

# Groundwater Remediation Strategy Using Global Optimization Algorithms

Shreedhar Maskey<sup>1</sup>; Andreja Jonoski<sup>2</sup>; and Dimitri P. Solomatine<sup>3</sup>

**Abstract:** The remediation of groundwater contamination by pumping and injection is generally a long-term and costly strategy. Aquifer cleanup time is a highly nonlinear and nonconvex function of pumping rates. The cleanup objective often involves minimizing or constraining the cleanup time or cleanup cost. Linear programming and nonlinear optimization cannot guarantee the global solution. In this study, four global optimization (GO) algorithms, including a popular genetic algorithm, are used to minimize both cleanup time and cleanup cost taking pumping rates and/or well locations as decision variables. Groundwater flow and particle-tracking models (MODFLOW and MODPATH) and a GO tool (GLOBE) are used. Real and hypothetical contaminated aquifers are considered for application. The results are satisfactory and show that GO techniques can be widely applied in groundwater remediation strategy and planning. The comparison of the performance of algorithms did not reveal a clear winner. The results also show that in the particle-tracking method, excluding few particles from removal can significantly reduce the cleanup time.

**DOI:** 10.1061/(ASCE)0733-9496(2002)128:6(431)

**CE Database keywords:** Ground water; Remedial action; Algorithms; Ground-water management.

## Introduction

The contamination of groundwater is a widespread problem and requires huge investments for its remediation. Many remedial approaches are applied to the problem. In the United States, for example, a removal action (i.e., contaminant removal to clean up the aquifer) is most typically sought by members of the public who are affected by a groundwater contamination problem. The objective of this approach is to remove as much contamination as possible by pumping and injection and return the groundwater to its original quality. Despite the fact that the cleanup approach is very expensive and generally takes many years to clean up, it is in many cases desirable and unavoidable, considering the human health and environment (Gorelick et al. 1993).

Gorelick et al. (1993) described two methods to find the optimum solution, namely: (1) combination of engineering judgement and trial-and-error iteration process; and (2) combination of simulation and optimization. The first approach lacks mathematically formalized checks and balances that might be used to ensure that the hydraulic design is optimal, both with respect to cost and physical control of the contaminant plume.

In the second approach, the simulation is carried out with available groundwater models for flow and transport, and the op-

timization is usually based on the standard linear programming and nonlinear optimization tools. Some examples of the application of this approach include Bogacki and Daniels (1989), Greenwald and Gorelick (1989), Chang et al. (1992), Jonoski et al. (1997); and Willis and Yeh (1987). However, the fact that the optimization problem of pumping strategy for the contaminant plume removal is highly nonlinear and nonconvex has led some researchers to use the so-called global optimization (GO) techniques. In other areas, for example in optimization of industrial design, various GO techniques have been successfully used since the 1960s. With high-speed workstations and PCs, the accessibility of GO increased enormously. In civil engineering, genetic algorithms (GA) (Michalewicz 1996) became very popular in the end of the 1980s and allowed many researchers and practitioners to use a relatively simple and effective optimization technique.

The study of Ahlfeld and Sprong (1998) examined the presence of nonconvexity and the multiple extrema in the feasible region defined by the concentration response function and showed that the response surface is nonconvex and contains multiple local extrema over a wide range of parameter values. Karatzas and Pinder (1993) presented an outer approximation method as an alternative to the traditional method of linear/nonlinear minimization algorithms. Aral and Guan (1996) used a differential genetic algorithm and applied to a hypothetical nonhomogeneous anisotropic aquifer. El Harrouni et al. (1996) presented a similar work by using genetic algorithm and a dual reciprocity boundary element method for groundwater pumping optimization and parameter estimation. Further, Wang and Zheng (1997) coupled groundwater flow and solute transport codes MODFLOW (McDonald and Harbaugh 1988) and MT3D with genetic algorithm for optimization and applied to a hypothetical and a field-scale problem. Some researchers, e.g., Rogers et al. (1995), applied artificial neural networks (ANN) together with genetic algorithms for the optimal design of groundwater remediation schemes. Yoon and Shoemaker (1999) presented the comparison of computational performance of eight optimization algorithms used to identify the most cost-effective policy for in situ bioremediation of contami-

<sup>1</sup>Research Fellow, International Institute for Infrastructural, Hydraulic and Environmental Engineering (IHE), P.O. Box 3015, Delft, The Netherlands. E-mail: maske@ihe.nl

<sup>2</sup>Lecturer, IHE, P.O. Box 3015, Delft, The Netherlands. E-mail: jon@ihe.nl

<sup>3</sup>Associate Professor, IHE, P.O. Box 3015, Delft, The Netherlands. E-mail: sol@ihe.nl

Note. Discussion open until April 1, 2003. Separate discussions must be submitted for individual papers. To extend the closing date by one month, a written request must be filed with the ASCE Managing Editor. The manuscript for this paper was submitted for review and possible publication on September 27, 1999; approved on October 4, 2001. This paper is part of the *Journal of Water Resources Planning and Management*, Vol. 128, No. 6, November 1, 2002. ©ASCE, ISSN 0733-9496/2002/6-431-440/\$8.00+\$0.50 per page.

nated groundwater. The algorithms used in the comparison were described under three classes: (1) evolutionary algorithms; (2) direct search methods; and (3) derivative-based optimization methods.

This paper presents the use of different GO algorithms to determine the optimized combination of pumping rates and well locations for the removal of a contaminant plume using particle tracking. Simulation is carried out considering the contaminant transport travel times based on advection. The four GO algorithms used are: (1) GA; (2) an adaptive cluster covering (ACCO); (3) the controlled random search (CRS4); and (4) a version of multistart clustering (called Multis). Two commonly used modeling codes MODFLOW and MODPATH (Pollock 1994) are used for groundwater flow and particle-tracking, respectively, and the GLOBE system is used for optimization. Use of GO algorithms with particle-tracking methods for contaminant plume removal has not been reported in the previous work. Also, all the publications cited above have used GO algorithms while considering aquifer *cleanup cost* as an objective function taking a fixed time period of remediation. Sometimes, however, it is necessary to clean up contamination as fast as possible for reasons not directly related to costs, such as a legal requirement or a public relations campaign (Greenwald and Gorelick 1989). Therefore, in this study, the simulation-optimization is carried out for both cases with: (1) aquifer *cleanup time*; and (2) aquifer *cleanup cost* as objective functions. In addition, while most of the previous work uses only variants of genetic algorithms (GAs), this paper presents comparative evaluations of various GO algorithms. The method is illustrated with applications to a hypothetical and a real aquifer.

