable and close-optimal solutions, although random resampling can render the objective function considerably noisy and thus jeopardize the performance of the search algorithm. *Singh and Minsker* [2008] further developed a multiobjective approach that borrows the noisy genetic algorithm procedure as presented by *Smalley et al.* [2000].

[8] This work deals with the so-called stack-ordering method: A small set of realizations ("evaluation stack") is sampled from the entire ensemble ("repository stack") for each objective function evaluation of the optimization algorithm. Only the realizations of this evaluation stack are sequentially used for approximating the model prediction. Depending on the objective function response, realization-specific weights are assigned to each (evaluated) realization, so that potentially critical realizations are automatically accentuated. These weights ("credits") are updated in each new function evaluation. On the basis of the collected individual weights, realizations in the evaluation subset are then probabilistically sampled for each objective function evaluation. This is the main difference to pure random sampling as utilized by *Smalley et al.* [2000].

[9] Subsequently, we further develop a stack-ordering variant introduced by *Bayer et al.* [2008] and examine in detail the appropriate setting of algorithm-specific parameters. Furthermore, two different evolutionary algorithms are chosen in order to analyze the performance of stack ordering with different solution concepts. As an application, a real-site-oriented case study is adopted, and well-layout problems with a single well and five wells are investigated. The purpose of this is to identify a robust algorithm configuration that achieves close-optimal and highly reliable solutions within a single-objective optimization framework. The transferability of the results is ultimately tested by four additional, different problem cases. A major concern for usefulness of the stack-ordering method is the savings in computational effort, and also the number of critical realizations that determine the solution of such a typical water management optimization problem.

2. Lauswiesen Well-Layout Problem

[10] In order to generate a realistic template site for the intended optimization studies under uncertainty, hydraulic parameters and the underlying geostatistical model were taken from the test field Lauswiesen located in the Neckar valley in the immediate vicinity of the city of Tübingen in southwest Germany [see *Sack-Kühner*, 1996]. The test site encompasses a 3×1 km^2 large part of a typical highly heterogeneous alluvial valley fill aquifer, which commonly serves as a local drinking water source for small to medium size communities within this area. Aquifer flow and transport parameters have been studied intensively in the past 15 years and geostatistical models developed [see, e.g., *Riva et al.*, 2006, and references therein]. From *Riva et al.* [2006] we adopt the so-called "single facies" model and the following parameters from the Lauswiesen site: the ambient hydraulic gradient of 0.1%, the (uncertain) geometric mean of the hydraulic conductivity distribution $K_G = 2.22 \times$

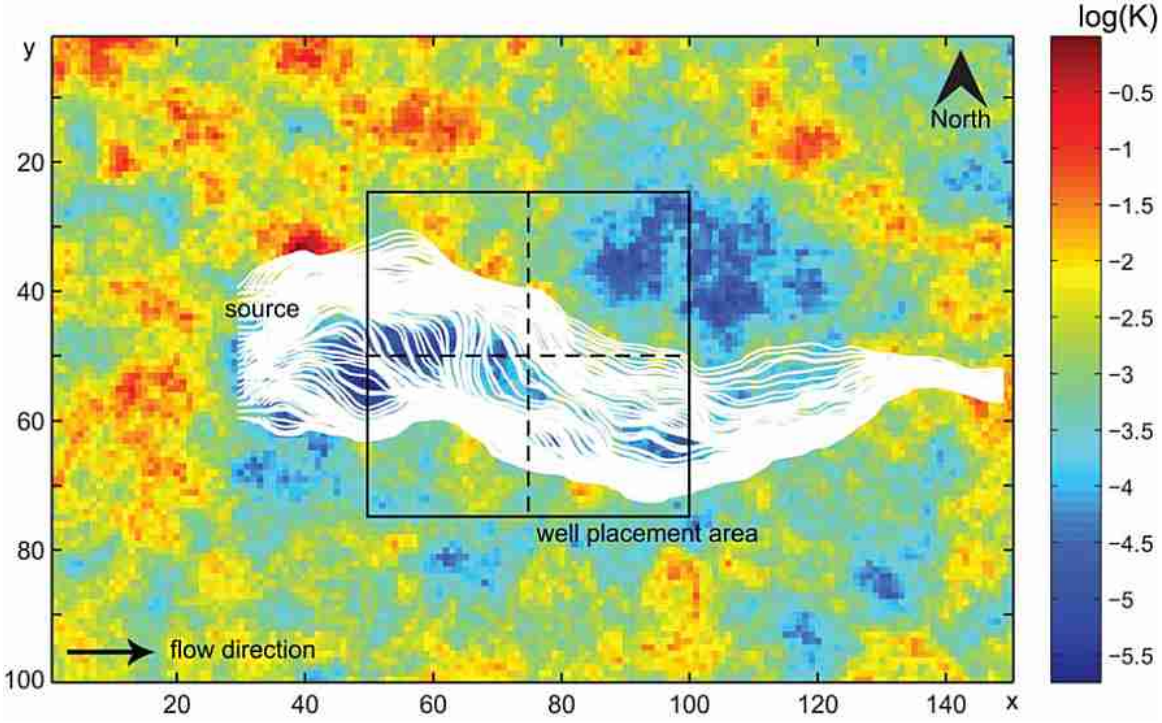

Figure 1. Top view of "true" hydraulic conductivity distribution at template site. From the source at column 25, particles move in groundwater flow direction and through the well placement area, which is subdivided into four quarters for the five-well case.

10^{-3} m s^{-1}, and the variance of the ln-conductivity field σ^2 = 2.91 (by "ln-conductivity" we refer to the natural logarithm of the hydraulic conductivity $Y = \ln K$). Furthermore, we assumed an isotropic exponential variogram model with a correlation length of 10 m. Please note that the intention of this study is not to simulate fluid flow and contaminant transport for the Lauswiesen site - parameter values measured or derived for this site simply serve as a basis for our site model setup.

[11] For the 2-D template site we set the model geometry to a gridded rectangular domain with the origin of the coordinate system in the upper left corner and coordinate axes parallel to the respective domain boundaries (see Figure 1). The x axis extends to the east by 150 m with a 1 m discretization. Aquifer thickness is assumed as unity. The y axis extends to the south for 100 m with a 1 m discretization. A "true" hydraulic conductivity field is generated by unconditional geostatistical simulation (see below), and a 2-D steady state confined flow field is established for the following boundary conditions: constant flow of Q_{tot} = 10^{-3} m^3 s^{-1} at the western domain boundary and constant head of h = 10 m at the eastern domain boundary.

[12] Within the true conductivity field and the true flow field, hypothetical conductivity measurements and head measurements are taken on the basis of the following rectangular grid layout: hydraulic conductivity measurements: at columns x ranging from 5 m to 145 m every 20 m and at rows y ranging from 10 m to 90 m every 20 m (40 in total); hydraulic head measurements: at x ranging from 15 m to 135 m every 20 m and at y ranging from 20 m to 80 m every 20 m (28 in total).

[13] On the basis of these measurements and the single facies model, a stack of 1000 realizations conditioned on hydraulic conductivity and head data was created by the quasi-linear method of geostatistical inversing presented by *Kitanidis* [1995] and further developed by *Nowak and Cirpka* [2004]. Conditional realizations are generated starting from unconditional simulations by the spectral method [see, e.g., *Gutjahr et al.*, 1994] for a zero mean and a covariance matrix according to the above mentioned geostatistical model. For each unconditional realization, an iterative estimation of an additive smooth correction term was carried out, which eventually ensured the reproduction of the measurements up to the accuracy of the measurement values while maintaining the statistics of the underlying geostatistical model. The uniform geometric mean of the ln-conductivity field was hereby assumed to be uncertain with a variance of 0.6. Additionally, we accounted for measurement errors allowing for an absolute error of 0.1 for ln-conductivity and an absolute error of 0.01 m for hydraulic head measurements.

[14] For this model setup, a typical well capture problem is implemented. The objective is to identify a single well and a five-well solution that minimize the total extraction rate while still guaranteeing capture for a given up-gradient aquifer zone. The pumping rate is used as a proxy for the maintenance cost, which is a simplification, but also realistic considering observations from practice [e.g., *Bayer and Finkel*, 2004]. Complexity is introduced by considering both well coordinates and pumping rates as decision variables. This results in a moving well problem, which is simulated by the USGS routine MODFLOW [*Harbaugh et al.*, 2000] and plume capture evaluated by particle tracking using MODPATH [*Pollock*, 1994]. The starting positions of the particles are assigned to a rectangular area, which could reflect an assumed contamination source zone (Figure 1). Neglecting dispersion

processes, hydraulic capture should be achieved by a set of down-gradient wells, if all particles reach the wells under steady state conditions. This simplification reduces the computational burden compared to full advective-dispersive transport simulations. Run times are about one second per model.

[15] The optimization problem can be formulated as an advective control problem [*Mulligan and Ahlfeld*, 1999; *Bayer and Finkel*, 2004; *Bürger et al.*, 2007]:

$$\min F = \varphi(v) \cdot f(q_{tot}) = \varphi(v) \cdot \sum_{w=1}^{W} q_w(x_w, y_w), \tag{1}$$

subject to

$$x_{w,low} < x_w < x_{w,up}, \tag{2}$$

$$y_{w,low} < y_w < y_{w,up}, \tag{3}$$

$$q_{w,low} < q_w < q_{w,up}, \tag{4}$$

where W is the number of wells, w is the index of a particular well, and x_w and y_w are column and row indices of the wells. Parameters q_{tot} and q_w stand for the absolute values of the total pumping rate and those of individual wells, respectively. The penalty term φ depends on v, the number of particles that are not captured. It is set to a high value ($\varphi(v) = 10^v$) in order to prevent invalid solutions from being favored. This means that even one particle that is not captured penalizes the objective function value by 1 order of magnitude.

[16] The position of the single well is optimized within a quadratic well placement area down-gradient from the contaminant source (Figure 1). This area is delimited by boundaries at $x_{1,low} = 50$, $x_{1,up} = 100$, $y_{1,low} = 25$, $y_{1,low} = 75$, yielding a moving well problem with 2601 possible positions. For the placement of multiple wells, we follow suggestions by *Huang and Mayer* [1997], who distinguish separate placement areas for each individual candidate well. As depicted in Figure 1, the quadratic zone of the single-well case is subdivided into four sectors of equal size. For the application, a five-well case is considered, where the first four wells are assigned to these sectors, and the fifth one is free as in the single-well case and may be moved into all sectors. By this subdivision, the decision space can be significantly reduced, while it is still possible to represent a sufficient range of potential well configurations.

[17] The limits for the pumping rates are set using the experience from a few preliminary trial runs. The single-well rate is constrained between $q_{1,low} = 5$ m^3/h and $q_{1,up} = 50$ m^3/h, thus is variable within 1 order of magnitude. The ranges for the five-well case are set lower, assuming that the individual extraction rates are smaller than in case of only one well ($q_{w,low} = 0.5$ m^3/h, $q_{w,up} = 25$ m^3/h).

3. Optimization Procedure

3.1. Differential Evolution

[18] Differential evolution (DE) [*Storn and Price*, 1995; *Price et al.*, 2005] is an evolutionary optimization algorithm that works on a population of N candidate solutions. It has recently been applied to hydrological management problems, for example, by *Karterakis et al.* [2007]. The actual optimization process can be described as a succession of generations, where the population $\mathbf{x}_i^{g+1}$ *(i = 1...N)*of each generation is generated through the subsequent application of the operator's mutation, crossover and selection on the candidate solutions of the previous generation $\mathbf{x}_i^g$ *(i = 1...N)*. The candidate solutions that form a generation are called individuals. For each individual the objective function is evaluated, and consequently the model is run with a set of realizations.

[19] DE starts from an initial generation g_0 consisting of N uniformly sampled real valued candidate solutions. To form the candidate solutions of the next generation $\mathbf{x}_i^{g+1}$, a trial vector $\mathbf{u}_i^g$ is generated for each individual $\mathbf{x}_i^g$ by a crossover of $\mathbf{x}_i^g$ with a mutant vector $\mathbf{v}_i^g$. $\mathbf{v}_i^g$ is composed of three randomly chosen points $\mathbf{x}_{ro}^g$, $\mathbf{x}_{r1}^g$, and $\mathbf{x}_{r2}^g$ from generation g with

$$\mathbf{v}_i^g = \mathbf{x}_{r0}^g + W(\mathbf{x}_{r1}^g - \mathbf{x}_{r2}^g) \quad 0 < W \leq 1, \tag{5}$$

where W is a user-defined constant that controls the evolution rate of the population. It has been set to 0.8 throughout this paper, which has been empirically shown to be successful on a wide range of optimization problems [*Price et al.*, 2005]. Larger W causes a larger impact of the difference vector $\mathbf{x}_{r1}^g - \mathbf{x}_{r2}^g$ on $\mathbf{v}_i^g$ and results in a higher diversity within the generated candidate solutions.

