WATER RESOURCES RESEARCH, VOL. 40, W06506, doi:10.1029/2003WR002675, 2004

Evolutionary algorithms for the optimization of advective control of contaminated aquifer zones

Peter Bayer and Michael Finkel

Center for Applied Geoscience, University of Tuebingen, Tuebingen, Germany

Received 11 September 2003; revised 13 April 2004; accepted 3 May 2004; published 24 June 2004.

[1] Simple genetic algorithms (SGAs) and derandomized evolution strategies (DESs) are employed to adapt well capture zones for the hydraulic optimization of pump-and-treat systems. A hypothetical contaminant site in a heterogeneous aquifer serves as an application template. On the basis of the results from numerical flow modeling, particle tracking is applied to delineate the pathways of the contaminants. The objective is to find the minimum pumping rate of up to eight recharge wells within a downgradient well placement area. Both the well coordinates and the pumping rates are subject to optimization, leading to a mixed discrete-continuous problem. This article discusses the ideal formulation of the objective function for which the number of particles and the total pumping rate are used as decision criteria. Boundary updating is introduced, which enables the reorganization of the decision space limits by the incorporation of experience from previous optimization runs. Throughout the study the algorithms' capabilities are evaluated in terms of the number of model runs which are needed to identify optimal and suboptimal solutions. Despite the complexity of the problem both evolutionary algorithm variants prove to be suitable for finding suboptimal solutions. The DES with weighted recombination reveals to be the ideal algorithm to find optimal solutions. Though it works with real-coded decision parameters, it proves to be suitable for adjusting discrete well positions. Principally, the representation of well positions as binary strings in the SGA is ideal. However, even if the SGA takes advantage of bookkeeping, the vital high discretization of pumping rates results in long binary strings, which escalates the model runs that are needed to find an optimal solution. Since the SGA string lengths increase with the number of wells, the DES gains superiority, particularly for an increasing number of wells. As the DES is a self-adaptive algorithm, it proves to be the more robust optimization method for the selected advective control problem than the SGA variants of this study, exhibiting a less stochastic search which is reflected in the minor variability of the found solutions. *INDEX TERMS:* 1829 Hydrology: Groundwater hydrology; 1831 Hydrology: Groundwater quality; 1832 Hydrology: Groundwater transport; 1899 Hydrology: General or miscellaneous; *KEYWORDS:* capture zone, optimization, pump-and-treat

Citation: 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.

1. Introduction

[2] The analysis of capture zones is a prominent topic in hydrogeology. For instance, it is important to guarantee reliable pump-and-treat systems as well as to delineate wellhead protection zones. Analytical and semianalytical algorithms for capture zone delineation have been developed by *Javandel and Tsang* [1986], *Shan* [1999], and *Christ and Goltz* [2002]. These techniques are essential aids but require certain simplifications of the prevailing conditions or are limited to specific situations, respectively. More generally applicable methods are based on numerical flow and transport modeling, either within a deterministic framework [e.g., *Mulligan and Ahlfeld*, 1999], where no uncertainty on the spatial parameter distribution is assumed, or by stochastic modeling. For the latter the classical method is the Monte Carlo analysis, where instead of one, several equally probable realizations of the transmissivity distribution are simultaneously considered [e.g., *van Leeuwen et al.*, 2000]. In the case of boundary conditions changing over time, transient simulations have been used to investigate time-related capture zones [*Bair et al.*, 1990; *Varljen and Shafer*, 1991].

[3] The major objective when planning hydraulic containment measures is to find pumping well configurations that accomplish the targeted effects most efficiently. With this, design optimization methods are essential. They ought to be fast and should concurrently exhibit the necessary robustness to be reliably successful independent of the respective characteristics of the problem given. The applicability and the meaningfulness of the results of such methods strongly depend on how capture, i.e., control of a given contaminated area, is formulated. A variety of formulations have been presented so far. According to *Mulligan and Ahlfeld* [1999], they can be subdivided into

0043-1397/04/2003WR002675

three groups: (1) concentration control, (2) hydraulic control, and (3) advective control. Concentration control means that feasible well configurations are found by complying with maximum concentration levels at control points [e.g., *Rogers and Dowla*, 1994; *Guan and Aral*, 1999]. Its application requires flow as well as concentration-based transport simulation and is therefore believed to be restricted to well-investigated sites where a comprehensive knowledge about the spatial distribution of the contaminants and transport relevant aquifer properties is available. Hydraulic control formulations are based on predefined head difference, gradient, or velocity constraints at selected points [e.g., *Colarullo et al.*, 1984; *Lefkoff and Gorelick*, 1986; *Gorelick*, 1987]. Hydraulic control is considered a fast method as it is exclusively based on flow modeling and allows utilization of the response matrix technique. However, a major disadvantage arises from the predefinition of constraints which requires anticipation of the flow regime, i.e., the shape of the capture zone of the optimal well configuration [*Mulligan and Ahlfeld*, 1999]. The advective control approach that has been used in this study makes use of particle tracking to delineate the capture zone [*Massmann et al.*, 1991; *Varljen and Shafer*, 1993; *Mulligan and Ahlfeld*, 1999; *Bayer et al.*, 2002]. In contrast to the two other approaches, advective control does not bias the results of optimization because no prejudgment concerning the capture zone must be made. However, since hydrodynamic dispersion is neglected, advective control should be applied only to cases dominated by macroscale heterogeneity sufficiently described in the flow model used. Several authors presented extensions to the advective control approach introducing travel time as an additional constraint [*Shafer and Vail*, 1987; *Massmann and Freeze*, 1987; *Greenwald and Gorelick*, 1989; *Varljen and Shafer*, 1993; *Ophori et al.*, 1998; *Maskey et al.*, 2002].

[4] Many sites allow considerable freedom in selecting the location of pumping wells and their extraction rates in order to reach hydraulic containment of contaminated aquifer zones. A typical task of optimization is to find the well configuration that guarantees complete capture at a minimum total pumping rate, which is of main concern economically due to its direct correlation to the operation costs. Since the objective function of the optimization is supposed to be highly nonlinear and nonconvex and have several local minima, conventional gradient-based nonlinear techniques are believed to be unsuitable. As a result, heuristic optimization techniques such as evolutionary algorithms (EAs) have gained considerable interest in the last decade to solve complex problems of groundwater flow control and remediation. [e.g., *Rogers and Dowla*, 1994; *Huang and Mayer*, 1997; *Zheng and Wang*, 1999; *Yoon and Shoemaker*, 1999; *Aly and Peralta*, 1999]. *Maskey et al.* [2002] successfully managed a time constraint plume remediation problem by advective control and the use of genetic algorithms. *Bayer et al.* [2001] used evolution strategies for the optimization of advective control by one-well pump-and-treat systems and the ideal adaptation of a funnel-and-gate system.

[5] The study presented here compares two forms of evolutionary algorithms which are based on different encodings of the decision variables and which realize a separate algorithm specific evolutionary search. Simple genetic algorithms (SGAs) [*Reed et al.*, 2000] work on binary parameter representations. (Completely) derandomized evolution strategies (DESs) [*Hansen and Ostermeier*, 2001] use real-valued parameter encodings. The performance of both algorithms in solving a hypothetical design optimization problem is analyzed under various settings of algorithm specific parameters. Hydraulic capture of a given area shall be achieved by a number of wells at a minimum total pumping rate. Candidate well configurations are represented by individual well positions and pumping rates leading to a mixed discrete-continuous problem. Since repeated runs of the simulation model during the optimization procedure may imply a remarkable computational burden, optimization algorithms are not only evaluated with respect to their suitability to detect a solution but also to the number of model runs needed.

2. Description of Example Containment Problem

[6] A hypothetical scenario of a contaminated site serves as the basis for the comparative analysis. The template site is simulated by a two-dimensional finite differences groundwater model (Figure 1). No-flow boundaries in the north and south as well as constant head boundaries along the western and eastern margins impose a regional hydraulic gradient of 10^{-3} from west to east on the confined flow regime. The aquifer thickness is assumed to be constant and set equal to 10 m. The finite differences resolution is given by a grid of 100 × 100 quadratic cells of 1 m side length each. The heterogeneous conductivity distribution is produced through indicator kriging, computed randomly, and unconditioned by the GSLIB software [*Deutsch and Journel*, 1992]. The aquifer is composed of equal portions of three sedimentological facies types with statistically isotropic spatial distribution, i.e., gravel (hydraulic conductivity $K_f = 10^{-2}$ m/s), sand (10^{-3} m/s), and silt (10^{-5} m/s). Figures 1 and 2 illustrate the spatial conductivity variation and the undisturbed flow paths when no pumping well is active. In total, 150 particles are equally distributed over a rectangular area (25 × 64 cells) representing the contaminated zone and traced by forward tracking along the locally varying head gradients. Figures 1 and 2 also depict the downgradient well placement zone (32 × 64 cells), where a number of pumping wells shall be installed for the hydraulic containment of the contaminated area.

3. Advective Control Optimization

[7] The major challenge one has to cope with when formulating a suitable objective function is that the optimum lies at the border between feasible and infeasible solutions. Feasible solutions are pumping scenarios that guarantee contaminant capture, whereas infeasible (invalid) solutions do not. Since for the feasible solutions the pumping rate can be taken as a measure of the quality of a well configuration [e.g., *Shafer and Vail*, 1987; *Maskey et al.*, 2002], it is an inappropriate criterion to rank invalid solutions. Given that a restriction of an objective function to the feasible decision space would result in a discontinuity at the minimal pumping rate of a given well configuration that hampers the applicability of most optimization algorithms, penalty functions are usually employed to transfer the constraint optimization problem into an unconstraint

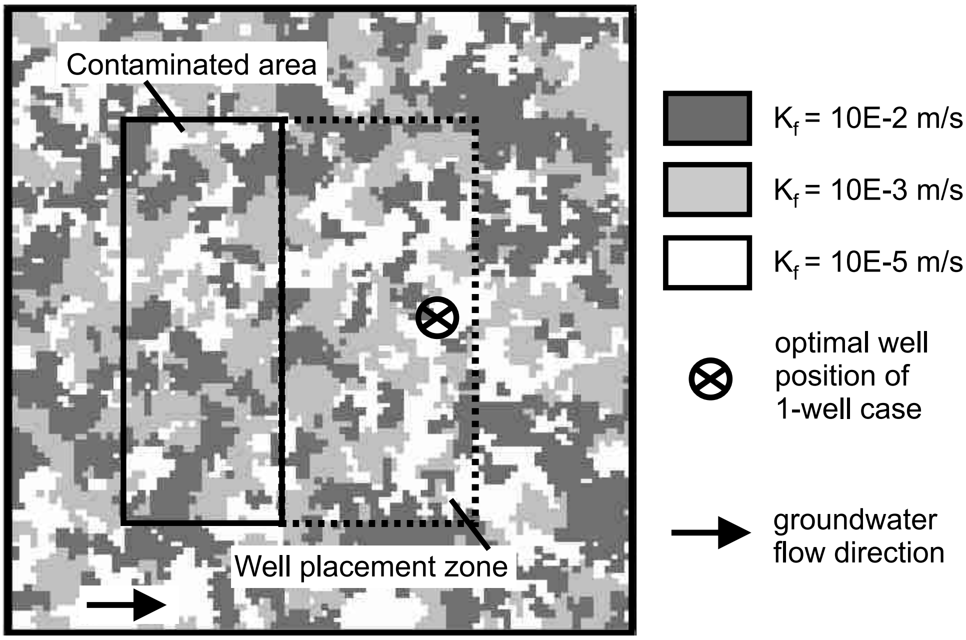

Figure 1. Spatial hydraulic conductivity distribution of template flow model with contaminated area (solid line) and well placement area (dotted line).

problem [e.g., *Mulligan and Ahlfeld*, 1999; *Yoon and Shoemaker*, 1999; *Chan Hilton and Culver*, 2000]. The penalty should properly quantify the degree of infeasibility of a given solution. Within the control optimization procedure proposed here the value of the penalty term involves the proportion of the contaminant plume that is captured, represented by the ratio of forward tracked particles ending up in one of the model cells including a pumping well. In order to cope with so-called "weak wells" [e.g., *Zheng*, 1994] the complete path lines have to be analyzed with respect to reaching an individual well. The "weak well problem" can occur when the well pumping rate is relatively small. In this case, the finite difference cell containing the well may not have completely inward gradients at the cell interfaces, so that the fate of a particle entering the cell is undefined.

[8] Following *Huang and Mayer* [1997], we express well positions as explicit decision variables. In fact, the approach proposed here is an extension to the formulation of *Huang and Mayer* [1997] because the search space for individual wells is not limited to well-specific subregions of a number of candidate wells but can be anywhere in the well placement zone. In doing so, optimization is not biased by any presumptions as any possible design alternative can potentially be considered and evaluated in the course of the optimization procedure. However, this results in an immense computational effort as every additional well triplicates the problem dimensionality. A technique to circumvent the explicit formulation is to separate the selection of well positions and the adjustment of pumping rates by the use of a two-step optimization procedure [*Zheng and Wang*, 1999; *Chang and Hsiao*, 2002]. Although this solution technique has the advantage that for each optimization step, different optimization algorithms may be applied, we preferred the explicit formulation. This is to keep the comparison of EAs simple, or rather, to avoid equivocalness of the results.