## Simulation-Optimization Approach

### Flow and Particle Tracking Models

In pump-and-treat remediation, pumping rates and well locations are major decision variables. For a given set of decision variables, a flow model updates the hydraulic head (a state variable) and a particle-tracking model computes the particle travel time and path lines. The equation describing the 3D movement of groundwater assuming constant density is expressed as

$$\frac{\partial}{\partial x} \left( K_{xx} \frac{\partial h}{\partial x} \right) + \frac{\partial}{\partial y} \left( K_{yy} \frac{\partial h}{\partial y} \right) + \frac{\partial}{\partial z} \left( K_{zz} \frac{\partial h}{\partial z} \right) + q_s = S_s \frac{\partial h}{\partial t} \quad (1)$$

where  $K_{xx}$ ,  $K_{yy}$  and  $K_{zz}$ =principle components of the hydraulic conductivity along  $x$ ,  $y$  and  $z$  coordinate axes;  $h$ =hydraulic head;  $q_s$ =source/sink term;  $S_s$ =specific storage; and  $t$ =time.

The total cleanup time of the contaminant plume can be viewed as a function of the transport of particles defined at the plume boundary. When every particle reaches a pumping well, the plume is said to be removed. Hence, the travel time of the slowest particle is assumed to be the total cleanup time. Considering the transport by advection only, the time it takes for an unretarded particle to flow to a pumping well is given by the integral along the particle flow path  $S(q)$  as follows (Greenwald and Gorelick 1989):

$$t(q) = \int_s \frac{1}{v(q)} ds \quad (2)$$

where  $q$ =vector of pumping and injection rates;  $t(q)$ =travel time of the particle;  $v(q)$ =velocity in the direction of flow; and  $ds$ =incremental distance in the direction of flow. The nonlinear-

ity of this function (travel time and pumping rates) is mainly due to: (1) the integration over the flow path  $S(q)$ , which is a function of pumping rates; and (2) the velocity in the denominator of the integral, which also is a function of pumping rates. This definition of cleanup time based on particle tracking accounts only for the advective transport of contaminants, and neglects dispersion and sorption. The readers are referred to Greenwald and Gorelick (1989) for a detailed explanation of the nonlinear and multiple extremum nature of the problem.

### Global Optimization Algorithms

Global optimization is aimed at finding the best solution of constrained optimization problems, which (may) have various local optima (Pinter 1996). The GO approach has particular advantage for problems in which other optimization techniques have difficulties due to the existence of multiple extrema and/or difficulties in defining functions analytically. A GO problem with box constraints can be considered as (Solomatine 1998): find an optimizer  $x^*$  such that

$$f^* = f(x^*) = \min_{x \in X} f(x) \quad (3)$$

where the objective function  $f(x)$  is defined in the finite interval (box) region of the  $n$ -dimensional Euclidean space as

$$X = [x \in \mathcal{R}^n: a \leq x \leq b \text{ (for each dimension)}] \quad (4)$$

This constrained optimization problem can be transformed to an unconstrained optimization problem by introducing a penalty function with a high value outside the specified constraints. In real-life cases, when the exact value of an optimizer is difficult to find, we speak about its estimate and, correspondingly, about the minimum estimate.

There are various possible algorithms to search for global minima in GO problems. Comprehensive coverage of these algorithms is found in Pinter (1996) and Törn and Zilinskas (1989). Probably the most popular GO algorithm is the GA. The four GO algorithms used in this study are briefly described below. These and other GO algorithms used in the applied GLOBE system and the issues of tuning their parameters are discussed in more detail by Solomatine (1998, 1999). Application of GLOBE in groundwater model calibration is discussed by Solomatine et al. (1999).

### Genetic Algorithms

Genetic algorithms (GA) belong to a wider class of evolutionary algorithms (EA) that are based on the idea of modeling a search process of natural evolution, although these models are crude simplifications of biological reality. EA search is directed with a random element to avoid local minima and uses the terminology from biology and genetics. For example, given a random sample at each iteration, pairs of parent individuals (points), selected on the basis of their "fitness" (function value), recombine and generate new "offspring." The best of these are selected for the next generation. Offspring may also "mutate," that is, randomly change their position in space. The idea is that fit parents are likely to produce even fitter children. EAs have been developed in three variations: (1) evolution strategies (ES); (2) evolutionary programming (EP); and (3) genetic algorithms (GA). In the last decade, GAs gained a lot of popularity as a general purpose optimization algorithm. In this work, the implementation of a canonic GA was used (Michalewicz 1996). Several versions of GAs and sets of parameters were compared (mutation rate, crossover probability, population size, etc.), including versions with

the “tournament” and “fitness rank” selection, and the one that performed the best was chosen for this study. This is a variant of GA with the “fitness rank” selection, one-point crossover, 15-bit coding of variables, bit mutation, and preservation of the best points (strings) discovered so far (elitism). To prevent redundant reevaluations, in each generation checks are made for the appearance of repetitive strings.

Termination condition appeared to be an important factor influencing the efficiency and effectiveness of GA (Michalewicz 1996, p. 67). Classical GA is normally stopped when a large proportion of the population has a similar minimum estimate. Our experience is that this simplified condition leads to a very large number of function evaluations without an improvement in accuracy. Several termination conditions have been tested and the one that was adopted involves the fractional proximity of the found minimum estimate to the averaged function value of the predetermined percentage of the population, the number of iterations without improvement in the function value, and the total number of iterations  $N_{iterMAX}$ . So GA is run while  $(N_{iter} < N_{iterMAX})$  and  $[(AvgBest_{Niter} - AvgBest_{Niter-1}) / AvgBest_{Niter} < 10^{-5}]$  and (no improvement in global minimum estimator in 15 successive generations) where  $AvgBest_{Niter}$  is the average function value for the 20% of the best strings in iteration  $N_{iter}$ .

