Multiscale Strategies for Solving Water Resources Management Problems with Genetic Algorithms

Meghna Babbar[1] and Barbara Minsker[2]

[1]Department of Civil and Environmental Engineering, University of Illinois at Urbana Champaign, Urbana, IL 61801; PH (217) 333-6979; email: mbabbar@uiuc.edu
[2]Department of Civil and Environmental Engineering, University of Illinois at Urbana Champaign, Urbana, IL 61801; PH (217) 333-9017; email: minsker@uiuc.edu

Abstract

Genetic Algorithms (GAs) face major computational bottlenecks when numerical models are used for estimating the fitness of the objective function. Especially in large-scale water resources design problems, where the scale of the spatial grids is important in determining the numerical accuracy of the design, a tradeoff exists in the precision of the fitness function and the computational expenses endured. This paper discusses multiscale strategies that can be utilized for improving the performance of GAs when working with spatial grid dependant fitness functions. The strategy uses fine grid and coarse grid fitness functions strategically to maintain the accuracy and computational speed of the problem and drive the GA towards better and more accurate solutions faster. The algorithm's efficacy is tested using a groundwater remediation design case study.

Introduction

Many engineering design problems can be solved using optimization techniques coupled with simulation models to identify effective designs. For example, in the groundwater remediation field genetic algorithms have been used extensively with numerical simulation models to analyze the performance of candidate remediation designs [e.g., Ritzel et al., 1994, Smalley et al., 2000, Wang, 1991, Wang and Zheng, 1997, Aksoy and Culver, 2000, Gopalakrishnan et al., 2003, Minsker et al, 1998]. The degree of accuracy of these numerical models is a function of many factors, including grid size. Finer grids usually improve accuracy, but can increase computational effort substantially.

Some work has been done in the past to study the effect of multiscale grid approximations on optimization models [see *Albert and Goldberg*, 2001, *Liu and Minsker*, 2001]. *Albert et al* used a simple genetic algorithm (SGA) to explore the use of coarse grid approximations for solving a numerical integration problem. *Liu et al* (2001) devised a full multiscale approach for in-situ bioremediation design using a derivative-based optimization method called SALQR. In this paper, we explore the potential for new multiscale approaches to improve the performance of a groundwater remediation design optimization problem described below.

Groundwater Remediation Problem and Objectives

This study examines a groundwater remediation design problem whose two major components are the numerical model and the optimization model. The main objective of the design problem is to identify the most cost effective groundwater treatment strategy that keeps the human health risk due to contamination within a specified limit. The numerical model simulates the flow of groundwater and transport of contaminants for each candidate remediation plan. The optimization model evaluates the objectives and constraints and identifies optimal designs using a simple genetic algorithm (SGA). Below is a description of the two models.

The groundwater management model was based on a confined, heterogeneous, isotropic aquifer with a BTEX spill, studied previously by *Smalley et al* (2000). The modeled aquifer is 480m long, 240m wide and 20m deep. The model assumed steady state flow of groundwater in the aquifer. In order to study solely the effect of noise due to grid size (which is the main objective of this research), the uncertainty in hydraulic conductivities and risk parameters (*Smalley et al*, 2000 and *Gopalakrishnan et al*, 2003) were assumed to be absent. See Babbar (2002) for details on the numerical model and test problem.

The initial plume was created using 10 BTEX contaminant sources, with contaminant concentrations varying from 50mg/l to 600mg/l, on a grid size of '144 by 72'. The numerical model was run for a stress period of 2 years to create the initial spatial distribution of the plume. Once the initial plume was created, the concentrations and hydraulic conductivities were spatially averaged to grid sizes of '16 by 8' and '48 by 24' elements, each having one layer. These averaged concentrations were then used as the starting concentrations for the plume on two coarser grids in the management model. Pump and treat technology was used for the remediation process, with a maximum of three remediation wells available for extraction/injection at the 58 candidate locations. Each well was assumed to have a maximum pumping capacity of 250 m^3/day. Since most BTEX compounds are semi-volatile, an ex-situ air stripping technology was selected for the extracted water. Once the concentration within the plume was estimated from the numerical model, a human health risk assessment model was used to predict human health risks at an assumed exposure point 200 m offsite. See *Smalley et al* (2000) for more details on the risk assessment model.

