Optimal Settings for a Hybrid Genetic Algorithm Applied to a Groundwater Remediation Problem

Felipe P. Espinoza*, Barbara S. Minsker** and David E. Goldberg***

* Department of Civil and Environmental Engineering, University of Illinois, 205 N. Matthews Ave., Urbana, IL 61801, email: fespinoz@uiuc.edu
** Department of Civil and Environmental Engineering, University of Illinois, e-mail:minsker@uiuc.edu
*** Department of General Engineering, University of Illinois, e-mail:deg@uiuc.edu

Abstract

Water resources management problems can be computationally intensive and improved methods are needed to allow solution of more complex applications. In this paper, we present a numerical algorithm designed to efficiently solve optimization applications such as groundwater management problems. The algorithm is a combination of a simple genetic algorithm and a local search method and is called a hybrid genetic algorithm (HGA). As a first step in the development of an effective HGA, this paper presents a new self-adaptive HGA (SAHGA) that can be used to competently solve the problem without extensive trial-and-error experimentation. This paper presents the SAHGA approach and compares its performance with the SGA and a non-adaptive HGA (NAHGA) for several test functions. The results show considerable promise for the SAHGA, which required less than 25% of the number of function evaluations required for the SGA at a 99% reliability level. The SAHGA algorithm was also more robust than the NAHGA, performing optimally across a broad range of parameter values. The next step will be to apply the SAHGA to the groundwater remediation design problem.

Introduction

One important problem in environmental engineering is the optimal design of a groundwater remediation system, for example the optimal location and pumping rates for an enhanced in situ bioremediation technology. The solution of this complex, nonlinear problem is computationally intensive. Different alternatives of solution have been proposed to solve this problem. Among them are mixed-integer linear approach [Sawyer et al., 1995], non-linear programming formulations [McKinney and Lin, 1996], and dynamic optimal control [Culver and Shoemaker, 1997]. From the previous formulations, dynamic optimal control is the most efficient for time-varying problems. The disadvantage of this method is the difficulty of managing discrete or discontinuous cost functions such as well installation costs. Minsker and Shoemaker (1998) showed that such costs can be quite important for in situ bioremediation design, which typically has shorter duration than pump-and-treat. To overcome this disadvantage and to improve performance, we propose the application of a hybrid genetic algorithm (HGA). An HGA is the coupling of two processes: the simple GA (SGA) and a local search algorithm. The SGA has the capability of managing in an efficient way the discrete part of the problem and the local search solves the time-varying portion of the problem. As a first step in the development of an effective HGA, this paper presents a new self-adaptive HGA (SAHGA) that can be used to competently solve the problem without extensive trial-and-error experimentation. The performance of the method is tested on several test functions using a steepest descent local search method. This paper presents the SAHGA approach and compares its performance with the SGA and a non-adaptive HGA (NAHGA) for several test functions. The results show considerable promise for the SAHGA, which required less than 25% of the number of function evaluations required for the SGA at a 99% reliability level. The SAHGA algorithm was also more robust than the NAHGA, performing optimally across a broad range of parameter values. The next step will be to apply the SAHGA to the groundwater remediation design problem, including changing the local search algorithm from the steepest descent to differential dynamic programming (DDP).

Hybrid Genetic Algorithm Methodology

Basic Elements

Genetic Algorithm. The simple Genetic Algorithm (SGA) used in this work is defined by three basic operators: tournament selection (with a tournament size of two), single point crossover, and mutation. The decision variables are encoded as a binary string ("chromosome") and a population of strings (possible solutions) are randomly generated. With the successive application of these three operators to the population of strings, the feasible space is searched in a more optimal way than with random search. Over time, the population evolves to higher and higher levels of fitness, where fitness is defined as the objective function value in an optimization problem. See Goldberg (1989) for an introduction to the SGA.

Local Search. The local search operator looks for the best solution starting at a previously selected point. For initial testing of the HGA using trial functions, the steepest descent method was chosen as the local search operator because of its ease of implementation and widespread use. This method moves along the direction of the steepest gradient until an improved point is found, from which a new local search starts. The algorithm ends when no new point can be found (this is equivalent to a gradient equal to zero). For functions with multiple local optima, the method finds one local optima but is not guaranteed to find the global minimum. For geometries with conical shape, for example, the method finds the local optimum in one local search starting from any point inside the basin of attraction. For other geometries, the local search operator requires more than one iteration to achieve the solution.