[9] The primordial objective is the reduction of the total pumping rate q_{tot}. Taking into account the constraint on particle capture as described above, the complete, now nonlinear, objective function is stated as

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

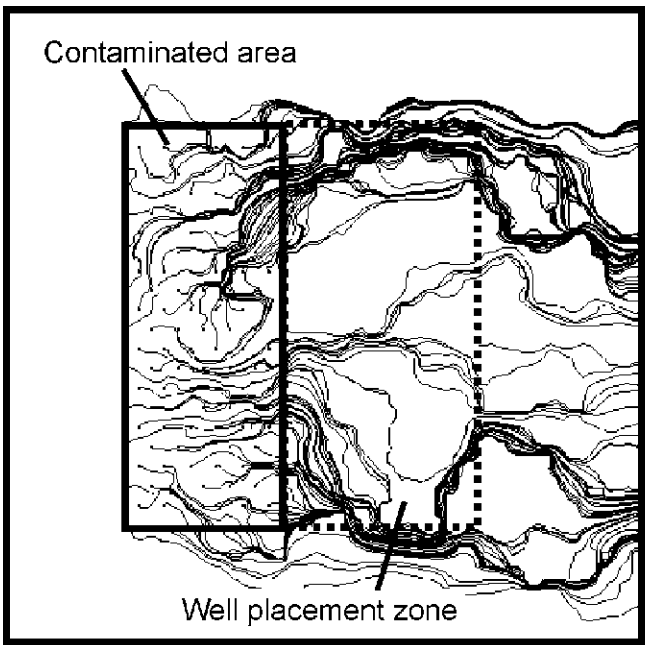

Figure 2. Path lines of particles starting from contaminated area (solid line) assuming undisturbed flow. The well placement area is marked by the dotted line. The groundwater flow direction is E–W.

subject to

$$0 \leq \nu \leq 1, \quad (2)$$

$$w = 1, \ldots, W, \quad (3)$$

$$x_{\text{low}} < x_w < x_{\text{up}}, \quad (4a)$$

$$y_{\text{low}} < y_w < y_{\text{up}}, \quad (4b)$$

$$q_{\text{low}} < q_w < q_{\text{up}}, \quad (5a)$$

$$q_{\text{tot,low}} < q_{\text{tot}} < q_{\text{tot,up}}, \quad (5b)$$

where q_{tot} and q_w are the absolute values of the total pumping rate and those of individual wells. W is the number of wells, w is the index of a particular well, and x_w and y_w are row and column indices of the wells. The penalty term φ depends on v, the ratio of particles that are not captured.

[10] The transformation of infeasible solutions into feasible ones by an appropriate penalty term formulation is a tricky step that is important for achieving a formulation of the objective function that maximizes the performance of the optimization algorithm. The penalty term φ should effect the rejection of invalid configurations and simultaneously discriminate between invalid settings according to the degree of constraint violation in terms of noncaptured particles. An ideal penalty term smoothes out the margins of the valid parameter space without an artificial creation of (sub)optimal invalid solutions. This "smoothness" seems to be desirable, especially for cases such as the one presented here, where marginal optima are expected, and the incorporation of the originally nonfeasible space allows for approaching the optimum from two "sides" instead of one during the search process. Here a multiplicative formulation is preferable over an additive, which would result in a sharp terraced objective function increase when moving from valid to penalized solutions [*Chan Hilton and Culver*, 2000]. Increasing the marginal smoothness stands for relaxing the acceptance of invalid solutions, which in our case would mean that unsatisfactory pumping strategies with low pumping rates that do not capture all particles gain superiority over valid solutions with relatively high pumping rates. This conflict limits the available range of smoothing. The crucial issue here is to value one noncaptured particle such that the objective function value is higher than for the respective valid solution having a higher pumping rate but captures all particles. Obviously, this task depends on hydraulic conditions given and has to be handled individually for each specific problem. For the purpose of a robust formulation, one has to deal with the worst case, which has to be anticipated on the basis of a preliminary modeling exercise. To give a further explanation, we consider a single well placed somewhere near the southern edge of the placement area (Figures 1 and 2). Precalculations revealed that remarkably low pumping rates are sufficient for capturing most but not all of the particles. The path lines of the noncaptured particles, however, are concentrated in a northern high-conductivity channel, and thus a disproportional increase of the pumping rate is necessary to capture the rest of the particles. This requires a relatively high penalty in order to assign the invalid solution a worse fitness than the valid solutions. Thus, it is not possible to obtain the desirable marginal smoothness of the objective function if unsatisfactory local optima in the form of invalid low-pumping rate solutions cannot be accepted. As a practicable compromise we used two penalty terms that turned out to be suitable for the problem given: an exponential (equation (6)) and an adaptive penalty term (equation (7)).

$$\varphi(v) = A^{(100v)^a}; \quad (6)$$

$$\varphi(v) = 1 + \frac{\left(\min F^{\text{valid}}_{0,..t-1} - \min F^{\text{all}}_{0,..t-1}\right)\left(\frac{v}{nft_v}\right)^{\kappa}}{\sum_{w=1}^{W} q_w(x_w, y_w)}; \quad (7a)$$

$$\forall\, t \in \{2, \ldots, T\}. \quad (7b)$$

[11] For the exponential penalty term, two driving parameters have to be set: the basis A that determines the value of φ if $\geq$1% of the particles is not captured ($v \geq 0.01$) and the upper exponent a that attenuates the exponential rise of the penalty factor. We recommend setting A in the range of 7–10 and $0.6 \leq a \leq 1$.

[12] *Coit et al.* [1996] formulated the adaptive term, which in the strict sense, results in an additive penalization. The adaptive penalty term was specially used for the application of evolutionary algorithms, which in general, are based on a number of consecutive function evaluations (i.e., generations t with a maximum number of generations T). After every new iteration, the penalty-driving parameters are dynamically updated. The setting is based on the best of all valid solutions so far, $F^{\text{valid}}_{0,\ldots,t}$, and the best of all non-penalized solutions so far, $F^{\text{all}}_{0,\ldots,t}$ (here all total pumping rates processed). The severity parameter κ and the problem-specific value of nft_v have to be defined by the user. The parameter nft_v determines the tolerance limit of invalid solutions, in our problem the accepted ratio of noncaptured particles. Beyond this limit, $\kappa > 1$ forces an increasingly severe penalty, while for $v < nft_v$, κ affects an attenuation of the penalty. We set $\kappa = 1.1$ in order to minimize the risk of finding invalid solutions. The tolerance limit nft_v was set to 5%.

4. Evolutionary Algorithms

4.1. Overview

[13] Since the 1960s, several different implementations of evolutionary algorithms have been presented. Despite their completely different origins the branches are based on the same idea: the imitation of processes of natural evolution for automated problem solving [e.g., *Bäck and Schwefel*, 1993]. This imitation is not aiming at a true copy of the biological archetype, but it is exploiting efficient evolutionary features in a self-serving, abstract manner. The main use of the subtypes genetic algorithms (GAs) and evolution

strategies (ESs) is for the probabilistic optimization of problems in various areas. Although their individual characteristics (e.g., representation of decision variables) predetermine dissimilar fields of application, in many cases, GAs and ESs compete with each other in terms of their specific suitability. In many cases, the term "suitability" is defined differently. For example, one procedure may produce the global optimum more reliably, while the other method quickly calculates a broad range of (sub)optimal results for a problem and may thus provide a more appropriate input for a subsequent decision analysis.

[14] Although the most suitable optimization routine depends on the individual problem characteristics, for practical applications such as advective control, the objective should be to find a robust and reliable routine which is easy to adapt to a broad range of scenarios. For example, it should be capable of dealing with a variable number of pumping wells and be suitable for a spectrum of hydraulic conditions. SGAs have proven their suitability by producing satisfactory solutions in diverse applications [e.g., *Rogers and Dowla*, 1994; *Huang and Mayer*, 1997]. However, in several cases, researchers have reported limits of their application that arise from a significant number of function evaluations necessary to find a solution [*Chang and Hsiao*, 2002], from the inefficiency to produce accurate solutions [*Yoon and Shoemaker*, 1999], or even from the improper adaptation to a specific problem [*Reed et al.*, 2000]. *Yoon and Shoemaker* [1999] compared SGAs to DESs (and other optimization methods). In their investigation of the optimal planning of a time-varying bioremediation measure they concluded that DESs seem to be preferable over the SGAs since DESs are more efficient in producing accurate solutions.

[15] Conceptually, both SGAs and DESs belong to the family of stochastic "generate-and-test" search algorithms [*Eiben and Schippers*, 1998]. Oriented at the natural evolution, solutions to a given optimization problem are represented by individuals in a population subject to step by step modification determined by probabilistic (and in some cases deterministic) operators. In each generation the created individuals are simultaneously evaluated by calculating the objective function value of the specific parameter configuration of the respective individual. For the advective control problem this means that every generation represents a number of well configurations that is processed by the numerical model. On the basis of the results from hydraulic modeling and particle tracking the objective function value (fitness) is assigned. After the evaluation (i.e., the "test"), recombination and mutation are conducted, leading to a probabilistic mixing and alteration within the population. The resulting individuals provide the mating pool, which is subject to a probabilistic or deterministic selection procedure to produce the next generation. To obtain an optimization algorithm, the driving idea is to repeatedly create new generations to evolve toward improving fitness values of the individuals [*Bäck and Schwefel*, 1993].

[16] Despite their conceptual similarity, SGAs and DESs are significantly different. One difference lies in the representation of individuals and, consequently, the manner in which the probabilistic operations are conducted. Another difference is that DESs, like ESs in general, are self-adaptive search methods compared to the rather static implementation of SGAs.

4.2. Simple Genetic Algorithms

[17] The individuals of SGAs are represented as binary strings or chromosomes [*Goldberg*, 1989]. This characteristic accounts for the ideal suitability of SGAs for pseudo-Boolean or discrete optimization problems. The number of bits b defines the number of different individuals that can be realized (2^b) by a string. It is apparent that the parameterization of a one-well case with explicitly formulated well locations leads to a three-dimensional (3-D) decision space with variables x_w, y_w, and q_w. For the example site used in this study, 32 rows and 64 columns have to be binary encoded. The binary mapping works out evenly, exactly 5 bits for the rows and 6 bits for the columns are required. Note that this is an ideal case, but in many situations the given discretization leads to unused bits which afford a specified handling [cf. *Goldberg*, 1998]. The coding of the real-valued q_w is achieved by subdividing the constraint parameter space into segments of equal length. The accuracy, i.e., the discretization of the representation of q_w, is determined by b and the parameter range $|q_{\mathrm{up}} - q_{\mathrm{low}}|$. The parameters are encoded in series on each string well by well. As such, the suggestions of *Huang and Mayer* [1997], who applied an SGA to solve a concentration control problem with moving wells, are followed.

[18] For the SGA of this study, recombination is worked out by single-point crossover. The predetermined crossover rate p_c fixes the probability that two randomly chosen individuals are selected, then cut at a same random bit position, and finally, exchange the cutoff portions to form completely new chromosomes. Mutation is carried out by flipping bits at random with probability p_m. The selection method of the SGA used here is tournament selection, where a number of strings are drawn at random from the population and the one with the best fitness will survive to form part of the next generation. Additionally, elitism is conducted, meaning that the best individual of the mating pool is always selected and thus preserved.

[19] Practical hints for the design of so-called competent (S)GAs that are perfectly adapted to a problem are proposed by several authors [e.g., *Goldberg*, 1998; *Bäck et al.*, 2000; *Reed et al.*, 2000; *van Dijk et al.*, 2002]. For this study the mutation probability p_m is customarily oriented at the recommendations by *DeJong* [1975] and set at the inverse of the population size $p_m = 1/ps_{\mathrm{SGA}}$. Following *Lobo* [2000] and *Reed et al.* [2000], a crossover probability p_c of 0.4 and a tournament size of $s = 4$ are chosen. Additionally, we tested binary tournament crossover with values of p_c between 0.4 and 0.8, which represent the typical range of so-called standard SGA settings [*Lobo*, 2000]. The population size $\mathrm{pop}_{\mathrm{SGA}}$, is identified as one of the most critical performance criteria. Although a relatively broad interval of the value of $\mathrm{pop}_{\mathrm{SGA}}$ is acceptable for the formulation of a successful SGA for an individual problem, using a number above the ideal value can result in a higher necessary total number of function evaluations (i.e., model runs). The selected population sizes are oriented at suggestions by *Thierens et al.* [1998] and *Reed et al.* [2000], who proposed lower limits to $\mathrm{pop}_{\mathrm{SGA}}$ of $\sim$1.4 b. This value is based on the assumption that an SGA must converge faster under selec-

tion than under the influence of genetic drift. Genetic drift occurs for smaller pop_{SGA}, where an insufficient selection pressure leads to the accumulation of nonoptimal strings and, finally, a premature convergence of the SGA (drift stall). Aside from these pop_{SGA} values, for comparison, slightly distinct values are tested in order to reveal sensitivities of SGA performance to pop_{SGA} and to detect potential better SGA variants.

