CONSTRAINT HANDLING IN GROUNDWATER REMEDIATION DESIGN WITH GENETIC ALGORITHMS

BY

MATTHEW JOHN ZAVISLAK

B.S., University of Illinois at Urbana-Champaign, 2002

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Environmental Engineering in Civil Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2004

Urbana, Illinois

ABSTRACT

Genetic algorithms (GAs) have been successfully used to solve a variety of water resource optimization problems. These problems often have constraints that, in a standard GA, are incorporated using penalty functions that require the user to configure parameters called penalty weights. To get the proper parameter values, considerable trial-and-error experimentation is usually required. This paper presents two methods, adapted from the genetic algorithm literature, that eliminate the need for user-specified penalty weights and the associated trial-and-error experimentation. The methods are applied to two groundwater remediation design case studies. The results show that the new methods can be successfully used for groundwater remediation design, and tended to find marginally better solutions (about 1% decrease in cost) in a shorter period of time (up to 10% faster, plus time spent adjusting user-specified penalty weights). Additional research is needed to further test these methods on other water resource problems.

Dedicated to the memory of Adam Blake

(1978-2004)

ACKNOWLEDGEMENTS

This work would have not been possible without the help of several past and present members of my research group (in alphabetical order): Meghna Babbar, Felipe Espinoza, Marcia Hayes, Abhishek Singh, Shengquan Yan. To all of you, I owe many thanks.

In addition, I would like to mention that this work would never have been possible without all the help my advisor, Barbara Minsker, has given me throughout my graduate career. Thank you, Barbara.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

Genetic algorithms (GAs) have shown much promise for optimizing water resources problems because of their ability to find globally optimal or near-optimal solutions in a large decision space typical of complex water resource problems. However, one of the difficulties in using GAs, as with other nonlinear optimization approaches, is the trial-and-error experimentation required to find appropriate GA parameter settings. Reed, et al (2000, 2003) developed guidelines for setting GA parameters that can be used to eliminate much of the trial-and-error experimentation. However, their guidelines do not address the issue of constraint handling.

Constraints are needed in water resource optimization for a variety of reasons. In the case of groundwater remediation design, for example, there are typically constraints specifying that the concentration of a pollutant be reduced to a certain level. The implementation of design constraints can be difficult because, if done improperly, the GA will not find a feasible solution (one that satisfies all design constraints) or it will find a suboptimal solution. Cieniawski et al (1995) notes that constraints are a shortcoming of GAs, specifically in the case of multiple constraints as is typical for water resource problems. Hilton & Culver (2000) study two methods for constraint handling in optimal remediation design, both of which are variations on the static method and still require significant experimentation to identify appropriate parameter settings. A good general listing of other papers on constraint-handling approaches for GAs is available at http://www.cs.cinvestav.mx/~constraint/.

In this paper we present two new approaches to implementing constraints that enable the GA to find a constrained optimal solution without significant trial-and-error experimentation. The goal of this study is to investigate whether these methods will enable the water resources community to overcome the difficulties associated with constraint handling methods and readily find optimal designs for any number of desired constraints. To test this hypothesis, two pump-and-treat design case studies are solved using both traditional and new constraint handling methods. The paper begins with a description of the two case studies, followed by a presentation of the traditional constraint method and the two new methods. Lastly, the results of the experimental trials are given, followed by a discussion of the conclusions of the study.

2. FORMULATION OF THE CASE STUDIES

To test the capabilities of the constraint-handling methods for water resources optimization, two case studies are considered involving remediation design for a contaminated aquifer. The main objective for both cases is to minimize remediation cost while meeting human health risk/concentration goals and any site-specific constraints such as limits on hydraulic head changes associated with groundwater pumping. The problems each incorporate numerical flow and transport models and objective functions that estimate remediation cost.

The first case study is a hypothetical case, whereas the second is a field-scale case. The hypothetical case is easy to compute and allows for a larger number of test runs, while the field-scale cannot be run many times because of its complexity but allows testing on a realistic problem. Further details on each case study are given in Sections 2.1 and 2.2 below.