Evolution: Lamarckian v/s Baldwinian. To combine the SGA and local search methods, HGAs typically use one of two approaches: Lamarckian or Baldwinian evolution [Hinton and Nolan, 1987], [Whitley et al., 1994]. Lamarck presented his theory of learned evolution in 1802 [Lamarck, 1802], in which direct learning passes the best characteristics of each individual from generation to generation. This means that both the change in the chromosome and fitness are passed to the individual as genotypic information at the end of local search (i.e., the chromosome of the individual is changed). Baldwinian evolution, also known as the Baldwin effect [Baldwin, 1896], is survival of the fittest following the direction of learning. In this case, only the improved fitness function value is changed after local search and not the chromosome itself. Lamarckian evolution has been shown to cause faster convergence than Baldwinian evolution, but sometimes causes premature convergence problems [Whitley et al., 1994].

Non-Adaptive Hybrid Genetic Algorithm (NAHGA). The NAHGA algorithm is a standard, non-adaptive hybrid genetic algorithm that combines an SGA with local search. The local search step is defined by three basic parameters: local search gap, probability of local search, and number of local search iterations. The first element for the definition of the algorithm is the local search gap, which is the switch between global and local search. In the NAHGA algorithm, this switch is performed every ΔG global search generations, where ΔG is a constant number called the generation gap. For example, if $\Delta G=3$, local search would be performed every 3 generations during the SGA. The second element of the algorithm is the probability of local search p, which is the probability that local search will be performed on each member of the SGA population in each generation where local search is invoked.

This probability is constant and is defined before the application of the algorithm. Finally, each time local search is performed, it is performed a constant number of local search iterations before local search is halted.

Self-Adaptive Hybrid Genetic Algorithm (SAHGA). The SAHGA algorithm works with the same operators as the NAHGA algorithm: local search, probability of local search and number of local search iterations. The major difference in the approaches is that the SAHGA adapts in response to algorithm performance as the algorithm converges to the solution. The details of the adaptations are given below.

Local Search versus Global Search. Instead of a generation gap ΔG, the switch between local and global search is performed using a threshold level given by the relative coefficient of variation of the fitness function between generations. The coefficient of variation is defined as the ratio of the mean and the standard deviation of the population fitness. Figure 1 shows the change in the coefficient of variation and the coefficient of variation itself for a particular experiment using the SGA alone. The trend for the coefficient of variation is decreasing and approaching to zero as the population converges to the optimal solution. Using the CV, we define a new parameter CV ratio (CVR) given in the next equation and shown in Figure 1:

$$CVR = \frac{CV(i)}{CV(i-1)}$$

where CV(i) is the coefficient of variation at generation "i." CVR represents the change in the coefficient of variation from one generation to the next. When CVR >1, the solution at generation "i" is worse than the solution at generation "i-1". Therefore, in order to improve the solution, local search must be performed. In Figure 1, a threshold of one is shown on the CVR curve to illustrate when the SAHGA would invoke local search.

Probability of Local Search Selection. In an HGA, local search typically operates over a small portion of the total population because the additional function evaluations required for local search can be very expensive. Therefore, when local search is achieving greater performance than the most recent global search iteration (using the criterion shown in the next section), the SAHGA algorithm is adapted to search a smaller portion of the population using the relationship:

$$P = P_0\,(1-\varepsilon)^{n-1} \tag{1}$$

In this equation, the local search probability P decreases in a constant way from the initial value. P_0 is the user-specified initial value of the local search probability, "n" is the local iteration number in the local search step, and "ε" is a user-specified parameter governing the rate of decrease in the local search probability. The probability P is reset to P_0 at the beginning of every local search step in order to start with the same sampling size at the beginning of every local search.

