Optimal design of aquifer cleanup systems under uncertainty using a neural network and a genetic algorithm

Alaa H. Aly
Utah State University Research Foundation, Logan

Richard C. Peralta
Department of Biological and Irrigation Engineering, Utah State University, Logan

Abstract. We present a methodology to account for the stochastic nature of hydraulic conductivity during the design of pump-and-treat systems for aquifer cleanup. The methodology (1) uses a genetic algorithm to find the global optimal solution and (2) incorporates a neural network to model the response surface within the genetic algorithm. We apply the methodology for a real example and different optimization scenarios. The employed optimization formulation requires few hydraulic conductivity realizations. The presented approach produces a trade-off curve between reliability and treatment facility size.

1. Introduction

A major objective of many groundwater remediation systems is to reduce contaminant concentrations to below certain levels. This is often accomplished by extracting contaminated groundwater and treating it at the surface. Then treated water can be injected to the aquifer via injection (recharge) wells. This is the pump-and-treat approach to groundwater remediation. Determining well locations and their pumping rates is most important for designing pump-and-treat systems.

Numerous simulation/optimization (S/O) models that combine groundwater flow and transport models with operations research methods have been developed to help design groundwater remediation systems [e.g., *Gorelick et al.*, 1984; *Ahlfeld*, 1990; *Whiffen and Shoemaker*, 1993; *Rogers and Dowla*, 1994; *Hegazy and Peralta*, 1997]. Reviews by *Gorelick* [1983] and *Wagner* [1995] describe S/O models developed for groundwater management.

Two approaches have been used for representing simulation constraints within optimization models. In the first approach, simulation equations are used as constraints inside the optimization model [e.g., *Aguado and Remson*, 1980; *McKinney and Lin*, 1993; *Gharbi and Peralta*, 1994; *Takahashi and Peralta*, 1995].

In the second approach, simple expressions are used to describe state variables (e.g., contaminant concentrations) as functions of pumping rates. These simple expressions can be obtained using Taylor series or curve-fitting methods [*Alley*, 1986; *Lefkoff and Gorelick*, 1990; *Sawyer et al.*, 1995; *Ejaz and Peralta*, 1995; *Cooper et al.*, 1998]. In this study we use a neural network to represent the simulation constraints inside the optimization model. Neural networks are described in a later section.

Several researchers applied nonlinear optimization to aquifer cleanup problems [*Gorelick et al.*, 1984; *Ahlfeld*, 1990; *Gharbi and Peralta*, 1994; *Peralta et al.*, 1995; *Peralta and Aly*, 1996]. Nonlinear programming techniques cannot guarantee global optimality when applied to large nonconvex problems. For real problems, where the time required to simulate the groundwater system is significant, nonlinear programming methods may need prohibitive amounts of CPU time.

The limitations of mathematical programming have motivated researchers to use alternative optimization techniques such as simulated annealing [*Rizzo and Dougherty*, 1996] and genetic algorithms (GAs) [*McKinney and Lin*, 1993; *Ritzel et al.*, 1994; *Rogers and Dowla*, 1994]. *Ritzel et al.* [1994] found that a GA performed better than mathematical programming for nonlinear and mixed-integer nonlinear problems. *McKinney et al.* [1994] found that using a GA to compute the starting point for a nonlinear gradient-based optimization algorithm provided significant advantages and allowed them to locate solutions that are approximately globally optimal.

A combination of neural networks and a GA was used by *Rogers and Dowla* [1994]. They found that this combination involved less computational burden and more flexibility than mathematical programming methods. However, *Rogers and Dowla* [1994] used a discrete representation of pumping rates. Wells were either pumping at their maximum capacity or not pumping. In the present study pumping rates are allowed to range between the upper and lower limits in prescribed small increments. *Aly and Peralta* [1997] used neural networks and a genetic algorithm in the design of an aquifer cleanup system to reduce the concentrations of two contaminants simultaneously. *Dowla and Rogers* [1995] provide a summary of neural networks' applications in hydrogeology.

Optimization methods rely on the prediction accuracy of flow and transport models used to represent the aquifer. Since accurate modeling of any aquifer can be very difficult, developed optimal strategies may not be optimal for the real aquifer system. There is a growing attention to considering the stochastic nature of aquifer parameters while designing remediation strategies. *Gorelick* [1990] discusses some techniques used to account for uncertainty in designing groundwater management systems. In the following section we describe the most significant proposed approaches and discuss their applicability.

Design of pump-and-treat systems is often complicated by the random nature of aquifer parameters. Three general techniques have been used for solving groundwater management problems under uncertainty. In the first the sources of uncertainty are not defined, but it is assumed that optimal pumping

Paper number 98WR02368.
0043-1397/99/98WR-02368$09.00

rates can be modified after a period of implementation and monitoring [*Jones et al.*, 1987; *Whiffen and Shoemaker*, 1993]. In this technique the differences between variable values predicted via optimization and the measured variable values (obtained from the field after the optimal strategy is implemented) are used to guide subsequent modification of the optimal strategy. The relation used to modify the computed optimal strategies is termed a feedback law. The process is continued as the modified optimal strategy is implemented.

In the second technique a probability distribution is either derived or assumed for the variables of interest. Then analytical relations are developed to relate the quantiles of this distribution to the decision variables. These analytical relations are used as constraints in the optimization problem. These constraints are termed chance constraints, and the resulting optimization model is known as the chance-constrained model [*Cantiller and Peralta*, 1989; *Peralta and Ward*, 1991].

In the third stochastic groundwater management technique a group of constraints is formulated, each for a different realization of the uncertain aquifer parameters [*Wagner and Gorelick*, 1987]. A realization is a set of the uncertain parameters' values. Typically, each realization is generated from the probabilistic model of the uncertain parameters. The resulting optimal strategy must satisfy all (or some) of the realizations simultaneously. The idea is to find optimal strategies that are robust (satisfy all management constraints) for a range of the uncertain parameters. Several studies tried to estimate the reliability of optimal strategies computed using the multiple-realization technique [*Morgan et al.*, 1993; *Chan*, 1993, 1994].

All cited studies concluded that in order to assure a design that has a high level of reliability, at least 50 to 100 realizations are needed [*Chan*, 1993, 1994; *Morgan et al.*, 1993]. For large problems, where the time required to simulate the system is significant, the time required to generate all the constraint equations can be prohibitive. However, since the response surfaces for different realizations can be evaluated simultaneously, one can greatly speed the process by computing them in parallel. Another possible remedy is to determine whether some realizations can be dropped without having to carry out the optimization [*Gomez-Hernandez and Carrera*, 1994; *Ranjithan et al.*, 1993]. *Karatzas* [1997] used a robust optimization approach and tried to develop an effective methodology for selecting critical realizations.

In this study we present and apply an approximation method that develops the trade-off curve between system size (total flow) and estimated reliability. The rest of the manuscript is organized as follows. In section 2 we describe the problem and outline the proposed methodology. In section 3 we provide an overview of neural networks and describe the neural network used in this study. In section 4 we introduce the genetic algorithm and its implementation. In sections 5 and 6 we show an application of the proposed approach and our conclusions.