4.3. Derandomized Evolution Strategies

[20] Contrary to the binary representation of SGAs, the object parameters of DESs (and ESs in general) are real-coded N-dimensional vectors [*Hansen and Ostermeier*, 2001; *Bäck and Schwefel*, 1993]. One population consists of λ individuals from which the fittest μ ($\lambda \geq \mu$) are selected to form the mating pool for the next generation (denoted as "(μ, λ)-DES"). New individuals evolve by repeated multi-parent recombination and mutation of the object parameters. In DES, mutation step sizes σ_t are introduced which control the generation-wise change, i.e., mutation of the object parameters. The tricky feature is that step sizes (i.e., their distribution) are adapted during the search process. This is accomplished by comparing the realized (successful) step length to the expected one. Assuming a population of individuals $\mathbf{p}_n$ (n = 1, ..., λ) representing a well configuration of W wells, the basic algorithm is stated as

$$\mathbf{p}_{n,t+1} = \langle \mathbf{p} \rangle_{\mu,t} + \sigma_t \, \mathbf{z}_{n,t+1}, \tag{8a}$$

$$\mathbf{s}_{t+1} = (1 - c_\sigma)\, \mathbf{s}_t + \sqrt{c_\sigma(2 - c_\sigma)} \left(\sum_{n=1}^{\mu} \omega_n^2 \right)^{-\frac{1}{2}} \langle \mathbf{z} \rangle_{\mu,t+1}, \tag{8b}$$

$$\sigma_{t+1} = \sigma_t \exp\left[\frac{1}{d_\sigma} \left(\frac{\| \mathbf{s}_{t+1} \| - \hat{\chi}_N}{\hat{\chi}_N} \right) \right], \tag{8c}$$

subject to

$$\mathbf{p}_{n,t} = (q_w, x_w, y_w)_{n,t}, \tag{9a}$$

$$w = 1, \ldots, W, \tag{9b}$$

$$N = 3W, \tag{9c}$$

$$t = 1, \ldots, T, \tag{9d}$$

$$\mathbf{z}_{n,t+1} \sim \mathbf{N}(0, \mathbf{1}), \tag{9e}$$

$$\langle \mathbf{p} \rangle_{\mu,t} = \sum_{n=1}^{\mu} \omega_n \mathbf{p}_{n'(n),t}, \tag{9f}$$

$$\langle \mathbf{z} \rangle_{\mu,t+1} = \sum_{n=1}^{\mu} \omega_n \mathbf{z}_{n'(n),t+1}, \tag{9g}$$

$$\sum_{n=1}^{\mu} \omega_n = 1, \tag{9h}$$

$$s_1 = 0, \tag{9i}$$

$$\hat{\chi}_N = E(\| \mathrm{N}(0, \mathbf{I}) \|) \cong \sqrt{N} - \sqrt{\frac{1}{4N}} - \sqrt{\frac{1}{21N^2}}. \tag{9j}$$

Table 1. Parameter Settings for the DES After *Hansen and Ostermeier* [2001]

Parameter	Value
λ	$4 + [3 \ln(N)]$
μ	$[\lambda/2]$
c_c	$4/(n + 4)$
c_{cov}	$2/(n + 2^{0.5})^2$
c_σ	$4/(n + 4)$
d_σ[a]	$c_\sigma^{-1}1+$

[a]Here d_σ is a damping parameter referred to the global step size.

Equation (8a) shows that for every new generation $t + 1$ the new individuals are produced through recombination of the parents and by adding mutation steps. The cumulation procedure (8b) introduced by *Ostermeier et al.* [1994] precedes the generation-wise step size adaptation given in equation (8c). In this way, instead of looking exclusively at the last step length, a weighted evolution path is considered. The relative weighting parameter c_σ defines the decreasing importance of the previous steps. The weighted sum s_t of differences of each two consecutive weighted means of parents (including the actual one) is then used for step size adaptation. Parameter $\hat{\chi}_N$ denotes the expectation of the length of a random vector with expectation 0 and covariance matrix $\mathbf{I}$ (equation (9j)), which is related to s_t. For the step size adaptation in equation (8c) a damping factor d_σ is introduced in order to obey *Rechenberg*'s [1994] rule of "mutating big but inheriting small."

[21] The weighted recombination is defined by equations (9f)–(9h), where $n'(n)$ is the index set of μ selected individuals which are sorted in descending order according to their fitness. The values of the weighting parameters ω_n determine the relative weighting of parent individuals which can be used to vary the selection pressure based on the relative fitness.

[22] For more than one individual in the mating pool (μ > 1) the simultaneous execution of several individual mutation steps and one global mutation step is conducted. A main feature of the DES is the covariance matrix adaptation (CMA) introduced by *Hansen and Ostermeier* [1996] to achieve a rotation invariant implementation of both individual and global step sizes. The CMA-based transformation is a rotation and scaling method to adapt the generation-specific mutation distribution to the local topology of the fitness function by the evaluation of the accomplished mutation steps. The complete algorithm is presented by *Hansen and Ostermeier* [2001] and will not be described here in detail.

[23] At first glance, a handicap of the DES formulation seems to be the appropriate setting of numerous (strategy) parameters. Beside μ, λ, d, c_σ and ω_n, the CMA comprises additional parameters such as the cumulation time for individual step size adaptation c_c and the change rate of the covariance matrix c_{cov}. *Hansen and Ostermeier* [2001] comprehensively discuss the ideal parameter setting and the parameter sensitivities and give default values suitable for a broad range of problems (Table 1). Starting from these default values, we tested a number of variable parameter configurations, but apart from stochastic outliers we could not identify reliably better settings. One central feature is that contrary to the SGA, the population size λ does not

linearly scale with problem dimension ($\lambda = 4 + 3\ln[N]$). *Hansen and Ostermeier* [2001] propose intermediate recombination (ω_n = const) as an alternative to weighted ($\omega_{n+1} < \omega_n$) recombination. Both schemes are compared throughout this study. The intermediate DES, denoted as (μ_I, λ)-DES, is characterized by

$$\omega_n = \frac{1}{\mu}. \tag{10a}$$

The (μ_W, λ)-DES includes weighted recombination, which means that relative weights are applied to ranked solutions with descending fitness values. Assuming $\mu \leq \lambda/2$, it is expressed here as

$$\omega_n = \frac{\ln\left(\frac{\lambda+1}{2}\right) - \ln(n)}{\sum_{n=1}^{\mu} \ln\left(\frac{\lambda+1}{2}\right) - \ln(n)}. \tag{10b}$$

[24] ESs are real-valued parameter optimization algorithms. Consequently, to obtain well location coordinates, DES-calculated real-valued positions are processed by parameter rounding. For discrete parameters, *Hansen* [1998] suggests setting the lower limit to the mutation step sizes:

$$\sigma_t \geq \frac{m}{10\sqrt{\frac{2}{3}N}}. \tag{11}$$

The parameter m incorporates the constant discretization step in each coordinate that has to be limited. Since well locations are parameterized as integer values (cell indices) in a finite differences scheme, m is set equal to 1 (and thus the limit is 0.122 $N^{-0.5}$). By incorporating this limit, premature stagnation due to too small mutation steps and thus stagnating well locations can be restricted.

[25] The object parameters, such as well coordinates and pumping rates, are scaled in the interval from 0 to 1 according to the problem-specific predefined validity limits (e.g., tolerance spectrum of pumping rates and well coordinate boundaries). This straightforward method simplifies the initialization of strategy and object parameters. For example, all initial individual and global step sizes $\sigma_{n,1}$ and σ_1 are set equal, in our case, to 0.5.

[26] In contrast to the predefined object parameter limits of SGA's binary coding, for the application of DES constraints on acceptable decision parameter values must be formulated in addition. From the many options available to restrict pumping strategies beyond the validity limits an exterior penalty method given by *Hansen and Ostermeier* [2001] was chosen.

5. Modeling and Evaluation Framework

[27] Groundwater flow is modeled with the finite difference based program MODFLOW [*McDonald and Harbaugh*, 1988]. The MODPATH software [*Pollock*, 1994] serves as a postprocessor to track particles in order to obtain the number of particles that are captured by the respective well configuration under consideration. The underlying governing equations describing steady state groundwater flow and for derivation of particle paths have been presented in various studies [*Pollock*, 1994; *Mulligan and Ahlfeld*, 1999] and shall be omitted here. Flow simulation and particle tracking are iteratively called by a MATLAB-based SGA and DES optimization driver program.

[28] Comparative evaluation of SGA and DES variants considers various parameter settings in order to find the optimization strategy that is capable of finding a given optimization target function objective value (fov) at a minimum of total model runs MR_{min}(fov). Because of the stochastic features of evolutionary algorithms, any randomly initialized optimization run (OR), even if parameter settings are not changed, shows a different search and provides different results, i.e., values of decision variables and objective function. Thus, to provide a reliable basis for the analysis, the optimization has been performed repeatedly (50 randomly initialized attempts) for each of the parameter settings. In doing so, we expect stochastic influences on the evaluation and fortuitousness of the interpretation of the results to be sufficiently reduced. The number of ORs might not be sufficient to obtain statistically representative results in the case examined but is believed to reveal the basic sensitivities of the EA to parameter settings and to facilitate the comparison of SGA and DES.

[29] For each optimization attempt the number of function evaluations, i.e., model runs (MR), was limited to a certain case-specific maximum number I. At first sight, the question to be answered is, How many model runs MR_{min}(fov) are needed on average to find a solution (well configuration) that meets a given target fov? As the progress of the optimization can vary remarkably in the course of the search process and may eventually reach a certain stagnation, a single OR does not necessarily have to be the best option. One also has to consider multiple ORs with a comparatively low number of MRs in order to find the optimization strategy that yields the lowest MR_{min}(fov). For example, it has to be examined whether one OR of 1000 function evaluations on average performs better than two ORs with 500 each. Accordingly, the above question has to be formulated more generally, How often does one have to repeat an OR of a certain number of function evaluations to find a solution that meets the target fov with a minimum of total function evaluations MR_{min}(fov)? For every consecutive function evaluation, during one attempt, the best performing well setting attained is saved. The probability of finding a specific (or better) fov at iteration i is then expressed by the rate of success over all attempts (k = 1, . . ., 50). If the number of function evaluations per OR is assumed to be i, the average number of total model runs to meet fov is derived by

$$\mathrm{MR}_i(\mathrm{fov}) = \frac{i}{\mathrm{prob}\left(\min\limits_{j=1,\ldots,i} f_{j,k} \leq \mathrm{fov}\right)}, \qquad \forall i = 1, \ldots, I, \tag{12}$$

where $f_{j,k}$ is the result of the function evaluation of MR number j in attempt k. The minimum of total model runs MR_{min}(fov) necessary to find a solution that meets the target follows as

$$\mathrm{MR}_{\min}(\mathrm{fov}) = \min_{i=1,\ldots,I} (\mathrm{MR}_i(\mathrm{fov})) = \mathrm{MR}_{I_{\mathrm{deal}}}(\mathrm{fov}), \tag{13}$$

where I_{deal} is the optimal (ideal) number of function evaluations per OR. Accordingly, the average number of required ORs n_{OR} is

$$n_{OR} = \frac{MR_{min}(fov)}{I_{deal}}. \quad (14)$$

The performance of EAs is further evaluated with respect to the optimization progress in the course of the evolutionary search which is quantified by the cumulative probability $\phi_{i'}(fov)$ of achieving a solution that meets the target after a certain number of function evaluations i':

$$\phi_{i'}(fov) = prob\left(\min_{j=1,\ldots,i'} f_{j,k} \leq fov\right), \forall i' = \Delta i, 2\Delta i, \ldots, I \quad (15)$$

with Δi representing a function evaluation interval that is chosen case specific between 250 and 500 MRs.

[30] Note that our approach neither includes a stop criterion [e.g., *Reed et al.*, 2000] nor evaluates the improvement during the search by a progress rate [e.g., *Hansen and Ostermeier*, 2001]. As such, a possible premature stagnation of the algorithm during the search is accepted, meaning a deficient production of improved individuals. Instead, a stagnating or, more generally, unsatisfactory search progress is reflected by the calculated (high) value of expected model runs $MR_{min}(fov)$. Even if a stop criterion based on convergence characteristics might be desirable in applications of the optimization algorithms to a specific site, we consider it as being unsuited for the purpose of this study, as it would complicate comparison of the optimization algorithms' performance.

[31] The assessment methodology considers two further important features: bookkeeping [e.g., *Chang and Hsiao*, 2002] and suboptimal solutions. Bookkeeping means that if individuals are preserved to the next generation, the saved fitness value is taken instead of executing a model run. Consequently, stagnation of the algorithm may lead to an increasing number of function evaluations but not of model runs themselves. While bookkeeping is not useful for DESs since repetitions of the real-valued individuals are unlikely, SGAs can benefit considerably from bookkeeping. This benefit was analyzed by comparing SGAs with and without bookkeeping. The number of function evaluations that refer to bookkeeping is addressed by the parameter I_{book} and is not included in $MR_{min}(fov)$. In order to terminate stagnated SGAs the maximum number of function evaluations ($I + I_{book}$) was limited to 50,000.

[32] Obviously, the most desirable optimization target fov is the ultimate minimum total pumping rate for a group of pumping wells. However, for the multiple well scenarios addressed here, this global optimum is not known. Targets are therefore referred to $q_{tot,min}$, the best value found in this study for the respective scenario. Since the SGAs' accuracy is determined by the specific bit coding, the predefined discretization of the search space determines its preciseness in identifying the optimal solution. Thus in order to ensure comparability between the DES and the SGA a search tolerance, i.e., a minimum objective value fov* slightly above $q_{tot,min}$, is introduced. The respective search tolerance varies depending on the problem dimension between 1% (one- and two-well cases) and 2.5% (four-well case). One may also be interested in suboptimal solutions within a broader tolerance interval to give the chance for expert selection based on not explicitly formulated "soft" criteria. Aside from this, highly resolved optima that severely restrict well locations and pumping rates are likely not practical given physical installation constraints at many sites as well as limited model accuracy. Because of this the performance of the EAs is also evaluated with respect to values of fov > fov*.

[33] Optimization strategies with more than one OR can potentially benefit from the experience gained in previous ORs through reorganization of the parameter settings resulting in an accelerated evolutionary search. An obvious resolution considered here is to reduce the range of the pumping rate by adjusting q_{up} according to the minimum pumping rate found in the previous OR. In this manner, decision space is reduced, which is believed to decrease the model runs required to achieve the target value. In order to estimate the benefit from such a boundary update, q_{up} of every second optimization attempt was adjusted in accordance to the results of the respective previous attempt. Accordingly, attempts are subdivided into "pioneer" and "updated" attempts. The performance of SGA may benefit from boundary update either by a higher accuracy of representing q or in terms of a reduced chromosome length while maintaining a targeted accuracy. The latter is believed to decrease the function evaluations required and has been considered here. The accuracy level was set to at least one per mil of q_{up} for the pioneer attempts, corresponding to 10 bits for the binary coding of the pumping rate, leading to 21 bits in total for each individual in the one-well case. The boundary update also improves the performance of real-coded DES by explicitly confining the decision space, thereby omitting unpromising model runs i.e., directing the evolutionary search to relevant regions.

6. Discussion of Advective Control Optimization Results

[34] With the framework described above, the performance of SGA and DES in minimizing the total pumping rate has been analyzed for one-, two-, four- and eight-well cases which are discussed individually in sections 6.1–6.4.