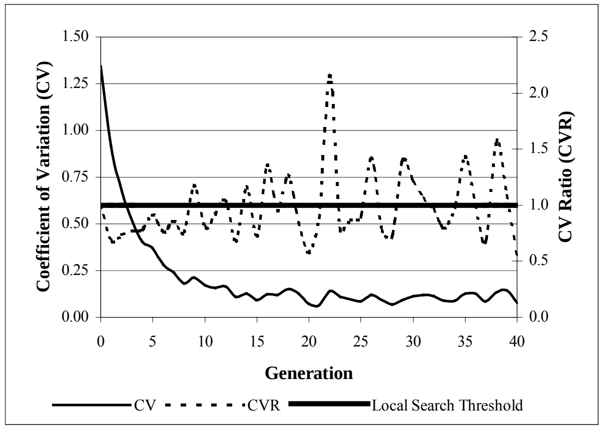

Figure 1. Global Search-Local Search Threshold Effect

Number of Local Search Iterations. One important issue for the application of the algorithm is how long the local search lasts before switching back to the global GA search. In order to make this decision, we compare the fitness improvement by local search with the fitness improvement by global search. This criterion is presented in equation 2:

$$\text{do local search if } \frac{\Delta \text{Global}}{\text{pop}} < \frac{\Delta \text{Local}}{\text{fev}} \tag{2}$$

where ΔGlobal is the improvement achieved between the two previous global search generations, ΔLocal is the current improvement in the local search step, pop is the population size (which is the number of function evaluations required for global search), and fev is the number of function evaluations required for the local search step.

When equation (2) is no longer true, the algorithm switches back to global GA search. However, the number of local search iterations is limited to a maximum value, which is equal to the number of local search iterations defined for the NAHGA algorithm.

Experiments

Test Function. To test the performance of the SAHGA with the NAHGA and SGA, we worked with two different multi-modal test functions with random geometry (radius and depth of multiple basins of attraction), whose equations are shown below. The coordinates ($x_{o,i}$,$y_{o,i}$) are the coordinates of the basin of attraction "i", with r_i the radius and d_i the depth of the basin of attraction. The basins of attraction for both functions are randomly distributed. Function 1 (f1) has conical basins of attraction and Function 2 (f2) has elliptical basins of attraction. Function 1 represents the best case for local search, in which only one local search is required to find the local minimum, and Function 2 represents a more realistic case in which multiple local searches are required to find the local minimum. The results are presented in term of "average number of fitness function evaluations" because generations take different amounts of time for the hybrid genetic algorithm approach, depending on how many local searches are done.

$$f_1(x_1, y_1) = \begin{cases} \dfrac{d_i}{r_i^2}\left(\overline{x}_1^2 + \overline{y}_1^2\right)\left(2 - \dfrac{\overline{x}_1^2 + \overline{y}_1^2}{r_i^2}\right) - d_i & \overline{x}_1^2 + \overline{y}_1^2 \le r_i^2 \\ 0 & \overline{x}_1^2 + \overline{y}_1^2 > r_i^2 \end{cases}$$

$$f_2(x_1, y_1) = \begin{cases} \dfrac{\overline{x}_1^2 + d_i\,\overline{y}_1^2}{r_i^2} - d_i & \overline{x}_1^2 + d_i\,\overline{y}_1^2 \le r_i^2 \\ 0 & \overline{x}_1^2 + d_i\,\overline{y}_1^2 > r_i^2 \end{cases}$$

$$\overline{x}_1 = x_1 - x_{0,i}$$
$$\overline{y}_1 = y_1 - y_{0,i}$$

Experiments. In order to evaluate the behavior of the SAHGA with respect to the NAHGA and SGA we performed several experiments to test the capabilities of the method. The settings for the parameters controlling the SGA for all of the experiments (population size of 800 and 1,200 individuals, respectively, for f1 and f2; probability of crossover of 0.4; and probability of mutation of 0.0013 and 0.0008, respectively, for f1 and f2) were identified using the 3-step methodology developed by Reed et al. (2000). For local search, we used a mixture of Baldwinian and Lamarckian evolution: 25% of the local searches worked with the Baldwinian effect and 75% with Lamarckian evolution. Our initial experiments found that this mixture represented the optimal choice, giving the speed of Lamarckian evolution without causing diversity problems. The stopping criterion for the algorithm was that at least 80% of the population had converged to the solution. In order to evaluate the reliability of the method for different conditions, we worked with 1,000 different initial populations from 1,000 random seeds. The results presented are averaged over the 1,000 realizations.