2. Problem Statement and Solution Methodology

Consider an aquifer having a dissolved contaminant plume to be addressed via a pump-and-treat (P&T) system. The P&T system will use a combination of extraction and injection wells along with a treatment facility. It is desirable to determine the size of the treatment facility, well locations, and pumping schedules. The design must assure with (1-α) reliability that concentrations at the end of the planning period are below a prescribed value, where α is a prescribed probability of failure.

We approximate the well-location-determination problem by selecting a number of potential locations for pumping wells. The optimization model will compute a pumping rate for each of these well locations. This pumping rate can be zero, indicating that no well is needed at this location. This common approximation greatly simplifies the analysis. Few studies have attempted to use well locations as decision variables [*Wang and Ahlfeld*, 1994; *Huang and Mayer*, 1997].

2.1. Optimization Problem Formulation and Process Overview

Assume M^P possible extraction and injection wells and a treatment facility of size P^{MAX}. Maximum total extraction rate equals treatment facility size. For a particular pumping strategy (set of pumping rates) define CMAX_i as the maximum concentration remaining in the aquifer at the end of the planning period for the ith realization. A solution of the formulated optimization problem is a pumping strategy that achieves acceptable aquifer cleanup by the end of the planning period with probability (1-α). The solution process overview is as follows:

1. Select a treatment facility size (P^{MAX}).
2. For the selected size, compute the optimal pumping rates that minimize $\mathrm{CMAX}_{(\mathrm{NR})}$ (defined below in (1)). This step is detailed in section 2.2.
3. Use Monte-Carlo simulation to determine the reliability of the pumping strategy (developed in step 2). Reliability is approximated as the fraction of simulations for which CMAX_i does not exceed a target concentration value (usually the maximum contamination limit, MCL).

Steps 1 through 3 are repeated for different selections of P^{MAX}. The results can be represented as a curve that shows reliability versus P^{MAX}. The appropriate value for P^{MAX} is then selected from the curve to achieve the desired reliability.

In step 2 we define $\mathrm{CMAX}_{(\mathrm{NR})}$ as the L_∞ norm of concentrations resulting from a single pumping strategy applied to NR realizations.

$$\mathrm{CMAX}_{(\mathrm{NR})} = \max\,(\mathrm{CMAX}_1, \mathrm{CMAX}_2, \cdots, \mathrm{CMAX}_{\mathrm{NR}}) \qquad (1)$$

The process is repeated for each set of evaluated pumping rates. This approach results in a single concentration constraint within the optimization model while assuring cleanup for all considered realizations simultaneously. In step 2, $\mathrm{CMAX}_{(\mathrm{NR})}$ minimization is motivated by the idea that once the treatment facility size is selected, it is desirable to use the available system to reduce contaminant concentrations as much as possible.

If NR is large (e.g., greater than 100) in step 2, there would be no need for step 3. However, for real problems, large NR values usually require prohibitive amounts of CPU time. In this study we used 250–300 simulations for each realization. If each flow and transport simulation requires 10 min of CPU time, then each realization requires more than 41 hours. Simulations for 100 realizations require more than 170 days of CPU time. Many real problems require more than 10 min of CPU time to simulate flow and transport. This discussion ignores the fact that these simulations can be run in parallel to reduce time requirements. We are merely trying to illustrate the size of the problem at hand.

Here we propose a new approximation approach to reduce the number of realizations used in step 2 (e.g., 5–20 instead of hundreds). According to the results of *Chan* [1994], if NR realizations are used and the pumping strategy developed in step 2 achieves a $\mathrm{CMAX}_{(\mathrm{NR})}$ value equal to target concentration value, then the reliability of this pumping strategy is approximately NR/(NR + 1). However, when the treatment facility size is large enough and a large number of potential wells is used, the resulting pumping strategies will usually achieve $\mathrm{CMAX}_{(\mathrm{NR})}$ values that are less than the target concentration. Therefore each pumping strategy is expected to have a greater reliability than NR/(NR + 1). If NR is 5, each pumping strategy is expected to have a reliability greater than 83%. However, the simulations in step 3 are needed to determine the actual reliability. This approach reduces the number of realizations by about 80% while assuring that developed pumping strategies achieve the desired reliability. In a later section we show how the choice of NR affects the design.

2.2. Optimization Problem for Each P^{MAX} Value

In the described formulation, the objective function is to minimize $\mathrm{CMAX}_{(\mathrm{NR})}$. One constraint limits total pumping from exceeding the maximum flow rate that the treatment facility can handle (P^{MAX}). Another constraint forces total extraction to equal total injection.

Minimize $\mathrm{CMAX}_{(\mathrm{NR})}$ subject to

$$p^L(\hat{e}) \le p(\hat{e}) \le p^U(\hat{e}) \qquad \hat{e} = 1, 2, \cdots, M^p \tag{2}$$

$$\sum_{\hat{e}=1}^{M^{ex}} |p(\hat{e})| \le P^{\mathrm{MAX}} \tag{3}$$

$$\sum_{\hat{e}=1}^{M^{ex}} p(\hat{e}) = \sum_{\hat{e}=M^{ex}+1}^{M^p} p(\hat{e}) \tag{4}$$

$$\mathrm{CMAX}_{(\mathrm{NR})} = f_C[p(1), p(2), \cdots, p(M^p)] \tag{5}$$

where M^{ex} is number of extraction wells; $p^L(\hat{e})$ and $p^U(\hat{e})$ are lower and upper bounds, respectively, for the pumping rate at location $\hat{e}$ [$L^3\ T^{-1}$]; and P^{MAX} is maximum allowed pumping from all extraction wells [$L^3\ T^{-1}$]. Here P^{MAX} is the flow capacity of the treatment facility.

We use a response-surface approach to define the function f_C within the optimization model. Few forms have been suggested in the literature for representing contaminant concentrations relations to pumping rates. *Alley* [1986] found that simple linear regression provided enough accuracy for predicting solute concentrations. However, in our study simple linear regression was not adequate to represent $\mathrm{CMAX}_{(\mathrm{NR})}$ as a function of pumping rates.

Lefkoff and Gorelick [1990] used regression to approximate transport of salt mass and found that this simplified the analysis. However, they did not show the employed functional form. *Ejaz and Peralta* [1995] used multiple linear regression to fit approximating expressions as surrogates for solute transport equations to solve a stream wastewater loading problem. *Cooper et al.* [1998] used a polynomial function to describe light nonaqueous phase liquid transient removal via an extraction well.

In this study, owing to the complex nature of the $\mathrm{CMAX}_{(\mathrm{NR})}$ surface as a function of pumping rates, we were unable to approximate the response surface using a polynomial equation with a reasonable number of terms. Instead, we used a neural network (NN) to represent $\mathrm{CMAX}_{(\mathrm{NR})}$.