6.1. One-Well Case

[35] For the one-well case, the best fitness that can be reached at each of the 2048 possible well positions within the placement zone corresponds to the minimum pumping rate $q_{w,min}(x_w, y_w)$ required to completely capture the given source area. The values of $q_{w,min}(x_w, y_w)$ have been determined using a bisection method [*Press et al.*, 1992]. Figure 3 shows the distribution of $q_{w,min}(x_w, y_w)$ and illustrates the nonconvex response surface. Several local optima are identified which can act as possible traps for the optimization algorithms during the search process. As both the pumping rate and the well positions are decision variables, the complete, three-dimensional representation of fitness f can be expected to be even more complex. Even so, Figure 3 gives insight into the nonlinearity of the selected advective control problem. The overall minimum total (here it equals individual) pumping rate $q_{tot,min}$ is calculated 7.15 m^3/d (fov* = 7.22 m^3/d) at x_w = 52 and y_w = 70.

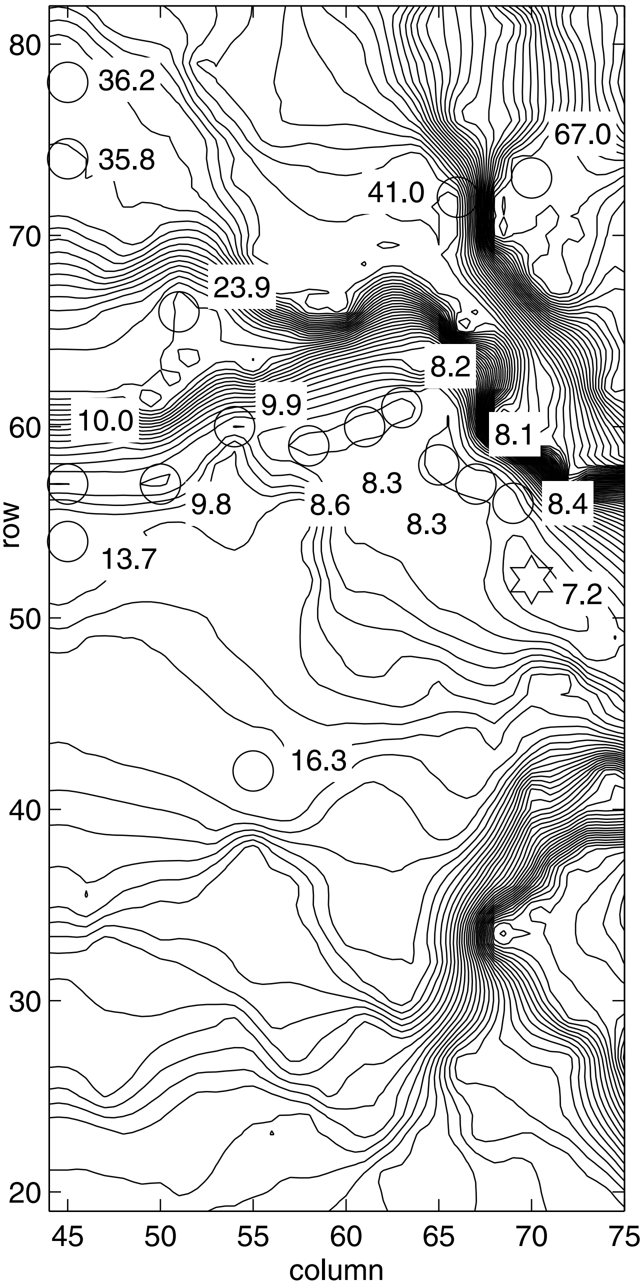

Figure 3. Isolines of minimum pumping rates $q_{w,\min}$ (m^3/d) for well placement area. Circles mark local optima, and the star represents the global optimum.

The value of $q_{w,\min}(x_w, y_w)$ is shown to vary over one order of magnitude between 7.15 and 75.7 m^3/d.

[36] In the pioneer attempts, only a limited knowledge of the $q_{w,\min}(x_w, y_w)$ distribution is presumed. The upper boundary q_{up} is set to the mean of all $q_{w,\min}(x_w, y_w)$, 26 m^3/d, while the value of q_{low} is fixed at q_{up}/1000. The updated upper boundary is set 20% higher than the best previously found f in order to relax the acceptance to a limited number of suboptimal solutions.

[37] The population size of the SGA was set to 30, with p_c = 0.5 and s = 4, using the exponential penalty term. Its comparison to other SGA variants that have been analyzed (pop_{SGA} = 10, 20, 50, and 75; p_c = 0.4, 0.6, and 0.8; s = 2) indicates that a broad range of SGA variants leads to similar results. A striking issue is the large fluctuations in SGA performance during any of the 50 attempts for each of the tested settings, reflected by a significant variability of the number of model runs needed to find a solution for a given fov. This aspect complicates the prediction of the performance of the SGA and also the comparison between different SGA variants. In general, small pop_{SGA} tend to give the best results. Table 2 lists the performance characteristics of the best SGA variant (pop_{SGA} = 20, p_c = 0.6, s = 2) for the three values of fov considered. Figure 4 depicts the calculated values of MR_i(fov) depending on the number of model calls i if fov = 8.6 m^3/d (representing a tolerance of 20% higher than fov*).

[38] Figure 4 not only illustrates the irregular curve shape for the MR_i(fov) values but also demonstrates the influence of bookkeeping and clarifies the benefit of boundary updating. The pioneer SGA with no bookkeeping is the worst option in this situation, resulting in a minimum number of total model runs $\mathrm{MR}_{\min}$(fov) = 2200, assuming ~3 ORs of

Table 2. One-Well Case: Success Rates and Ideal Configuration of Optimization Runs for Best SGA Variant Considering an Exponential Penalty Term

SGA[a]	fov, m^3/d	Success Rate,[b] %	I_{deal}	$\mathrm{MR}_{\min}$	n_{OR}
		No Bookkeeping			
Pioneer SGA (20, 0.6)	7.22	0	-	-	-
Pioneer SGA (20, 0.6)	7.5	12	2050	14,500	7.1
Pioneer SGA (20, 0.6)	8.6	68	700	2200	3.1
Update SGA (20, 0.6)	7.22	8	1000	12,000	12.0
Update SGA (20, 0.6)	7.5	68	750	1750	2.3
Update SGA (20, 0.6)	8.6	92	400	800	2.0
		Bookkeeping			
Pioneer SGA (20, 0.6)	7.22	8	-	>15,000	-
Pioneer SGA (20, 0.6)	7.5	16	2050	12,000	5.9
Pioneer SGA (20, 0.6)	8.6	72	550	1800	3.3
Update SGA (20, 0.6)	7.22	8	750	9500	12.7
Update SGA (20, 0.6)	7.5	72	600	1450	2.4
Update SGA (20, 0.6)	8.6	92	400	700	1.8

[a]Here s = 2.
[b]I = 3000.

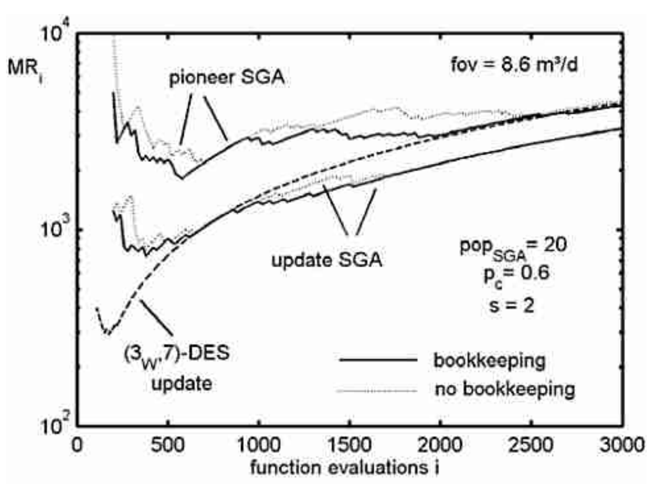

Figure 4. One-well case: Calculated total model runs MR_i(fov) for number of model runs i per OR, assuming an objective value fov = 8.6 m^3/d (exponential penalty term).

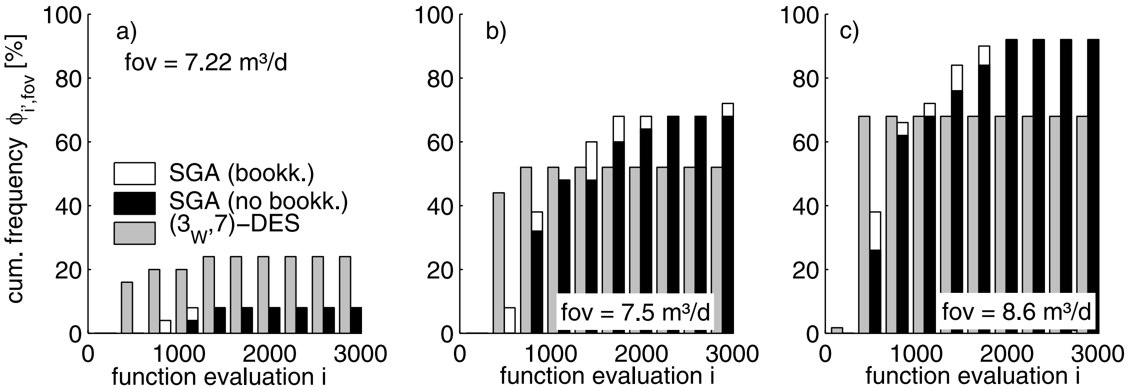

Figure 5. Cumulative frequencies of finding one-well case solutions when applying the boundary update method (SGA: pop_{SGA} = 20, p_c = 0.6, s = 2; [3_W,7] - DES; exponential penalty term).

I_{deal} = 700 model runs each. Remarkably, the pioneer SGA without bookkeeping was not able to find the optimum at fov* within 50 ORs of I = 3000 model runs each. The value of q = 7.5 m^3/d characterizes solutions of wells which are positioned (nearly) at the optimal position, but the pumping rates are not adapted sufficiently (see Figure 3). Twelve percent achieved success in finding solutions for fov = 7.5 m^3/d, and 68% could identify solutions for fov = 8.6 m^3/d.

[39] By using the boundary update method the efficiency of the SGA can be improved significantly. The reduced decision space, together with the savings of bits, leads to a decrease of MR_{fov}. Instead of 2200, now 800 model runs are needed to obtain a solution for fov = 8.6 m^3/d, assuming two ORs with I_{deal} = 400 model runs each. To identify the near-optimal well position (fov = 7.5 m^3/d), I_{deal} has to be nearly doubled, and then 8% of the ORs could find the optimum. The total number of model runs required to guarantee solutions for fov* adds up to 12,000. Increasing the tournament size to s = 4 (pop_{SGA} = 30, with p_c = 0.5) leads to similar results for fov = 8.6 m^3/d but to slightly higher numbers of model runs at lower values of fov.

[40] For a more detailed view, Figure 5 depicts the cumulative frequency histogram for the three given values of fov (pop_{SGA} = 20, p_c = 0.6, s = 2). It is noteworthy that the main improvement of the fitness values occurs during the first 1500 function evaluations. A higher number of model runs barely increases the (cumulative) frequency $\phi_{i',fov}$, suggesting that overall, the maximum number of model runs I was sufficient for this study and that most of the SGA-ORs would hardly find a better solution assuming I > 3000.

[41] Figures 4 and 5 reveal the reduction of the MR_{min}(fov) if bookkeeping conditions are conducted for the SGA. The values of MR_{min}(fov) are now calculated by considering new model runs exclusively. The SGA with boundary update, pop_{SGA} = 20 and p_c = 0.6, requires MR_{min}(fov) = 9500 model calls to find a solution for fov*. This means a savings of 2500 model calls compared to the implementation without memorizing previous model runs. Since the repetition of identical individuals from one generation to the next is erratic, the relative savings through bookkeeping as listed in Table 2 are nonuniform though they depend on the parameter setting of the SGA. The higher p_c and the smaller pop_{SGA}, the more the individuals are altered during the search progress. Figure 6 depicts the MR_{min}(fov) for the two extreme variants, the SGA with pop_{SGA} = 75 and p_c = 0.4 compared to the SGA with pop_{SGA} = 10 and p_c = 0.8. Without bookkeeping the values of MR_{min}(fov) are significantly higher for the first setting. Neglecting repeated individuals decreases the MR_{min}(fov), and the resulting curve is only slightly different from the one for the second setting. For the latter, bookkeeping has only a minor effect.

[42] The MR_{min}(fov) plots for the $(3_I,7)$-DES and the $(3_W,7)$-DES are given in Figure 7. A characteristic of both algorithms is that for a broad range of fov the calculated MR_{min}(fov) differs depending on the type of recombination and on the condition of whether the pumping rate boundaries are updated or not. The $(3_W,7)$-DES performs better than the $(3_I,7)$-DES in the pioneer runs. Because of the higher weighting of good individuals the weighted recombination is a more well-directed way of searching the decision space, while the intermediate type implicitly has a higher acceptance of less valued configurations. Therefore weighted recombination might easily lead to getting trapped in suboptimal solutions, although it finds the optimum potentially quicker. The DES with intermediate weighting

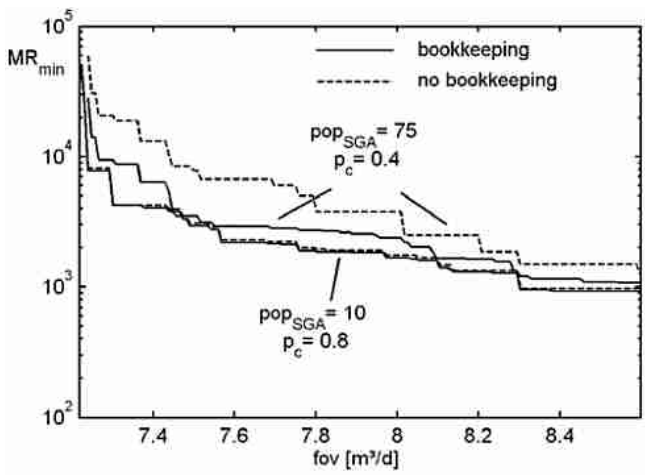

Figure 6. One-well case: Calculated minimum total model runs MR_{min}(fov) of two SGA variants assuming an exponential penalty term, pioneer boundary setting, and s = 2.