### Multistart and Clustering

The basic idea of the family of multistart methods is to apply a search procedure several times and then to choose the global optimum. One of the popular versions of multistart used in GO is based on clustering, that is creating groups of mutually close points that, hopefully, correspond to relevant regions of attraction of potential starting points (Törn & Zilinskas 1989). The region (area) of attraction of a local minimum  $x^*$  is the set of points in  $X$  starting from which a given local search procedure  $P$  converges to  $x^*$ . For the GLOBE systems applied in the present study, a multistart algorithm called *Multis* was implemented. For local search it uses gradient-free Powell-Brent search (Press et al. 1990).

### Adaptive Cluster Covering (ACCO)

This algorithm (Solomatine 1999) is designed as a workable combination of generally accepted ideas of reduction, clustering, and covering. First groups of mutually close points in search space are identified (they represent the most promising subdomains in which to continue the global search) and within each of them rectangular subdomains are associated. Then each subdomain is covered randomly, i.e., a certain number of points are sampled. The values of the objective function are then assessed at each point. Covering is repeated multiple times; each time the subdomain is progressively reduced in size and shifted towards concentrations of points with smallest function values. Variation of the ACCO algorithm, named ACCOL, continues with several local searches started from the “best” points found in ACCO phase.

### Controlled Random Search (CRS4)

The algorithms (Price 1983; Ali and Storey 1994) of controlled random search (CRS) are based on the principle in which the new trial point in search (parameter) space is generated on the basis of a randomly chosen subset of previously generated points. At each iteration, a simplex is formed from a sample, and a new trial point is generated as a reflection of one point in the centroid of the other points in this simplex. If the worst point in the initially generated set is worse than the new one, it is replaced by the latter. The ideas of CRS algorithms have been further extended by Ali and Storey (1994) producing CRS4 and CRS5. In CRS4, if a

new best point is found, it is “rewarded” by continuing search around it; new points are sampled from the multidimensional beta-distribution with the peak “above” the found best point.

### Optimization Problem Formulation

Two optimization problems are formulated: (1) minimization of cleanup cost (installation cost and operation plus maintenance cost of pumping and wells); and (2) minimization of cleanup time. In both cases, the pumping rates and the well locations are decision variables. The upper and lower limits in pumping rates and the specified area for well locations are considered as constraints. In addition, in the case of cost minimization, the limitation in cleanup time is also introduced as a constraint.

### Optimization of Cleanup Time

If the aquifer cleanup time is to be minimized the objective function and the constraints can in general be defined as

$$\begin{aligned} \text{minimize } t &= f(q_1, q_2, \dots, q_n, c_1, r_1, c_2, r_2, \dots, c_n, r_n) \\ \text{subject to } q_{\min} &\leq (q_1, q_2, \dots, q_n) \leq q_{\max} \\ c_{\min} &\leq (c_1, c_2, \dots, c_n) \leq c_{\max} \\ r_{\min} &\leq (r_1, r_2, \dots, r_n) \leq r_{\max} \end{aligned} \quad (5)$$

where  $t$  = cleanup time;  $q_1, q_2, \dots, q_n$  = pumping rates in wells 1, 2, ...,  $n$ ;  $c_1, c_2, \dots, c_n$  = column number (on grid) of wells 1, 2, ...,  $n$ ;  $r_1, r_2, \dots, r_n$  = row number (on grid) of wells 1, 2, ...,  $n$ ;  $q_{\min}, q_{\max}$  = minimum and maximum ranges in pumping rates;  $c_{\min}, c_{\max}$  = ranges in column number for well locations; and  $r_{\min}, r_{\max}$  = ranges in row number for well locations.

### Optimization of Cost

The well installment cost (capital cost) and the operation and maintenance cost per year (annual cost) can be expressed as a function of total pumping rates as

$$\text{capital cost} = C_1 \sum_{i=1}^n q_i^m \quad (6)$$

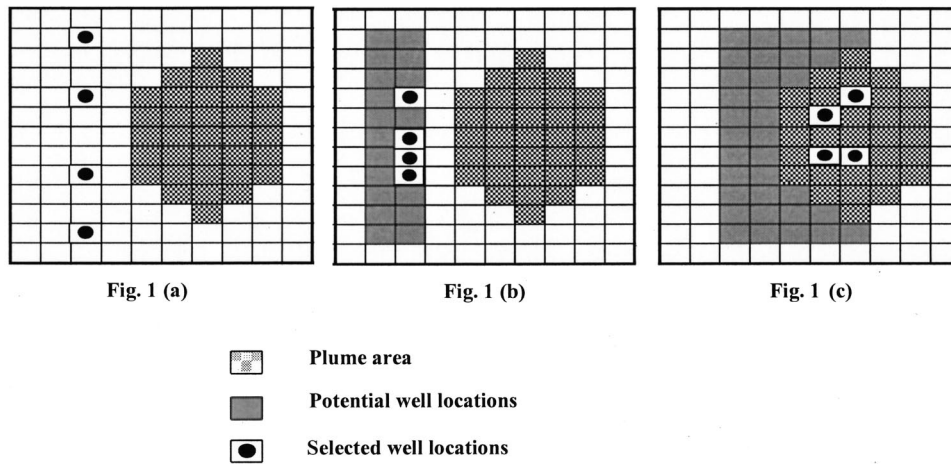
$$\text{operation and maintenance cost per year} = C_2 Q^r \quad (7)$$

The constants  $C_1$  and  $C_2$  depend on the unit rates (per unit pumping rate) of capital cost and annual cost, respectively. The exponents  $m$  and  $r$  are generally less than unity, and they account for the rate of change (generally decrease) in per unit capital and annual costs, respectively, with respect to the increase in total pumping rate. The  $Q$  is the total pumping rate of all wells, and  $q_i$  is the pumping rate of individual wells with  $n$  being the number of wells. Thus, for the optimization of the total cost of well installment and pumping the objective function can be expressed as a function of pumping rates. Expressing the total cost in present worth the objective function and the constraints are defined as

$$\begin{aligned} \text{minimize } C_1 \sum_{i=1}^n q_i^m + C_2 \sum_{k=0}^t \frac{Q^r}{(1+D)^k} \\ \text{subject to } t \leq t_{\max} \end{aligned} \quad (8)$$

The constraints in pumping rates and position arrays apply similarly as in the optimization of cleanup time.