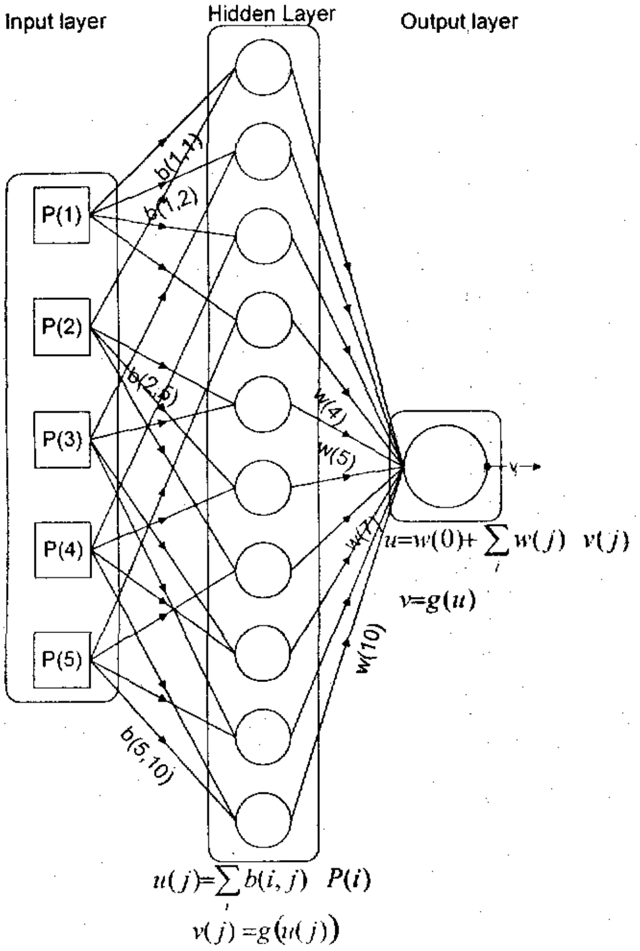


Figure 1. Neural network architecture for five wells.

3. Neural Networks

Neural networks (NNs) have received much attention in many disciplines. Their use has grown owing to their widespread acceptance as powerful and flexible forecasting tools as well as to their applicability to almost any problem. Initially, neural networks were developed as an attempt to emulate the parallel processing nature of the human brain. Biophysics suggests that man's cognizant power can be attributed to our own biological neural networks. Billions of neurons, making thousands of chemical and electrical connections, endow us with sensory perception, rationalization, and adaption skills. An NN attempts to perform the same functions, although not nearly as efficiently. Much the same way that humans learn by pattern recognition, synaptic training, and experience, an NN is trained to rationalize through repetitive learning and generalization.

In this study we use a multilayer, feed-forward, error back-propagation neural network (Figure 1). The network is composed of multiple processing elements organized in a series of two or more mutually exclusive layers [*Hertz et al.*, 1991].

The first layer is called the input layer and is used to receive the stimuli (pumping rates). Pumping rates are scaled before they are used as inputs to the first layer. The scaling is linear and is used to make any individual pumping rate generate a value that is smaller (in magnitude) than 0.9 when used as input to the sigmoid function described below. The last layer (output layer) is used to receive the responses of the network and produce a value between −1 and 1. This value is then linearly scaled to compute $\mathrm{CMAX}_{(\mathrm{NR})}$. Input $\mathrm{CMAX}_{(\mathrm{NR})}$ values are linearly scaled to be between −0.9 and 0.9.

There is one hidden layer between the input and output layers. Each node in the hidden layer receives input from two nodes in the input layer. That is, if there are M^p nodes in the input layer (M^p wells), there are $\binom{M^p}{2}$ (M^p choose 2) nodes in the hidden layer. There is only one node in the output layer. This node receives input from all nodes in the hidden layer and produces a single value.

More hidden layers can increase the predictive capability of the network, especially to classify patterns that are not linearly separable [*Zurada*, 1992], but one hidden layer is sufficient for most applications [*Cybenko*, 1988]. *Poggio* [1983] showed that a neural network with two hidden layers can represent any continuous function. However, designing the structure of a neural network for a specific application is best decided by the problem at hand. In this study a single hidden layer provided a good approximation of the predicted variable ($\mathrm{CMAX}_{(\mathrm{NR})}$) for all tested scenarios.

In any layer, all nodes work independently and concurrently. All nodes in the network (except those in the input layer) perform two kinds of computations: determining the net-input value to the node and computing the output value. The net-input value to the ith node is

$$\mathrm{net}_i = \sum_j p(j) W_{ij} + \theta_i \tag{6}$$

where j represents all input connections to the node, W is the weight of input pumping rate p, and θ is the bias. In this study, only the output node has a bias weight. Each node converts the net input to an activation value

$$a_i = F_i(\mathrm{net}_i) \tag{7}$$

Then the output value is computed by the output function

$$O_i = f_i(a_i) \tag{8}$$

Usually $a_i = \mathrm{net}_i$ and the output value is

$$O_i = f_i(\mathrm{net}_i) \tag{9}$$

The output function is usually a sigmoid function. A function is sigmoid if it is bounded, monotonic, continuous, and smooth [*Smith*, 1993]. In this study we selected a sigmoid function that is bounded between −1 and 1. Our selected function is

$$O_i = f_i(\mathrm{net}_i) = \frac{2}{1 + e^{-\mathrm{net}_i}} - 1 \tag{10}$$

Each time O_i is computed (from the output layer), it is compared with the desired response (supervised learning). A learning algorithm is used to adjust the weights of the interconnections according to the error obtained from the comparison. A training sample set is applied to the network repeatedly until an equilibrium state is reached or a predefined training period expires. The values of the weights are saved and can be later used to simulate response to any set of pumping rates.

We used backpropagation [*Rumelhart et al.*, 1986] with a supervised learning algorithm. The learning algorithm uses gradient descent to achieve training (or learning) by adjusting the weights to minimize the error measured by the difference between the desired and actual network outputs.

$$E_p = \tfrac{1}{2}(\mathrm{CMAX}_p - O_p)^2 \tag{11}$$

$$E = \sum_p E_p \tag{12}$$

where p is an index for a training sample. The weights are then adjusted by a rule of the form

$$W_i(t) = W_i(t-1) - \lambda_i d_i(t) \tag{13}$$

$$d_i(t) = \frac{\delta E}{\delta W_i(t-1)} \tag{14}$$

where λ_i is the learning rate for weight i and t is an iteration counter. This rule, the steepest-descent method, has two limitations. First, there are no guidelines for selecting a learning rate(s). Second, the method can easily converge to a local minimum.

To avoid the first limitation, we used the delta-bar-delta adaptive learning rule [*Jacobs*, 1988]. This method yields faster convergence than steepest descent and avoids the learning rate selection dilemma. It is not unusual to achieve a target error level in one tenth the number of iterations that would be required using steepest descent and an optimal learning rate [*Smith*, 1993].

The concept of the delta-bar-delta rule is simple. There is a learning rate for each weight in the network. If the direction in which the error decreases as this weight changes is the same as the direction it has been decreasing recently, increase the learning rate. If the direction is opposite of the recent direction, decrease the learning rate.

The direction in which the error decreases is determined by the sign of d_i. If d_i is positive, the error decreases as the weight goes down; if d_i is negative, the error decreases as the weight goes up. The direction of weight change is defined as the average of current and past derivatives. This average change c at iteration t is

$$c(t) = \theta c(t-1) + (1-\theta)d(t) \tag{15}$$

$$0 \le \theta < 1 \tag{16}$$

where θ is a parameter that controls how long "recently" means.

The learning rate for a weight is

$$\begin{aligned} \lambda(t) &= \lambda(t-1) + \kappa \qquad d(t)c(t) > 0 \\ \lambda(t) &= \lambda(t-1)\phi \qquad d(t)c(t) \le 0 \end{aligned} \tag{17}$$

where κ and ϕ are parameters. Once λ is selected, the actual weight update rule is

$$w_m(t) = w_m(t-1) - \lambda_m(t) d_m(t) \tag{18}$$

In practice, the NN performance is not highly sensitive to the choice of values for κ, ϕ, and θ [*Smith*, 1993]. For the training sessions discussed in this paper, actual values that worked well across a variety of problems are $\kappa = 0.1$, $\phi = 0.5$, and $\theta = 0.7$.