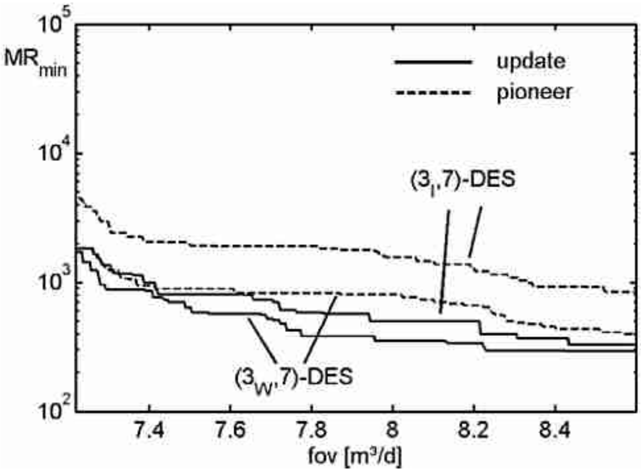

Figure 7. One-well case: Calculated minimum total model runs MR_{min}(fov) of the DES variants assuming pioneer and updated boundary settings (exponential penalty term).

is less susceptible to being deviated to suboptimal solutions, simultaneously requiring a higher number of function evaluations to achieve an objective value. This aspect is reflected for the pioneer runs by the higher ideal number of model runs I_{deal} per OR at intermediate recombination (Table 3). The rate of success of the $(3_I,7)$-DES in finding a solution for the given fov is lower than that of the $(3_W,7)$-DES.

[43] Using the boundary update method leads to a priori exclusion of a broad range of unsatisfactory solutions. When included, these might act as obstacles during the search process, especially in the case of intermediate recombination, as inferior individuals play a more important role during the search process than during weighted recombination. This explains the significant effect of boundary update on the rate of success of $(3_I,7)$-DES: it increases significantly compared to the results for the pioneer ORs and is mostly even higher than that for the $(3_W,7)$-DES. Since, however, the I_{deal} under intermediate recombination typically is greater than under weighted recombination, the MR_{min}(fov) give better results for the $(3_W,7)$-DES.

[44] The boundary update method improves the performance of the $(3_I,7)$-DES but apparently has only a minor effect on the $(3_W,7)$-DES, when the objective value is fov*. We assign this feature especially to the self-adaptive search of the DES. Higher values of fov afford a lower number of model runs, but the self-adaptation of the algorithm to the problem has not emerged very far. Thus the boundary update clearly bears the highest potential to improve the evolutionary search. Increasing the number of model runs leads to a status where solutions which would be excluded with boundary update are only scarcely evaluated. This is especially true if the selection pressure of the good individual is high, such as under weighted recombination. Consequently, boundary update loses importance when increasing the number of model runs.

[45] Comparing the results for the SGA and DES variants, two features are evident: the significantly lower number of minimum model runs MR_{min}(fov) and the uniform shape of the DES trade-off curve compared to the irregular SGA plots. The latter points out the higher reliability of the DES, indicating a less stochastic way of searching. Figure 5 illustrates this aspect for the three selected fov values. Characteristically, for each fov there is one threshold at which the algorithm finds a solution. The SGA performance is hard to predict and has a broad threshold interval.

Table 3. One-Well Case: Success Rates and Ideal Configuration of Optimization Runs for DES Variants Considering an Exponential Penalty Term

	fov, m^3/d	Success Rate,[a] %	I_{deal}	MR_{min}	n_{OR}
Pioneer $(3_W,7)$-DES	7.22	30	400	1850	4.6
Pioneer $(3_W,7)$-DES	7.5	42	300	900	3.0
Pioneer $(3_W,7)$-DES	8.6	88	250	400	1.6
Update $(3_W,7)$-DES	7.22	24	350	1700	4.9
Update $(3_W,7)$-DES	7.5	52	350	650	1.9
Update $(3_W,7)$-DES	8.6	68	200	300	1.5
Pioneer $(3_I,7)$-DES	7.22	20	700	4600	6.6
Pioneer $(3_I,7)$-DES	7.5	34	650	1900	2.9
Pioneer $(3_I,7)$-DES	8.6	72	450	700	1.6
Update $(3_I,7)$-DES	7.22	40	750	1850	2.5
Update $(3_I,7)$-DES	7.5	76	500	800	1.6
Update $(3_I,7)$-DES	8.6	88	200	300	1.5

[a] $I = 3000$.

[46] One of the most important differences between DES and SGA is their performance considering fov* as the objective value. The success rate of DES is several times higher than for SGA and reaches 40% for the $(3_I,7)$-DES compared with only a maximum of 8% for the SGA. Consequently, the number of model runs to find the optimum of the one-well case is significantly lower for DES. Assuming boundary update, the $(3_W,7)$-DES requires five ORs of $I_{deal} = 350$ model runs each, the SGA needs >12 ORs of 750 each. If the objective value is fov >7.5 m^3/h, the SGA has a higher reliability than the $(3_W,7)$-DES. This is reflected by cumulative frequencies $\phi_{i'}$(fov) of up to 92% compared to 68% for the DES variant (fov = 8.6 m^3/d; Figure 5c). To attain $\phi_{i'}$(fov) = 92%, the number of iterations has to be beyond 1500. Table 2 reveals that in this case it is advantageous to restart the SGA at $I_{deal} = 400$ model runs in order to minimize the total required model runs MR_{min}.

[47] The use of the adaptive penalty term yields similar results to those obtained with the exponential penalty term (Table 3), as it is shown in Table 4 for the DES application. In most cases, the calculated MR_{min}(fov) for the adaptive

Table 4. One-Well Case: Success Rates and Ideal Configuration of Optimization Runs for SGA Variants Considering an Adaptive Penalty Term

	fov, m^3/d	Success Rate,[a] %	I_{deal}	MR_{min}	n_{OR}
Pioneer $(3_W,7)$-DES	7.22	30	500	3100	6.2
Pioneer $(3_W,7)$-DES	7.5	38	350	1200	3.4
Pioneer $(3_W,7)$-DES	8.6	80	300	500	1.7
Update $(3_W,7)$-DES	7.22	56	650	1700	2.6
Update $(3_W,7)$-DES	7.5	50	400	1100	2.8
Update $(3_W,7)$-DES	8.6	71	200	500	2.5
Pioneer $(3_I,7)$-DES	7.22	12	650	5250	8.1
Pioneer $(3_I,7)$-DES	7.5	24	500	2050	4.1
Pioneer $(3_I,7)$-DES	8.6	64	350	700	2.0
Update $(3_I,7)$-DES	7.22	40	750	2050	2.7
Update $(3_I,7)$-DES	7.5	48	600	1350	2.3
Update $(3_I,7)$-DES	8.6	74	100	300	3.0

[a] $I = 3000$.

penalty term slightly exceeds the values computed for the exponential term. The main drawback of the adaptive term is that invalid solutions are not rejected sufficiently, even if a rigorous setting of the penalty-specific parameters (κ = 1.1) is used, which results in a severe punishment of invalid solutions. The inspection of the solutions showed that the best well configurations, which had been found by the adaptive penalty term, in part did not guarantee capture of all particles. Four percent of the pioneer DES solutions and 16% of the (ideal) pioneer SGA were invalid. These numbers increase when the decision space is reduced by boundary update: 12% for the (3_W,7)-DES, 30% for the (3_I,7)-DES, and 34% for the SGA. Because of the better results, unless otherwise mentioned, the exponential penalty term is used in the remainder of this study.

6.2. Two-Well Case

[48] In general, there is a chance to decrease $q_{tot,min}$ by using more than one well. Inspection of the distribution of the particle path lines for the undisturbed flow regime (Figure 2) suggests that two wells are likely to yield a considerable reduction. The particles delineate two main flow channels for contaminant transport in the north and south of the well placement area, which points toward an advantage with two extraction wells. In fact, the $q_{tot,min}$ found for the two-well case was 6.15 m^3/d, which means a reduction of ~15% compared to the one-well case. The best found positions for two wells are located near the eastern margin of the well placement area. Figure 8 illustrates the calculated alternatives of optimal well positions. Assuming again fov* = 1.01$q_{tot,min}$ (6.22 m^3/d), the optimization results not only in one single optimum but in a spectrum of two-well configurations. All configurations consist of one well in the NE corner (two alternative positions) and one in the south (nine alternative positions). Note that these represent the solutions found throughout all ORs and there might be more acceptable pumping scenarios. Although the positions are very similar, it is apparent that several (local) optima exist. This usually complicates the optimization procedure. Aside from this, a particularly fine discretization is necessary for the SGA to identify or at least approximate the optimum solution(s). Our investigation showed that by slightly extending the tolerance for fov* to 5% the areas of possible well positions are significantly widened, creating hundreds of different acceptable pumping scenarios.

[49] The problem dimension for the two-well case is N = 6. The reference population size was set to 75 assuming s = 4. We additionally chose pop_{SGA} values of 150 and 300, s = 2, and assumed the same range of crossover rates as for the one-well case in order to test SGA alternatives and identify a problem-specific optimal variant. The string lengths varied, depending on the boundary settings for the pumping rates, between 38 and 42. Since we know from the one-well case that a solution with q_{tot} = 7.22 m^3/d is possible, the upper bound on both individual pumping rates was set q_{up} = 7.22 m^3/d (and for comparison q_{up} = 26 m^3/d). Even if individual pumping rates of the two wells are expected to be lower, a more restrictive formulation by decreasing q_{up} is problematic. An extreme case can exist, where one pumping well is particularly dominating the other ($q_w \approx$ fov). If one wants to prevent disproportionate pumping rates, a lower boundary q_{low} on the individual pumping rates has to be stated, and the individual q_{up} can be

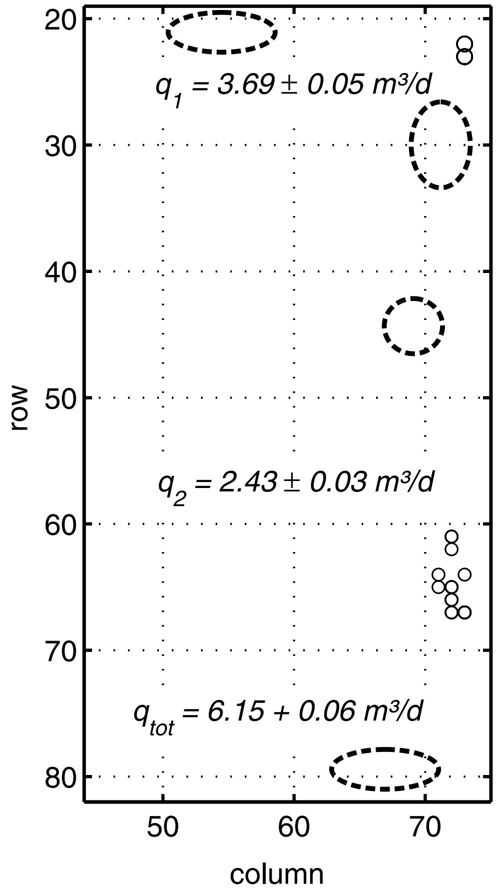

Figure 8. Well placement area with positions of wells for two-well case assuming fov* = 6.22 m^3/d. The dashed circles mark the (additional) preferential areas of well positions when the number of wells is four.

reduced by q_{low}. However, this would not guarantee that the global optimum will be found because the decision space is a priori restricted. Since the primordial objective is to reduce the total pumping rate without further restrictions and since we also wish to find alternative, suboptimal solutions, q_{up} was set to 7.22 m^3/d. In accordance with the one-well case and in order to exclude the one-well case as a local optimum, q_{low} was set to q_{up}/100 = 0.072 m^3/d.

[50] Figure 9 depicts the MR_{min}(fov) for SGA variants with pop_{SGA} = 75 and p_c = 0.4, when boundary update is applied. It is revealed that increasing the tournament size here leads to an improvement for fov > 6.9 m^3/d but performs worse for fov < 6.9 m^3/d. Neither SGA variant could find a solution for fov < 6.22 m^3/h in any OR, no matter whether bookkeeping was conducted or not. In the course of the investigation, pop_{SGA} was raised and pioneer search was compared to the use of boundary update. Figure 10 shows that pioneer search with pop_{SGA} = 150 and p_c = 0.4 gives only slightly worse results than the OR with boundary update for fov > 6.5 m^3/d (assuming bookkeeping, s = 2). This was also true for other SGA variants that have been examined. For lower and thus more demanding values of fov < 6.5 m^3/d, however, boundary update appreciably improved the SGA performance, especially for fov < 6.32 m^3/d, where the pioneer search could not even find solutions in any of the ORs. The significant advantage

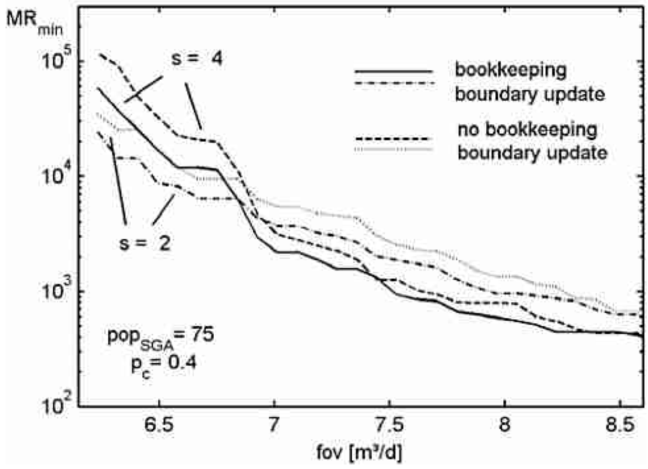

Figure 9. Two-well case: Influence of tournament size and bookkeeping on the calculated minimum total model runs MR_{min}(fov) of two SGA variants.

of boundary update here is that the discretization of the pumping rate changes. As such, the influence of the discretization on the capability of the SGA to hit a fov with the solutions diminishes.