[20] Each component of $\mathbf{u}_i^g$ is then randomly chosen from either $\mathbf{v}_i^g$ or $\mathbf{x}_i^g$ according to

$$u_{j,i}^g = \begin{cases} v_{j,i}^g & \text{if } U(0,1) \leq C_{DE} \text{ or } j = j_r \\ x_{j,i}^g & \text{otherwise} \end{cases}, \tag{6}$$

where j denotes the current component of the vector and $U(0,1)$ a uniformly sampled random variable. To ensure that at least one component of $\mathbf{u}_i^g$ differs from $\mathbf{x}_i^g$, the j_r-th component of $\mathbf{u}_i^g$ is always taken from $\mathbf{v}_i^g$. Because of this convention, C_{DE} which is a user define- constant and has been set to 0.9 throughout this paper only approximates the crossover probability p_c of conventional evolutionary algorithms [*Price et al.*, 2005].

[21] After crossover, DE selects either the $\mathbf{u}_i^g$ or $\mathbf{x}_i^g$ for the next generation, depending on their objective function values. In case of a minimization problem, as in this paper, it reads

$$\mathbf{x}_i^{g+1} = \begin{cases} \mathbf{u}_i^g & \text{if } F(\mathbf{v}_i^g) \leq F(\mathbf{x}_i^g) \\ \mathbf{x}_i^g & \text{otherwise} \end{cases}. \tag{7}$$

The resulting individuals $\mathbf{x}_i^{g+1}$, one of which is generated for each individual $\mathbf{x}_i^g$ of the current generation, comprise the population of the next generation. The entire sequence of mutation, crossover, and selection to form subsequent generations is repeated until a stopping criterion, usually a predefined number of function evaluations, is met.

3.2. Evolution Strategy With Covariance Matrix Adaptation

[22] The second optimization algorithm, Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [*Hansen and Ostermeier*, 2001], is from the family of evolution

strategies. It has been applied to groundwater management problems, for example, by *Bayer et al.* [2005, 2008] and *Bürger et al.* [2007]. Like DE, CMA-ES works on a population of individuals $\mathbf{x}_i^g$, but differs greatly in the way the individuals $\mathbf{x}_i^{g+1}$ of the next generation are generated. The basic idea of CMA-ES is the efficient adaptation of the search distribution to the contour lines of the target function F. This is done through the approximation of a covariance matrix $\mathbf{C}$ which can also be interpreted as an approximation of the inverse Hessian of the objective function [*Hansen*, 2008].

[23] On the basis of $\mathbf{C}$, CMA-ES generates λ offspring for each generation according to

$$x_i^{g+1} = m^g + \sigma^g N(0, \mathbf{C}^g) \quad i = 1...\lambda, \tag{8}$$

where σ^g is the overall standard deviation which controls the step size and $N(0,\mathbf{C}^g)$ is a normal distribution with zero mean and the covariance matrix $\mathbf{C}^g$. The parameter m^g represents the weighted mean of the sorted μ best candidate solutions (denoted by $\mathbf{x}_{i:\lambda}^g$) of the current generation with

$$\begin{gathered} m^g = \sum_{i=1}^{\mu} w_i x_{i:\lambda}^g \\ \sum_i^{\mu} w_i = 1 \quad w_1 \geq w_2 \geq \cdots \geq w_\mu > 0 \quad \mu < \lambda \end{gathered}. \tag{9}$$

Typical values for w_i [*Hansen and Ostermeier*, 2001; *Hansen*, 2008], which have also been used throughout this paper, are defined by

$$w_i = \frac{w_i'}{\sum_j^{\mu} w_j'} \quad w_i' = \ln\left(\frac{\lambda+1}{2}\right) - \ln(i). \tag{10}$$

The combination of equations (9) and (10) assigns higher weights to better candidate solutions, which leads to a better convergence of the optimization process [*Hansen and Ostermeier*, 2001; *Bayer and Finkel*, 2004]. As only the μ best candidate solutions of the entire stock of λ candidates are used to compute the weighted mean, equation (9) also implicitly acts as the selection operator of CMA-ES. The same selection procedure is also used throughout the computation of $\mathbf{C}^g$ and σ^g, which is briefly described in the following paragraphs.

[24] The dimension of the problem is defined by the number of decision parameters. Since μ is usually much smaller than the dimension, $\mathbf{C}^g$ cannot adequately be approximated solely from the population of the current generation. To overcome this problem, several updating schemes which incorporate information from previous generations have been developed for CMA-ES. For this paper we used a combination of rank-1 update [*Hansen and Ostermeier*, 2001] and rank-μ update [*Hansen et al.*, 2003].

[25] The rank-1 update incorporates information from previous generations using the evolution path p_c^g, which is based on the sum of the weighted difference of points from the generations g and g-1, as in

$$p_c^g = (1-c_c)p_c^{g-1} + \sqrt{c_c(2-c_c)}\frac{m^g - m^{g-1}}{\sigma^{g-1}} \quad 0 < c_c \leq 1, \tag{11}$$

where c_c controls the backward time horizon of p_c^g and is usually chosen between $1/n$ and $1/\mathrm{sqrt}(n)$, where n is the dimensionality of the optimization problem. Because of the weighting with c_c, the influence of a certain search step depends on the numbers of generation in between. The farther in the past a search step occurred the lower is its impact on p_c^g.

[26] Combined with the rank-μ update, $\mathbf{C}^g$ is approximated by the weighted sum of the covariance matrix of the previous generation $\mathbf{C}^{g+1}$, the covariance of the evolution path from the rank-1 update $p_c^g\ (p_c^g)^{\mathrm{T}}$, and a covariance matrix of rank μ $\mathbf{C}_\mu^g$, which is determined from the μ best candidate solutions of the current generation:

$$\begin{aligned} \mathbf{C}^g = (1-c_{\mathrm{cov}})\mathbf{C}^{g-1} + \frac{c_{\mathrm{cov}}}{\mu_{\mathrm{cov}}} p_c^g (p_c^g)^{\mathrm{T}} \\ + c_{\mathrm{cov}}\left(1 - \frac{1}{\mu_{\mathrm{cov}}}\right)\mathbf{C}_\mu^g \quad 0 \leq c_{\mathrm{cov}} < 1. \end{aligned} \tag{12}$$

The determination of $\mathbf{C}_\mu^g$ is closely related to the definition of an empirical covariance matrix from μ sampling points [*Hansen*, 2008] with

$$\begin{gathered} \mathbf{C}_\mu^g = \sum_{i-=1}^{\mu} w_i y_{i:\lambda}^g (y_{i:\lambda}^g)^T \\ y_{i:\lambda}^g = \frac{x_{i:\lambda}^g - m^{g-1}}{\sigma^{g-1}} \end{gathered}. \tag{13}$$

Please note, since $\mathbf{C}_\mu^g$ uses the weighted mean of the previous generations m^{g-1} (see equation (9)) instead of the mean of the actually sampled points, $\mathbf{C}_\mu^g$ is an estimator for the distribution of the selected μ best steps and not an estimator of the variance within the realized candidate solutions. To further improve convergence, the rank-μ update assigns higher weights to better candidate solutions in a process similar to the determination of m^g. Therefore, $\mathbf{C}_\mu^g$ has a bias toward the reproduction of steps that resulted in an improved objective function value [*Hansen*, 2008].

[27] For faster adaptation, the overall step size σ^g is determined independently from $\mathbf{C}^g$. As for the rank-1 update, CMA-ES uses the evolution patch to incorporate information from previous generations. Since the direction of the evolution path influences its expected length, a conjugate evolution path p_σ^g is constructed with

$$\begin{gathered} p_\sigma^g = (1-c_\sigma)p_\sigma^{g-1} + \sqrt{c_\sigma(2-c_\sigma)}\mathbf{C}^{g^{-1/2}}\frac{m^g - m^{g-1}}{\sigma^{g-1}} \quad 0 < c_\sigma \leq 1 \\ \mathbf{C}^{g^{-1/2}} = \mathbf{B}^g \mathbf{D}^{g^{-1}} \mathbf{B}^{g\mathbf{T}} \end{gathered}, \tag{14}$$

where $\mathbf{C}^g = \mathbf{B}^g\ \mathbf{D}^{g^2}\ \mathbf{B}^{g^{\mathrm{T}}}$ is the eigendecomposition of $\mathbf{C}^g$. Therefore, $\mathbf{B}^g$ contains an orthonormal basis of eigenvectors of $\mathbf{C}^g$ and $\mathbf{D}^g$ is a matrix, the diagonal elements of which contain the square roots of the corresponding eigenvalues. The parameter $\mathbf{C}^{g-1^{1/2}}$ can be interpreted as a subsequent rotation of the principal axes of $N(0,\mathrm{C})$ onto the coordinate axes with $\mathbf{B}^{g^{\mathrm{T}}}$, a rescaling of all axes to equal size with $\mathbf{D}^{g^{-1}}$, and a back rotation of the rescaled axes into the original coordinate system with $\mathbf{B}^g$. The combination of all three transformations within $\mathbf{C}^{g-1^{1/2}}$ makes the expected length of p_σ^g independent of the actual rotation of the evolution path [*Hansen and Ostermeier*, 2001]. Parameter c_σ controls the influence of older generations on p_σ^g, where $1/c_\sigma$ is the backward time horizon which contains approximately 63% of the overall weight [*Hansen*, 2008].

Table 1. Parameter Settings for CMA-ES

Parameter	Value
λ	$4 + 3\ln(n)$
μ	$\lambda/2$
c_c	$\frac{4}{4+n}$
μ_{cov}	$\frac{1}{\sum_i^{\mu} w_i^2}$
c_σ	$\frac{\mu_{\text{cov}}+2}{n+\mu_{\text{cov}}+3}$
d_σ	$1 + 2\max\left(0, \sqrt{\frac{\mu_{\text{cov}}-1}{n+1}}\right) + c_\sigma$
c_{cov}	$\frac{1}{\mu_{\text{cov}}} \frac{2}{(n+\sqrt{2})^2} + (1 - \frac{1}{\mu_{\text{cov}}}) \min\left(1, \frac{2\mu_{\text{cov}}-1}{(n+2)^2+\mu_{\text{cov}}}\right)$

[28] On the basis of p_σ^g, σ^g is then updated according to

$$\sigma^g = \sigma^{g-1} \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|p_\sigma^g\|}{\mathrm{E}\|N(0,\mathbf{I})\|} - 1\right)\right) \quad d_\sigma \geq 1, \tag{15}$$

where $\mathrm{E}\|N(0,\mathbf{I})\|$ is the expectation of the Euclidian norm of a random vector drawn from an isotropic normal distribution with zero mean and variance 1. Equation (14) can be interpreted as a comparison of the actual length of $\|p_\sigma^g\|$ with its expected length$\mathrm{E}\|N(0,\mathbf{I})\|$. This means, if$\|p_\sigma^g\| > \mathrm{E}\|N(0,\mathbf{I})\|$, subsequent steps of the optimization process have canceled each other out, indicating that σ^g is too large and should be decreased. $\|p_\sigma^g\| > \mathrm{E}\|N(0,\mathbf{I})\|$ on the other hand indicates that σ^g is too small since all previous successful steps had a similar direction. The strategy parameter d_σ is usually set to values slightly larger than 1 to damp changes within σ^g.

[29] Like DE, CMA-ES repeats all steps to generate a succession of generations until a certain stopping criterion is met. Typical values for the parameters of CMA-ES [*Hansen and Ostermeier*, 2001], which have also been used within the paper, are given in Table 1.

3.3. Stack Ordering

[30] We define a full application of an evolutionary algorithm as an optimization run (Figure 2). The optimization run encompasses a fixed number of successive generations and thus a fixed number of individuals. Each individual represents a candidate solution that is evaluated by the objective function and thus requires iteration with the (flow and transport) model. By iteration we refer to running the model sequentially for a set of realizations.