In Eq. (8),  $t$  is the cleanup time in years, and  $D$  is the discount rate (discounted from  $k$ th year). The  $t_{\max}$  is the maximum limit in cleanup time. If the resulting cleanup time is greater than the maximum limit, instead of using Eq. (8), high cost called a pen-



**Fig. 1.** Hypothetical aquifer: Plume area and pumping wells with (a) initially fixed well locations; (b) optimized well locations when potential well locations are outside the plume area (CRS4 solution); and (c) optimized well locations when potential well locations are both inside and outside the plume area (GA solution)

alty cost is used. In this case, the penalty cost was set greater than the cost obtained when all the pumps are operated to their full capacities for the period of  $t_{\max}$ .

The capital cost, Eq. (6) can further be separated into fixed cost and varying cost. But this fixed cost is more site specific and requires more detailed information about the cost in the area concerned. For this type of study, however, it is reasonable to express the capital cost as a function of pumping rates.

## Tools and Techniques

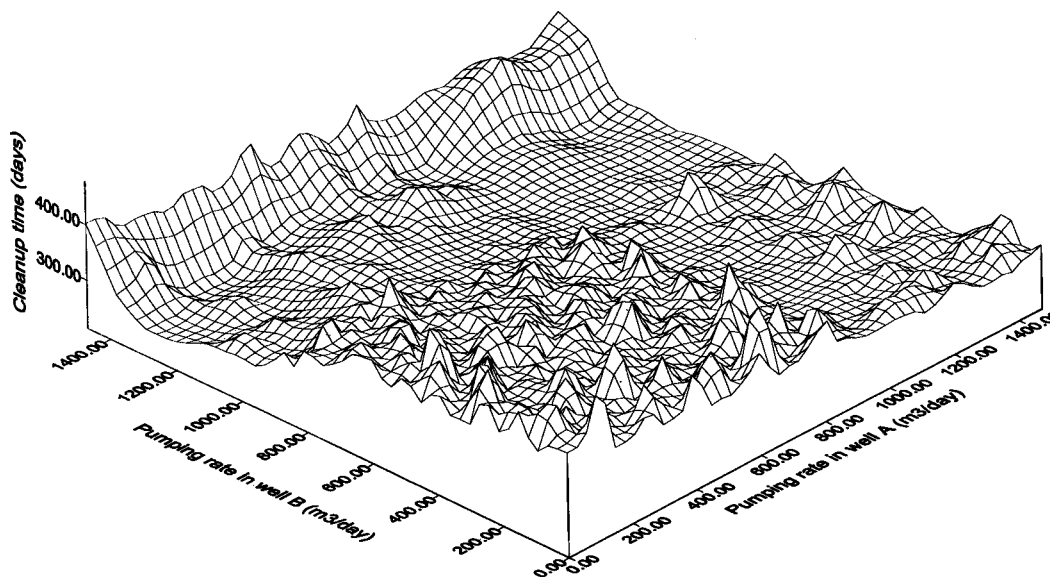
### Tools Used

The groundwater simulation codes MODFLOW (McDonald and Harbaugh 1988) and MODPATH (Pollock 1994) developed by the U.S. Geological Survey are used for flow and particle-tracking, respectively. The global optimization tool GLOBE (Solomatine

1998, 2002) is used for optimization. Brief descriptions about these codes are given below.

MODFLOW is a 3D finite-difference groundwater flow model. It has a modular structure that allows it to be easily modified to adapt the code for a particular application. MODFLOW simulates steady and nonsteady flow in an irregular shaped flow system in which aquifer layers can be confined, unconfined, or a combination of confined and unconfined. Flow from external stresses, such as flow to wells, areal recharge, evapotranspiration, flow to drains, and flow through riverbeds can be simulated. Specified head and specified flux boundaries can be simulated. It is currently one of the most used numerical models for groundwater flow problems.

MODPATH is a particle-tracking postprocessing package that was developed to compute 3D flow paths using output from steady-state or transient groundwater flow simulation by MODFLOW. MODPATH uses a semianalytical particle-tracking



**Fig. 2.** Illustration of nonconvex function of aquifer cleanup time and pumping rates

scheme that allows an analytical expression of the particle's flow path to be obtained within each finite-difference grid cell. Particle paths, their velocities and the travel time are computed by tracking particles from one cell to the next until the particle reaches a boundary, an internal sink/source, or satisfies some other termination criterion.

GLOBE is a PC-based tool built to apply GO techniques originally to the problems of model calibration. GLOBE can be configured to use an external program as a supplier of the objective function values. Currently, GLOBE has nine GO algorithms, which include: two versions of controlled random search (CRS2 and CRS4), two multistart algorithms (Multis and M-Simplex), adaptive cluster covering (ACCO), adaptive cluster covering with local searches (ACCOL), adaptive cluster covering with descent (ACD), adaptive cluster covering with descent and local searches (ACDL), and a version of GA, which performed best as discussed earlier.

### Coupling of MODFLOW and MODPATH with GLOBE

Using the GLOBE as an optimizer requires coupling it with the simulation model (in this case MODFLOW and MODPATH) so that they execute as a single application without the necessity of interactive input during computation. To do this, two sets of executable programs are needed. The first program converts the GLOBE output file (searched values of parameters) as an input file to MODFLOW, whereas the second program takes the output from MODPATH and computes the objective function value to feed to GLOBE. The coupled model starts from the GLOBE and runs in a loop until the selected algorithms generate an acceptable solution and a stopping criterion is met.

### Application Examples

The method is illustrated by applying it to two contaminated aquifer systems—one hypothetical and one real. The hypothetical case is used only for testing the proposed technique. Therefore, it is formulated as a very simple aquifer that can be remediated in a relatively short period. The description of the aquifer and the results obtained in each are discussed separately.

#### Hypothetical Aquifer System

##### General Description

A hypothetical unconfined aquifer system is formulated. The size of this aquifer is 1,500×1,500 m with an average depth of 34.5 m. In plan, the square uniform grid size of 20×20 m is used. Two types of boundary conditions—impervious (or zero flux) at two sides and constant head boundaries at the other two sides—are

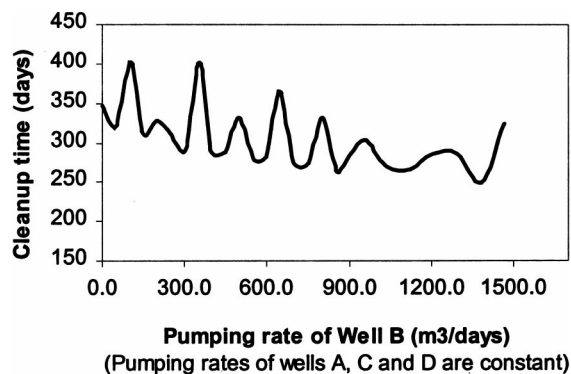


Fig. 3. Illustration of cleanup time variation with pumping rate

specified. The hydraulic conductivity varies from 2 to 30 m/day. The effective porosity of the soil is taken as 0.2, and recharge of 0.0005 m/day is specified throughout the area. A contaminant plume area is defined in the middle of the aquifer, which covers 28 grids in plan. A portion of the aquifer showing the plume area and pumping wells is shown in Fig. 1(a). The plume boundary is defined by placing particles on all external faces. In total, there are 846 particles with 9 particles on each such face.