The objective of the remediation design problem is to minimize total treatment cost such that constraints for risk, drawdown and pumping rates are met. The total treatment cost consists of remediation well costs, monitoring costs and ex-situ treatment costs [see *Gopalakrishnan et al*, 2003, and Babbar, 2002, for details]:

$$\textbf{\textit{Min}}\ C_{TOT} = C_{REM} + C_{MON} + C_{SYST} \qquad (1)$$

The constraints for this problem include human health risk limits, pumping rate limits, and hydraulic drawdown limits (See Babbar, 2002, for details).

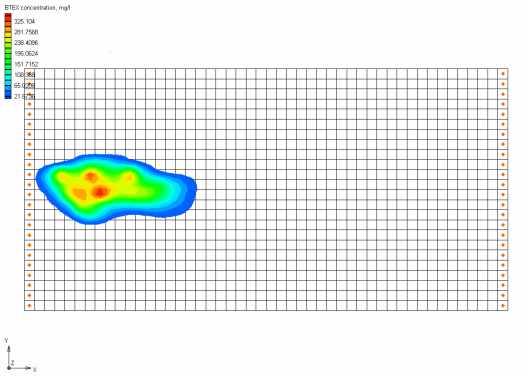

Figure 1. BTEX plume on '48 by 24' grid size

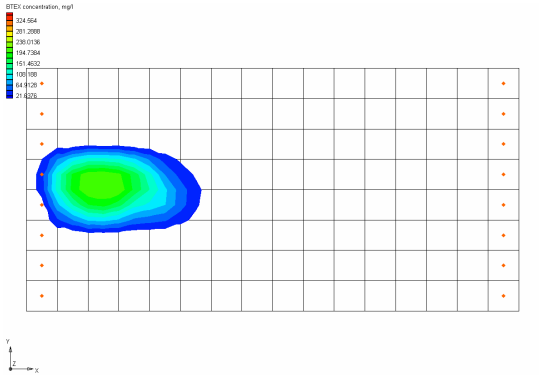

Figure 2. BTEX plume on '16 by 8' grid size

For the genetic algorithm application, decision variables for the remediation design are well locations and pumping rates for three remediation wells, a variable to decide whether the wells perform extraction or injection, and a variable to determine whether the wells are installed or not. These decision variables are encoded into a binary string with 48 bits. Unlike the usual GA operations where fitness functions are maximized, the fitness function here is minimized:

$$\textit{Fitness function} = C_{TOT} + w_1 * \textit{Risk violation} + w_2 * \textit{head violation}. \quad (2)$$

The penalty weights for risk (w_1) and head (w_2) violations were each set to 1000. Penalties are based on violations with respect to a risk limit of 10^{-5} (or a 1 in 100,000 increased lifetime cancer risk), and an allowable maximum drawdown of 0.12 m.

Simple Genetic Algorithm Design Methodology

Simple genetic algorithms (SGAs) are good optimization tools for groundwater remediation design problems, which are discrete, non-convex, and discontinuous. Reed et al. (2000) have shown that effective selection of various GA parameters is important to ensure that the GA converges to optimal or near-optimal solutions for practical engineering problems. In their work, they also showed that theory from the current GA literature can be used in the GA design methodologies for identifying realistic parameter values. A brief summary of their suggestions that we have used in our work is given below.

The building block theory says that GAs find good solutions by assembling building blocks, which are high quality "pieces" of chromosomes. Goldberg et al. (1992) have emphasized the importance of GA population sizing to ensure a sufficient initial supply of building blocks. The population size can be estimated using the equation developed by *Harik et al* [1997] which is based on a random walk model for the initial supply of building blocks (BBs) and the gambler's ruin model for representing the interaction of competing BBs:

$$N \geq -2^{k-1} \ln(a) \frac{s_{bb} \sqrt{p\,(m-1)}}{d} \qquad (3)$$

where,

N = population size, k = BB order, a = probability of failure (less than 5%), d = signal difference, s_{bb} = standard deviation or noise interference between competing BBs.