2.1 Hypothetical Remediation Case Study

This case study has been examined in several previous papers (Smalley et al, 2000, Gopalakrishnan et al., 2003) and only a brief description will be given here. The site is a heterogeneous confined isotropic aquifer with dimensions of 480 m by 240 m, and 20 m deep. The contamination is from a BTEX spill and is based on characteristics of the

Borden site (see Smalley et al, 2000 for more information). The case study assumes steady state groundwater flow with an average conductivity of 2255.7 m/yr, mean hydraulic gradient of 0.00146, and porosity of 0.3. RT3D, a fate and contaminant transport model (see Clement et al., 1998), in combination with Modflow 2000 (see Harbaugh et al., 2000) was used to evaluate potential solutions. As in the case considered by Gopalakrishnan et al. (2003), for simplicity no reactions (such as biodegradation) are included in this case study.

The remediation technology to be designed in the optimization is pump and treat, with a maximum of three remediation wells available to be installed at 58 possible locations in the model, as shown in Figure 1. The dots in the center of the figure represent the candidate pumping well locations and the dots on the left and right sides represent constant head boundary conditions. In both cases, dots with overlaid squares signify monitoring well locations. The lines in the figure show the grid that is used in the numerical flow and transport model. Pumping rates were allowed to be set between 0 and 200 m^3/d. Treatment of the extracted water is through air stripping, since most BTEX compounds are semi-volatile.

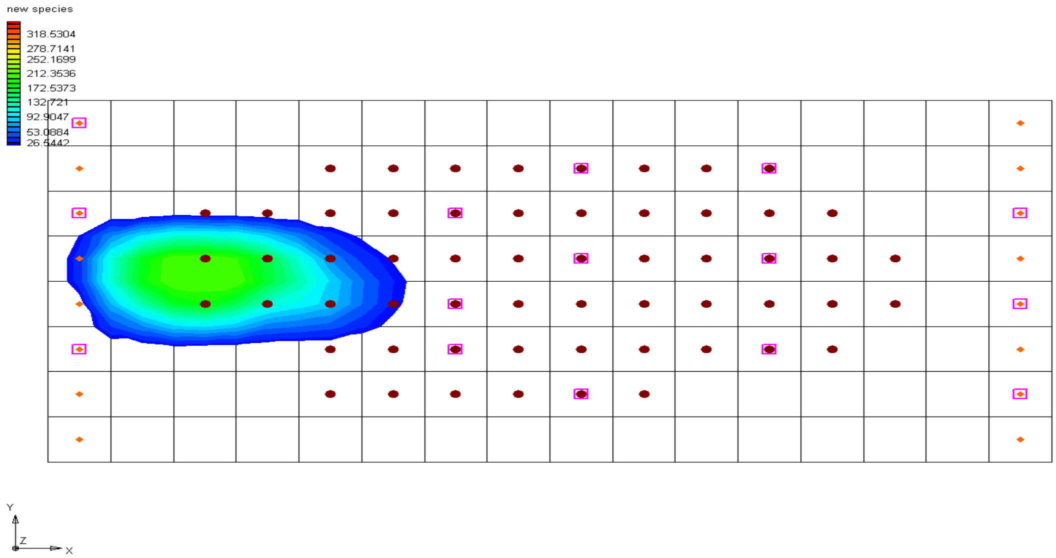

Figure 1. BTEX plume and numerical grid used for the hypothetical case study

The overall objective of the optimization is to identify well locations and pumping rates that minimize the overall cost of treatment, as follows:

$$\text{Min } C_{\text{total}} = C_{\text{remediation}} + C_{\text{monitoring}} + C_{\text{ex-situ}} \tag{1}$$

where C_{total} is the overall cost of the remediation plan, which is the sum of several cost components. $C_{remediation}$ incorporates capital and operating costs, $C_{monitoring}$ is the cost of on-site monitoring, and $C_{ex\text{-}situ}$ accounts for the cost of the air stripping technology used to treat any extracted water. More details on the first two terms are given by Smalley et al. (2000); Gopalakrishnan et al. (2003) gives more detail on the last term.