#### Illustration of the Relationship Between Pumping Rates and Cleanup Time

To illustrate graphically the relationship between pumping rates and plume removal time, a plot of cleanup time against two varying pumping rates is shown in Fig. 2. The pumping was done by four pumping wells positioned in a line, A, C, D, and B, respectively. A constant pumping rate of 1,000 m<sup>3</sup>/day is applied in the two inner pumping wells C and D, whereas the pumping rates in the two outer wells (A and B) are varied in the interval of 50 m<sup>3</sup>/day. The range of pumping rates in these wells is between 0 to 1,500 m<sup>3</sup>/day. Furthermore, the cleanup time is also computed to illustrate the variation in cleanup time by varying the pumping rate in only well B (Fig. 3). In this case, the constant pumping rate of 1,000 m<sup>3</sup>/day is used in all other three wells. Figs. 2 and 3 clearly indicate the nonlinear and nonconvex nature of the cleanup time response function when more than one pumping well is used. Similar results have been shown by Greenwald and Gorelick (1989).

#### Optimization and Results

Optimization is carried out to minimize: (1) cleanup time and (2) cleanup cost with pumping rates and well locations as decision variables. In both cases, four wells are used for pumping the aquifer with initially assumed positions across the regional flow

Table 1. Hypothetical Aquifer: Cleanup Time Optimization—Pumping Rates as Decision Variable

GO algorithms	Number of model runs	Cleanup time (days)	Pumping Rates (m <sup>3</sup> /day)				Total
			Well-1	Well-2	Well-3	Well-4	
Without optimization		1,039	1,500	1,500	1,500	1,500	6,000
GA	509	192	1,210	1,070	1,100	740	4,120
ACCO	353	191	1,480	1,000	950	1,190	4,620
CRS4	658	188	1,170	1,120	1,070	760	4,120
Multis	1,583	207	390	1,290	930	970	3,580

**Table 2.** Hypothetical Aquifer: Cleanup Time Optimization—Pumping Rates and Well Locations as Decision Variables

GO algorithms	Number of model runs	Cleanup time (days)	Pumping Rates (m <sup>3</sup> /day) and Well Locations (row, column)				Total
			Well-1	Well-2	Well-3	Well-4	
Wells outside the plume							
GA	948	153	1,310 (35, 34)	730 (38, 34)	970 (40, 34)	580 (43, 34)	3,590
ACCO	630	153	1,310 (35, 34)	970 (38, 34)	1,020 (41, 34)	150 (42, 34)	3,450
CRS4	1,400	151	1,160 (36, 34)	730 (38, 34)	390 (39, 34)	630 (40, 34)	2,910
Wells inside & outside the plume							
GA	1,352	45	1,000 (36, 37)	740 (37, 37)	710 (39, 37)	520 (39, 38)	2,970
ACCO	759	54	1,000 (34, 35)	900 (37, 37)	860 (39, 37)	480 (40, 35)	3,240
CRS4	982	51	550 (37, 38)	1,020 (38, 37)	260 (39, 36)	760 (40, 37)	2,590

Note: The values in parentheses are row and column numbers of the well position.

direction as shown in Fig. 1(a). For each well, the range of pumping rates are defined from 0.0 (no pumping) to 1,500 m<sup>3</sup>/day. The results are discussed below.

Cleanup time is calculated without using the optimization technique keeping the maximum pumping rate (1,500 m<sup>3</sup>/day) in all wells with the initially assumed locations. The cleanup time for this case is found to be 1,080 days. Then, the GO technique is applied by varying only the pumping rates while maintaining the initial fixed well locations. Four algorithms (GA, ACCO, CRS4, and Multis) are used. The best optimal solutions (aquifer cleanup time) given by all four algorithms are approximately 200 days with the total pumping rates (sum of pumping rates of four wells) between 3,580 and 4,620 m<sup>3</sup>/day (Table 1). Interestingly, all optimal solutions and corresponding total pumping rates are substantially lower than the cleanup time of 1,080 days resulting from the total pumping rate of 6,000 m<sup>3</sup>/day. In this case, CRS4 gives the best result (188 days) requiring 658 model runs and ACCO uses the fewest model runs (353), resulting in 191 days of cleanup.

Optimization is also carried out varying both the pumping rates and well locations. Two cases are used, one with potential well locations outside the plume area and one with potential locations both inside and outside the plume area. The results clearly show that well locations are equally important for the optimized design of aquifer cleanup system and that the cleanup time can be

reduced considerably by selecting appropriate well positions (Table 2). Huang and Mayer (1997) explicitly used well locations as decision variables and showed that it is important. The result of this study coincides with their conclusion. The results obtained from three algorithms (GA, ACCO, and CRS4) are shown in Table 2, and the well locations for the best solutions are presented in Figs. 1(b) and 1(c). The algorithm Multis is dropped from this and further experiments, as it could not terminate the optimization process.

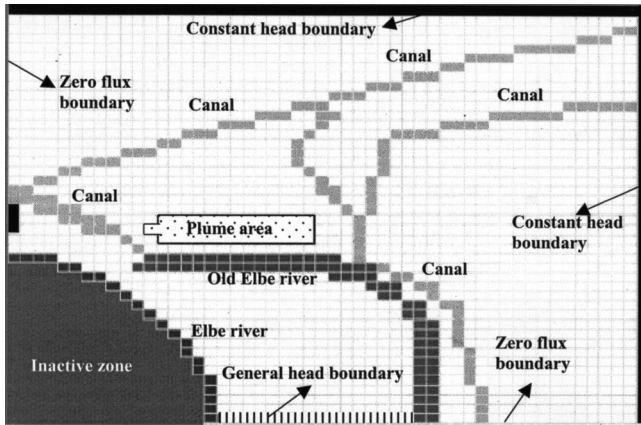
The aquifer system is then optimized with respect to the total cost of cleanup, Eq. (8). In this case, only the pumping rates are taken as decision variables. The coefficients  $C_1$  and  $C_2$  are assumed to be 300 and 730 respectively. The coefficient  $m$  is taken as 0.75 whereas the  $r$  is taken equal to 1. The annual discount rate  $D$  is taken as 8.0%. The resulting cost will have arbitrary monetary unit (MU). In addition to the limit in maximum pumping rate of 1,500 m<sup>3</sup>/day, the maximum limit in cleanup time of 5 years is introduced as a constraint.