In this equation, the term '$s_{bb} \sqrt{p\,(m-1)}$' can be approximated as the standard deviation of fitness s_f for a large initial population (*Reed et al*, 2000). For the '16 by 8 grid' size problem and '48 by 24 grid' size problem, fitness evaluations were done for an initial population of 1000 random individuals. Using the fitness of this population, the standard deviation 's_f' was estimated to be 1,472,000 for the '16 by 8 grid' problem and 1,731,000 for the '48 by 24 grid' problem.

The signal difference 'd', which represents the difference in fitness between best and second best building blocks in the population, is unknown in real-world applications because the building blocks cannot be explicitly identified. We estimate the signal difference using a probabilistic approach that is based on the statistical frequency analysis theory (*Hogg et al, 1978*). This approach sorts the fitness evaluations for the initial random population and organizes them into classes that are probabilistically distinct. Based on this frequency analysis theory, the number of bins created is approximately equal to $\sqrt{n}$, where n is the total number of individuals (1000 in this case). The number of individuals per bin is evaluated based on the condition that np_i should be ≥ 5, where p_i is the percentage of total number of individuals in that bin. See *Hogg et al* for more details on this theory. Once the histograms are created, the signal difference is estimated to be the difference in fitness of the first two bins (i.e. second lowest and lowest fitness in this case, where lower fitness is better). Signal difference for this problem was estimated to be 157,821 for the '16 by 8 grid' and 198,541 for the '48 by 24 grid'. Figure 3 and Figure 4 show the frequency analysis histograms for the '16 by 8' and '48 by 24' cases.

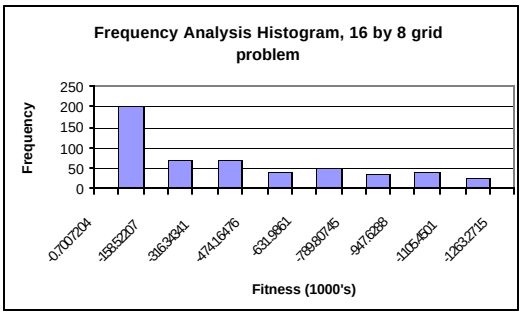

Figure 3. Histogram of fitness for initial population,16 by 8 grid problem

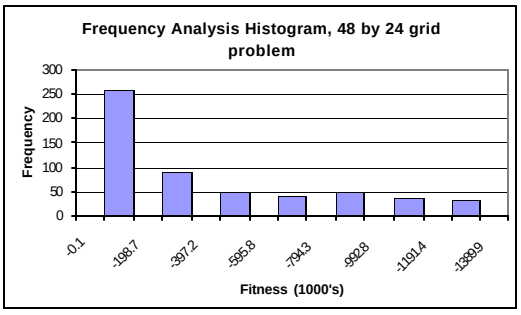

Figure 4. Histogram of fitness for initial population,48 by 24 grid problem

Since we cannot identify the actual building block order 'k' in equation 6 for real-world problems, the population sizes for both grid problems are calculated using the above parameters for k=1, k=2, k=3, and k=4. Higher building block orders are

usually disrupted by the crossover operation (*Reed et al*., 2000) and hence are not considered.

The $m + l$ selection scheme, without replacement, was used for selecting the best individuals for successive generations, where m represents the child population and l represents the parent population. In this selection scheme the parent and child population are combined and the best individuals from the joint population are selected as the parent population for the next generation.

Tournament selection without replacement with a tournament size of 's', equal to 2, was used to select individuals for mating. In the 'without replacement' selection condition, any individual selected is not placed in the contest again. Based on the work of Harik et al (1997), this selection scheme was found to be more effective for this problem than a policy with replacement.