[51] Comparison between the two SGA variants of Figure 10 further reveals that both yield similar results without bookkeeping, indicating a certain insensitivity of the SGA performance to the parameter settings. However, as a smaller crossover rate reduces the number of new individuals in the offspring of each generation, bookkeeping is more efficient for p_c = 0.4 (pop_{SGA} = 150) than for p_c = 0.8 (pop_{SGA} = 300; see Table 5).

[52] Figures 11 and 12 depict the computed values of MR_{min}(fov) for the $(4_I,9)$-DES and the $(4_W,9)$-DES, respectively. As in the one-well case, the best performing optimization algorithm is the DES with weighted recombination. Figures 11 and 12 illustrate the advantage of boundary update and include the experience from the one-well case by setting q_{up} = 7.22 m^3/d. The worst performance can be observed for the DES if a high upper bound of q_{up} = 26 m^3/d

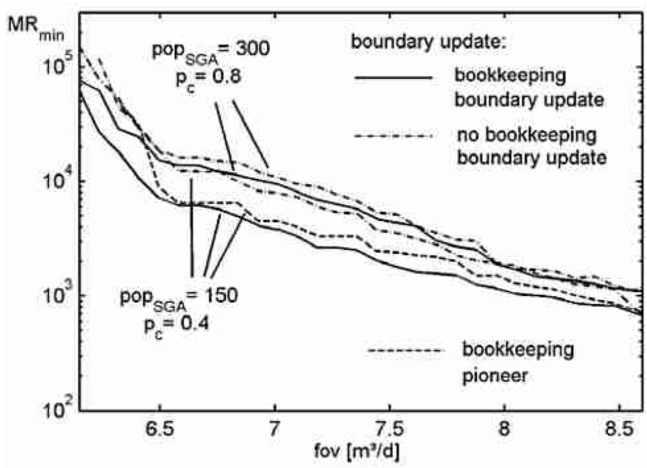

Figure 10. Two-well case: Influence of boundary setting and bookkeeping on the calculated minimum total model runs MR_{min}(fov) of two SGA variants (s = 2).

Table 5. Two-Well Case: Success Rates and Ideal Configuration of Optimization Runs for Four SGA Variants Assuming an Exponential Penalty Term and for One Variant Assuming an Adaptive Penalty Term[a]

	fov, m^3/d	Success Rate,[b] %	I_{deal}	I_{book}	MR_{min}	n_{OR}
SGA (75, 0.4), s = 4	6.22	4	-	-	>50,000	-
SGA (75, 0.4), s = 4	6.45	24	2000	1200	22,000	11
SGA (75, 0.4), s = 4	7.22	56	700	300	1880	2.7
SGA (75, 0.4), s = 2	6.22	10	900	420	25,000	27.8
SGA (75, 0.4), s = 2	6.45	30	2000	1700	8950	4.5
SGA (75, 0.4), s = 2	7.22	48	1100	600	3250	3.0
SGA (150, 0.4), s = 2	6.22	10	2300	5900	25,000	10.9
SGA (150, 0.4), s = 2	6.45	45	2600	7100	7400	2.8
SGA (150, 0.4), s = 2	7.22	80	1500	1300	2650	1.8
SGA (300, 0.8), s = 2	6.22	10	7200	3300	65,000	9.0
SGA (300, 0.8), s = 2	6.45	44	6500	2400	17,000	2.6
SGA (300, 0.8), s = 2	7.22	96	5000	1000	6900	1.4
SGA* (150, 0.4), s = 2	6.22	4	-	-	>50,000	-
SGA* (150, 0.4), s = 2	6.45	34	2500	1550	8600	3.4
SGA* (150, 0.4), s = 2	7.22	88	1800	2950	2550	1.4

[a]The adaptive penalty term is denoted by an asterisk. The results are given for the calculations with updated boundaries.
[b]I = 10,000.

is defined. Especially for the $(4_W,9)$-DES, a substantial improvement is attained if boundary update is applied. For fov* = 6.22 m^3/d the MR_{min}(fov) decreases from over 20,000 to only 5500 model runs for q_{up} = 26 m^3/d and from 7450 to 3500 for q_{up} = 7.22 m^3/d (see Table 6).

[53] Figure 13 plots the histograms of the cumulative frequency $\phi_{i',fov}$ for the updated DES runs. As observed in the one-well case (Figure 5), the $(4_W,9)$-DES is the most reliable algorithm. For q_{up} = 7.22 m^3/d, solutions for fov* are identified at <1000 model runs within approximately two ORs (Table 6). A different picture emerges when preexisting experience of the one-well case is not taken into account and q_{up} is set to a more conservative value of 26 m^3/d. The wider decision space yields a remarkable performance loss for the $(4_W,9)$-DES, while the $(4_I,9)$-DES reveals to be less sensitive to the change of the extent of the

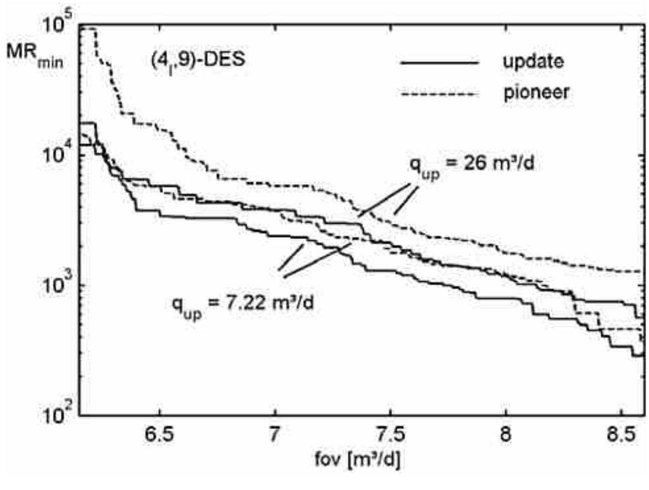

Figure 11. Two-well case: Effect of the (initial) maximum pumping rate q_{up} and of dynamic boundary setting on minimum total model runs MR_{min}(fov) calculated for the DES with intermediate recombination.

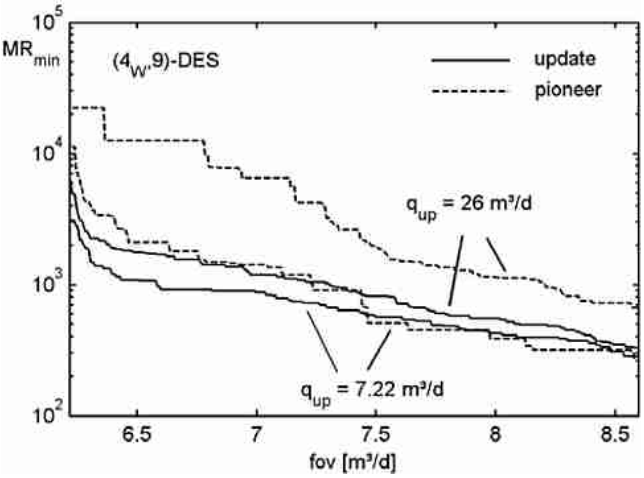

Figure 12. Two-well case: Effect of the (initial) maximum pumping rate q_{up} and of dynamic boundary setting on minimum total model runs MR_{min}(fov) calculated for the DES with weighted recombination.

decision space. The histograms indicate a continuously increasing $\phi_{i',fov}$ with the number of model runs for q_{up} = 26 m^3/d over the related values of $\phi_{i',fov}$. A further increase of $\phi_{i',fov}$ can be expected for I > 4000. Apparently, the success rate of the two DESs with intermediate recombination is not directly correlated to the values of q_{up}. Assuming fov = 6.45 m^3/d, $\phi_{i',fov}$ is larger for q_{up} = 26 m^3/d than for q_{up} = 7.22. For fov = 7.22 m^3/d the reverse relationship is observed.

[54] Although the optimization procedures by DES were conducted without bookkeeping, the number of ideal model runs I_{deal} are generally lower than those of the SGA. As a result, the number of model runs of the investigation per OR was set lower to I = 4000 for the DES application. Only the SGA with pop_{SGA} = 75 and p_c = 0.4 requires similar numbers of I_{deal} as the DES variants. A more stochastic search of the computed SGA results compared to the DES results might be the reason for the higher variability of the SGA. In this context, the 50 ORs per each SGA variant that are conducted in this study seem to be insufficient to clarify the general performance.

[55] As for the one-well case, the performance of the different DES implementations is similar for both penalty term formulations (Tables 5 and 6) and invalid solutions were among the optima, exclusively when the adaptive penalty term was used. Assuming boundary update and q_{up} = 7.22 m^3/d, 40% of the (4$_I$,9)-DES and 48% of the (4$_W$,9)-DES were invalid solutions. Only 10% of the SGA with pop_{SGA} = 150, p_c = 0.4, and s = 2 were invalid, and for the SGA with pop_{SGA} = 300, p_c = 0.4, and s = 2, no invalid solutions were found.

6.3. Four-Well Case

[56] Doubling the number of wells from two to four yields a decline of the minimum total pumping rate from 6.15 to 5.65 m^3/d. This slight decrease is not reached by a completely different (optimal) positioning of the pumping wells. When relaxing fov* by 2.5% over the minimum (fov* = 5.79 m^3/d), several different optimal solutions were found which are dominated by wells positioned at the same coordinates as those of the two-well case. The additional wells are positioned at several preferential areas as illustrated in Figure 8.

[57] The investigated SGA parameter settings for the four-well case are the same as for the two-well case. For all further implementations of the SGA, boundary update was applied and q_{up} is set to 7.22 m^3/d. The number of bits per well, assuming a discretization of >q$_{up}$/100, was 18, and thus each individual consisted of 72 bits. For all SGA variants examined, no more than 3 of 50 ORs were successful in finding solutions for fov*. Again, the efficiency of bookkeeping increases with pop_{SGA}, especially if p_c is low. For the four-well case the SGA with pop_{SGA} = 100, p_c = 0.5, and s = 4 revealed to be the most promising variant for the spectrum of fov (Figure 14).

[58] The DES variants performed much better than the SGA implementations and detected solutions for fov* at a success rate of 36% (Table 7). Again, boundary update means a significant reduction of the required model runs, especially for the (5$_I$,11)-DES, to find solutions for the considered spectrum of fov (see Figure 15). As for the one- and two-well cases, the DES with weighted recombination required the least MR_{min}(fov).

[59] Tables 7 and 8 demonstrate that both SGA and DES reliably find solutions for fov = 7.22 m^3/d(=fov* of the one-well case). The fov* of the two-well case (6.22 m^3/d) represents no major complicacy for the (best) SGA and the DES variants (rate of success is 70-90% for I = 10,000). The difference lies in the required number of model runs: For the SGA with pop_{SGA} = 100, p_c = 0.5, and s = 4 an ideal number of model runs per OR I_{deal} = 2800 is calculated, whereas the I_{deal} of the (5$_W$,11)-DES is only 1800. As a consequence, the (5$_W$,11)-DES requires the lowest MR_{min}(fov) (1600) for this fov. For the best SGA variant, 7300 model runs are necessary.

[60] While the SGA only sporadically finds solutions for fov* = 5.79 m^3/d, the (5$_W$,11)-DES needs three to four ORs of 1600 model runs each to find an optimum. Figure 15 depicts the frequency histograms for the considered values

Table 6. Two-Well Case: Success Rates and Ideal Configuration of Optimization Runs for DES Considering an Exponential Penalty Term[a]

	fov, m^3/d	Success Rate,[b] %	I_{deal}	MR_{min}	n_{OR}
Pioneer (4$_I$,9)-DES	6.22	14	1900	13,800	7.3
Pioneer (4$_I$,9)-DES	6.45	32	1300	5600	4.3
Pioneer (4$_I$,9)-DES	7.22	60	1000	2550	2.6
Update (4$_I$,9)-DES	6.22	18	1600	12,000	7.5
Update (4$_I$,9)-DES	6.45	30	800	3700	4.6
Update (4$_I$,9)-DES	7.22	66	700	2050	2.9
Pioneer (4$_W$,9)-DES	6.22	16	1100	7450	6.8
Pioneer (4$_W$,9)-DES	6.45	44	750	1800	2.4
Pioneer (4$_W$,9)-DES	7.22	70	600	1100	1.8
Update (4$_W$,9)-DES	6.22	30	900	3500	3.9
Update (4$_W$,9)-DES	6.45	60	700	1100	1.6
Update (4$_W$,9)-DES	7.22	72	400	700	1.8
Update (4$_W$,9)-DES*	6.22	30	900	3100	3.4
Update (4$_W$,9)-DES*	6.45	44	650	1750	2.7
Update (4$_W$,9)-DES*	7.22	80	400	750	1.9

[a]For (4$_W$,9)-DES the results are given also for the adaptive penalty term (denoted by an asterisk).
[b]I = 4000.