Among the three algorithms used, CRS4 provides the most cost effective solution with the cleanup cost of 15,437 (Table 3). This solution removes the plume in 1,482 days (4.06 years). The best solution given by the same algorithm when optimizing in cleanup time (Table 1) requires 24,913, but removes the plume in 188 days only.

**Table 3.** Hypothetical Aquifer: Cleanup Cost Optimization—Pumping Rates as Decision Variables

GO algorithms	Cleanup time (days)	Pumping Rates (m <sup>3</sup> /day)				Cleanup Cost (MU)	
		Well-1	Well-2	Well-3	Well-4	Optimization in cost	Optimization in time <sup>a</sup>
GA	462	55	83	1,350	0	17,368	25,261
ACCO	455	28	1,461	55	28	18,079	27,985
CRS4	1,482	28	28	413	28	15,437	24,913

<sup>a</sup>This cost is calculated for the cleanup time and pumping rates from Table 1.



**Fig. 4.** Real aquifer: Plume area, boundary conditions and rivers/canals positions

### Real Aquifer System

#### General Description

A contaminated aquifer in Europe is considered. This aquifer is contaminated due to the waste disposal from industries, mainly the chemical works. Two dumping sites, one for solid waste of about 300 by 60 m in size and the other for distillation liquid residues of about 60 m in diameter, are responsible for the contamination of the aquifer. The site was used for waste disposal without any protection against leachate for 15 years. Due to the lack of information on the exact plume dimensions, the entire area underneath these dumping sites is considered as the plume. The area of interest comprises quaternary fluvial sediments of the river. The sediments consist of clayey loam with a varying content of sand, sandy clay, and locally clayey fine-grained sand. Their thickness varies between 2.5 and 8.0 m. The groundwater flow is affected by the seepage from the old Elbe River and drainage caused by a canal.

#### Assumptions, Hydraulic Parameters, Boundary Conditions and the Plume Boundary

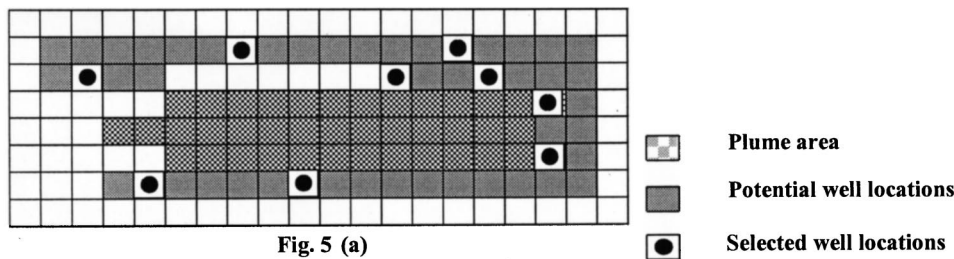
A single-layered unconfined heterogeneous isotropic aquifer is assumed. The total area of 1,040 by 880 m is considered for simulation with square uniform grid of 20×20 m. The boundary condition along the north and east (Fig. 4) is defined as the constant head boundary. On the West Side, a zero-flux boundary is used except for the three cells next to the canal where a constant head boundary is specified. On the southern side, zero-flux boundary and constant head boundary are defined. All rivers and canals are incorporated for the simulation specifying appropriate fluxes.

An extreme variation in the hydraulic conductivity is observed with the minimum of 0.75 m/day to the maximum of 75 m/day. In terms of recharge rates, the area is divided into three sectors. Sector 1 reflects the regional recharge rate of  $2.074 \times 10^{-4}$  m/day. Sector 2 is the plume area with the recharge rate of  $1.037 \times 10^{-4}$  m/day (half of the regional value), whereas Sector 3 covers the landfill area with an impervious liner, and no recharge is considered.

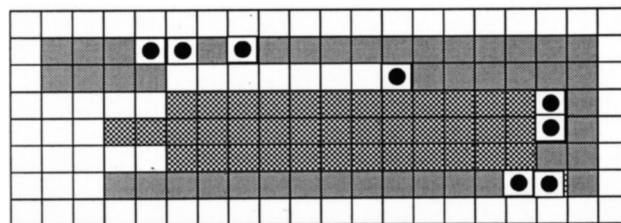
The plume boundary is defined by specifying particles on the external faces of all grid cells containing the plume. Each such face has 9 particles, and altogether 990 particles envelop the plume. The illustration of the plume area together with the initial locations of the pumping wells is shown in Fig. 5(a). As seen in this figure, all potential well locations are considered outside the plume area. The area inside the plume is assumed inaccessible for well construction.

#### Optimization and Results

In this aquifer area, as given by test wells, the maximum yield from each well is only 25.92 m<sup>3</sup>/day. Therefore, to consider pumping rates as a decision variable has no significance. It is for this reason that the optimized solution is computed considering only the well locations as a decision variable. Nine pumping wells with the initial positions shown in Fig. 5(a) are assumed, and the constant pumping rate of 25.92 m<sup>3</sup>/day is assigned to each well. In this case, optimization in cleanup time is carried out using two algorithms, GA and ACCO. For the initially located well posi-



**Fig. 5 (a)**



**Fig. 5 (b)**

**Fig. 5.** Real aquifer: Plume area, potential well locations and pumping wells with (a) initially fixed well locations, and (b) optimized well locations (GA solution)

**Table 4.** Real Aquifer: Cleanup Time Optimization—Well Locations as a Decision Variable

Pumping well	Well Location (row, column)		
	Initial (assumed)	Given by GA	Given by ACCO
Well-1	(21, 16)	(21, 13)	(21, 14)
Well-2	(21, 23)	(21, 14)	(21, 16)
Well-3	(22, 11)	(21, 16)	(21, 21)
Well-4	(22, 21)	(22, 21)	(22, 12)
Well-5	(22, 24)	(22, 21)	(22, 21)
Well-6	(23, 26)	(23, 26)	(22, 26)
Well-7	(25, 26)	(24, 26)	(26, 22)
Well-8	(26, 13)	(26, 25)	(26, 23)
Well-9	(26, 18)	(26, 26)	(26, 27)
Cleanup time (years)	35	13	14.8
Number of model runs		1,158	860

tions, the resulting cleanup time is 35 years. The optimal solutions given by GA and ACCO are 13 and 14.8 years, respectively. The result is shown in Table 4. In this example, GA gives the best solution but ACCO performs faster. The optimized well locations found by GA are shown in Fig. 5(b).