Since $m + l$ is an elitist strategy, it was assumed safe to have a higher probability of crossover (Pc) of 0.75 than the theoretical crossover probability of 0.5 for mating individuals. Test runs for both the crossover probabilities proved that a higher Pc of 0.75 was able to explore for solutions faster and performed better. This was done to ensure faster explorative search while saving the best solutions found. Probability of mutation (P_m) is set equal to 1/N, where N is the population size (*De Jong et al*., 1975). For tournament selection, convergence time for the SGA is estimated to be less than twice the chromosome length (*Thierens et al*., 1998).

An 'injection' method proved to be very effective in solving the simple GA problem for different building block orders (Espinoza et al., submitted 2002). The SGA was first solved for building block order 'k=1', which is the smallest population size. The best individual found in this problem was then injected into the starting population of the next order (i.e. 'k=2') problem. This method of injecting best individuals from previous 'building block order' problems into starting random population of the next higher 'building block order' problem was repeated successively for higher order problems. Using this approach, we insure that the search always moves towards improving the solution quality in sequential runs. The population size was sequentially increased in this manner until more than 90% of the population converged to the same or sufficiently similar solutions. Espinoza et al found that the SGA using the injection method was approximately 50% faster than the SGA not using the injection method. For this reason, the injection method was used for all subsequent cases in this work. Building block order 'k=3' was selected as the most suitable population size for all subsequent analysis. The time spent on computation for 'k=3' is much less than that for 'k=4', but the solution quality varies less than 2% from the best solution identified in each case.

Multiscale Strategies for Genetic Algorithms

Fine numerical grids improve the numerical accuracy of solutions, thus helping the GA find better and more accurate designs. However, evaluating fitness functions with

fine grid numerical models is often computationally intensive, with each fitness function evaluation requiring a large amount of processor time. Coarse numerical grid problems, though computationally less intensive, tend to produce less reliable solutions because of the numerical inaccuracies faced in calculating the fitness function.

In this work, we propose two strategies that can be implemented with a range of grid sizes to identify more accurate solutions with less computational effort. For the groundwater remediation problem, as stated earlier, we used two grid sizes of '48 by 24' cells and '16 by 8' cells to implement the strategies. The two strategies use information from both the grid sizes while searching for optimal solutions as follows:

1) The **first multiscale** strategy begins the search with the coarsest grid numerical model for the optimization problem, and after a certain fixed number of generations it changes the numerical model to a finer grid numerical model. For more than two numerical grids the epochs, which are the generations during which a particular grid size is used, could be appropriately distributed over the total number of generations for which the GA is run.
2) The **second multiscale** strategy uses the coarse grid numerical model for the entire population except for a certain percentage of best individuals in the total population that are tested on the fine numerical grid. For crossover, tournament selection uses a biased technique to select the individual for mating. The biased technique prefers individuals tested on the fine grid over the individuals tested on the coarse grid, even if the apparent fitness of individual tested on the coarse grid is better than that tested on the fine grid. For individuals tested on similar grids, the better one is selected. For this strategy, $m + l$ selection scheme for the final population is done separately for individuals tested on different grid sizes. The proportion of individuals selected from the fine grid size group is slowly increased in the final population to allow the individuals tested on the fine grid to takeover the population after sufficient exploration using the coarse grid numerical model. Individuals already tested on the fine grid in previous generations is not tested again, and the 'true' fitness is updated for all similar copies of individuals in the population.

The following cases were run on 5 different random number seeds to compare the different strategies:

- Case 1: The GA is run entirely on the coarse grid (16 by 8) numerical model. The final optimal solution obtained on the coarse grid is then tested on the finest grid numerical model to assess the 'true' value of the solution.
- Case 2: The GA is run entirely on the fine grid (48 by 24) numerical model.
- Case 3: The GA tests the **first multiscale** strategy by changing the coarse grid numerical model to the fine grid numerical model at the '20^{th}' and the '30^{th}' generations.
- Case 4: The GA tests the **second multiscale** strategy by testing the best '5%', '10%', '15%', '20%', '25%' of the total population on the fine grid numerical model.