The second problem of gradient-based learning is convergence to local minima. A possible remedy to this problem is to start the training algorithm from different starting points (initial guess values for the weights). However, this substantially increases the learning time because one learning session will be required for each initial guess.

Another remedy is to break the data set into batches of similar or different sizes. Batches are used in cycles to compute the weight updates in each iteration. For example, if the training set contains 100 observations, one can break the observations into 5 batches of 20. In the training session the first iteration uses the gradient of the error computed only for observations in the first batch. The second iteration uses the gradient of the error for observations in the second batch, and so forth. The 11th iteration uses the first batch again to find the error derivative. The error is computed for all sets of pumping rates each iteration, and the weights that score the lowest error are retained. Weight updates are computed for different batches until many iterations fail to find weight values that have a lower error than the best weights.

We used 80–85% of total simulations for training and retained the rest for testing. Each training set was separated into 10 equal-size batches. For testing, the test data was introduced into the network and its mean squared error was computed. Then we computed the ratio of this mean squared error to the training set's mean squared error. For all tested problems this ratio was less than 1.45 (for most tested problems this ratio was less than 1.05). This indicates that the NN has been adequately trained and it can be used for simulating the system. A large ratio indicates lack of training or, worse, overtraining. Overtraining is unlikely in our case. When the number of wells is 5, the largest number of weights is 31 and the size of the training set is greater than 250. When the number of wells is 10, the largest number of weights is 46 and the size of the learning set is greater than 350. Overtraining means that the network has memorized the training data set. This problem is unlikely in a network where the number of weights is small compared to the size of the training set.

4. The Genetic Algorithm

Genetic algorithms (GAs) are heuristic rules for searching a solution space to identify the best solution. The use of GAs was first suggested by *Holland* [1975], who based his search on a survival-of-the-fittest rule. Since then, GAs have been used in many disciplines. *Davis* [1991] reviews many important applications of GAs, and *Goldberg* [1989] presents a comprehensive introduction to GAs. In groundwater management, GAs have been used by *McKinney and Lin* [1993], *Ritzel et al.* [1994], *Rogers and Dowla* [1994], *Cieniawski et al.* [1995], and others. In this manuscript we focus on how the GA is implemented to address the subject problem.

The major advantage of GAs is that they are independent of the particular problem being analyzed. A GA requires only an objective (fitness) function that can be evaluated for any set of the control variables. This function can be nonlinear, nondifferentiable, or discontinuous. GAs require only that system performance can be evaluated for any set of the decision variables. In this study the fitness value is the reciprocal of $\mathrm{CMAX}_{(\mathrm{NR})}$. Therefore the GA tries to find the pumping rates that will result in the smallest $\mathrm{CMAX}_{(\mathrm{NR})}$.

We used a GA with the basic reproduction, crossover, and mutation operators. The GA we used is similar to the simple genetic algorithm (SGA) of *Goldberg* [1989]. However, instead of the roulette-wheel selection in the SGA, we use tournament selection [*Goldberg*, 1990].

For the presented problem, one problem with GAs is that they do not provide an explicit method to handle constraints. Instead of explicitly considering constraints, penalty terms are added to the objective (fitness) function. In our formulation, one constraint limits total pumping. An efficient method to handle such a constraint in a GA is to assign a very low fitness value for any set of pumping rates whose sum exceeds the upper bound on total pumping. After a few generations the GA hardly tries to evaluate the fitness value for any set of pumping rates whose sum exceeds P^{MAX}.

We used binary coding where the pumping rate from each well is represented by L digits of the chromosome. For example, when we tried to optimize the pumping rates from five extraction wells, the chromosome length was $5L$. The chromosome length, L, is determined from the desired representation accuracy. For example, if the pumping rate from one well can range between P^L and P^U and the desired accuracy is ε, then

$$L = \left[\log\left(1 + \frac{|P^U - P^L|}{\varepsilon}\right)\right] \Big/ (\log 2) \qquad (19)$$

where the logarithm is taken to any base. For example, when P^U is 800, P^L is zero, the required accuracy is 0.5, and the chromosome length is 11. If we have five such pumping rates, the final chromosome length is 55. Notice that different pumping rates can have different accuracies if desired. Longer chromosomes can be used to achieve the desired accuracy at the expense of more GA run time.

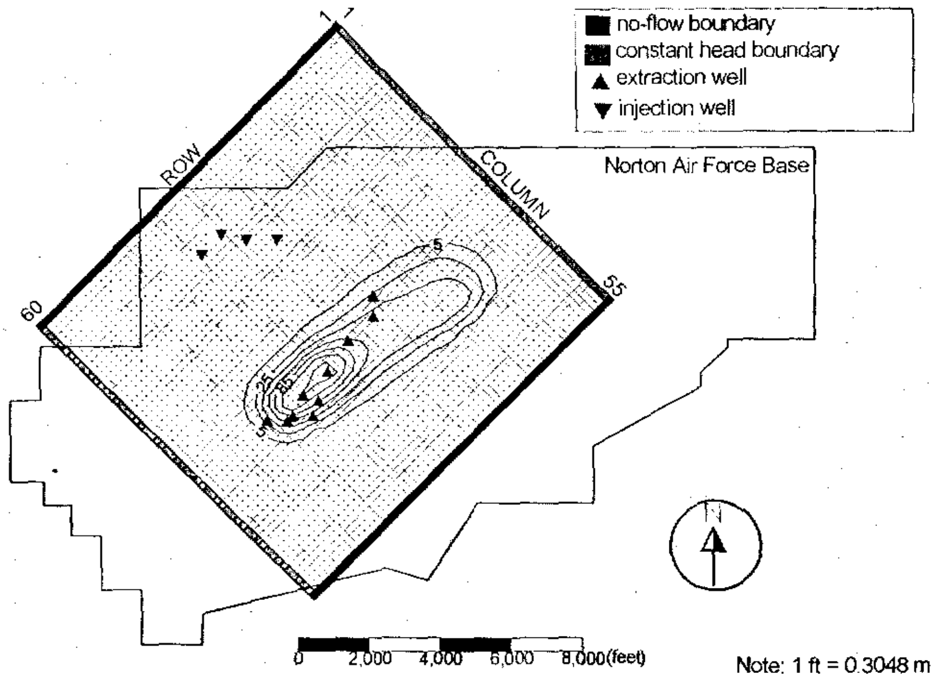
Control parameter selection greatly affects the answer computed by the GA. However, there are no published general guidelines for selecting these parameters. Many studies have attempted to evaluate parameter values that work well under a variety of conditions [*De Jong*, 1975; *Schaffer et al.*, 1989]. However, their results are problem-specific and depend on how the GA is implemented. A major advantage of our proposed methodology is that the GA itself takes very little time. This is because the size of the study area affects only the time required to evaluate the response functions. The neural networks can evaluate the response function in significantly less time than the full simulations. Therefore one can afford to use a robust method like the GA despite the fact that the GA is often considered slow because of the large number of function evaluations. After the response functions are evaluated, the GA takes very little time to find the best set of pumping rates. This allowed us to use the GA for several GA control parameter values.