The constraints for the problem (shown in equations 2 through 4) are to maintain acceptable human health risk (based on concentration at an assumed exposure location

200 m offsite), operate within pumping rate limits, and maintain hydraulic heads within drawdown limits.

Equation 2 gives the risk constraint for time *t* and exposure location *k*. It incorporates, respectively, risk due to drinking water ($\mathrm{Risk}_{t,k}^{w}$), showering ($\mathrm{Risk}_{t,k}^{shw}$), and other non-consumptive uses ($\mathrm{Risk}_{t,k}^{nc}$). Smalley et al (2000) gives full details on how the terms of equation 2 are calculated. TR, total risk, is set to 10^{-5}.

$$\mathrm{Risk}_{t,k}^{TOTAL} = \mathrm{Risk}_{t,k}^{w} + \mathrm{Risk}_{t,k}^{shw} + \mathrm{Risk}_{t,k}^{nc} \leq \mathrm{TR}\,\forall t, \forall k \tag{2}$$

The constraint in Equation 3 gives the pumping rate, *u*, minimum and maximum values for each well *i*. The constraint in Equation 4 gives the minimum and maximum values on hydraulic heads for each well *i* at location *l*.

$$u_{min,i} \leq |u_i| \leq u_{max,i}\,\forall i \tag{3}$$

$$h_{min,l} \leq h_i \leq h_{max,l}\,\forall i \text{ at each } l \tag{4}$$

Equations 2 and 4 require constraint handling of the sort discussed in this paper. The constraint in equation 3, however, can be handled by the encoding of the decision variables within the GA.

2.2 Field-Scale Remediation Case Study

The Umatilla site is described in detail in other documents (Ren et al, 2003, Minsker et al, 2003) and will only be summarized here. Umatilla, an Army depot, has a total area of 19,728 acres and is contaminated with RDX and TNT that were disposed into unlined lagoons onsite. The Army Corps of Engineers (USACE, 1996 and 2000) created the simulation models for the site that are used in this study, which have 5 model layers and a numerical grid of 132 by 125 elements (see Figure 2). Modflow 2000 (Harbaugh et al., 2000) and MT3DMS (Zheng and Wang, 1998) were used to solve the simulation models. Hydraulic conductivities were obtained from pumping tests and divided into 17 zones. An existing pump-and-treat system is in place with three functioning extraction wells and three infiltration basins, and can be seen in Figure 2.