[31] Figure 3 illustrates the procedure. Realizations are identified with indices $r = 1, \ldots, S$, which are arbitrarily set at the beginning and kept constant during the optimization procedure for the entire set (repository stack) of S realizations. Each realization is initialized to a credit of $C_r(t = 0) = 0$, and a small random sample of S_{eval} realizations (the evaluation stack S_e^1) is selected for objective function evaluation of the first candidate solution ($t = 1$). For each objective function evaluation t, the model is run once for each of the S_{eval} realizations in the evaluation stack S_e^t, where the index t indicates that the content of the evaluation stack may change according to the algorithm specified below for each objective function evaluation t. The evaluation stack S_e^t is to be understood as a list of S_{eval} indices r referring to those realizations that are subject to evaluation within objective function evaluation t. Since the purpose of evaluating a whole stack of realizations is to maximize the reliability of a management solution or technology, the objective function value of the candidate solution is set equal to the worst model response during this procedure. For a (cost) minimization problem, this reads

$$F^t = \max F(r), r \in S_e^t. \tag{16}$$

This creates an opportunity to save computation time. Instead of evaluating the whole evaluation stack, it is suggested to fully examine only potentially optimal solutions. If one of the model simulations of objective function evaluation t indicates a constraint violation, an infeasible solution, or if it does not satisfy any other case-specific criterion, the evaluation of the S_{eval} realizations is stopped. Otherwise, the evaluation is continued until all S_{eval} realizations have been tested.

[32] This procedure is repeated for the successive objective function evaluations. If a realization indicates an unsatisfactory objective function value, it is recognized as a potentially critical realization. In the subsequent objective function evaluations this information is exploited by accentuating the sampling of this realization: instead of uniform random sampling, a dynamically adapted probability distribution is constructed. This is done as follows: First, the realization is awarded a so-called credit by

$$C_r^{t+1} = C_r^t + \ln(pos_r^t), \tag{17}$$

where pos_r^t denotes the current position of the realization in the evaluation stack during an objective function evaluation. For example, consider an evaluation stack size of 25. If the 10th realization that is tested indicates constraint violation, its credit is increased by ln(10) = 2.3. The logarithmic dependency on the position is based on the rationale that the later a potentially critical realization is found, the higher its expected relevance compared to other realizations will be. An appropriate measure is the expected number of realizations r which perform worse than all realizations 1…(r-1) evaluated before. The probability of such a realization at position pos_r is $1/r$. Therefore the expected number of realizations which fulfill this criterion until the realization at position pos_r is reached can be calculated by $1/1 + 1/2 + \ldots + 1/pos_r$ which is the harmonic series that can be approximated by $\ln(pos_r)$. Please note that the logarithmic notation replaces the former square root formulation as presented by *Bayer et al.* [2008].

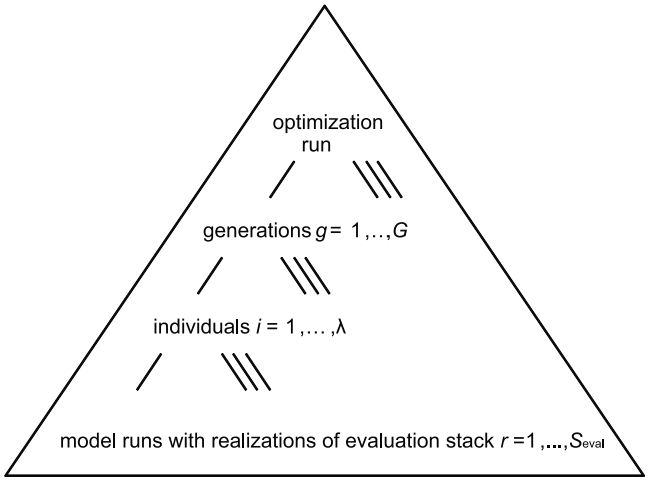

Figure 2. Hierarchical terminology used for the combination of evolutionary algorithms with stack ordering.

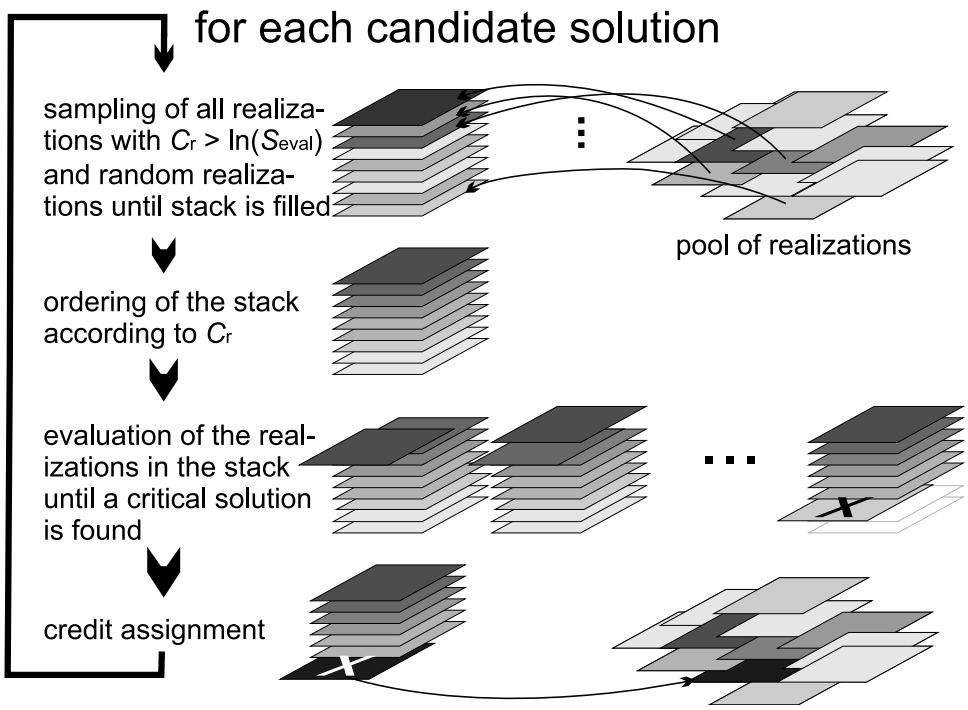

Figure 3. Stack-ordering procedure for one objective function evaluation (testing of one candidate well layout).

[33] For each new candidate solution the repository stack is sorted in ascending order according to the (accumulated) credits of each individual realization. All realizations that have zero credits are randomly shuffled. We start from top of the stack, pick the realizations one by one and decide on the basis of the credit whether it should be selected for the new evaluation stack. If certain realizations have accumulated a high credit C_r above a threshold, that is, the maximal credit per complete stack evaluation $\ln(S_{eval})$, these are automatically included in the evaluation stack. The remaining positions in the evaluation stack are then sequentially filled by probabilistic sampling, where the probability to be included in the evaluation stack is taken as

$$p_r^{t+1} = \frac{C_r^t + 1}{\ln(S_{eval}) + 1}. \tag{18}$$

Sampling will be stopped as soon as the evaluation stack is filled. For the next objective function evaluation t+1 the realizations are then taken from the evaluation stack in the new order established by the above threshold credit hierarchy and the subsequent probabilistic sampling. Consequently, the potentially most critical realizations are evaluated first.

[34] During the course of optimization the sequence of realizations typically does not remain constant, as different realizations may appear to be critical for the diverse candidate solutions tested. Consequently, realizations found critical in an early stage of the evolutionary search may not be critical anymore in the later phase when totally different well configurations are inspected.

[35] As an extension to the above presented method, a decay factor $k \in (0,1)$ can be included. In each objective function evaluation t, where no credit is assigned and no new critical realization is found, the credits of all realizations are reduced by k. This increases the probability of sampling new realizations for evaluation and prevents an early domination of the evaluation stack by a few realizations:

$$C_r^{t+1} = (1 - k)\ C_r^t. \tag{19}$$

The structure of equation (19) causes C_r to behave like many natural decay processes. Such an approach is commonly used in evolutionary algorithms like CMA-ES to increasingly limit the influence of function evaluations on the optimization process as their occurrence becomes further back in the past. The overall procedure can be easily implemented, and there exist only two free parameters that have to be set: the evaluation stack size S_{eval} and the decay factor k. In the following, a detailed analysis of the role of these parameters will be presented. For both, their effect on the optimization procedure will be tested by evaluating a range of parameter values. Three variants will be compared: random sampling of S_{eval}, stack ordering, and stack ordering with decay.

[36] Additionally, we also expand on the following idea: Instead of setting a fixed and constant value for S_{eval}, the stack size could be dynamically adapted during the course of the evolutionary search. The underlying rationale is that during the initial, more explorative, phase of the evolutionary search, a few realizations could be sufficient to approximate the objective function. During further convergence to an optimum, higher precision could be achieved by an increase of S_{eval}. Since also the number of potentially critical, credited realizations rises, it is logical to dynamically increase S_{eval} along with the rising number of credited realizations.

[37] Again, a compromise has to be found in order to achieve the desired effect at low computational demand. S_{eval} has to be adjusted so that only a small stack of critical realizations is evaluated while also new realizations are

inspected to improve the maturity of the stack. One way to achieve this balance is to set S_{eval} to twice the number of currently credited realizations, r_c^t:

$$S_{\mathrm{eval}}^{t+1} = 2\, r_{C_r>0}^{t}. \tag{20}$$

A more restrictive criterion would be to set it to twice the number of those realizations that are sampled with certainty owing to their high credits; that is, their credits are higher than $\ln(S_{\mathrm{eval}})$:

$$S_{\mathrm{eval}}^{t+1} = 2\, r_{C_r>\ln\left(S_{eval}^{t}\right)}^{t}. \tag{21}$$

All realizations with lower credits are potentially ignored and thus offer space for new realizations in the stack. The outcome is a stack of which one half is a set of critical realizations which gradually increases. The other half undergoes dynamic changes as the search evolves. Please note that each time the stack size is increased, also the criterion $\ln(S_{eval}^{t})$ gets stricter. This automatism is desirable as the credits per realization also increase. However, this can also result in an alternation of the stack size. For both variants, the evaluation stack is initialized with $S_{eval}^{1} = 1$, which also serves as lower limit for S_{eval} as long as no critical realizations are found. Subsequently, both variants with conservative and restrictive dynamic evaluation stack sizes will be compared to those with a priori defined static size.

3.4. Assessment of Results

[38] The nominal reliability R of a well layout is expressed as

$$R = \frac{S - r_v}{S} \cdot 100\%, \tag{22}$$

where r_v is the number of realizations, out of the entire set of those realizations, for which at least one particle is not captured. If all realizations indicate capture, R is maximal and reaches 100%. After each full optimization run, the reliability of the best solution is analyzed by testing it on the entire set S.

[39] Please note that in practice, hydraulic conductivity may be regarded as the most important source of uncertainty in hydrogeological modeling. However, uncertainty and imprecision are also inherent to numerous other model parameters; conceptual model shortcomings also have to be considered in order to cover the entire range of potential uncertainties that can affect the optimized solution. To keep the study concise, we only focus on hydraulic conductivity, although alternative parameters or conceptualizations could be included. The term nominal reliability is used herein to discriminate between the reliability based on the evaluation of multiple realizations and the true reliability that would cover all sources of uncertainty [e.g., *Ng and Eheart*, 2008].

[40] To be specific, in this application we are interested in a solution that maximizes the nominal reliability while system cost, that is, the total pumping rate, is minimized. The underlying assumption is that insufficient capture threatens potential down-gradient receptors, a risk that has to be held as low as possible. Solutions of high reliability commonly are hard to find, since the closer the nominal reliability R has to be to 100%, the more a solution is controlled by a few outliers, that is, critical realizations. For example, a well layout of 50% reliability usually can be well approximated by a small subset of realizations [e.g., *Smalley et al.*, 2000]. This number disproportionally grows the closer the final solution is supposed to be to the 100% criterion.

[41] In the following analysis, we focus on three aspects: (1) How close is the suggested solution to the global optimum? (2) How high is its reliability? (3) What is the number of model runs required to detect this solution? Since the presented procedure involves stochastic elements in evolutionary search and sampling of realizations, repeated optimization runs with different (random) starting points are conducted. This means for each problem and solver, 25 optimization runs are inspected. The results are statistically analyzed and median and 10%/90% quantiles are presented.

[42] We exclusively focus on the best solution of each optimization run that is found in the later stage of the search, that is, during the last 10% of objective function evaluations (after 630 function evaluations for single-well case, after 2700 for the five-well case). Considering solutions with best results from the entire search would increase the chance of "premature" candidate well layouts with good objective function values but low reliability. At the late search stage, it is expected that those solutions accumulate which are fit with respect to both the minimization of the objective function value and the validity for a high number of realizations.