Local Search. The first experiment was designed to evaluate the effect of local search frequency on the solution of the problem. For the NAHGA algorithm, local search was performed at a pre-defined interval ΔG; for the SAHGA algorithm, local search followed the threshold requirements previously explained. Figures 2 a) and b) show the results for the NAHGA and for the SAHGA algorithms, respectively. For both algorithms, the maximum number of local search iterations was three and the initial probability of local search was 0.1. For the NAHGA, Figure 2 a) shows that the optimal results are achieved only for one value of the variable in study, a generation gap of 1. On the other hand, for the SAHGA, Figure 2 b) shows that the optimal results are achieved for a set of different values of the "local search threshold", so the algorithm is more robust. The performance of the algorithms for different generation gaps and threshold parameter values was similar for both functions.

Probability of Local Search. The second experiment tested the effect of the probability of local search parameter on performance. In the SAHGA, the probability of local search is adapted using the parameter ε in equation 1. Figure 3 a) shows the effect of the adaptive parameter for a specific probability of local search (P_0=0.1). The other parameters for the algorithm are a threshold of 0.6 and a maximum of 3 local search iterations. This experiment shows that there is almost no change in the number of function evaluations for different values of the adaptive parameter for the two different functions. Again, this is another indication of the robustness of the SAHGA algorithm.

a) b)

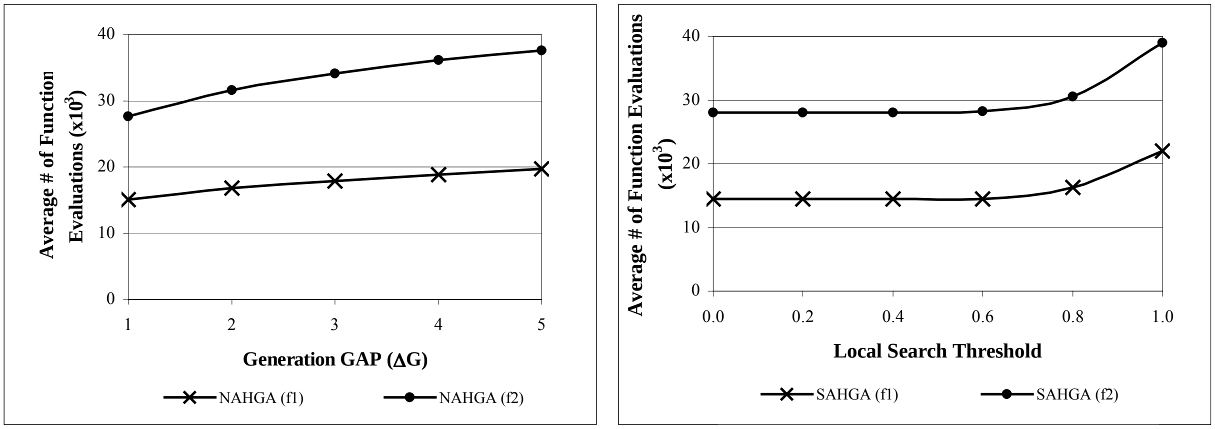

Figure 2. Local Search Effect for NAHGA Algorithm(a) and SAHGA Algorithm (b)

The next step is to evaluate the behavior of both algorithms for different probabilities of local search. For the NAHGA, the generation gap selected was 1, which gave the best performance in the first experiment. Figure 3 b) shows the results of this experiment, which indicates that the minimum number of function evaluations occurs at almost the same probability for both the NAHGA and the SAHGA algorithm. The major difference is that the NAHGA achieves the minimum for only one probability of local search and the SAHGA achieves optimal or very near optimal performance for a broad range of initial probabilities of local search due to its adaptation of P_0 during the run. This effect is achieved for both functions.

Maximum Number of Local Search Iterations. The final experiment analyzes the number of iterations in the local search step. For this analysis, we worked with a probability of local search equal to 0.1 for both algorithms, a ΔG equal to 1 for the NAHGA, and a threshold of 0.6 and an adaptive parameter (ε) equal to 0.2 for the SAHGA. Figure 4 shows the results of this experiment. These results indicate that the number of function evaluations for the NAHGA algorithm increases with the number of local search iterations allowed, but for the SAHGA algorithm the number of function evaluations remains constant because of the adaptive stopping criterion in the SAHGA local search algorithm

Comparison to SGA. To complete the analysis, we performed a final experiment to investigate the performance of the SAHGA relative to the SGA for different population sizes and levels of reliability. The analysis was performed only for the SAHGA algorithm because, as shown in the previous experiments, this algorithm worked for a broader range of parameters than the NAHGA. Figure 5 a) shows the reliability of each algorithm for different population sizes, where reliability is defined as the percentage of the 1,000 different initial populations that found the optimal solution. It is clear that the SAHGA achieves much higher levels of reliability at smaller population sizes than the SGA. Figure 5 b) shows reliability versus number of fitness function evaluations. From this plot, it is clear that the SAHGA is able to achieve much higher reliability with far fewer function evaluations.