At least 100 sets of control parameters were tested for each problem discussed below. For the tested problems the best results were obtained using a population size between 50 and 100. Our experience is that larger population sizes do not affect the solution but do require extra time. However, if the number of wells is larger or if only a relatively small subspace provides a feasible solution, a larger population size will probably be needed.

A crossover probability between 0.8 and 0.9 and a mutation probability between 0.08 and 0.12 consistently lead to the best results. Generally, a crossover rate less than 0.7 provided an inferior answer. A mutation rate greater than 0.12 increased the number of infeasible evaluations without improving the final answer. The worst performance of the GA was when the mutation probability was zero. This is expected since mutation prevents the GA from getting trapped at local optima.

5. Implementing the Groundwater Management Model

A data set for a single-layer aquifer contaminated by dissolved trichloroethylene (TCE) is used to apply the proposed



Note: 1 ft = 0.3048 m.

Figure 2. Base boundary, finite difference grid, boundary conditions, well locations, and initial TCE concentrations.

approach. TCE is moving toward nearby municipal wells. One of the suggested approaches is to install a pump-and-treat system near the contaminant source. In the original problem the treatment facility size was determined a priori. However, in this study we determine the treatment facility size and the corresponding pumping rates that will assure cleanup of the aquifer within 3 years with a 95% probability. Figure 2 shows the initial TCE concentrations, the finite difference grid, and boundary conditions. *Peralta and Aly* [1995] provide a more-detailed description of the study area.

MODFLOW [*McDonald and Harbaugh*, 1988] and MT3D [*Zheng*, 1990] are used to simulate groundwater flow and plume migration, respectively. Ten extraction wells and four existing injection wells can be used. Injection wells are near existing pipelines far from the plume center (Figure 2). Preliminary analysis indicated that injection rates have little effect on groundwater hydraulics near the center of the plume. Therefore, injection rates are fixed and we only determine optimal extraction rates. We consider two sets of scenarios. In the first set we find optimal extraction rates from five wells. In the second set we determine optimal extraction rates from 10 wells.

To test the suggested approach, we base our design on 5, 10, or 20 transmissivity realizations and compare the results. For one scenario we consider up to 100 realizations.

The procedure for generating transmissivity realizations is as follows. The natural logarithm of transmissivity, denoted Y, is assumed to follow a multivariate normal distribution with mean $\mu = 9.5$ (corresponding to a transmissivity of 13,360 feet^2/d, or = 1241 m^2/d) and exponentially decaying covariance:

$$\mathrm{Cov}\,(Y_i, Y_j) = \sigma^2 \exp\,(-d_{i,j}/\lambda) \tag{20}$$

where σ (square root of variance) and λ (correlation length) are parameters and $d_{i,j}$ is the Euclidean distance between points i and j. In our study $\sigma^2 = 2.92$ and $\lambda = 300$ feet (91.4 m). The coefficient of variation for Y is 0.18. For the generated transmissivity realizations, the coefficient of variation ranged between 0.23 and 0.28.

Several methods are available for generating Y realizations. The most straightforward, but computationally intensive, is the matrix inversion method [*Davis*, 1987]. The simple nearest neighbor method [*Smith and Freeze*, 1979] uses linear equations to describe the dependence of the conductivity in a given block on conductivity values in surrounding blocks. This method can handle both statistically isotropic and anisotropic covariance functions.

We used the more efficient turning-bands method (TBM) developed by *Mantoglou and Wilson* [1982] for generating 2-D Y realizations (see also work by *Dietrich* [1995] and *Gneiting* [1996]).

6. Results and Discussion

The response surface for CMAX$_{(\mathrm{NR})}$ as a function of pumping rates becomes more complex as NR increases. For example, Figures 3a–3c show the response surfaces for an optimization problem using NR = 5, 10, and 15. To visualize the response surfaces, only three extraction wells are considered, and total extraction is fixed (equals the treatment facility flow capacity of 2000 gallons/min, or 10,811 m^3/d).

Figures 3a–3c show contours of CMAX$_{(\mathrm{NR})}$ generated using 250 simulations per realization (for each figure). For example, 1250 and 2500 simulations are used to generate the data for Figures 3a and 3b, respectively. Each simulation is carried out by changing the pumping rates and running the flow and transport models for a given set of aquifer hydraulic conductivity values.

The response surfaces in Figures 3a–3c are highly nonlinear with several local minima. Gradient-based optimization methods cannot guarantee finding the global optimal solution for such problems. A GA provides an attractive alternative. Note in Figures 3a–3c that the global optimal solution does not change as the number of realizations increases from 5 to 15. This supports our suggestion that a few realizations may provide sufficient accuracy.

The procedure suggested in section 3 is approximate because it uses a relatively small number of realizations while solving the optimization problems. To test the effect of this approximation, we use NR = 5, 10, and 20 to define $CMAX_{(NR)}$. The first P^{MAX} value is usually selected to be a reasonable guess of the optimal value. If the developed optimal pumping strategy achieves a higher reliability level than desired, the second P^{MAX} value is smaller than the first, and vice versa.

In our application the first P^{MAX} value was selected to be 2000 gallons/min (10,811 m^3/d). Then optimal pumping rates from the five extraction wells were determined to minimize $CMAX_{(5)}$ (defined using five realizations). The reliability of the developed pumping strategy estimated using 500 Monte Carlo simulations was found to be 100%. The second P^{MAX} value was set to 1600 gallons/min (8649 m^3/d; 1 gallon/min equals 5.4054 m^3/d) and the developed pumping strategy had an estimated reliability value of 33%. Clearly, the P^{MAX} value that achieves a 95% reliability is closer to 2000 than to 1600 gallons/min. Subsequent P^{MAX} values were set to 1950, 1900, and 1850 gallons/min.

Figure 4 shows the results of Monte Carlo testing of pumping strategies. It shows the proportion of postoptimization realizations that achieve prescribed CMAX values. Figures 5 and 6 contrast total pumping with reliability for 5 and 10 extraction wells, respectively. Here, considering additional wells resulted in very little improvement for developed pumping strategies.

Figures 5 and 6 suggest that for the tested problem, there is no need to use more than 10 realizations. To further test this conclusion, we solved the $CMAX_{(NR)}$ minimization problem (for P^{MAX} = 1900 gallons/min) for different values of NR. Results in Figure 7 show that any number of realizations greater than 10 results in pumping rates that are within 5% of those computed using 100 realizations.

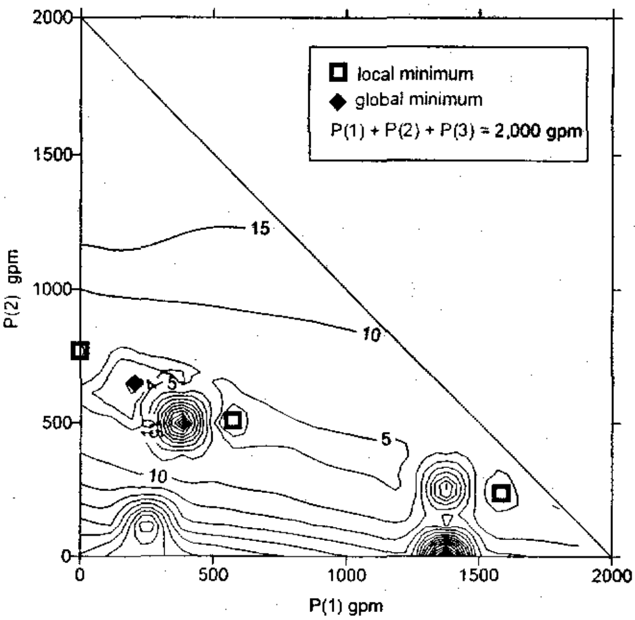


Figure 3b. $CMAX_{(10)}$ for 10 realizations.