[43] In the following, static and dynamic stack size variants are comprehensively examined only for the Lauswiesen case to compare their specific performance. In this manner, ideal settings of governing parameters (S_{eval}, k) are derived. In order to validate the results and test a general applicability of the method, four alternative well positioning problems are set up. The average performance of the solution procedure is then tested by ten separate optimization runs for each problem. Results are interpreted on the basis of reliability of results and computational effort required.

4. Results and Discussion

4.1. Characterization of Optimal Well Layouts

[44] For the assessment of the results, it is not intended to find the global optimum exactly. The latter is not known without a full enumeration of all possible well configurations, and taking into account the limited reliability of hydrologic models, not practically relevant. As commonly found in similar research studies, a compromise is made; we seek solutions that are supposedly close to optimal, that is, preferable well positions and (relative) extraction rates. Especially for multiple well layouts, it is commonly observed that several favorable well combinations exist, and that these typically differ most regarding wells with comparatively low extraction rates, as those are the spatially most variable. This may indicate a redundancy of such low-rate wells; however, in some cases exactly those wells exert an important influence on the local hydraulic regime and thus become a linking part of the system [e.g., *Bayer and Finkel*, 2007].

[45] During the presented examination of the stack-ordering procedure with the study site, millions of Lauswiesen model runs were performed in total and a huge number of well-layout solutions were inspected. This gave substantial insight into preferable well locations, while discriminating these from less beneficial regions in the well placement area. For example, the contaminant particles pass the well

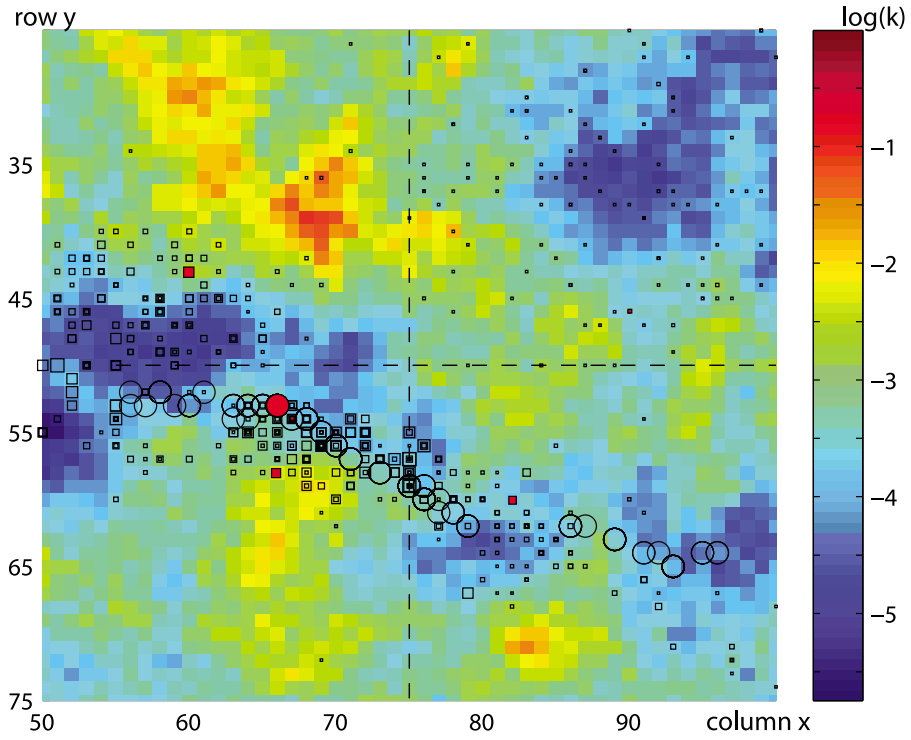

Figure 4. Zoomed section of the "true" hydraulic conductivity distribution in the well placement area. Circles denote (close) optimal well positions of the single-well case, and squares denote those for the five-well case. The relative individual extraction rates are expressed by different marker sizes. Best solutions are accentuated by red markers.

placement area particularly in the southern part. Hence, the northern half of the area appears to be less attractive. This is also illustrated by the undisturbed flow conditions as depicted for the "true realization" in Figure 1. Particle paths that originate from the source cross the well placement area with a preferred direction toward southeast, apparently diverted by the broad low-conductivity zone that dominates the northeast sector.

[46] In addition to the analysis with small evaluation stack sizes, a few (3–5) trial runs were performed with $S_{eval} = 250$, 500 and 1000. The purpose of this brute force approach was to examine whether better well layouts could be identified, since an increase of the number of realizations for each objective function evaluation will reduce the noise and thus enhance the optimization. Although these few computationally demanding trials are not sufficient for a rigorous conclusion, the fitness of the best well layouts found were very close to those of the highly reliable solutions detected during the stack-ordering application. We interpret this finding as a first indication of the efficiency of the new approach with small stack sizes. Furthermore, the best solutions seem to be hardly improvable and thus can be set as reference for the (close) optimal solutions searched for.

[47] For the single-well problem, the best solution with 100% nominal reliability that has been found during the entire study is a well at column $x_1 = 66$ and row $y_1 = 53$ that operates at $q_{tot} = q_1 = 15.0$ m^3/h. It is located in the SW sector of the well placement area in a local area of higher conductivity (Figure 4). Convergent flow here mitigates the pumping rate required for capture. Further close-optimal solutions appear along a narrow range up-gradient and down-gradient and delineate the central flow path that is simulated by the particles in Figure 1. Only the most eastern close-optimal well positions seem to be shifted somewhat to the north within a low-conductivity zone.

[48] Despite the small number of conditioning points, the ensemble of realizations obviously approximates characteristic features of the spatial conductivity of the hypothetical truth and hence also the flow conditions very well. This may indicate that the set of realizations could have been potentially reduced without consequences for the obtained results. Nevertheless, the realizations appear very different when inspected visually. In some test runs with randomly chosen subsets of 100 and 500 realizations in the repository stack, best solutions achieved only 93% and 97% reliability when cross-checked with the whole ensemble of 1000 realizations.

[49] The best well configurations with five wells at 100% nominal reliability also stress the preference area as identified for the single-well problem (Figure 4). Again the dominating wells are concentrated along the southern fringe of the extended low-conductivity area. Moreover, the southern part of the NW sector becomes attractive, apparently for capturing those particles that are deviated here to the north. The individual wells of the best configuration detected are specified as $q_1 = 3.3$ m^3/h, $x_1 = 60$, $y_1 = 43$ (NW sector), $q_2 = 0.5$ m^3/h (90/46, NE), $q_3 = 0.5$ m^3/h (67/54, SW), $q_4 = 2.3$ m^3/h (83/61, SE), $q_5 = 2.8$ m^3/h (66/58, free). A common observation is that the sector-free well is also assigned to the SW quarter in the vicinity of the best single-well locations. In contrast, the extraction rates of wells in the NE sector are mostly turned to the lower limit, indicating that wells in this region play no or only a minor role. Compared to the best single-well layout the total extraction rate could be further reduced to $q_{tot} = 13.8$ m^3.

Table 2. Nominal Reliability of Results for Different Evaluation Stack Sizes S_{eval}[a]

	S_{eval}							
	2	5	10	15	25	50	Conservative S_{eval}	Restricted S_{eval}
				No Ordering (*Single Well*)				
CMA-ES, median	51.4	70.9	85.8	89.2	92.8	96.4		
CMA-ES, 10%	36.1	59.4	80.9	84.7	91.1	94.5		
CMA-ES, 90%	66.3	79.4	88.7	92.1	95.6	97.9		
DE, median	40.8	70.9	82.0	92.9	95.2	98.3		
DE, 10%	22.1	51.5	76.2	85.0	93.0	94.3		
DE, 90%	58.6	80.6	91.0	96.3	98.0	99.3		
				Stack Ordering (*Single Well*)				
CMA-ES, median	43.2	90.1	99.6	99.9	99.9	100.0	99.9	99.8
CMA-ES, 10%	29.9	67.4	98.2	99.7	99.8	99.9	99.7	99.6
CMA-ES, 90%	75.9	98.5	99.8	100.0	100.0	100.0	100	100
DE, median	59.0	94.1	99.3	99.6	99.8	99.9	99.7	99.4
DE, 10%	26.2	85.9	97.6	99.2	99.5	99.8	99.3	98.1
DE, 90%	84.9	98.8	99.9	99.9	100.0	100.0	99.9	99.9
				No Ordering (*Five Wells*)				
CMA-ES, median	42.1	70.5	82.6	87.4	93.3	96.0		
CMA-ES, 10%	33.8	62.1	76.1	82.1	90.2	94.5		
CMA-ES, 90%	100.0	79.9	88.3	90.7	97.5	97.2		
DE, median	63.1	81.7	94.0	94.6	96.4	98.6		
DE, 10%	34.1	68.0	82.3	87.9	94.0	97.9		
DE, 90%	84.7	90.2	96.9	97.6	98.8	99.6		
				Stack Ordering (*Five Wells*)				
CMA-ES, median	54.5	79.8	96.5	98.9	99.9	100.0	100	100
CMA-ES, 10%	15.2	54.6	87.4	95.5	99.2	99.9	99.9	99.6
CMA-ES, 90%	89.1	89.0	98.6	99.7	100.0	100.0	100	100
DE, median	66.6	93.4	97.9	99.1	99.6	99.8	99.9	99.9
DE, 10%	45.2	79.8	93.2	95.8	97.8	99.3	99.6	99.6
DE, 90%	88.3	97.0	99.1	99.9	99.9	100.0	100	100

[a]Listed are median reliabilities as well as, from these values, ranges for 10% and 90% quantiles.

Apparently, it is beneficial to use multiple wells that share particle capture. However, it can be expected from the findings that more than three wells achieve only slight improvement.

4.2. Optimization Without Ordering of Realizations

[50] First, we cover the outcomes of optimization runs using a fixed number of purely randomly selected realizations without setting up an ordered stack. Each objective function value was obtained for a random set of S_{eval} realizations. These realizations were modeled sequentially until an invalid solution was found. In that case, the objective function value was set to the value obtained for this realization plus a penalty; within this objective function evaluation no further realizations were tested. The underlying idea is that a more exact objective function value of such undesirable candidate solutions is not required, since they will barely influence the evolutionary search. The none-valuation of the remaining realizations in the stack saves computation time: for this example application only about 2/3 of S_{eval} realizations needed to be tested on average, an advantage that is also exploited in the approach where the realizations are ordered.

[51] The only parameter in this nonordered setup is the (evaluation) stack size, which was set to S_{eval} = 2, 5, 10, 15, 25, 50. Each different S_{eval} required 25 randomly initialized optimization runs, both for the CMA-ES and the DE. These were repeated for the single and the five-well case. For each optimized well configuration, the nominal reliability was obtained in a postinspection with the full set of 1000 realizations.

[52] As expected, a general trend could be detected. Increasing the realizations per objective function evaluation raises the reliability of the best solutions obtained, while the 10% and 90% quantiles continuously tighten (Table 2). The comparison of results from the two different optimization algorithms does not yield a clear ranking. For the single-well case, the outcome of the DE appears to be more variable, which is indicated by overall broader quantile ranges. In contrast, the DE mostly outperforms the CMA-ES in median reliability. This could be interpreted as being less sensitive to the significant noise from random replacement of realizations. Apparently, the noise conflicts with the self-adaptive search mechanism of the CMA-ES. Since the CMA-ES tries to approximate the contour lines of the objective function as the search progresses it is less prone to handle very noisy objective functions compared to DE where all evolutionary operators are kept constant throughout the search.