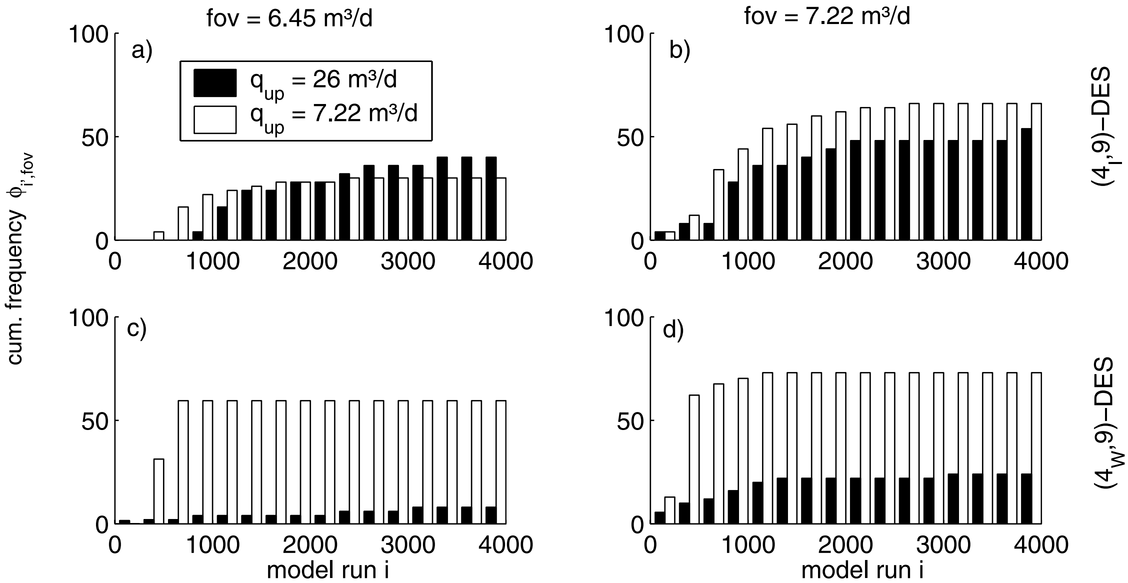

Figure 13. Influence of initial setting of q_{up} on cumulative frequencies of finding two-well case solutions by DES (boundary update).

of fov for the $(5_W,11)$-DES. All solutions are found before 4000 function evaluations. A typical feature has already been observed for the cases with fewer numbers of wells: The variability of the performance of the DES is lower, leading to a relatively steep incline of the cumulative frequency $\phi_{i',fov}$ within a short interval of the number of model runs.

6.4. Eight-Well Case

[61] For eight wells the design approach of *Reed et al.* [2000] predicts a minimum population size of 200 for SGA. Since inspection of the SGA suitability would therefore be extremely time consuming and because of its low reliability in finding optimal solutions for the four-well case, the SGA is not applied for the optimization of the eight-well case. Exclusively used was the DES with weighted recombination and boundary update which generally showed the best performance. The initial upper boundaries were set at q_{up} = 7.22 m^3/d for each well. In a first step the lower boundary was set to q_{low} = 0 m^3/d, accepting solutions with less than eight wells. The minimum total pumping rate ever found, $q_{tot,min}$, was the same as for the four-well case (Figure 16), indicating that no further reduction of the pumping rate is possible by installing more than four wells. Table 9 lists the MR_{min}(fov) for the same target values fov as for the four-well case. Compared with the $(6_W,13)$-DES, the $(5_W,11)$-DES of Table 7 shows that nearly the same success rate can be observed (for I = 10,000). The doubled dimension of the problem, however, leads to a higher number of model runs until solutions are found (Figures 14 and 16). The ideal model runs per OR increases significantly from I_{deal} = 1600 for the four-well case to 3500 for the eight-well case. Accordingly, the MR_{min}(fov*) more than doubles to 13,900. The results for the suboptimal fov, as given in Table 9, show a similar trend at comparable success rates.

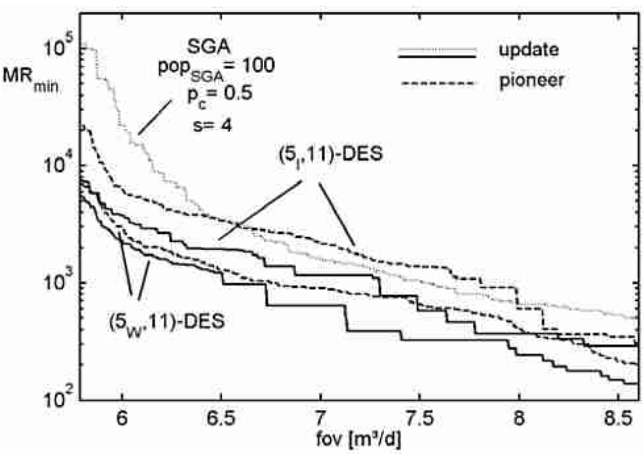

Figure 14. Four-well case: Influence of the setting of q_{low} on the calculated minimum number of model runs MR_{min} and efficiency of boundary update for DES (q_{up} = 7.22 m^3/d) and the best SGA variant with bookkeeping.

Table 7. Four-Well Case: Success Rates and Ideal Configuration of Optimization Runs (With Boundary Update) for the Two DES Variants[a]

	fov, m^3/d	Success Rate,[b] %	I_{deal}	MR_{min}	n_{OR}
Update $(5_I,11)$-DES	5.79	36	2800	7450	2.7
Update $(5_I,11)$-DES	6.22	86	1900	2700	1.4
Update $(5_I,11)$-DES	7.22	100	900	1150	1.3
Update $(5_W,11)$-DES	5.79	36	1600	5500	3.4
Update $(5_W,11)$-DES	6.22	90	1200	1600	1.3
Update $(5_W,11)$-DES	7.22	100	250	400	1.6
Update $(5_W,11)$-DES*	5.79	20	2000	14,400	7.2
Update $(5_W,11)$-DES*	6.22	84	1400	2050	1.5
Update $(5_W,11)$-DES*	7.22	100	350	700	2.0

[a]The results are shown for an exponential penalty term and for the DES with weighted recombination also assuming an adaptive penalty term (denoted by an asterisk).
[b]I = 10,000.

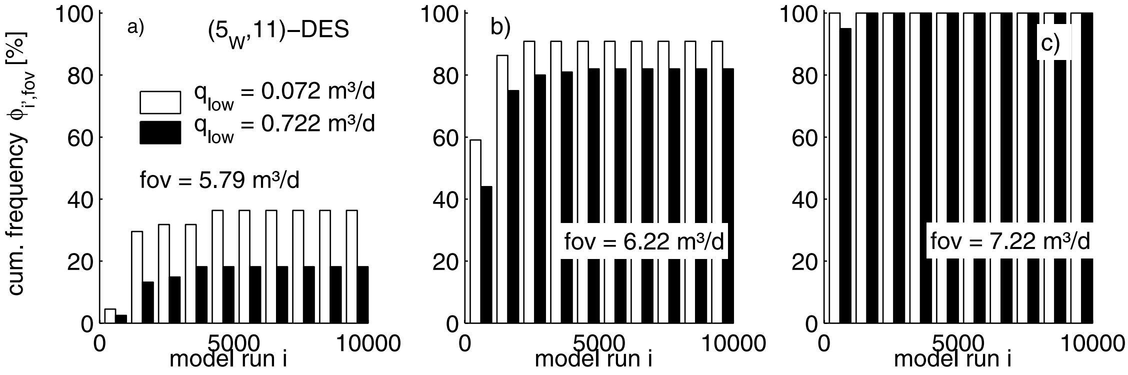

Figure 15. Four-well case: Cumulative frequency of finding solutions for the DES with weighted recombination depending on the lower boundary of the individual pumping rates q_{low} (with boundary update q_{up} = 7.22 m^3/d).

[62] The optimal well configurations of the eight-well case were similar to the ones obtained for the four-well case. Generally, at least two of the eight wells were not active (q_w = 0 m^3/d), two dominated, and the rest were irregularly distributed over the well placement area.

[63] In a second step the DES was forced to find an eight-well solution by setting q_{low} = 0.072 m^3/d. Figure 16 suggests that this reduction of the decision space considerably speeds up the search of suboptimal solutions. Especially for fov = 7.22 m^3/d, not only the success rate rises, but also the MR$_{\min}$(fov) decreases from 1250 to 250 model runs. Obviously, the (6$_\text{W}$,13)-DES was not able to find solutions for fov* = 5.79 m^3/d. This might signify that no case with eight active wells exists, which leads to a lesser total pumping rate than for the four-well case. Since for q_{low} = 0 m^3/d for several optimal solutions more than four wells were active, it seems that at least alternatives to the four-well optimum with more than four wells are possible. Figure 16 shows that if a fov above the minimum pumping rate (5.94 m^3/d) is chosen, solutions are still found with a lower total pumping rate down to 5.82 m^3/d. This proves the assumption that optimally more than four wells have to be installed but eight wells are too many. Though the variable optimal solutions for the eight-well case with q_{low} = 0.072 m^3/d characteristically have several wells with very low pumping rates, these wells must not be omitted as they are necessary to minimize the total pumping rate.

Table 8. Four-Well Case: Success Rates and Ideal Configuration of Optimization Runs for SGA Variants Considering an Exponential Penalty Term (With Boundary Update)

	fov, m^3/d	Success Rate,[a] %	I_{deal}	I_{book}	MR$_{\min}$	n_{OR}
SGA (100, 0.5), s = 4	5.79	2	-	-	>50,000	-
SGA (100, 0.5), s = 4	6.22	72	2800	1100	7300	2.6
SGA (100, 0.5), s = 4	7.22	94	1000	250	1800	1.8
SGA (75, 0.6), s = 2	5.79	6	-	-	>50,000	-
SGA (75, 0.6), s = 2	6.22	80	4800	750	7600	1.6
SGA (75, 0.6), s = 2	7.22	100	1500	150	1900	1.3
SGA (150, 0.4), s = 2	5.79	4	-	-	>50,000	-
SGA (150, 0.4), s = 2	6.22	64	5500	3300	11,600	2.1
SGA (150, 0.4), s = 2	7.22	100	1600	700	2100	1.3
SGA (300, 0.8), s = 2	5.79	0	-	-	>50,000	-
SGA (300, 0.8), s = 2	6.22	52	10,000	3000	15,000	1.5
SGA (300, 0.8), s = 2	7.22	100	4000	500	4500	1.1

[a]I = 10,000.

7. Summary and Conclusions

[64] Optimization of hydraulic capture of contaminated zones in heterogeneous aquifers by pumping wells based on an advective control scheme represents a demanding and time-consuming task. Repetitive numerical modeling of the flow field and particle tracking have to be applied to delineate the pathway of contaminants for each well setting representing a specific solution. This study discusses the formulation of an objective function, which includes the location of a given number of wells and the total pumping rate as well as the feasibility of the respective solution in terms of the degree of capture. Therefore invalid solutions are valued based on the number of particles which are not captured by the selected pumping wells. An exponential and an adaptive penalty term have been tested for their suitability to quantify capture, i.e., to incorporate the number of lost particles into the objective function. Throughout the study it is clarified that assessing capture by particles is an effective but also highly sensitive method. Because of their nonlinear relationship the number of lost particles cannot be brought

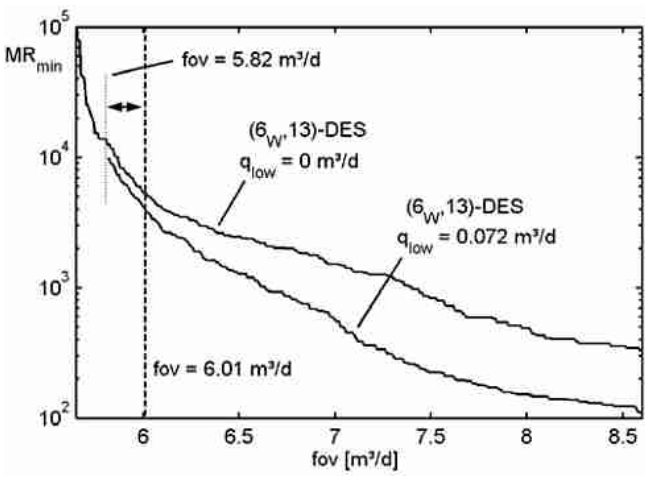

Figure 16. Eight-well case: Influence of the setting of q_{low} on the calculated minimum number of model runs MR$_{\min}$(fov) (boundary update q_{up} = 7.22 m^3/d).

Table 9. Eight-Well Case: Success Rates and Ideal Configuration of Optimization Runs (With Boundary Update) for the DES With Weighted Recombination

	fov, m^3/d	Success Rate,[a] %	I_{deal}	MR_{min}	n_{OR}
$(6_W,13)$-DES[b]	5.79	34%	3500	13,900	4.0
$(6_W,13)$-DES[b]	6.22	88%	1800	3500	1.9
$(6_W,13)$-DES[b]	7.22	98%	600	1250	2.1
$(6_W,13)$-DES[c]	5.79	0%	-	-	-
$(6_W,13)$-DES[c]	6.22	96%	1750	2250	1.3
$(6_W,13)$-DES[c]	7.22	100%	250	350	1.4

[a] I = 10,000.
[b] Here q_{low} = 0 m^3/d.
[c] Here q_{low} = 0.072 m^3/d.

in directly as surrogates for the pumping rate, which would have to be added to achieve complete capture. The results obtained for the example cases considered here indicate that optimization works properly only if a severe penalization of lost particles is applied. Since this can be ideally reached by the exponential penalty term, it has to be preferred over the chosen adaptive formulation, which possibly leads to invalid solutions as optima.

[65] As classical gradient-based techniques are unqualified as solvers for the highly nonconvex objective functions we focus on, evolutionary algorithms are used. SGAs are especially widely accepted as proper optimization tools for pump-and-treat systems, while the DESs represent a promising innovative alternative. In order to compare their performances, their capability of finding the optimal well setting and acceptable suboptimal solutions was analyzed on the basis of the model runs required on average. For this purpose, one issue was included that is omitted in several studies: Although evolutionary algorithms are accepted to be robust problem solvers, they carry out a probabilistic search and, consequently, generally require more than one run of the optimization procedure. During the presented study, for each algorithm-specific parameter setting, 50 model runs are conducted in order to attain a representative picture of the performance. In doing so, the ideal number of model runs per a calculated number of optimization runs and the minimum total number of model runs MR_{min}(fov) could be identified.