7. Summary

We have presented and demonstrated a new stochastic optimization approach for complex nonlinear problems. The suggested approach is based on the multiple-realization method and uses a neural network to model complex response surfaces. The neural network was trained using one set of simulations and then tested on another set of simulations. The neural network was able to approximate $CMAX_{(NR)}$ surfaces with a high accuracy, as indicated by its performance on the testing set.

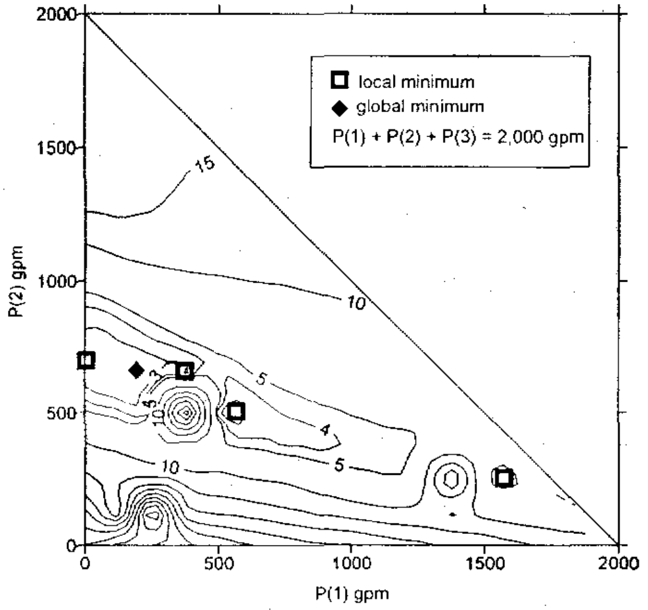


Figure 3a. $CMAX_{(5)}$ for five realizations.

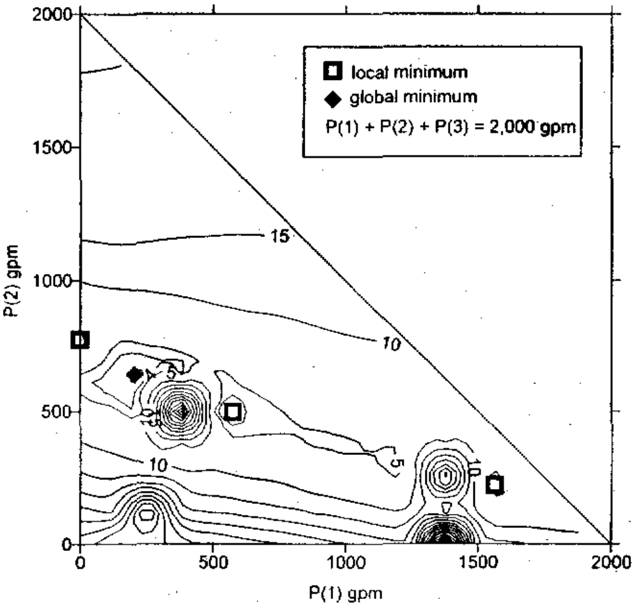


Figure 3c. $CMAX_{(15)}$ for 15 realizations.

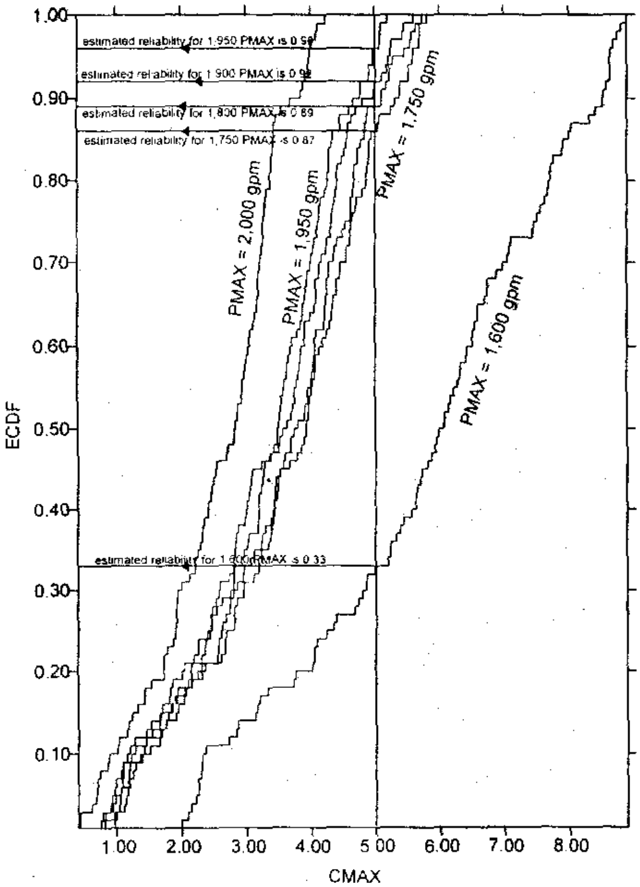


Figure 4. Empirical cumulative distribution functions for CMAX based on 10-realization design for five wells.

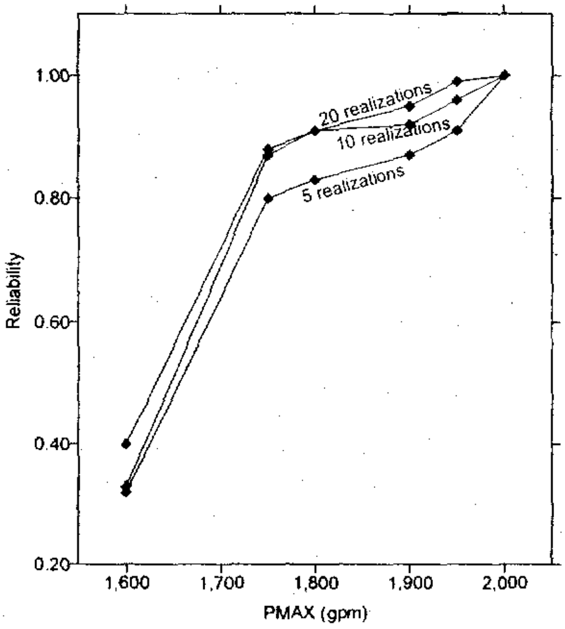


Figure 5. Reliability versus treatment facility size for a five-well design.