In particle-tracking methods, some particles may enter cells with very low or no outflow or travel through very long paths and thereby take a very long time to be removed. Because of these particles, the cleanup strategy may be unreasonably expensive. In practice, however, it is almost impossible to remove 100% contamination from an aquifer. Therefore, to exclude the effect of such particles in total cleanup time, the computation is repeated considering only 98% particle removal for the well configuration shown in Fig. 5(a). Significantly enough, more than 20% reduction in cleanup time is achieved by excluding 2% of the particles. The comparison is shown in Table 5. It is to be noted that there is no established relationship between the percentage of particles removed and the concentration of the contaminants.

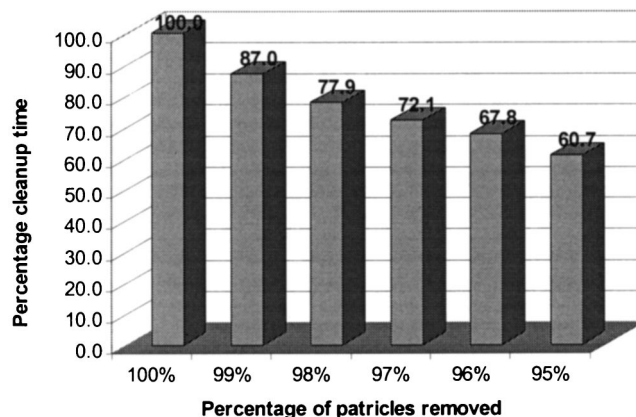
It is difficult to say how many particles to exclude, and the decision should be made based on the designer's judgment and the particular situation. To better understand the effect of excluding some particles, the travel time is computed for different percentages of particle removal for the same well configuration, Fig. 5(a). In this example, the cleanup time is reduced to 87% considering 99% removal of the particles, and it is reduced to 60.7% for 95% particle removal. The result is illustrated in Fig. 6.

### Comparison of Algorithms

It would be important for a practitioner to have an idea which GO algorithm to use. Comparison of algorithms is always a difficult task. It depends not only on the criteria used, but also on the characteristics of the particular case studies used, details of imple-

**Table 5.** Real Aquifer: Comparison of Cleanup Time for 100 and 98% Particle Removal

GO algorithms	Particles Removal 100%		Particles Removal 98%		Reduction in cleanup time (%)
	Number of model runs	Cleanup time (years)	Number of model runs	Cleanup time (years)	
GA	1,158	13.0	744	10.0	23.0
ACCO	860	14.8	989	11.3	23.6



**Fig. 6.** Illustration of the reduction in plume removal time when some particles are excluded

mentation, proper tuning of algorithms, etc. The authors did their best to ensure equal "starting conditions" for all four algorithms. Solomatine (1998, 1999) compared the performance of several GO algorithms using three performance indicators: (1) Effectiveness (how close the algorithm gets to the global minimum, or, if the global minimum is unknown, how low its found estimate is); (2) Efficiency (running time) of an algorithm measured by the number of function evaluations needed (assuming the running time of the algorithm itself is negligible compared to the time required for the function evaluation); and (3) Reliability (robustness) of the algorithm measured by the number of successes in finding the global minimum, or at least approaching it sufficiently closely. The comparison of computational performance of optimization methods carried out by Yoon and Shoemaker (1999) are also based on similar criteria.

In this study, four algorithms are compared based only on "effectiveness" and "efficiency" criteria. Due to the popularity of GA, special attention was given to the selection of the best performing version and its proper tuning (covered above). The hypothetical aquifer with the well locations shown in Fig. 1(a) was used as an example. The optimum solutions obtained from different algorithms are quite close but require noticeably different number of model runs. It is necessary to note that in different cases or runs the best results are obtained from different algorithms. In all cases, ACCO was the fastest to reach the solution with 315 to 759 model runs which is, on an average, less than two thirds the model runs of GA and CRS4. The GA and CRS4 required 509 to 1,352 and 473 to 1,400 model runs, respectively. Such differences in model runs (computation time) may become critical when more complex groundwater models or systems are used. However, CRS4 and GA were more effective, i.e., were able to find optimum more accurately.

Fig. 7 illustrates a typical run. It must be stated that the comparison presented here is based only on one run. Proper compari-



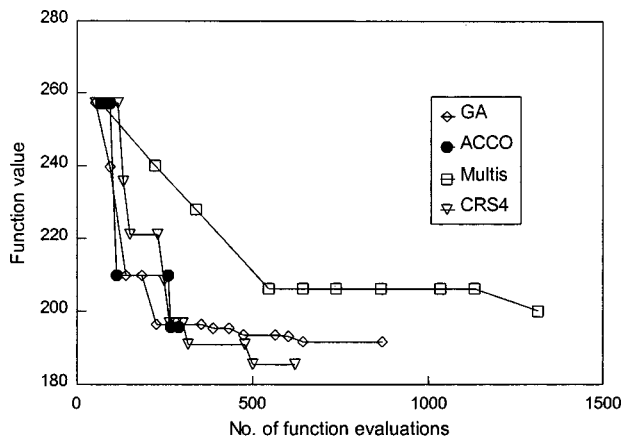


Fig. 7. Comparison of the performance of algorithms

son should involve more algorithms and their variations and be based on averaging on several runs. It should also be stated that ACCO and CRS4 algorithms are optimized for real-valued problems or problems with high discretization. GA, by design, is oriented at discrete-valued problems, where it shows higher performance.