[66] The analysis reveals key characteristics of and notable differences between the DES and the SGA for the advective control problem.

[67] 1. The advective control problem comprises two types of decision variables: discrete well coordinates and continuous pumping rates. Since the DESs are real-value based and the SGAs work with a binary representation of decision variables, none is perfectly suitable for the problem. The DES requires a minimum mutation step size for positioning wells; for the SGA, a compromise has to be found between accuracy and representability for the discrete formulation of the pumping rates. During our investigation, the minimum step size exerted only a minor influence on the well positioning by the DES. However, the performance of the SGA is constricted concerning the numerical resolution of the pumping rates. This is demonstrated especially for the two-well case, where a low percentage increase of the minimum total pumping rate leads to a significantly raised number of potential well settings. A considerable fine discretization of the pumping rate interval is necessary, which means expanded binary strings and, accordingly, a more demanding evolutionary search. One beneficial manipulation was revealed in the boundary update approach incorporating the experience from previous optimization runs into the following run. By reducing the decision space and thus the bit length for each pumping rate and by an imposed rediscretization for each new optimization run, in nearly all instances, boundary update led to a reduction of MR_{min}(fov).

[68] 2. Because of the discrete representation of the SGA, bookkeeping of solutions that were already analyzed during the optimization process can be conducted. This is especially applicable for high population sizes and low crossover rates since bookkeeping turned out to be an adequate method to lower the number of model runs required to find a solution. Though genetic drift has to be omitted, stagnation of the algorithm is not implicitly a disadvantage when bookkeeping is conducted. In this case algorithm stagnation as a measure of the search process is not an appropriate stop criterion.

[69] 3. One major difference between the analyzed DES and SGA variants is the number of model runs needed to find a solution. For all cases that have been analyzed the DES with weighted recombination was particularly superior to the best of the herein considered SGA variants. The (μ_W, λ)-DES increasingly outperformed the SGA with problem dimension and optimality of the objective value fov. For the example considered here the (μ_W, λ)-DES shows an approximately linear time complexity, which is also shown for other recent DES implementations [*Hansen et al.*, 2003]. Typically, significantly more optimization runs with also a higher number of model runs I_{deal} have to be conducted to find a solution with the SGA. The SGA seems to be useful in detecting suboptimal solutions but is revealed to be inefficient in fine-tuning the decision parameters. Especially in the case of a high number of wells, the use of SGAs proves not to be favorable because of the increasing size of the binary strings. The reliability decreases significantly even if the number of iterations per optimization run is extended beyond 10,000. These observations reflect conclusions of the analysis on temporal salience by *Thierens et al.* [1998], who derive an exponential increase of the upper boundary of the estimated convergence time for SGA with the string length.

[70] Especially for high-dimensional problems, alternatives to the common SGAs are GA implementations with a decreased time complexity, such as the messy GA by *Goldberg et al.* [1993]. Modified SGA implementations are available that depending on the particular problem, can represent considerable enhancements in the algorithm's performance. For instance, *Minsker and Reed* [2002] suggest using multiple population sizes instead of one "ideal" size. Starting from arbitrarily small pop_{SGA} ($\sim$5 to 10), for every restart of the SGA, pop_{SGA} is doubled and the best previously found individual is injected into the random initial population of the next OR. In this way, relatively small population sizes are exploited in order to precondition SGA's search and obtain a more reliable algorithm.

[71] The DES has a huge potential to compete with other algorithms conventionally used for optimal groundwater

remediation design. Particularly because of the self-adaptive and accordingly reliable search of solutions, it should be considered as a true alternative to other optimization algorithms, especially in the case of complex, high-dimensional problems with real-valued decision variables. However, since each advective control problem has an individual complexity, further investigations are necessary in order to enable an ideal application. During this study, algorithm termination through stop criteria was omitted, which might be a useful extension to simplify the general use.

[72] **Acknowledgments.** Support for this project is provided by the German Department of Education, Science, Research and Technology (BMBF), contract 02WT0019. The authors wish to thank Nikolaus Hansen and two anonymous reviewers for their constructive comments regarding this work.

References

Aly, A. H., and R. C. Peralta (1999), Optimal design of aquifer cleanup systems under uncertainty using a neural network and a genetic algorithm, *Water Resour. Res.*, *35*(8), 2523–2532.

Bäck, T., and H.-P. Schwefel (1993), An overview of evolutionary algorithms for parameter optimization, *Evol. Comput.*, *1*(1), 1–23.

Bäck, T., D. B. Fogel, and Z. Michalewicz (2000), *Evolutionary Computation: Basic Algorithms and Operators*, 384 pp., Inst. of Phys., Bristol, U. K.

Bair, E. S., R. A. Sheets, and S. M. Eberts (1990), Particle-tracking analysis of flow paths and travel times from hypothetical spill sites within the capture zone of a well field, *Ground Water*, *28*(6), 884–892.

Bayer, P., M. Morio, C. Bürger, B. Seif, M. Finkel, and G. Teutsch (2001), Funnel-and-gate vs. innovative pump-and-treat systems: A comparative economical assessment, in *Groundwater Quality: Natural and Enhanced Restoration of Groundwater Pollution*, edited by S. F. Thornton and S. Oswald, *IAHS Publ.*, *275*, 235–244.

Bayer, P., M. Finkel, and G. Teutsch (2002), Reliability of hydraulic performance and cost estimates of barrier-supported pump-and-treat systems in heterogeneous aquifers, in *Calibration and Reliability in Groundwater Modelling: A Few Steps Closer to Reality*, edited by K. Kovar and Z. Hrkal, *IAHS Publ.*, *277*, 290–294.

Chang, L.-C., and C.-T. Hsiao (2002), Dynamic optimal ground water remediation including fixed and operation costs, *Ground Water*, *40*(5), 481–490.

Chan Hilton, A. B., and T. B. Culver (2000), Constraint-handling methods for genetic algorithms in optimal pump-and-treat design, *J. Water Resour. Plann. Manage.*, *126*(3), 128–137.

Christ, J. A., and M. N. Goltz (2002), Hydraulic containment: Analytical and semi-analytical models for capture zone curve delineation, *J. Hydrol.*, *262*, 224–244.

Coit, D., A. Smith, and D. Tate (1996), Adaptive penalty methods for genetic optimization of constrained combinatorial problems, *INFORMS J. Comput.*, *8*(2), 173–182.

Colarullo, S. J., M. Heidari, and T. Maddock III (1984), Identification of an optimal groundwater strategy in a contaminated aquifer, *Water Resour. Bull.*, *20*(5), 747–760.

DeJong, K. A. (1975), *An analysis of the behavior of a class of genetic adaptive systems*, Ph.D. thesis, Dep. of Comput. and Commun. Sci., Univ. of Mich., Ann Arbor.

Deutsch, C. V., and A. G. Journel (1992), *GSLIB: Geostatistical Software Library and User's Guide*, Oxford Univ. Press, New York.

Eiben, A. E., and C. A. Schippers (1998), On evolutionary exploration and exploitation, *Fundam. Inf.*, *35*(1–4), 35–50.

Goldberg, D. E. (1989), *Genetic Algorithms in Search Optimization and Machine Learning*, Addison-Wesley-Longman, Reading, Mass.

Goldberg, D. E. (1998), The race, the hurdle, and the sweet spot: Lessons from genetic algorithms for the automation of design innovation and creativity, *ILLIGAL Rep. 98,007*, Dep. of Gen. Eng., Univ. of Ill. at Urbana-Champaign, Urbana.

Goldberg, D. E., K. Deb, H. Kargupta, and G. Harik (1993), Rapid, accurate optimization of difficult problems using fast messy genetic algorithms, in *Proceedings of the Fifth International Conference on Genetic Algorithms*, edited by S. Forrest, pp. 56–64, Morgan Kaufmann, San Mateo, Calif.

Gorelick, S. M. (1987), Sensitivity analysis of optimal groundwater contaminant capture curves: Spatial variability and robust solutions, in *Proceedings of the National Water Well Association Conference: Solving Groundwater Problems With Models*, pp. 133–146, Natl. Water Well Assoc., Denver, Colo.

Greenwald, R. M., and S. M. Gorelick (1989), Particle travel times of contaminants incorporated into a planning model for groundwater plume capture, *J. Hydrol.*, *107*, 73–98.

Guan, J., and M. M. Aral (1999), Optimal remediation with well location and pumping rates selected as continuous decision variables, *J. Hydrol.*, *221*, 20–42.

Hansen, N. (1998), Verallgemeinerte individuelle Schrittweitenregelung in der Evolutionsstrategie. Eine Untersuchung zur entstochastisierten, koordinatensystemunabhängigen Adaptation der Mutationsverteilung, Ph.D. thesis, 107 pp., Tech. Univ. of Berlin, Berlin.

Hansen, N., and A. Ostermeier (1996), Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation, in *Proceedings of the 1996 IEEE Conference on Evolutionary Computation (ICEC '96)*, pp. 312–317, IEEE Press, Piscataway, N. J.

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

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.

Huang, C., and A. S. Mayer (1997), Pump-and-treat optimization using well location and pumping rates as decision variables, *Water Resour. Res.*, *33*(5), 1001–1012.

Javandel, I., and C. F. Tsang (1986), Capture-zone type curves: A tool for aquifer cleanup, *Ground Water*, *24*(5), 616–625.

Lefkoff, L. J., and S. M. Gorelick (1986), Design and cost of rapid aquifer restoration systems using flow simulation and quadratic programming, *Ground Water*, *24*(6), 777–790.

Lobo, F. (2000), The Parameter-less genetic algorithm: Rational and automated parameter selection for simplified genetic algorithm operation, Ph.D. thesis, Univ. Nova de Lisboa, Lisbon.

Maskey, S., A. Jonoski, and D. P. Solomatine (2002), Groundwater remediation strategy using global optimization algorithms, *J. Water Resour. Plann. Manage.*, *128*(6), 431–440.

Massmann, J., and R. A. Freeze (1987), Groundwater contamination from waste management sites: The interaction between risk-based engineering design and regulatory policy: 1. Methodology, *Water Resour. Res.*, *23*(2), 351–367.

Massmann, J. W., R. A. Freeze, L. Smith, T. Sperling, and B. James (1991), Hydrogeological decision analysis: 2. Applications to ground-water contamination, *Ground Water*, *29*(4), 536–548.

McDonald, M. G., and A. W. Harbaugh (1988), A modular three-dimensional, finite difference ground-water flow model, *U.S. Geol. Surv. Tech. Water Resour.*, *Book 6*, Chap. A1.

Minsker, B. S., and P. Reed (2002), Making genetic algorithms work in the real world: Guidelines from competent GA theory, paper presented at 2002 Genetic and Evolutionary Computation Conference, Morgan Kaufmann, New York.

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.

Ophori, D. U., T. Chan, and F. W. Stanchell (1998), Hydrologic response to pumping and contaminant advection in a fractured rock environment, *J. Am. Water Resour. Assoc.*, *34*(1), 57–72.

Ostermeier, A., A. Gawelczyk, and N. Hansen (1994), A derandomized approach to self-adaptation of evolution strategies, *Evol. Comput.*, *2*(4), 369–380.

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*.

Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling (1992), *Numerical Recipes in FortranThe Art of Scientific Computing*, 992 pp., 2nd ed. Cambridge Univ. Press, New York.

Rechenberg, I. (1994), *Evolutionsstrategie '94*, Frommann-Holzboog, Stuttgart, Germany.

Reed, P., B. S. Minsker, and D. E. Goldberg (2000), Designing a competent simple genetic algorithm for search and optimization, *Water Resour. Res.*, *36*(12), 3757–3761.

Rogers, L. L., and F. U. Dowla (1994), Optimization of groundwater remediation using artificial neural networks with parallel solute transport modeling, *Water Resour. Res.*, *30*(2), 457–481.

Shafer, J. M., and L. W. Vail (1987), A screening method for contaminant plume control, *J. Water Resour. Plann. Manage.*, *113*(3), 336–352.

Shan, C. (1999), An analytical solution for the capture zone of two arbitrarily located wells, *J. Hydrol.*, *222*, 123–128.

Thierens, D., D. E. Goldberg, and A. G. Pereira (1998), Domino convergence, drift, and the temporal-salience structure of problems, in *Proceedings of the 1998 IEEE Conference on Evolutionary Computation*, pp. 535–540, IEEE Press, Piscataway, N. J.

van Dijk, S., D. Thierens, and M. de Berg (2002), On the design and analysis of competent Gas, *Tech. Rep. TR-2002-15*, Utrecht Univ., Utrecht, Netherlands.

van Leeuwen, M., A. P. Butler, C. B. M. te Stroet, and J. A. Tompkins (2000), Stochastic determination of well capture zones conditioned on regular grids of transmissivity measurements, *Water Resour. Res.*, *36*(4), 949–957.

Varljen, M. D., and J. M. Shafer (1991), Assessment of uncertainty in time-related capture zones using conditional simulation of hydraulic conductivity, *Ground Water*, *29*(5), 737–748.

Varljen, M. D., and J. M. Shafer (1993), Coupled simulation-optimization modeling for municipal ground-water supply protection, *Ground Water*, *31*(3), 401–409.

Yoon, J.-H., and C. A. Shoemaker (1999), Comparison of optimization methods for ground-water bioremediation, *J. Water Resour. Plann. Manage.*, *125*(1), 54–63.

Zheng, C. (1994), Analysis of particle tracking errors associated with spatial discretization, *Ground Water*, *32*(5), 821–828.

Zheng, C., and P. P. Wang (1999), An integrated global and local optimization approach for remediation system design, *Water Resour. Res.*, *35*(1), 137–148.

P. Bayer and M. Finkel, Center for Applied Geoscience, University of Tuebingen, Sigwartstrasse 10, D-72076 Tuebingen, Germany. (peter.bayer@uni-tuebingen.de)