For each level of reliability, Figure 6 shows the average number of function evaluations required for the SAHGA as a percentage of the number required for the SGA. For a reliability of 99%, the number of function evaluations required in the SAHGA for function f1 is approximately 7% of the number of function evaluations for the SGA algorithm. For function f2, the SAHGA requires 23% of the number of function evaluations required for the SGA. The higher number of function evaluations for function f2 over f1 occurs because f2 is much more complex than f1 and requires more local search iterations. These results were achieved for a population size equal to 15% and 35% of the optimal population size for the SGA, for f1 and f2, respectively.

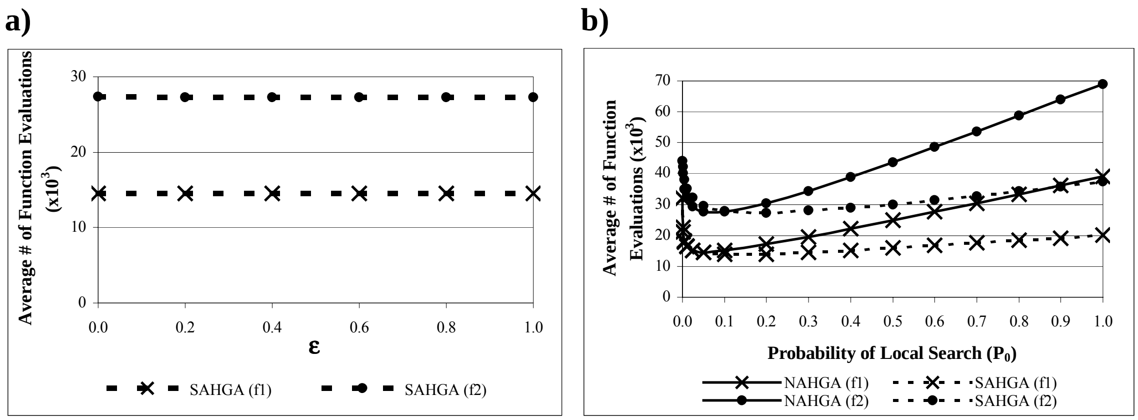

Figure 3. Adaptive Parameter Effect on Probability of Local Search (a) and Probability of Local Search Effect (b)

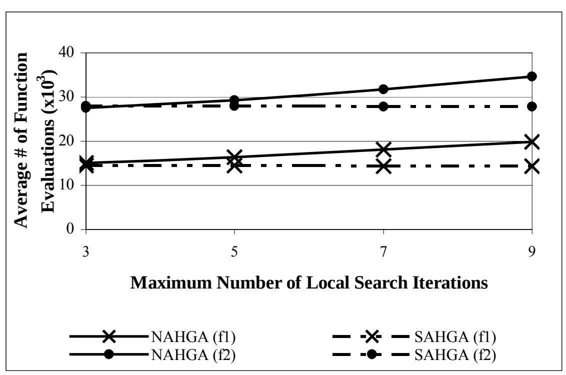

Figure 4. Maximum Number of Local Search Iterations Effect for NAHGA and SAHGA Algorithm

Groundwater Management Problem

For the application of the SAHGA to the solution of a groundwater management problem, the local search algorithm (steepest descent) will be replaced with the DDP algorithm and, in the SGA, the fitness function will be the sum of the cost of installation and operation of a bioremediation pumping system and a penalty function for violating water quality constraints. The operation of the pumping system will be evaluated using a simulation model.