4.3. Optimization With Ordering of Realizations

[53] As shown in Figure 5 and Table 3, reliability increases at the expense of higher median objective function values. Accordingly, random sampling without ordering of realizations results in lowest total pumping rates, q_{tot}. On average, DE detects well configurations with higher reliability, which corresponds to higher values of q_{tot}. If stack ordering is applied, raising the stack size is especially influential up to S_{eval} = 15. This is illustrated by the median convergence curves for application of CMA-ES in Figure 6. These curves also delineate typical trends of convergence curves: initially good fitness values are obtained, because only a few nonrepresentative realizations are inspected. Accordingly, the reliability of such solutions is small. Then

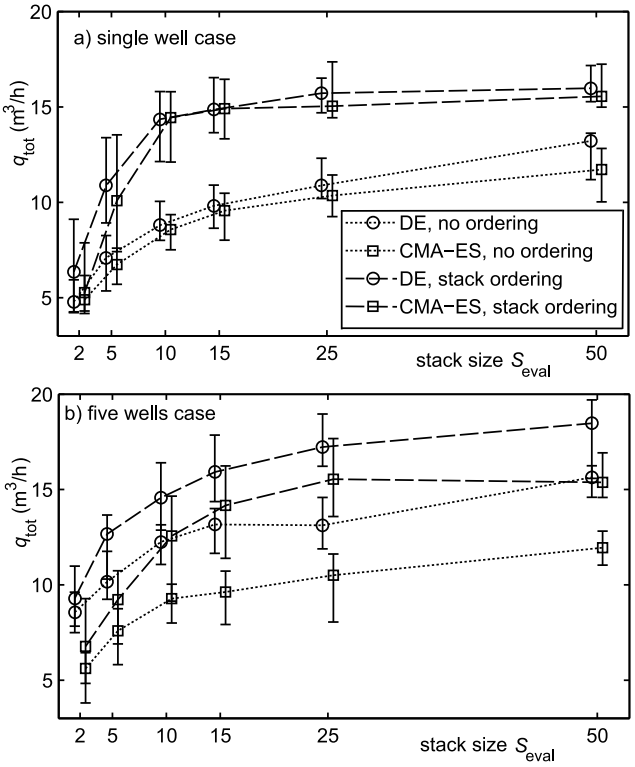

Figure 5. (a, b) Dependency of optimized total pumping rates, q_{tot}, on the evaluation stack sizes, S_{eval}.

the generation-wise best fitness first increases with the rising number of realizations examined. When the stack becomes more mature and therefore stable, the focus of the optimization process shifts from exploratory behavior to a local improvement of the solution.

Table 3. Nominal Reliabilities in Percent for Results From Well-Layout Optimization With Stack Ordering Depending on Stack Size S_{eval} and Decay Factor k[a]

	k \ S_{eval}	2	5	10	15	25	50
		Stack Ordering (*Single Well*)					
CMA-ES	0	43.2	90.1	99.6	**99.9**	**99.9**	**100.0**
	0.01	68.4	97.0	99.7	**99.9**	**99.9**	99.9
	0.05	85.1	98.7	**99.8**	99.8	99.8	99.8
	0.1	85.4	**98.9**	99.6	99.7	99.7	99.5
	0.2	**87.9**	97.9	99.5	99.6	99.6	99.3
DE	0	59.0	94.1	99.3	99.6	**99.8**	**99.9**
	0.01	65.9	95.4	**99.5**	**99.9**	**99.8**	**99.9**
	0.05	70.6	**97.3**	**99.5**	99.7	99.8	99.8
	0.1	**86.9**	95.8	99.1	99.8	**99.8**	99.6
	0.2	76.8	95.3	99.3	99.3	99.4	99.2
		Stack Ordering (*Five Wells*)					
CMA-ES	0	**54.5**	79.8	96.5	98.9	99.9	**100.0**
	0.01	37.6	94.8	**99.8**	**99.9**	**100.0**	**100.0**
	0.05	43.4	**98.4**	99.7	99.8	99.8	99.8
	0.1	46.4	97.8	99.6	99.8	99.7	99.6
	0.2	38.8	97.7	99.4	99.6	99.7	99.4
DE	0	66.6	93.4	97.9	99.1	99.6	**99.8**
	0.01	**79.7**	97.0	98.6	**99.6**	**99.8**	**99.8**
	0.05	75.5	**97.1**	**99.3**	99.4	99.7	**99.8**
	0.1	71.5	96.8	99.1	**99.6**	99.5	99.2
	0.2	72.3	96.0	98.3	99.0	99.7	99.4

[a]Highest reliabilities for each value of S_{eval}, problem case, and solution algorithm are in boldface.

[54] The similarity in performance of DE and CMA-ES in the single-well case is lost in the higher complexity of the five-well case (Figure 5b). The results are more variable and the differences between the evolutionary solvers become more obvious. What we can see is that CMA-ES appears to be slightly more efficient for minimizing q_{tot} (Figure 5b) while reaching high-reliability solutions (Table 3, $k = 0$). Stack ordering apparently reduces noise compared to purely random sampling of realizations and the CMA-ES can exert the advantages of self-adaptation. Median reliabilities are higher for $S_{eval} \geq 10$ (single) and $S_{eval} \geq 25$ (five wells) and converge closer to the desired maximum of 100%. At the same time, the objective function values are mostly smaller, ranging around the expected optimal values. Since an increase of S_{eval} means reduction of noise, the CMA-ES seems to develop its full search capability only above a certain number of realizations per stack.

[55] As an illustrative example, Figure 7 depicts results for the individual optimization runs for the five-well case and $S_{eval} = 25$. The graph shows the ranges of reliability and optimized total pumping rate among the different runs, and the resulting median values. Even if some outliers exist, most of the best well layouts found group closely around the median. Nevertheless, the spreading also indicates the existence of several close-optimal solutions. Hence, the optimization algorithms are hardly able to detect exactly the same solution within each independent optimization run.

4.4. Optimization With Ordering and Decay

[56] Repeatedly multiplying a decay factor to the credits of realizations should increase the probability of sampling realizations that have not been tested so far. It also represents an adaptive mechanism, as it puts less emphasis on "old" credits and thereby accentuates the current search: as the optimization algorithm explores different parts of the decision space until it converges, realizations that appear critical in the beginning may not be relevant anymore in the later stage of the search.

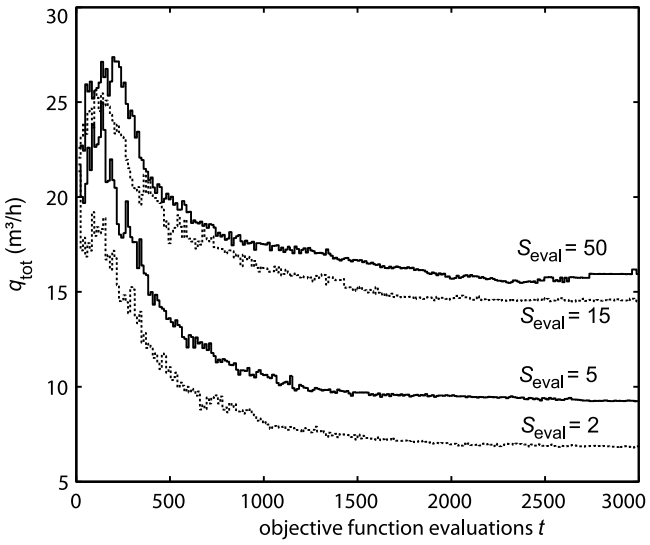

Figure 6. Median convergence curves (best fitness per generation) for applying CMA-ES to the five-well case with stack ordering (no decay, $k = 0$).

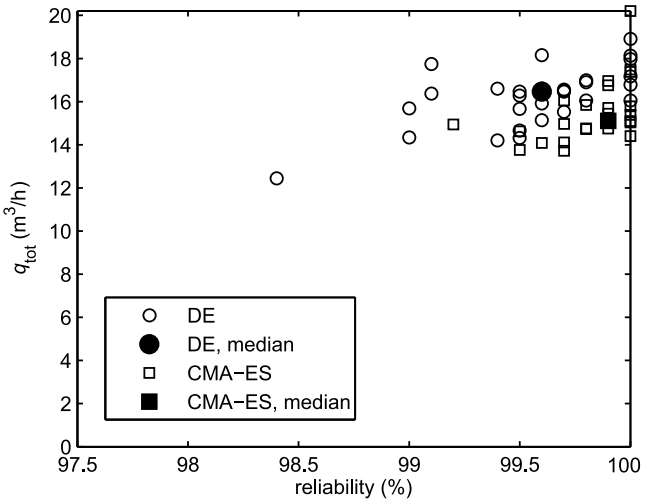

Figure 7. Total pumping rate versus reliability for 25 optimization runs to solve the five-well case; this example shows the results for an evaluation stack size of S_{eval} = 25 and no decay (k = 0).

[57] Table 3 lists the achieved median reliabilities for the two well cases for decay factors of k = 0, 0.01, 0.05, 0.1, and 0.2. The variant with k = 0 is equivalent to the implementation without decay as discussed above. A general observation is that the use of a small decay factor is especially beneficial for small stack sizes. Roughly speaking, for $S_{eval} \leq$ 10 (single) and $S_{eval} \leq 25$ (five wells) this results in a higher reliability. These values for S_{eval} appear to be case-specific thresholds. If properly sampled and ranked, there is no need for a higher number of realizations to maximize reliability while minimizing the pumping rates. These are very small numbers compared to the total of 1000 realizations considered for a full reliability determination.

[58] Including a decay factor enhances the replacement of realizations that would otherwise pile up in the evaluation stack and render it stable rather than dynamic. Hence, the decay factor k can also be used to further decrease the required S_{eval}. Generally, k seems to be most sensitive for small evaluation stack sizes. However, there is no clear indication which value of k is best. For the application with CMA-ES, an evaluation stack size of S_{eval} = 2 is not sufficient and only delivers low-reliable solutions. Above this value, reliabilities are mostly higher than 90% and even over 94% when employing the decay factor. The higher the $S_{eval,}$ the lower the ideal value of k. For very high S_{eval}, a decay factor even compromises the search and results in unnecessary noise. Too large values of k (e.g., k = 0.2) always generate too much variation in the evaluation stack and counteract the ordering procedure.

[59] For further insight, Figure 8 depicts the median number of credited realizations after a full optimization run with the CMA-ES. Figure 8 is similar for DE, with slightly higher numbers of credited realizations for large evaluation stacks. It illustrates the effect of increasing the decay factor for different values of S_{eval}. For the single-well case, the numbers of credited realizations tend to stagnate at a value of 10. Once more this indicates that, independent of the evaluation stack size, S_{eval}, a small number of properly selected realizations is sufficient to constrain the search within the valid search space. If k is higher than 0.05, more realizations are credited but without a positive effect for the optimization results (see Table 3). The five-well case does not deliver similarly consistent results. However, when compared with the high-reliability configurations in Table 3, the number of credited realizations seems to be ideal for values between 25 and 30 for this case. Apparently, the higher values of S_{eval} required when raising the number of wells are due to the increase in problem dimension.

[60] All in all, DE is only preferable for very small stack sizes. If reliabilities higher than 95% are sought after, CMA-ES outperforms DE in terms of median pumping rates, again indicating a slightly better efficiency in case of merely moderate noise. The ideal values of k seem to be comparable. As a rough rule of thumb, a recommended value for the decay factor can be determined by $k = 0.05 - S_{eval}/1000$ for $S_{eval} \leq 50$ for this example problem.

4.5. Optimization With Dynamic Stack Size Adjustment

[61] The effect of a dynamically adjusted stack size is examined for both well cases, starting with an initial value of S_{eval} = 1. In accordance with the study on fixed stack sizes, each algorithm implementation, DE or CMA-ES, single-well or five-well case, and conservative or restricted stack size was examined by 25 randomly initialized optimization runs. No ordering has been found to be unsuitable for fixed stack sizes and thus is not further studied. Aside from this, in particular for the conservatively adapted stack

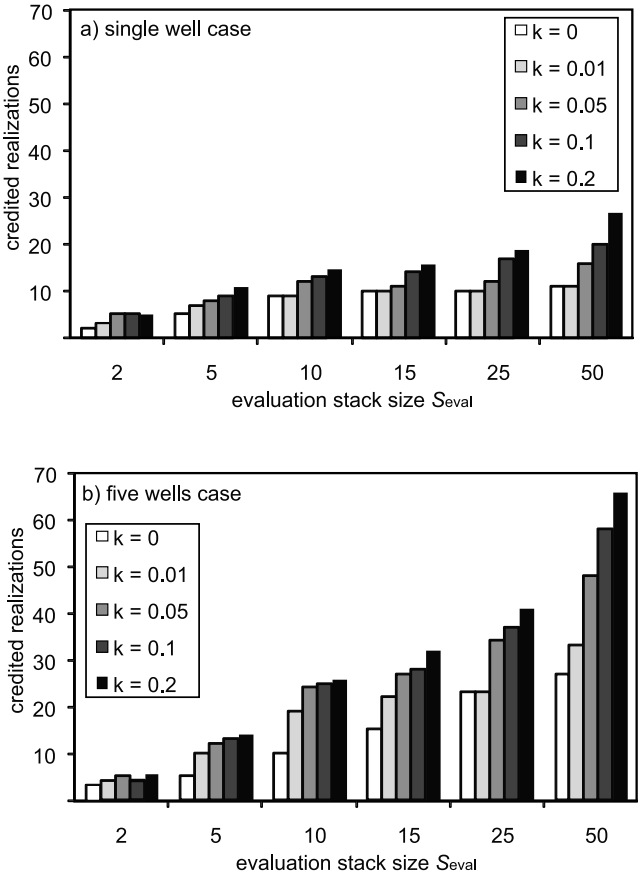

Figure 8. (a, b) Average number of credited realizations after 25 optimization runs (CMA-ES).