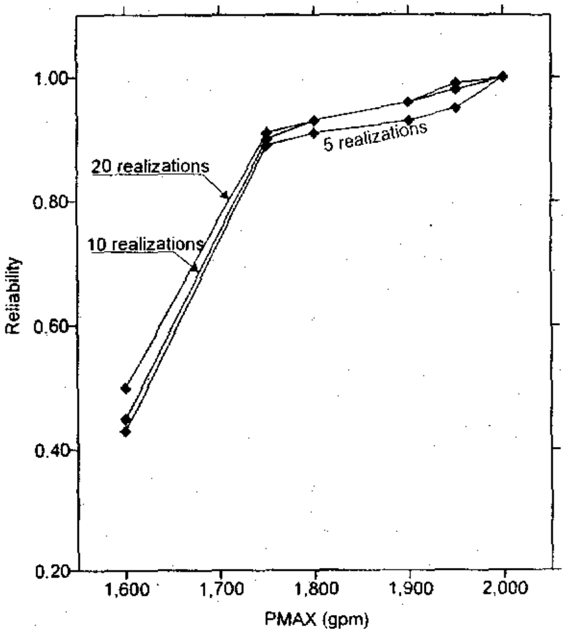


Figure 6. Reliability versus treatment facility size for a 10-well design.

A genetic algorithm efficiently identified optimal pumping rates. The suggested approach makes it easy to find the best control parameters for the genetic algorithm. We found that a crossover probability between 0.85 and 0.9, a mutation proba-

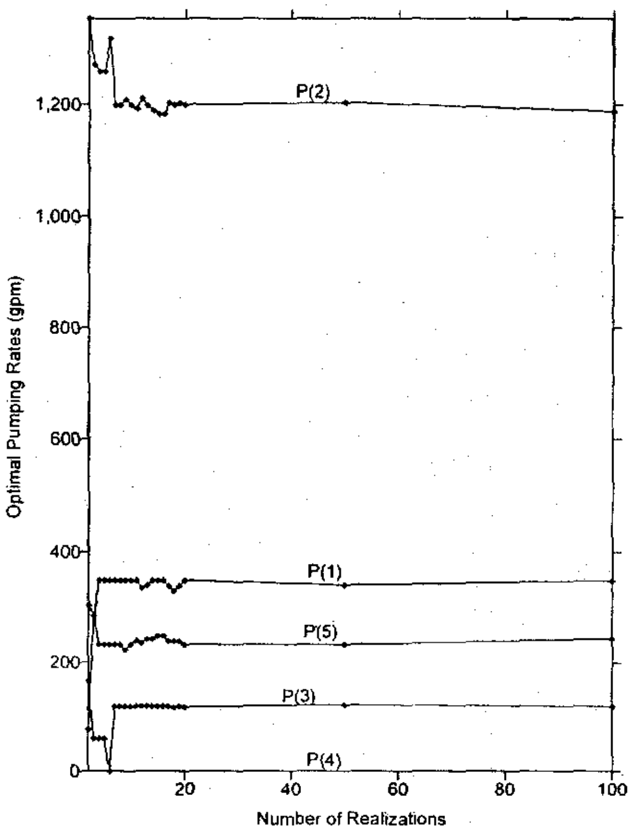


Figure 7. Optimal pumping rates versus number of simultaneously considered realizations.

bility between 0.08 and 0.12, and a population size between 50 and 100 always provided the best answer from the genetic algorithm. However, it must be noted that the formulated optimization problem is not highly constrained. If the optimization problem is highly constrained, then the genetic algorithm may need more generations to find optimal pumping rates. Also, other values for the crossover and mutation probabilities might lead to better performance.

For all tested problems, 10 realizations were adequate for finding optimal pumping rates. This is important since the number of considered realizations significantly affects the CPU time needed to train and test the neural network.

Extension of the suggested approach to handle other groundwater management problems is straightforward. The approach does not depend on the specific flow and transport simulators or the state variables of interest.

Acknowledgments. This paper was supported by the Utah Agricultural Experiment Station, Utah State University. This is approved as journal paper 7064. The authors wish to thank Donna Rizzo at the University of Vermont and an anonymous reviewer for their contribution to the clarity of this manuscript.

References

Aguado, E., and I. Remson, Groundwater management with fixed nonlinear optimization problems charges, *J. Water Resour. Plan. Manage.*, *106*(WR2), 375–382, 1980.

Ahlfeld, D. P., Two-stage groundwater remediation design, *J. Water Resour. Plan. Manage.*, *116*(4), 517–529, 1990.

Alley, W., Regression approximations for transport model constraint sets in combined simulation-optimization studies, *Water Resour. Res.*, *22*(4), 581–586, 1986.

Aly, A. H., and R. C. Peralta, Modification of a flow model and optimal pumping strategies for cleanup and capture of TCE and DCE plumes near Mission Drive, Wurtsmith Air Force Base, Michigan, *ERC Rep. 97-09*, Air Force Cent. for Environ. Excellence, Environ. Restoration Dir., U.S. Air Force, Brooks Air Force Base, Tex., 1997.

Cantiller, R. R. A., and R. C. Peralta, Computational aspects of chance-constrained sustained groundwater yield management, *Trans. Am. Soc. Civ. Eng.*, *32*(3), 939–944, 1989.

Chan, N., Robustness of the multiple realization method for stochastic hydraulic aquifer management, *Water Resour. Res.*, *29*(9), 3159–3167, 1993.

Chan, N., Partial infeasibility method for chance-constrained aquifer management, *J. Water Resour. Plan. Manage.*, *120*(1), 70–89, 1994.

Cieniawski, S. E., J. W. Eheart, and S. Ranjithan, Using genetic algorithms to solve a multiobjective groundwater monitoring problem, *Water Resour. Res.*, *31*(2), 399–409, 1995.

Cooper, G., R. C. Peralta, and J. J. Kaluarachchi, Optimizing separate phase light hydrocarbon recovery from contaminated unconfined aquifers, *Adv. Water Resour.*, *21*(5), 339–350, 1998.

Cybenko, G., Continuous valued neural networks with two hidden layers are sufficient, technical report, Dep. of Comput. Sci., Tufts Univ., Medford, Mass., 1988.

Davis, L. (Ed.), *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, 1991.

Davis, M. W., Production of conditional simulation via the LU triangular decomposition of the co-variance matrix, *Math. Geol.*, *19*(2), 91–98, 1987.

De Jong, K. A., An analysis of the behavior of a class of genetic adaptive systems, Ph.D. dissertation, Univ. of Mich., Ann Arbor, 1975.

Dietrich, C. R., A simple and efficient space domain implementation of the turning band method, *Water Resour. Res.*, *31*(1), 147–156, 1995.

Dowla, F. U., and L. L. Rogers, *Solving Problems in Environmental Engineering and Geosciences With Artificial Neural Networks*, MIT Press, Cambridge, Mass., 1995.

Ejaz, M. S., and R. C. Peralta, Modeling for optimal management of agricultural and domestic wastewater loading to streams, *Water Resour. Res.*, *31*(4), 1087–1096, 1995.

Gharbi, A., and R. C. Peralta, Integrated embedding optimization applied to Salt Lake valley aquifers, *Water Resour. Res.*, *30*(3), 817–832, 1994.

Gneiting, T., Comment on "A simple and efficient space domain implementation of the turning band method" by C. R. Dietrich, *Water Resour. Res.*, *32*(11), 3391–3396, 1996.

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

Goldberg, D. E., A note on Boltzmann tournament selection for genetic algorithms and population-oriented simulated annealing, *Complex Syst.*, *4*, 445–460, 1990.