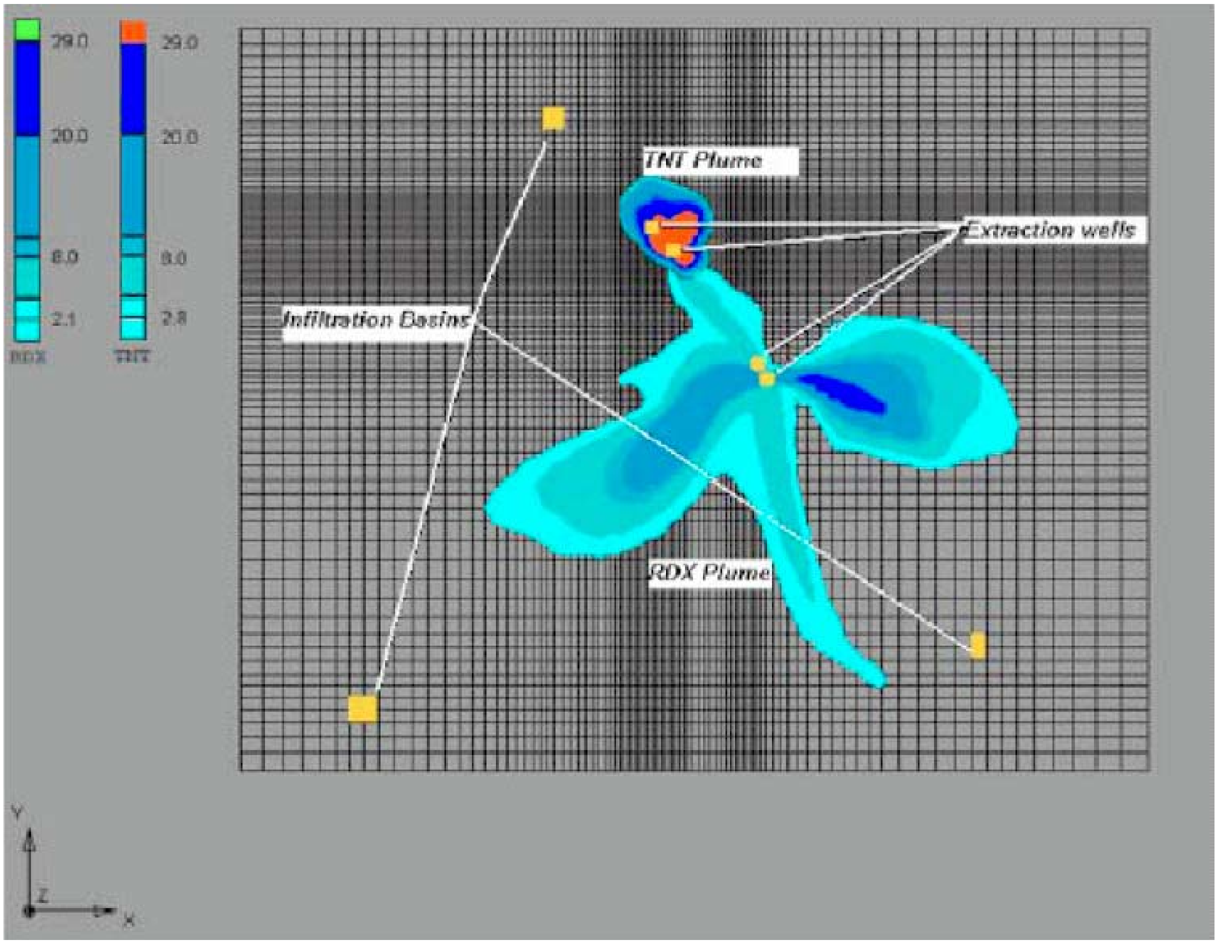

Figure 2. Modeled RDX and TNT plumes on numerical grid for Umatilla site.

Similar to the hypothetical case study, the optimization objective at the Umatilla site is to minimize overall remediation cost. This is done by specifying the pumping rates for the existing wells at the site, and by specifying locations and pumping rates for a maximum of four new extraction wells and three new injection basins. The objective (cost) function, which was created for a recently-completed flow-and-transport optimization demonstration project (Minsker et al. 2003), is as follows:

$$\text{Minimize}\,(\text{Total Cost} = \text{CCW} + \text{CCB} + \text{FCL} + \text{FCE} + \text{VCE} + \text{VCG} + \text{VCS}) \tag{5}$$

CCW: Capital costs of new wells

CCB: Capital costs of new recharge basins

FCL: Fixed costs of labor

FCE: Fixed costs of electricity

VCE: Variable costs of electricity

VCG: Variable Costs of Changing GAC Units

VCS: Variable costs of sampling

More details on these cost terms are given by Minsker et al (2003).

The optimization constraints in this case study include limits on the concentrations of each of the two contaminants at the end of the remediation (Equations 9 and 10), a re-injection condition to ensure that total injection equals total extraction (Equation 11), maximum limits on the pumping rates in each of two hydraulic zones (Equations 6 and 7), and maximum limits on the total amount pumped, due to treatment plant capacities (Equation 8). Of these six constraints, the first two (Equations 6 and 7) can be handled through the GA encoding, while the rest (Equations 8 through 11) must be handled with the methods described in section 3.