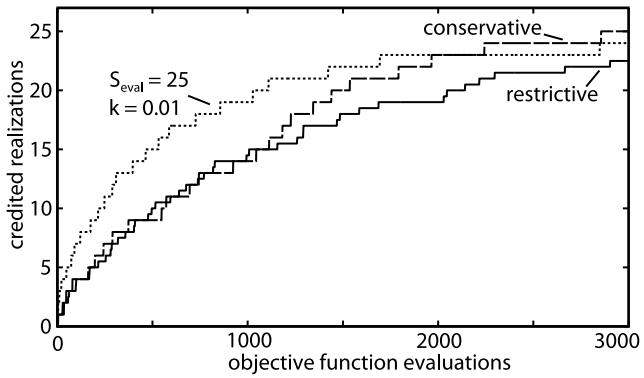

Figure 9. Median increase of number of credited realizations with objective function evaluations (five-well case) for fixed stack size (S_{eval} = 25) and dynamic stack size with restrictive and conservative setting.

size, the huge number of credited realizations would render such a procedure computationally relatively inefficient.

[62] Figure 9 demonstrates for the five-well case that the increase of credited realizations is comparable to a favorable fixed stack size setting with (S_{eval} = 25, k = 0.01). Owing to the smaller initial stack sizes, credited realizations are fewer in the initial search phase. In the long run, independent of the implementation, all variants converge to a number between 20 and 25. This indicates the adaptive potential of the dynamic stack size adjustment. Figure 10 demonstrates that the associated stack sizes have different trends. The conservative implementation effects a continuous increase, the restrictive works with smaller stack sizes that can even drop and thus cause a temporal up-down mode.

[63] Of major interest is the reliability of the best solutions found. As listed in Table 2 for the single-well case, the bigger stack sizes of the conservative method produce better results. This trend can be expected already from the findings of the previous analysis with different fixed stack sizes. Median nominal reliabilities increase from 99.8% to 99.9% for the CMA-ES and from 99.4% to 99.7% for DE. However, for the five-well case, this trend can hardly been seen. Also, they are (nearly) maximal, with 100% achieved with CMA-ES and 99.9% with DE. This indicates that, independent of the method used, a sufficient number of realizations is inspected to achieve high-reliable solutions. Apparently, a major role is played by the higher number of model runs for the five-well case: the evaluation stack evolves during 3000 objective function evaluations in comparison to only 700 for the single-well problem of this study. Since the dynamic stack size is an adaptive approach, it appears in particular robust for high-dimensional problems with a significant number of sequential objective function evaluations needed. This is not only confirmed by the small differences between two dynamic stack size variants, but also by the small quantile ranges in Table 2.

[64] The better results with CMA-ES in comparison with DE denote slightly better convergence properties, although they seem negligible for the five-well case. Probably by increasing the number of objective function evaluations per optimization run, even more similar results would be obtained. This is also indicated by the achieved objective function values, which are on average less optimal and higher with the DE: an observation already made for high fixed stack sizes (see Figure 5). Therefore further discussion of the exact numbers is not presented.

4.6. Computational Savings

[65] A major incentive for sampling only a small number of realizations per objective function evaluation is to keep the computational effort low or to find an approximate solution to a problem that cannot be solved within a reasonable computation time at all. Independent of the sampling procedure, random or probabilistic, only 2/3 of the evaluation stack size is evaluated on average. As mentioned above, this is achieved by stopping further testing as soon as a realization of the stack denotes constraint violation. This means, for example, that for a single-well case with 700 objective function evaluations and S_{eval} = 15, about 2/3 x 15 x 700 = 7000 model runs are required per optimization run (compared to 15 x 700 = 10,500 model runs otherwise).

[66] Figure 11 illustrates the trade-offs between nominal reliabilities of optimized solutions and the model runs necessary for each algorithm configuration considered. Both well cases and outcomes from random as well as probabilistic sampling are shown. As a representative setting, we chose to pinpoint only the stack-ordering variant with a small decay factor of k = 0.01 and fixed S_{eval}. The graphs in Figure 11 reflect the better performance with DE at small stack sizes and for random sampling, whereas the advantages from CMA-ES lie in the achievement of highly reliable solutions (see Table 3).

[67] Of major importance is the high reliability that can be achieved with both algorithms even for small fixed stack sizes. For reliabilities very close to 100%, about 4500 model runs are needed for the single-well case (S_{eval} = 10), and about 20,000 model runs for the five-well case. In contrast to the number of model runs required for a full enumeration of the 1000 realizations for each objective function evaluation (700,000 and 3.0 Mio. per full optimization run), this means a saving of > 99.3% for both well cases. This figure is enormous and reflects the efficiency of stack ordering. An underlying assumption is that the objective function noise introduced by only considering small subsets of realizations does not substantially influence the optimization algorithm performance. Also compared with the results for random sampling of realizations, the number of model runs is rather

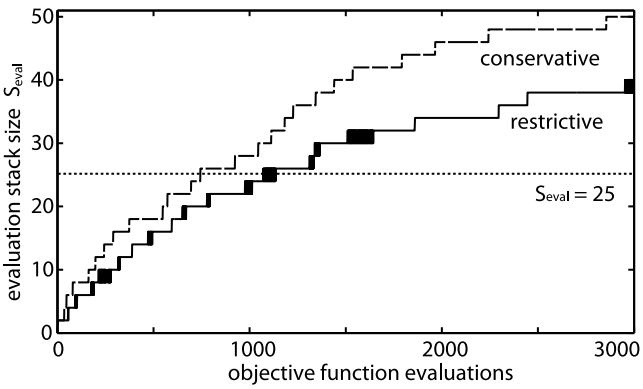

Figure 10. Median evaluation stack size S_{eval} change for dynamic implementations (see Figure 9).

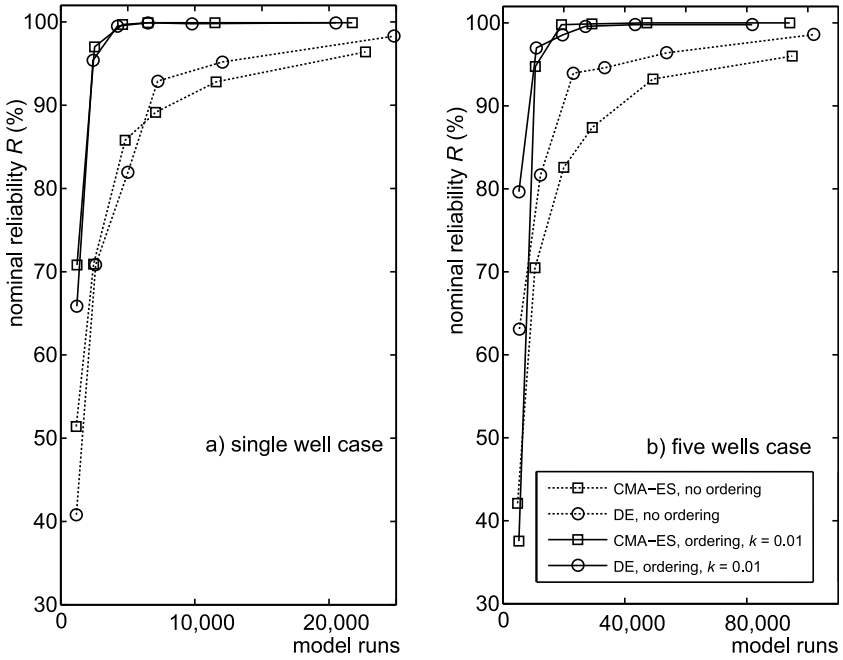

Figure 11. (a, b) Required number of model runs for different implementations and evolutionary algorithms to achieve certain reliability. These trends stem from calculating the median from optimization runs with S_{eval} = 2, 5, 10, 15, 25, and 50.

small. Without ordering, high-reliability solutions are not obtained for the range of S_{eval} (≤50) considered. A rough extrapolation of the trade-offs in Figure 11 points toward a number of model runs that would be about 1 order of magnitude higher to achieve a solution of close to 100% reliability.

[68] The advantage of a dynamic stack size is that in the beginning of the search only a small number of realizations are inspected. Depending on the update rule (equations (20) and (21)), the number of credited realizations is increased and the stack size can substantially grow. Thus the early savings are outmatched by the rather big stack sizes at the later phase of the search. This is also reflected in the relatively high computational burden for the five-well case. In comparison to the single-well case, the given number of function evaluations per optimization run is more than four times higher (3000 versus 700) and hence the evaluation stack size becomes much larger. On the average, we obtain savings of 99.3% for the restrictive and 99.5% savings for the single-well case. This is better than for the best fixed stack size implementation, but therefore the achieved reliabilities are not maximal (see Table 2). This is different for the five-well case. 100% reliability is achieved, but on the expense of relatively high number of model runs: for the conservative variant, this is 77,000 per optimization run (97.5% savings) and for the restrictive 53,000 (98.2%).

4.7. Validation With Other Case Studies

[69] Of major interest is to what extent the findings in the Lauswiesen case and the derived control parameter settings can be generalized. To test the transferability of the results and to examine the performance of stack ordering to alternative problems, four completely different ensembles of conductivity realizations (cases A, B, C, and D; see Figure 12) are generated.

[70] For the ensemble of case A the variance of the ln-conductivity field was increased to 4.5, a value reported for the MADE site [e.g., *Rehfeldt et al.*, 1992] and characteristic of a highly heterogeneous aquifer. The correlation lengths, as well as the number and positions of conditioning points (40 conductivity values and 28 head values) were kept the same as in the Lauswiesen case. The ensemble of A represents a case of much more pronounced features like high- and low-conductivity zones compared to the Lauswiesen case.

[71] For case B (and the subsequent cases C and D) the number of conditioning points was significantly reduced. A total of only 9 measurement points conditioned on both hydraulic conductivity and hydraulic head (x coordinates: 25, 65, 105; y coordinates: 30, 50, 70) were chosen to keep the nonstationarity of conditioned random fields and at the same time allow for a much higher variability in hydraulic conductivity between conditioning points. The variance of the ln-conductivity field was kept at 2.91 for case B. Nevertheless, in terms of the objective function B represents a more noisy case compared to the Lauswiesen case owing to the reduction of conditioning points.

[72] Case C combines the higher variance of case A and the reduced number of conditioning points of case B. This translates to an even more noisy objective function, as individual high- and low-conductivity zones are given by larger conductivity contrasts and are only minorly pinned to conditioning points. Finally, the size of heterogeneous aquifer features was also increased by setting the isotropic correlation lengths to 20 m for case D (variance and conditioning points as in C).