Figures 5, 6, 7, 8, and 9 compare the different strategies for five different random number seeds. Case 1 is shown on these figures ('Case1_coarse') by plotting the 'true' value of the final optimal solution for the coarse grid problem. The curve for 'Case2_fine_0' depicts the trend for Case 2. Curves for 'Case3_fine_20' and 'Case3_fine_30' depict Case 3 when the change in grid size occurs at the 20th and 30th generation, respectively. Curves 'Case4_best5%', 'Case4_best10%', 'Case4_best15%', 'Case4_best20%', and 'Case4_best25%' test the second multiscale strategy (Case 4) for different percentages of the population tested on the fine grid in each generation.

From the five figures we can observe that 'Case1_coarse' performs the worst in terms of best solution found compared to all other cases. 'Case2_fine_0' is on the other hand able to find a much better solution than 'Case1_coarse'. However, the computational time invested in finding this improved solution is more than 5 times the 'Case1_coarse' for most of the random seeds. Thus, this tradeoff between solution quality achieved and computational expenses endured implies that the obvious choice of using a fine grid numerical model alone to get the best and most accurate solution is not the best strategy when a GA is used. Case 3, the first multiscale strategy, is able to achieve solution quality close to that of 'Case2_fine_0' but with similar computational expenses. For some random number seeds, the solution quality for Case 3 was much better than using the fine grid alone, but the CPU time invested for these seeds were larger than the 'Case2_fine_0'. Case 4, the second multiscale strategy, proved to be the most effective in finding better solutions within shorter time intervals. For most of these cases the GA was able to find solutions closer to the quality of the fine grid case 'Case2_fine_0', within fractions of the time that invested for 'Case2_fine_0'.