$$Q_1 \leq 360 \text{ gpm} \tag{6}$$

$$Q_2 \leq 900 \text{ gpm} \tag{7}$$

$$Q_1 + Q_2 \leq 1170 \text{ gpm} \tag{8}$$

$$C_{RDX} \le 2.1\,\mu g/l \tag{9}$$

$$C_{TNT} \le 2.8\,\mu g/l \tag{10}$$

$$Q_{injection} = Q_{extraction} \tag{11}$$

3. METHODOLOGY

This study uses a standard simple genetic algorithm (SGA), as described by Goldberg (1989), with the following configurations: tournament selection (with replacement for the hypothetical case and without replacement for the field-scale case, which performed best in previous research with these cases), $\mu + \lambda$ selection, and uniform crossover. Guidelines outlined in Reed (2000) were used for finding the GA parameter settings. For those without much GA background, we will define some important GA terms used in this study. An *individual* is a possible solution that is being considered. The *population* is the group of individuals that is present in the GA at any one time (*generation*). *Fitness* is a measure of an individual's desirability, that is, how well it performs in terms of objectives and constraints. In the cases studied here, lower fitness is better. *Tournament selection* is a method for choosing which members of the population (*parents*) will mate to create new individuals (*children*), a process favoring those with better fitness. *Replacement*, with respect to tournament selection, means whether or not an individual that has already mated will be allowed to mate again. *Crossover*, which can be random-point or uniform, is a way of combining two parent individuals to create a new child individual. *$\mu + \lambda$ selection* means that the parent population and the child population are combined and sorted, with the top 50% of individuals in this group forming the next parent population. *Convergence* occurs when the population becomes highly homogeneous and the GA is considered to be finished with the optimization. Convergence criteria were defined separately for each case and are as follows. For the hypothetical case, if 95 % of the population is within 0.1% of the minimum fitness in the generation, it is considered converged. For the field scale site, if 90% of the population is within 0.1% of the minimum fitness in the generation, it is considered converged. The difference in approaches is not significant, as convergence criteria are set by the GA user.

3.1 Traditional Penalty-Based Constraint Handling

Normally, constraints are implemented within nonlinear optimization methods through the use of penalty functions. An example of a linear penalty, which is used in this work as the base penalty-handling case, is given below.

$$Fitness = Cost + \sum_{j} k_j \cdot V_j \tag{12}$$

In equation 12, V_j is the violation of constraint j. Penalty weights, the k_j terms, are coefficients for each constraint j, and must be found through experimentation. For example, in the hypothetical case study, fitness is calculated as follows, using the constraints presented in Equations 2 and 4:

$$Fitness(x) = Cost(x) + \sum_{j=1}^{2} k_j . v_j(x) \tag{13}$$

$$v_1 = \sum_{t} \sum_{k} \begin{Bmatrix} \mathrm{Risk}_{\mathrm{t,k}}^{\mathrm{TOTAL}} - Risk_{t\arg et}, \text{ } if \text{ } \mathrm{Risk}_{\mathrm{t,k}}^{\mathrm{TOTAL}} > Risk_{t\arg et} \\ 0, \text{ } otherwise \end{Bmatrix} \tag{14}$$

$$v_2 = \sum_i \begin{Bmatrix} h_i - h_{\max}, if\ h_i > h_{\max} \\ h_{\min} - h_i, if\ h_i < h_{\min} \\ 0, otherwise \end{Bmatrix} \tag{15}$$

3.2 Adaptive Penalty Method

The first new approach is an adaptive penalty method that is similar to the standard penalty function approach, but computes the penalty weight automatically, thus avoiding trial-and-error parameter setting. This method was originally proposed by Barbosa & Lemonge (2003) with a fitness function calculation as follows.