[73] For each of the cases A, B, C, and D a new true conductivity field had to be chosen owing to the changes in

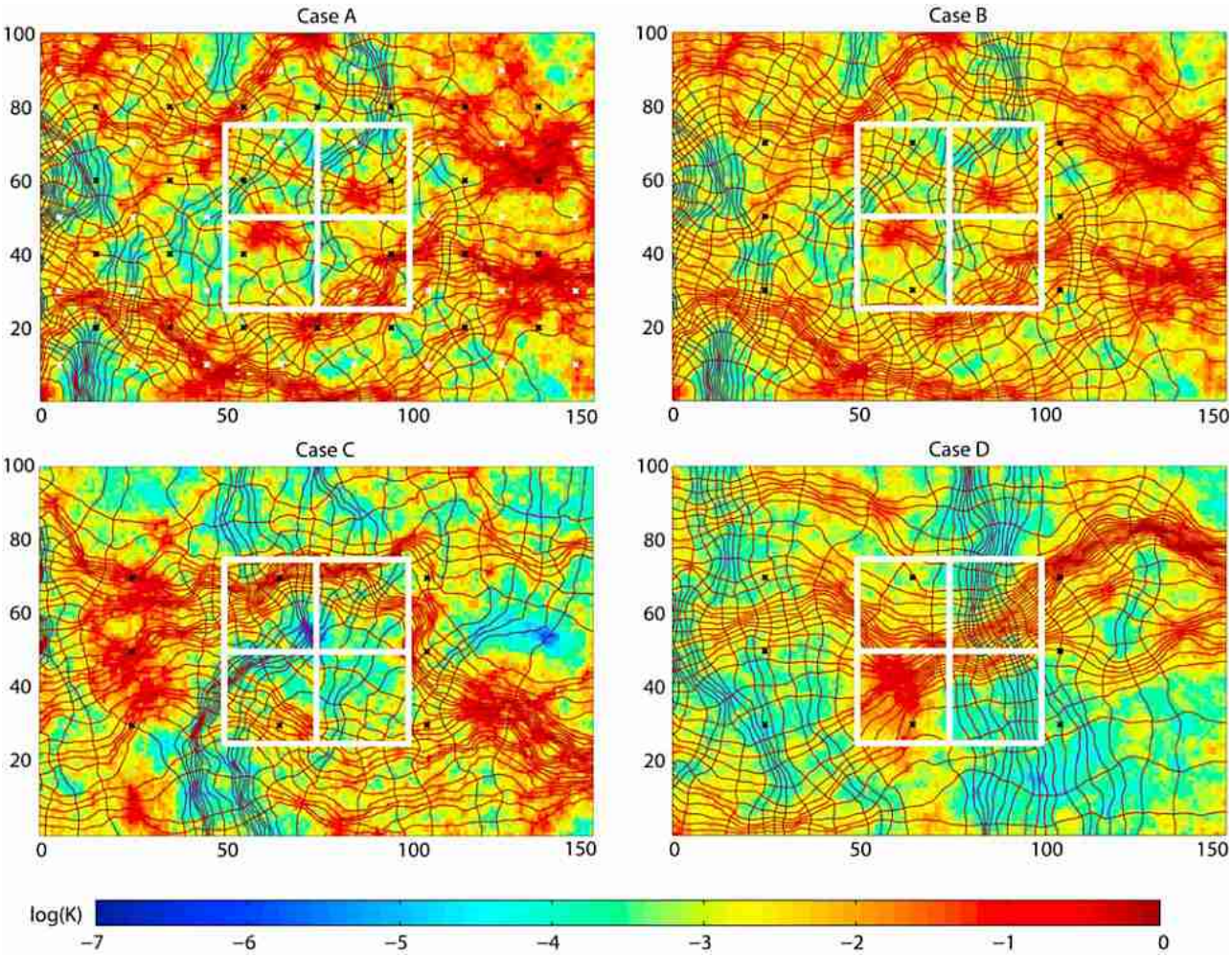

Figure 12. Hydraulic conductivity distribution maps of four synthetic validation cases with N-S oriented head contours and W-E oriented hypothetical pathlines. Crosses denote conditioning points. (For case A, black crosses denote conductivity, and white crosses denote head. For cases B, C, and D, black crosses denote head and conductivity.)

geostatistical parameters. In order to keep the features of the true fields alike for the cases A and B, the same seed was chosen for the random path. For cases C and D the seed for the random generator was also altered. This easily becomes evident in Figure 12. Note, however, that the random path was only fixed for the generation of the true fields for cases A and B. For the ensembles the random generator seeds were different.

[74] To keep the scope of this work in a reasonable shape, the same conceptual groundwater model is used and identical optimization criteria are defined. Again, the objective is to minimize the pumping rate of a single-well and a five-well case while achieving capture of a contaminant plume that originates from the same source as depicted in Figure 1. We also set the same specifications for the optimization algorithms that are repeatedly applied for each problem. Only those specifications are used that revealed to be favorable from the Lauswiesen study: a fixed stack size of S_{eval} = 15 for the single-well case, and S_{eval} = 25 for the five-well case, with decay factors of 0.035 and 0.025 ($k = 0.05 \cdot S_{eval}/1000$), respectively. Additionally, (restrictive) dynamic stack sizes are implemented without decay. Since it is the algorithms' average performance that it is of interest and not the statistical analysis, only 10 randomly initialized optimization runs are conducted per individual ensemble. Again, median values for nominal reliability of optimized solutions and related computational effort are investigated.

[75] Tables 4a and 4b summarize the results with respect to average nominal reliability of results and the related computational savings. Overall, the numbers clearly demonstrate a very similar performance of the optimization algorithms. Even if conductivity distributions are changed, higher heterogeneity is considered and less conditioning points are used, especially the dynamic stack size coupled with CMA-ES is very efficient. Reliabilities of optimal solutions consistently reach values close to or even of 100%, and computational savings are -on the average - slightly higher than those calculated for the Lauswiesen site. For example, a total number of 4122 model runs for solving the single-well case and of 46,128 for the five-well case are

Table 4a. Results From Validation With Four Problem Cases of Different Hydraulic Conductivity Distribution for the Single-Well Case\ S_{eval} [a]

	Median Nominal Reliability (%)				Median Number of Model Runs			
	CMA-ES, 15	CMA-ES, Dynamic	DE, 15	DE, Dynamic	CMA-ES, 15	CMA-ES, Dynamic	DE, 15	DE, Dynamic
Lauswiesen, single well	99.9	99.8	99.9	99.4	5838	4347	6363	3269
Problem A, single well	99.9	99.8	100	99.5	6468	3927	5635	3199
Problem B, single well	99.8	99.8	99.8	99.0	6314	3934	6531	3129
Problem C, single well	99.8	99.9	99.3	99.6	6356	4900	5796	3458
Problem D, single well	99.8	99.9	99.9	99.8	6398	3500	5222	2674
Mean	99.8	99.8	99.8	99.5	6275	4122	5909	3146

[a]Dynamic stack size with restrictive implementation.

computed, which means only about 6 and 15 model runs per objective function evaluation.

5. Summary and Conclusions

[76] In this study, we have further explored the stack-ordering concept for finding highly reliable solutions to complex model-based hydrological problems. This method works well when a large number of model alternatives or parameter field realizations are used to quantify the uncertainty inherent to the problem simulation. When an iterative optimization procedure is applied, evaluation of the entire set of model variants would hardly be possible, as one would easily end up with millions of model simulations. Instead, only a small fraction of realizations is used to approximate objective function values of candidate solutions to the optimization problem. The presented approach benefits from the noise robustness of population-based heuristic optimization algorithms such as differential evolution (DE) and evolution strategy (CMA-ES). These represent promising candidates for the family of evolutionary algorithms, which up to now supplies the most common algorithms for combined simulation-optimization problems in hydro(geo)logy.

[77] As a basic scenario, we selected a moving well optimization problem generated from parameter values obtained from field observations at the Lauswiesen site close to Tübingen in Germany. It is formulated as an advective control problem that involves the adjustment of both well positions and extraction rates of a single well or a gallery of five wells. The objective is the minimization of the total extraction rate, which is a typical task in the design of pump-and-treat systems in practice. The uncertainty in the groundwater model prediction is considered to be due to the variability of the spatially distributed hydraulic conductivity. We assumed a hypothetical situation with a limited number of borehole measurements of aquifer properties and hydraulic head. These measurements were then employed for the conditioning of 1000 equally probable conductivity fields.

[78] The optimization of the complex moving-well problems is carried out repeatedly for obtaining statistically representative results. Of major interest are optimality, reliability and computational demand. The best solution per each optimization run is examined in a postaudit with the entire ensemble of realizations to determine its reliability. We compared random sampling of constant, small sets of realizations to our probabilistic approach and identified substantial improvements: random sampling hardly delivers high-reliability solutions, even if the evaluation stack size is large (>25). In contrast, by stack ordering, reliabilities converge to the maximum of 100% already at small stack sizes of 10–15, while the identified well configurations are close to the optimal solutions. However, this number appears to be problem-specific, and rises with problem dimensionality. A small decay factor that forces sampling of new realizations to the evaluation stack is recommendable for small stack sizes. For the example application of this study, we found that - if included at all - the decay factor should decrease with increasing evaluation stack size. The presented empirical relationship can serve as first reference for other applications.

[79] In general, the presented procedure is shown to be very robust for a broad range of the two controlling parameters, the evaluation stack size and the decay factor. If a very small stack size does not yield a satisfactory solution, it may be increased in a subsequent optimization run. This sounds like an inconvenience for a couple of unfavorable trial runs, but please note that the computational demand for trials with small stack sizes is very low. For the well-layout problem of this study, we could demonstrate that even subideal settings for stack ordering yield well-optimized and highly reliable solutions. The computational savings were in a range of values above 99%. This number of course depends on the total number of realizations considered, and

Table 4b. Results From Validation With Four Problem Cases of Different Hydraulic Conductivity Distribution for the Five-Well Case\ S_{eval} [a]

	Median Nominal Reliability (%)				Median Number of Model Runs			
	CMA-ES, 25	CMA-ES, Dynamic	DE, 25	DE, Dynamic	CMA-ES, 25	CMA-ES, Dynamic	DE, 25	DE, Dynamic
Lauswiesen, five wells	100	100	99.8	99.9	46,860	60,570	43,170	46,140
Problem A, five wells	99.8	100	99.6	99.9	57,180	40,620	61,740	36,660
Problem B, five wells	99.9	100	99.9	99.9	58,260	45,060	55,920	44,220
Problem C, five wells	99.8	100	98.7	99.9	56,280	44,880	54,870	37,200
Problem D, five wells	99.8	100	99.6	99.7	57,600	39,510	66,570	25,260
Mean	99.9	100	99.5	99.9	55,236	46,128	56,454	37,896

[a]Dynamic stack size with restrictive implementation.

in other cases a much smaller number than 1000 realizations may be representative. However, assuming same performance of the solution procedure, computational savings still would be around 90% when, for example, only 100 realizations are selected. This makes even complex multiple-realization problems tractable on personal computers. Not the total number of model realizations but their variability determines the required effort.

[80] As an alternative to a priori fixed evaluation stack size, also a straightforward dynamic stack size adjustment is presented. Its major advantage is that is quasi parameter-free procedure, for which two different variants are suggested, a restrictive and a conservative. Both perform well in particular for higher-dimensional problems such as the examined five-well case. Computational savings for the study site are still in the order of 98% and higher.

[81] For the development of the method, a moving well problem was selected and comprehensively studied. In order to assess the general validity of the results, alternative optimization problems are set up and solved by the derived method specification. The results of this test confirm a robust applicability of stack ordering to related problems, but also they indicate its potential for totally different complex reliability-based problems of water management. In fact, we selected this well placement problem because it arises commonly in practice and is usually of high mathematical complexity owing to the spatial dependency of the hydraulic effect on the individual wells. Furthermore, multiple well problems are hardly separable and in most cases nonconvex. It can be expected that stack ordering is also effective, if further nonlinearities are introduced, for example by badly behaved cost functions or time dependencies. A major advantage is that the presented procedure does not make any assumptions about the source of uncertainty.

[82] One clear-cut result is that stack ordering can be applied with different optimization algorithms. As revealed in this study, a high noise level can represent an obstacle for the self-adaptive CMA-ES: however, to a lesser extent to the static search mechanism of the DE. In contrast, already for moderate stack sizes (>5) the CMA-ES provided slightly better results in terms of both optimality and reliability. This may be interpreted as a better performance of the evolution strategies in case of small (or no) noise.

[83] **Acknowledgments.** This study was funded by the German Research Foundation (Deutsche Forschungsgesellschaft, DFG, contract BA 2850/1-2) and supported by the GW-LCA project within the 7th framework program (contract PIEF-GA-2008-220620) as well as the 6th framework program AquaTerra (contract GOCE 505428). The authors wish to thank three anonymous reviewers for their constructive comments. Thanks to Margaret Hass for her assistance in preparing the manuscript.

References

Ahlfeld, D. P., and Y. Hoque (2008), Impact of simulation model solver performance on groundwater management problems, *Ground Water*, *46*(5), 716–726, doi:10.1111/j.1745-6584.2008.00454.x.

Ascough, J. C., II, H. R. Maier, J. K. Ravalico, and M. W. Strudley (2008), Future research challenges for incorporation of uncertainty in environmental and ecological decision-making, *Ecol. Modell.*, *219*(3–4), 383–399, doi:10.1016/j.ecolmodel.2008.07.015.

Bayer, P., and M. Finkel (2004), Evolutionary algorithms for the optimization of advective control of contaminated aquifer zones, *Water Resour. Res.*, *40*, W06506, doi:10.1029/2003WR002675.

Bayer, P., and M. Finkel (2007), Optimization of concentration control by CMA-ES: Formulation, application, and assessment of remedial solutions, *Water Resour. Res.*, *43*, W02410, doi:10.1029/2005WR004753.