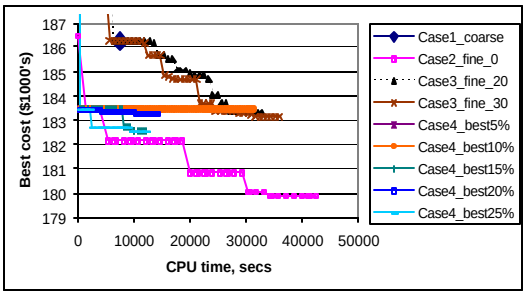

Figure 5. Multiscale strategies for random number seed 1

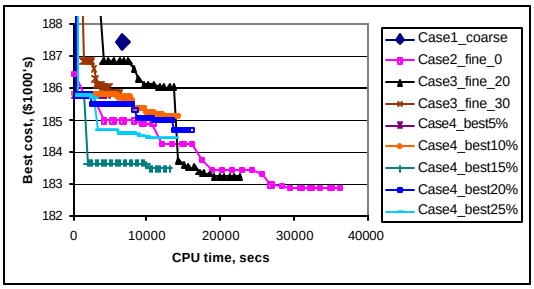

Figure 6. Multiscale strategies for random number seed 2

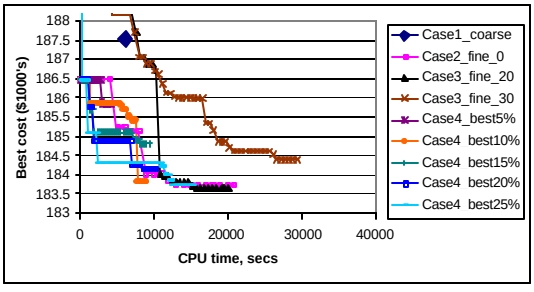

Figure 7. Multiscale strategies for random number seed 3

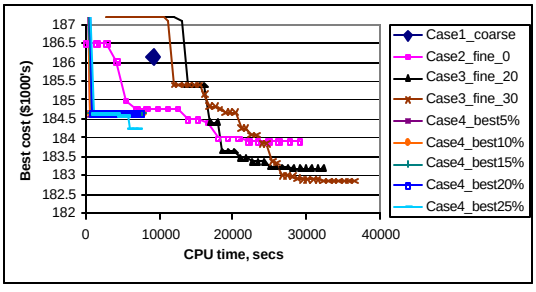

Figure 8. Multiscale strategies for random number seed 4

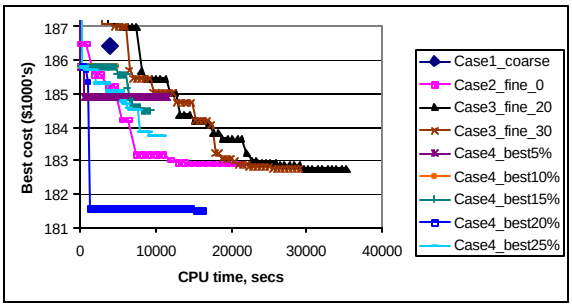

Figure 9. Multiscale strategies for random number seed 5

A comparison of average values of solution quality and CPU time invested (see Figures 10 and 11) indicates interesting results. Both the multiscale strategies worked much better than 'Case1_coarse', producing solutions with costs closer to 'Case2_fine_0' than 'Case1_coarse' (Figure 10). However, the computational costs involved (Figure 11) were substantially improved for the second multiscale strategy. Case 3 (the first multiscale strategy) ended up taking CPU time comparable to that of 'Case2_fine_0', because the entire population was evaluated on the fine grid and convergence required more generations than 'Case2_fine_0'. Case 4 (second multiscale strategy), on the other hand, used computational time less than half that of 'Case2_fine_0' for all of its tests: 'Case4_best5%', 'Case4_best10%', 'Case4_best15%', 'Case4_best20%', and 'Case4_best25%'. This can be attributed to the fact that only a portion of the population was tested on the fine grid every generation, thus decreasing the computational effort tremendously. The choice of what fraction of the population should be tested on the fine grid depends on the problem type and how much computational resources are available to the practitioner. The most useful observation apparent from Case 4 is that using fractions of the population to test on more accurate grids proved effective than other cases in two ways:

- Only the best individuals on the coarse grid were allowed to be further tested on the fine grid, thus preventing unnecessary investment in evaluating less fit individuals on the fine grid.
- The chromosomes tested on the coarse grid were allowed to exchange building blocks with those tested on the fine grid, with a biased preference for fine grid when selecting individuals for mating. This ensured that the chances of recombining individuals tested on fine grid with those tested on the coarse grid improved.

Conclusion

The above work highlights some of the crucial issues involved in optimizing problems that use information from numerical models with varying spatial grid sizes. As observed in our results, coarse grids and fine grids can drive the optimization search towards different sub-spaces of the solution space because they solve the numerical model with different inaccuracies. These numerical inaccuracies can impart variation in fitness functions evaluations for the same candidate designs. This

phenomenon affects the optimization search and the time taken for the GA to converge to an optimal or near optimal solution. In this problem, the two multiscale strategies (Case 3 and Case 4) of using coarse grid along with the fine grid proved to be very useful for saving computational effort (on an average) as compared to 'Case2_fine_0'and improving solution quality as compared to the 'Case1_coarse'. The second multiscale strategy (Case 4), which allowed only a fraction of the population to be tested on the fine grid, proved to be the most effective. This approach reduced computational effort substantially while maintaining solution quality close to 'Case2_fine_0'.

Multiscale optimization strategies such as those proposed in this paper can also be helpful for improving understanding of how sensitive an optimization search is to the grid design of the numerical model. For example, if the optimal design changes substantially when the fine grid approximation is incorporated into the solution, then that fine grid is clearly necessary. Our strategies will help to allow computationally intensive numerical models to be used with coarse grid versions of the same model to attain solutions with numerical accuracies and fitness close to those obtained when fine grid is used alone. The next stage of our research will verify this conclusion on a field-scale application.