Gomez-Hernandez, J. J., and J. Carrera, Using linear approximations to rank realizations in groundwater modeling: Application to worst case selection, *Water Resour. Res.*, *30*(7), 2065–2072, 1994.

Gorelick, S. M., A review of distributed parameter groundwater management modeling methods, *Water Resour. Res.*, *19*(10), 305–319, 1983.

Gorelick, S. M., Large scale nonlinear deterministic and stochastic optimization: Formulations involving simulation of subsurface contamination, *Math. Prog.*, *48*, 19–39, 1990.

Gorelick, S. M., C. I. Voss, P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, Aquifer reclamation design: The use of contaminant transport simulation combined with nonlinear programming, *Water Resour. Res.*, *20*(4), 415–427, 1984.

Hegazy, M. A., and R. C. Peralta, Feasibility considerations of an optimal pumping strategy to capture TCE/PCE plume at March Air Force Base, *Rep. SS/OL 97-1*, Syst. Simul./Optim. Lab., Dep. of Biol. and Irrig. Eng., Utah State Univ., Logan, 1997.

Hertz, J., A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation*, Addison-Wesley, Reading, Mass., 1991.

Holland, J. H., *Adaption in Natural and Artificial Systems*, Univ. of Mich. Press, Ann Arbor, 1975.

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

Jacobs, R. A., Increased rates of convergence through learning rate adaption, *Neural Networks*, *1*(4), 295–308, 1988.

Jones, L., R. Willis, and R. W. Cottle, Optimal control of nonlinear groundwater hydraulics using differential dynamic programming, *Water Resour. Res.*, *30*(7), 2065–2072, 1987.

Karatzas, G., An effective procedure to incorporate uncertainty into subsurface remediation models using a multi-scenario approach, in Fourth SIAM Conference on Mathematical and Computational Issues in the Geosciences, Soc. for Indust. and Appl. Math., Albuquerque, N. M., June 16–19, 1997.

Lefkoff, J. J., and S. M. Gorelick, Simulating physical processes and economic behavior in saline, irrigated agriculture: Model development, *Water Resour. Res.*, *26*(7), 1359–1369, 1990.

Mantoglou, A., and J. L. Wilson, The turning bands method for simulation of random fields using line generation by a spectral method, *Water Resour. Res.*, *18*(5), 1379–1394, 1982.

McDonald, M. G., and A. W. Harbaugh, A three-dimensional finite-difference groundwater model, *U.S. Geol. Surv. Open File Rep. 83-875*, 1988.

McKinney, D. C., and M.-D. Lin, Genetic algorithm solution of groundwater management models, *Water Resour. Res.*, *30*(6), 3775–3789, 1993.

McKinney, D. C., G. B. Gates, and M.-D. Lin, Groundwater resource management models: A comparison of genetic algorithms and nonlinear programming, in *Computational Methods in Water Resources*, vol. 2, edited by A. Peters et al., Kluwer Acad., Norwell, Mass., 1994.

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

Peralta, R. C., and A. H. Aly, Optimal pumping strategies to maximize contaminant extraction of TCE plume at Central Base Area, Norton AFB, California, *ERC Rep. 2*, U.S. Air Force, Air Force Cent. for Environ. Excellence, Environ. Restoration Dir., U.S. Air Force, Brooks Air Force Base, Tex., 1995.

Peralta, R. C., and A. H. Aly, Optimal pumping strategies to maximize contaminant extraction of TCE plume at Mather AFB, California, *ERC Rep. 3*, Air Force Cent. for Environ. Excellence, Environ. Restoration Direct., U.S. Air Force, Brooks Air Force Base, Tex., 1996.

Peralta, R. C., J. Solaimanian, and G. R. Musharrafiah, Optimal dispersed groundwater contaminant managementa: MODCON method, *J. Water Resour. Plan. Manage.*, *121*(6), 490–498, 1995.

Peralta, R. C., and R. Ward, Short-term plume containment: Multiobjective comparison, *Ground Water*, *29*(4), 526–535, 1991.

Poggio, J., *Physical and Biological Processing of Images*, Springer-Verlag, New York, 1983.

Ranjithan, S., J. W. Eheart, and J. H. Garrett Jr., Neural network-based screening for groundwater reclamation under uncertainty, *Water Resour. Res.*, *29*(3), 563–574, 1993.

Ritzel, B. J., J. W. Eheart, and S. Rajithan, Using genetic algorithms to solve a multiple objective groundwater pollution containment problem, *Water Resour. Res.*, *30*(5), 1589–1603, 1994.

Rizzo, D. M., and D. E. Dougherty, Design optimization for multiple management period groundwater remediation, *Water Resour. Res.*, *32*(8), 2549–2561, 1996.

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

Rumelhart, D. E., G. E. Hinton, and R. J. Williams, Learning internal representation by error propagation, *Parallel Distrib. Process.*, *1*, 318–362, 1986.

Sawyer, C. S., A. P. Ahlfeld, and A. J. King, Groundwater remediation design using a three-dimensional simulation model and mixed-integer programming, *Water Resour. Res.*, *31*(5), 1373–1385, 1995.

Schaffer, J. D., R. A. Caruana, L. J. Eshelman, and R. Das, A study of control parameters affecting online performance of genetic algorithms for function optimization, in *Proceedings of the Third International Conference on Genetic Algorithms*, edited by J. D. Schaffer, Morgan Kaufmann, Los Altos, Calif., 1989.

Smith, L., and R. A. Freeze, Stochastic analysis of steady state groundwater flow in a bounded domain, 2, Two-dimensional simulations, *Water Resour. Res.*, *15*(6), 1543–1559, 1979.

Smith, M., *Neural Networks for Statistical Modeling*, Van Nostrand Reinhold, New York, 1993.

Takahashi, S., and R. C. Peralta, Optimal perennial yield planning for complex nonlinear aquifers: Methods and examples, *Adv. Water Resour.*, *18*, 49–62, 1995.

Wagner, B. J., Recent advances in simulation optimization groundwater management modeling, *Rev. Geophys.*, *33*, 1021–1028, 1995.

Wagner, B. J., and S. M. Gorelick, Optimal groundwater quality management under parameter uncertainty, *Water Resour. Res.*, *23*(7), 1162–1174, 1987.

Wang, W., and D. P. Ahlfeld, Optimal groundwater remediation with well location as a decision variable, *Water Resour. Res.*, *30*(5), 1605–1618, 1994.

Whiffen, G. J., and C. A. Shoemaker, Nonlinear weighted feedback control of groundwater remediation under uncertainty, *Water Resour. Res.*, *29*(9), 3277–3289, 1993.

Zheng, C., A modular three-dimensional transport model for simulation of advection, dispersion and chemical reactions of contaminants in groundwater systems, Robert S. Kerr Environ. Res. Lab., U.S. Environ. Prot. Agency, Ada, Okla., 1990.

Zurada, J. M., *Introduction to Artificial Neural Systems*, West, Eagan, Minn., 1992.

A. H. Aly, Department of Biological and Irrigation Engineering, Utah State University, Logan, UT 84322-4105. (e-mail: alaa@ssol.agirrig.usu.edu)

R. C. Peralta, Department of Biological and Irrigation Engineering, Utah State University, Logan, UT 84322-4105. (e-mail: peralta@cc.usu.edu)

(Received September 2, 1997; revised July 1, 1998;
accepted July 14, 1998.)