## Conclusions

This study introduces the optimization of pumping and injection strategy for contaminated groundwater remediation as a problem of multiple extrema (GO problem) and shows the applicability of GO techniques in such problems. Such techniques do not require computation of derivatives as required in more traditional optimization methods and can be applied in groundwater modeling problems where commercial off-the-shelf (black-box) software is used. In these cases, the details of the underlying equations are not known. The comparison of the four global optimization algorithms did not reveal a clear winner. ACCO was the fastest, but CRS4 and GA were more accurate. Due to the popularity of GA, its code could be easily obtained from the Internet, and if this is the case, GA would be a good choice. GA could be also the best choice in case of discrete-valued variables. However, if time allows to apply several GO algorithms, this should be done. It would be beneficial for a decision maker to have several solutions. It is important to note that all of the presented algorithms use randomized search, so the solutions found are normally different in different runs and for different algorithms. Fig. 3 shows that many points (solutions) have close function values (cleanup time and consequently cost). The role of the modeler and decision maker would then be to choose an appropriate engineering solution from several "good" solutions found by various algorithms/runs.

This study also shows that the cleanup time and, therefore, the cleanup cost are very sensitive to both pumping rates and well locations. Satisfactory results are obtained both in the minimization of cleanup time and cleanup cost taking pumping rates and/or well locations as decision variables.

In particle-tracking method, the estimated cleanup time may be very high due to some particles which may enter cells with very low or no outflow or travel through very long paths. This study shows that excluding the effect of few such particles can significantly reduce the cleanup time. But the decision of how

many particles to exclude is solely dependent on the designer's judgment and the particular situation.

## References

- Ahlfeld, D. P., and Sprong, M. P. (1998). "Presence of nonconvexity in groundwater concentration response function." *J. Water Resour. Plan. Manage.*, 124(1), 8–14.
- Ali, M. M., and Storey, C. (1994). "Modified controlled random search algorithms." *Int. J. Comput. Math.*, 53, 229–235.
- Aral, M. M., and Guan, J. (1996). "Optimal groundwater remediation design using differential genetic algorithm." *Computational Methods in Water Resources*, XI (1), 349–357.
- Bogacki, W., and Daniels, H. (1989). "Optimal design of well location and automatic optimisation of pumping rates for aquifer cleanup." *Proc., Int. Symposium on Contaminant Transport in Groundwater*, Balkema, Rotterdam, The Netherlands, 363–370.
- Chang, L.-C., Shoemaker, C. A., and Liu, P. L.-F. (1992). "Optimal time-varying pumping rates for groundwater remediation: Application of a constrained optimal control algorithm." *Water Resour. Res.*, 28(12), 3,157–3,173.
- El Harrouni, K., Ouazar, D., Walters, G. A., and Cheng, A. H.-D. (1996). "Groundwater optimization and parameter estimation by genetic algorithm and dual reciprocity boundary element method." *Eng. Anal. Boundary Elem.*, 18(4), 287–296.
- Gorelick, S. M., Freeze, R. A., Donohue, D., and Keely, J. F. (1993). *Groundwater contamination: Optimal capture and containment*. Lewis, Boca Raton, Fla.
- Greenwald, R. M., and Gorelick, S. M. (1989). "Particle travel times of contaminants incorporated into a planning model for groundwater plume capture." *J. Hydrol.*, 107(2), 73–98.
- Huang, C., and Mayer, A. S. (1997). "Pump-and-treat optimization using well locations and pumping rates as decision variables." *Water Resour. Res.*, 33(5), 1,001–1,012.
- Jonoski, A., Zhou, Y., Nonner, J., and Meijer, S. (1997). "Model-aided design and optimization of artificial recharge-pumping systems." *Hydro. Sci. J.*, 42(6), 937–953.
- Karatzas, G. P., and Pinder, G. F. (1993). "Groundwater management using numerical simulation and the outer approximation method for global optimization." *Water Resour. Res.*, 29(10), 3371–3378.
- McDonald, M., and Harbaugh, A. W. (1988). *Techniques of water-resources investigations of the U.S. Geological Survey: A modular three-dimensional finite-difference groundwater flow model*, 6, Scientific Software Group, Washington, D.C., 8.1–8.6.
- Michalewicz, Z. (1996). *Genetic algorithms+data structures=evolution programs*, Springer, Berlin.
- Pinter, J. D. (1996). *Global optimization in action. Continuous and Lipschitz optimization: Algorithms, implementation and applications*. Kluwer, Dordrecht, The Netherlands.
- Pollock, D. W. (1994). *MODPATH/MODPATH Plot user's guide, version 3: A particle tracking post-processing package for MODFLOW*, USGS, Reston, Va.
- Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. (1990). *Numerical recipes in Pascal. The art of scientific computing*. Cambridge Univ. Press, Cambridge, Mass.
- Price, W. L. (1983). "Global optimization by controlled random search." *J. Optim. Theory Appl.*, 40, 333–348.
- Rogers, L. L., Dowla, F. U., and Johnson, V. M. (1995). "Optimal field-scale groundwater remediation using neural networks and the genetic algorithm." *Environ. Sci. Technol.*, 29(5), 1,145–1,155.
- Solomatine, D. P. (1998). "Genetic and other global optimization algorithms—comparison and use in calibration problems." *Proc., 3rd Int. Conf. on Hydroinformatics*, Balkema, Rotterdam, The Netherlands, 1,021–1,027.

- Solomatine, D. P. (1999). "Two strategies of adaptive cluster covering with descent and their comparison to other algorithms." *J. Global Optim.*, 14(1), 55–79.
- Solomatine, D. P. (2002). Global optimization tool GLOBE, ([www.ihe.nl/hi/sol/global.htm](http://www.ihe.nl/hi/sol/global.htm)).
- Solomatine, D. P., Dibike, Y. B., and Kukuric, N. (1999). "Automatic calibration of groundwater models using global optimization techniques." *Hydro. Sci. J.*, 44(6), 879–894.
- Törn, A., and Zilinskas (1989). *Global optimization*, Springer, Berlin.
- Yoon, J.-H., and Shoemaker, C. A. (1999). "Comparison of optimization methods for ground-water bioremediation." *J. Water Resour. Plan. Manage.*, 125(1), 54–63.
- Wang, M., and Zheng, C. (1997). "Optimal remediation policy selection under general conditions." *Ground Water*, 35(5), 757–764.
- Willis, R., and Yeh, W. W-G. (1987). *Groundwater systems planning and management*, Prentice-Hall, Englewood Cliffs, N.J., 337–342.