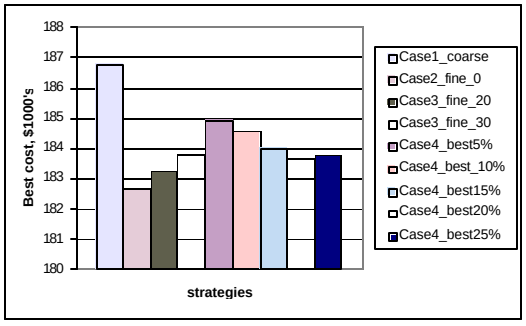

Figure 10. Average best costs of different strategies over different random number seeds

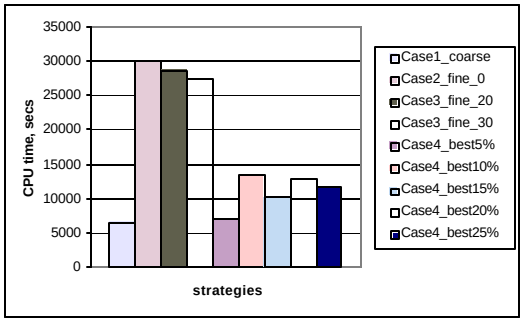

Figure 11. Average CPU time of different strategies over different random number seeds

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. BES 99-03889, and the U. S. Army Research Office under grant numbers DAAD19-00-1-0389 and DAAD19-001-1-0025.

References

Aksoy, A., and T.B. Culver, Effect of sorption assumptions on aquifer remediation designs, *Groundwater*, 38(2), 200-208, 2000.

Albert L.A., and Goldberg, D.E., Efficient Evaluation Genetic Algorithms under Fitness Functions, IlliGAL Report No. 2001024, July 2001.

Babbar, M. Multiscale Parallel Genetic Algorithms for Optimal Groundwater Remediation Design , M.S. Thesis, University of Illinois, 2002.

Cantu'-Paz, E., A survey of Parallel Genetic Algorithms, *Calculateurs Paralleles, Reseaux et Systems Repartis*, Vol. 10, No. 2, pp. 141-171, Paris: Hermes, 1998.

Cantu'-Paz, E., Designing efficient and accurate parallel genetic algorithms, PhD thesis, 1999.

Goldberg David E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison–Wesley, New York, NY, 1989

Gopalakrishnan G., Minsker B., and Goldberg D.E., Optimal sampling in a Noisy Genetic Algorithm for Risk-Based Remediation Design, *Journal of Hydroinformatics*, 2003.

Grefenstette J.J. and Fitzpatrick J.M., Genetic search with approximate function evaluations, In Grefenstette, J.J. (Ed.), *Proceedings of an International Conference on Genetic Algorithms and their Applications*, pp 112-120, Hillsdale, NJ, 1985.

Harik G. R., Cantu-Paz E., Goldberg D. E., and Miller B. L., The gambler's ruin problem, genetic algorithms and the sizing of populations, In Proceedings of the 1997 IEEE Conference on Evolutionary Computation, pp 7-12, IEEE press, New York, NY, 1997

Hogg, R., and Craig, A., *Introduction to Mathematical Statistics*. Macmillan Publishing Co., Inc., New York, 1978.

Liu, Y., and B. S. Minsker, "Efficient multiscale methods for optimal in situ bioremediation design." *Journal of Water Resources and Planning Management*, in press, 2001.

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

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

Reed, P. Striking the Balance: Long-Term Groundwater Monitoring Design for Multiple Conflicting Objectives, Ph. D. Thesis, University of Illinois, 2002.

Ritzel, B.J., J.W. Eheart, and S. Ranjithan, Using genetic algorithms to solve a multiple objective groundwater pollution containment problem, *Water Resources Research*, 30(5), 1589-1603, 1994.

Smalley J. B., Minsker B. S., and Goldberg D. E., Risk-based In Situ bioremeditation design using a noisy genetic algorithm, *Water Resources Research*, 36(20), 3043-3052, 2000.

Wang, Q.J., The genetic algorithm and its application to calibrating conceptual runoff models, *Water Resource Research*, 27(9), 2467-2471, 1991.

Wang, M. And C. Zheng, Optimal remediation policy selection under general conditions, *Groundwater*, 35(5), 757-764, 1997.