Bayer, P., C. M. Bürger, and M. Finkel (2008), Computationally efficient stochastic optimization using multiple realizations, *Adv. Water Resour.*, *31*(2), 399–417, doi:10.1016/j.advwatres.2007.09.004.

Bayer, P., M. Finkel, and G. Teutsch (2005), Cost-optimal contaminant plume management with a combination of pump-and-treat and physical barrier systems, *Ground Water Monit. Rem.*, *25*(2), 96–106.

Bayer, P., E. Duran, R. Baumann, and M. Finkel (2009), Optimized groundwater drawdown in a subsiding urban mining area, *J. Hydrol. Amsterdam*, *365*(1–2), 95–104, doi:10.1016/j.jhydrol.2008.11.028.

Becker, D., B. Minsker, R. Greenwald, Y. Zhang, K. Harre, K. Yager, C. Zheng, and R. Peralta (2006), Reducing long-term remedial costs by transport modeling optimization, *Ground Water*, *44*(6), 864–875, doi:10.1111/j.1745-6584.2006.00242.x.

Bürger, C., P. Bayer, and M. Finkel (2007), Algorithmic funnel-and-gate system design optimization, *Water Resour. Res.*, *43*, W08426, doi:10.1029/2006WR005058.

Chadalavada, S., and B. Datta (2008), Dynamic optimal monitoring network design for transient transport of pollutants in groundwater aquifers, *Water Resour. Manage.*, *22*(6), 651–670, doi:10.1007/s11269-007-9184-x.

Chang, L. C., H. J. Chu, and C. T. Hsiao (2007), Optimal planning of a dynamic pump-treat-inject groundwater remediation system, *J. Hydrol.*, *342*(3–4), 295–304.

Cui, L., and G. Kuczera (2005), Optimizing water supply headworks operating rules under stochastic inputs: Assessment of genetic algorithm performance, *Water Resour. Res.*, *41*, W05016, doi:10.1029/2004WR003517.

De Jong, K. A. (2006), *Evolutionary Computation: A Unified Approach*, MIT Press, Cambridge, Mass.

Espinoza, F. P., and B. S. Minsker (2006), Development of the enhanced self-adaptive hybrid genetic algorithm (e-SAHGA), *Water Resour. Res.*, *42*, W08501, doi:10.1029/2005WR004221.

Franssen, H. J. H., and W. Kinzelbach (2009), Ensemble Kalman filtering versus sequential self-calibration for inverse modelling of dynamic groundwater flow systems, *J. Hydrol. Amsterdam*, *365*(3–4), 261–274, doi:10.1016/j.jhydrol.2008.11.033.

Guo, X., C.-M. Zhang, and J. C. Borthwick (2007), Successive equimarginal approach for optimal design of a pump and treat system, *Water Resour. Res.*, *43*, W08416, doi:10.1029/2006WR004947.

Gutjahr, A., B. Bullard, S. Hatch, and L. Hughson (1994), Joint conditional simulations and the spectral method approach for flow modeling, *Stochastic Hydrol. Hydraul.*, *8*(1), 79–108, doi:10.1007/BF01581391.

Hansen, N. (2008), The CMA evolution strategy: A tutorial, technical report, Tech. Univ. Berlin, Berlin. (Available at http://www.bionik.tu-berlin.de/user/niko/cmatutorial.pdf)

Hansen, N., and A. Ostermeier (2001), Completely derandomized self-adaptation in evolution strategies, *Evol. Comput.*, *9*(2), 159–195, doi:10.1162/106365601750190398.

Hansen, N., S. D. Müller, and P. Koumoutsakos (2003), Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES), *Evol. Comput.*, *11*(1), 1–18, doi:10.1162/106365603321828970.

Harbaugh, A. W., E. R. Banta, M. C. Hill, and M. G. McDonald (2000), MODFLOW-2000, the U.S. Geological Survey modular ground-water model, User's guide to the modularization concepts and the ground-water flow process, *U.S. Geol. Surv. Open File Rep.*, *00-92*, 121 pp.

Hsiao, C. T., and L. C. Chang (2005), Optimizing remediation for an unconfined aquifer using a hybrid algorithm, *Ground Water*, *43*(6), 904–915.

Hu, Z., C. W. Chan, and G. H. Huang (2007), Multi-objective optimization for process control of the in-situ bioremediation system under uncertainty, *Eng. Appl. Artif. Intel.*, *20*(2), 225–237, doi:10.1016/j.engappai.2006.06.008.

Huang, C., and A. S. Mayer (1997), Pump-and-treat optimization using well locations and pumping rates as decision variables, *Water Resour. Res.*, *33*(5), 1001–1012, doi:10.1029/97WR00366.

Karterakis, M. S., G. P. Karatzas, I. K. Nikolos, and M. P. Papadopoulou (2007), Application of linear programming and differential evolutionary optimization methodologies for the solution of coastal subsurface water management problems subject to environmental criteria, *J. Hydrol. Amsterdam*, *342*(3–4), 270–282, doi:10.1016/j.jhydrol.2007.05.027.

Kirsch, B. R., G. W. Characklis, K. E. M. Dillard, and C. T. Kelley (2009), More efficient optimization of long-term water supply portfolios, *Water Resour. Res.*, *45*, W03414, doi:10.1029/2008WR007018.

Kitanidis, P. (1995), Quasi-linear geostatistical theory for inversing, *Water Resour. Res.*, *31*(10), 2411–2419, doi:10.1029/95WR01945.

Ko, N. Y., and K. K. Lee (2008), Reliability and remediation cost of optimal remediation design considering uncertainty in aquifer parameters, *J. Water Resour. Plann. Manage.*, *134*(5), 413–421, doi:10.1061/(ASCE)0733-9496(2008)134:5(413).

Kourakos, G., and A. Mantoglou (2008), Remediation of heterogeneous aquifers based on multiobjective optimization and adaptive determination of critical realizations, *Water Resour. Res.*, *44*, W12408, doi:10.1029/2008WR007108.

Mantoglou, A., and G. Kourakos (2007), Optimal groundwater remediation under uncertainty using multi-objective optimization, *Water Resour. Manage.*, *21*(5), 835–847, doi:10.1007/s11269-006-9109-0.

Mayer, A. S., T. Kelley, and C. T. Miller (2002), Optimal design for problems involving flow and transport phenomena in subsurface systems, *Adv. Water Resour.*, *25*(8–12), 1233–1256.

Meyer, P. D., and E. D. Brill Jr. (1988), A method for locating wells in a groundwater monitoring network under conditions of uncertainty, *Water Resour. Res.*, *24*(8), 1277–1282, doi:10.1029/WR024i008p01277.

Morgan, D. R., J. W. Eheart, and A. J. Valocchi (1993), Aquifer remediation design under uncertainty using a new chance constrained programming technique, *Water Resour. Res.*, *29*(3), 551–561, doi:10.1029/92WR02130.

Mulligan, A. E., and D. P. Ahlfeld (1999), Advective control of groundwater contaminant plumes: Model development and comparison to hydraulic control, *Water Resour. Res.*, *35*(8), 2285–2294, doi:10.1029/1999WR900106.

Ng, T. L., and J. W. Eheart (2008), A multiple-realizations chance-constrained model for optimizing nutrient removal in constructed wetlands, *Water Resour. Res.*, *44*, W04405, doi:10.1029/2007WR006126.

Nowak, W., and O. A. Cirpka (2004), A modified Levenberg-Marquardt algorithm for quasi-linear geostatistical inversing, *Adv. Water Resour.*, *27*(7), 737–750, doi:10.1016/j.advwatres.2004.03.004.

Nunes, L. M., E. Paralta, M. C. Cunha, and L. Ribeiro (2004), Groundwater nitrate monitoring network optimization with missing data, *Water Resour. Res.*, *40*, W02406, doi:10.1029/2003WR002469.

Papadopoulou, M. P., G. F. Pinder, and G. P. Karatzas (2003), Enhancement of the outer approximation method for the solution of concentration-constrained optimal-design groundwater-remediation problems, *Water Resour. Res.*, 39(7), 1185, doi:10.1029/2002WR001541.

Pollock, D. W. (1994), User's guide for MODPATH/MODPATH-PLOT, Version 3: A particle tracking post-processing package for MODFLOW, the U.S. Geological Survey finite-difference ground-water flow model, *U. S. Geol. Surv. Open File Rep.*, 94–464.

Price, K. V., R. M. Storn, and J. A. Lampinen (2005), *Differential Evolution: A Practical Approach to Global Optimization*, 538 pp., Springer, New York.

Rehfeldt, K. R., J. M. Boggs, and L. W. Gelhar (1992), Field study of dispersion in a heterogeneous aquifer: 3. Geostatistical analysis of hydraulic conductivity, *Water Resour. Res.*, *28*(12), 3309–3324, doi:10.1029/92WR01758.

Ricciardi, K. L., G. F. Pinder, and G. P. Karatzas (2007), Efficient groundwater remediation system design subject to uncertainty using robust optimization, *J. Water Resour. Plann. Manage.*, *133*(3), 253–263, doi:10.1061/(ASCE)0733-9496(2007)133:3(253).

Riva, M., L. Guadagnini, A. Guadagnini, T. Ptak, and E. Martac (2006), Probabilistic study of well capture zones distribution at the Lauswiesen field site, *J. Contam. Hydrol.*, *88*(1–2), 92–118, doi:10.1016/j.jconhyd.2006.06.005.

Sack-Kühner, B. (1996), Einrichtung des Naturmessfeldes "Lauswiesen Tübingen", Erkundung der hydraulischen Eigenschaften, Charakterisierung der Untergrundheterogenität und Vergleich der Ergebnisse unterschiedlicher Erkundungsverfahren, diploma thesis, Geol. Inst., Univ. of Tübingen, Germany.

Schütze, N., and G. H. Schmitz (2009), A new stochastic framework based on evolutionary algorithms for evaluating potential yield and corresponding risks of optimal deficit irrigation strategies, in *Hydroinformatics in Hydrology, Hydrogeology and Water, IAHS Publ.*, *331*, 472–479.

Singh, A., and B. S. Minsker (2008), Uncertainty-based multiobjective optimization of groundwater remediation design, *Water Resour. Res.*, *44*, W02404, doi:10.1029/2005WR004436.

Smalley, J. B., B. S. Minsker, and D. E. Goldberg (2000), Risk-based in situ bioremediation design using a noisy genetic algorithm, *Water Resour. Res.*, *36*(10), 3043–3052, doi:10.1029/2000WR900191.

Storn, R., and K. Price (1995), Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces, *Tech. Rep. TR-95-012*, Int. Comput. Sci. Inst., Berkeley, Calif.

Tilmant, A., and R. Kelman (2007), A stochastic approach to analyze trade-offs and risks associated with large-scale water resources systems, *Water Resour. Res.*, *43*, W06425, doi:10.1029/2006WR005094.

Vasquez, J. A., H. R. Maier, B. J. Lence, B. A. Tolson, and R. O. Foschi (2000), Achieving water quality system reliability using genetic algorithms, *J. Environ. Eng.*, *126*(10), 954–962, doi:10.1061/(ASCE)0733-9372(2000)126:10(954).

Wagner, B. J., and S. M. Gorelick (1989), Reliable aquifer remediation in the presence of spatially variable hydraulic conductivity: From data to design, *Water Resour. Res.*, *25*(10), 2211–2225, doi:10.1029/WR025i010p02211.

Wu, J., C. Zheng, C. C. Chien, and L. Zheng (2006), A comparative study of Monte Carlo simple genetic algorithm and noisy genetic algorithm for cost-effective sampling network design under uncertainty, *Adv. Water Resour.*, *29*(6), 899–911, doi:10.1016/j.advwatres.2005.08.005.

Zhang, Y., G. F. Pinder, and G. S. Herrera (2005), Least cost design of groundwater quality monitoring networks, *Water Resour. Res.*, *41*, W08412, doi:10.1029/2005WR003936.

Zheng, C., and P. P. Wang (2002), A field demonstration of the simulation-optimization approach for remediation system design, *Ground Water*, *40*(3), 258–265, doi:10.1111/j.1745-6584.2002.tb02653.x.

P. Bayer, Ecological System Design, Institute for Environmental Engineering, ETH Zurich, Schafmattstr. 6, CH-8093 Zurich, Switzerland. (bayer@ifu.baug.ethz.ch)

C. M. Bürger, Center for Applied Geosciences, University of Tübingen, Sigwartstr.10, D-72076 Tübingen, Germany.

M. de Paly, Center for Bioinformatics Tübingen, University of Tübingen, Sand 1, D-72076 Tübingen, Germany.