$$Fitness(x) = \begin{Bmatrix} \overline{f}(x) + \sum_{j=1}^{m} k_j \cdot v_j(x), if\ x\ is\ not\ feasible \\ \cos t(x), if\ x\ is\ feasible \end{Bmatrix} \tag{16}$$

$$\overline{f}(x) = \begin{Bmatrix} \cos t(x), & if\ \cos t(x) > \langle f(x) \rangle \\ \langle f(x) \rangle, & if\ \cos t(x) < \langle f(x) \rangle \end{Bmatrix} \tag{17}$$

$$k(j) = \langle f(x) \rangle \cdot \frac{\langle v_j(x) \rangle}{\sum_{l=1}^{m} [v_l(x)]^2} \tag{18}$$

When individual *x* is feasible, no constraint violations would occur and the fitness simply becomes the cost of the design, *cost(x)*. If individual *x* is infeasible, its fitness is calculated using Equations 16, 17, and 18. In Equations 17 and 18, *<f(x)>* is the average cost of all individuals in the population, m is the number of constraints in the problem, $v_j(x)$ is the violation of the jth constraint for individual x, and $\langle v_j(x) \rangle$ is the average violation of constraint j over the entire population. Note that the information needed for this method does not require any external parameters to be set; rather, all needed information can be gathered from the current GA population.

This method is similar to the static penalty approach, with several differences. First, it recalculates the penalty weights (k_j) every generation. Second, the penalty weights are not user-supplied; instead, they are calculated based on the average penalty violation of each constraint (see Equation 18). Third, a baseline penalty is given for infeasible solutions, $\overline{f}(x)$, which insures that infeasible individuals will have a fitness greater than the average cost of the members of the population.

In implementing this method on the groundwater remediation case studies, the approach failed to converge to feasible solutions. When feasible solutions were found during the GA runs, they were pushed out of the mating population by cheaper, but infeasible, solutions. This problem occurs because finding feasible solutions groundwater remediation design problems (and other complex water resources optimization problems) can be difficult. Unlike typical GA test functions where these methods were developed and tested, a random groundwater remediation design population is not likely to contain initial feasible solutions. Therefore, careful constraint handling is critical to guiding the GA to find feasible individuals.

To address this issue, the above method was modified to use the average fitness of all *feasible* solutions instead of the average fitness of *all* members of the population, $<f(x)>$, in equations 17 and 18. The reasoning behind this modification is that the average fitness acts as a baseline penalty for infeasible solutions in equation 16. Raising this baseline to the average of the feasible solutions ensures that at least some feasible solutions will be maintained in future generations once they are found. In the cases where no feasible solutions have been found, the average fitness of all members of the population is used.

While the above modification was sufficient for the adaptive method to work well for the hypothetical case study, the method was still not able to find feasible solutions for the field-scale case because of scaling issues associated with multiple constraints. In particular, the constraint given in Equation 11 (total injection = total extraction) had violations with much higher magnitude than the other constraints, causing the adaptive method to create penalties that were too small when the other three constraints were violated. This issue frequently arises with penalty-based methods and was addressed by scaling the violations of each constraint to the range of 0 to 1. That is, at every generation, the maximum violation of each constraint over the population was found, and for every individual, their violations were scaled relative to those maximum violations.

3.3 Niched Approach

This method was developed by Deb and Agrawal (2003) and does not compute any penalties; instead, modifications are made to the selection mechanisms of the GA.

During tournament selection, a competition takes place so that feasible solutions are preferred over infeasible ones, infeasible solutions with smaller violations are preferred over solutions with larger violations, and feasible solutions with lower objective function values (in this case cost) are preferred over other feasible solutions. The same principle applies to the $\mu + \lambda$ process, in which the parent (μ) and child (λ) populations are combined and the best members are chosen for the final population according to these same rules.

4. RESULTS

Figures 3 and 4 (below) show the results of the experimental trials performed to test the constraint-handling methods for the hypothetical and field-scale case studies. The results for each case are described in more detail below.

4.1 Hypothetical Case Study

The hypothetical case study was designed to have realistic parameter values but is simple enough to allow evaluation of each solution rapidly, enabling the penalty methods to be tested on a large number of random initial populations. In addition, four population sizes were used for all three methods to identify any variations in performance for different population sizes. For each of the population sizes of 20, 40, 60, 80, and 100, each constraint-handling method was tested using 20 different random initial populations.