The DDP algorithm in its original form is labor intensive because it needs the analytical evaluation of the derivatives of the simulation model and the inclusion of the operation cost of the pumping system. Now, in the SAHGA algorithm, we will include include a new version of DDP capable of more efficient numerical evaluation of the derivatives of the simulation model (Liu and Minsker, submitted 2000). Results of the implementation should be available at the conference.

a) **b)**

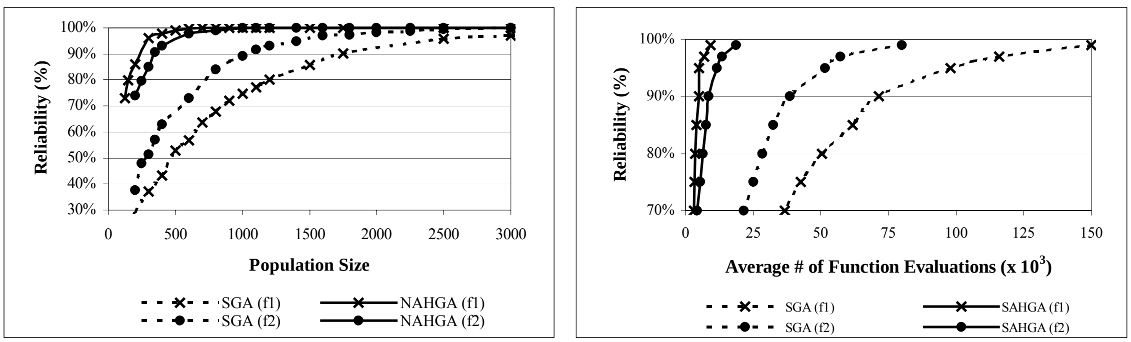

Figure 5. Reliability v/s Population Size a) and Reliability versus Number of Function Evaluations b)

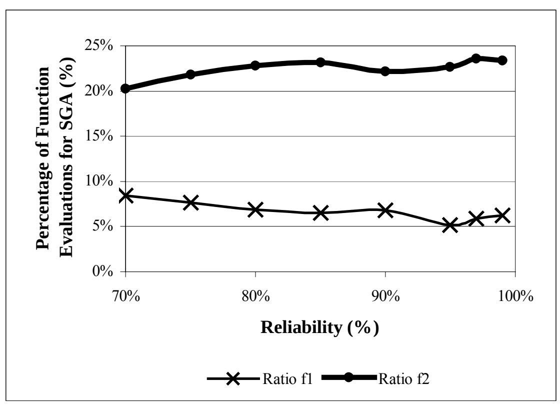

Figure 6. Average Number of Function Evaluations for the SAHGA as a Percentage of the Number for the SGA at each Level of Reliability

Conclusions

The results presented in this paper clearly indicate that the first step of the process is a success. We have developed a new algorithm with adaptive capabilities, the SAHGA algorithm, which enabled robust solution of complex, multi-modal problems for a much greater range of parameter settings than the NAHGA. Compared with the SGA, the SAHGA was able to solve complex problems much faster because of the combined effect of smaller population sizes and increased information from local search. For the same level of reliability, the SAHGA required as much as 95% fewer function evaluations than the SGA for function f1 and as much as 75% fewer function evaluations for function f2. For the next step, we expect that the new algorithm will solve the groundwater remediation design problem in a very efficient way because of the combined effect of the SAHGA algorithm and the local search (DDP).

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. BES 97-34076 CAR.

References

Baldwin, J. M. (1896). "A new factor in evolution". *American Naturalist*, 30:441-451.

Culver, T. B. and Shoemaker, C. A. (1997). "Dynamic optimal groundwater reclamation with treatment capital cost". *J. Water Resources Planning and Management,* ASCE, 123(1), 23-29.

Hinton, G. E., and Nolan, S. J. (1987). "How learning can guide evolution". *Complex Systems*, 1:495-502.

Goldberg, D. E. (1989). "Genetic algorithms in search, optimization, and machine learning". Addison-Wesley.

Lamarck, J. (1809). "Zoological Philosophy".

Liu, Y., and B. S. Minsker, (in press)." Efficient multi-scale method for optimal in-situ bioremediation design". *Journal of Water Resources and Planning Management*.

McKinney, D. C., and Lin, M. D. (1996). "Pump-and-treat groundwater remediation system optimization", *J. Water Resources Planning and Management,* ASCE, 122(2), 128-136.

Minsker, B. S., and Shoemaker, C. A. (1998). "Dynamic optimal control of in-situ bioremediation of groundwater", *Journal of Water Resources Planning and Management,* ASCE, 124(3), 149-161.

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

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

Whitley, D., Gordon, V. S., and Mathias, K. (1994). "Lamarckian evolution, the Baldwin effect and function optimization", *Parallel Problem Solving from Nature- PPSN III,* 6-15.