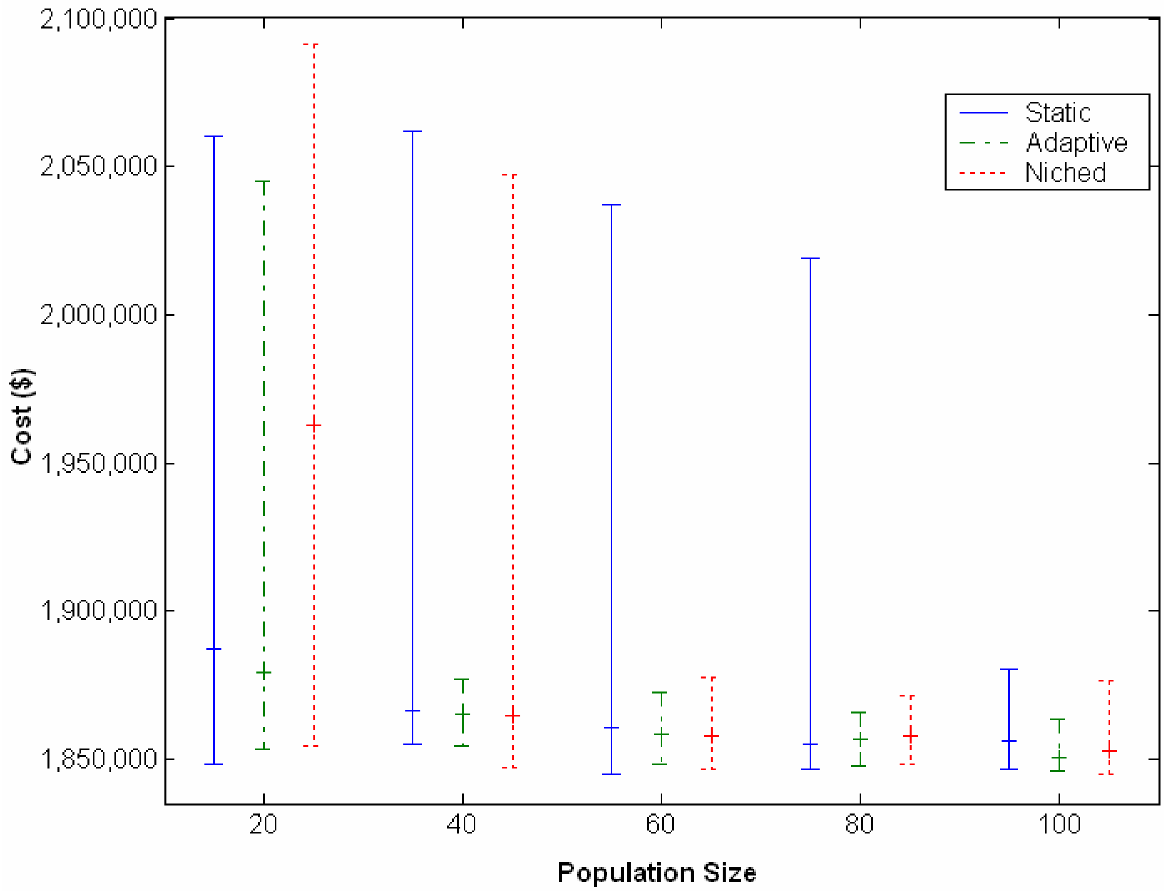

Figure 3. Fitness results for hypothetical case study. Bars show minimum, median, and maximum values.

From the above figure, it is clear that a population size of 100 is needed to use the traditional static method. The two new methods, however, are able to utilize smaller population sizes with consistent results. The adaptive method was the most efficient and was able to use a population size of just 40 individuals with highly consistent results; the niched approach performed nearly as well, with a minimum population of 60. Averaged over all population sizes and initial random populations, the niched method performed 0.4% better than the static method, and the adaptive method performed 1.1% better than the static method.

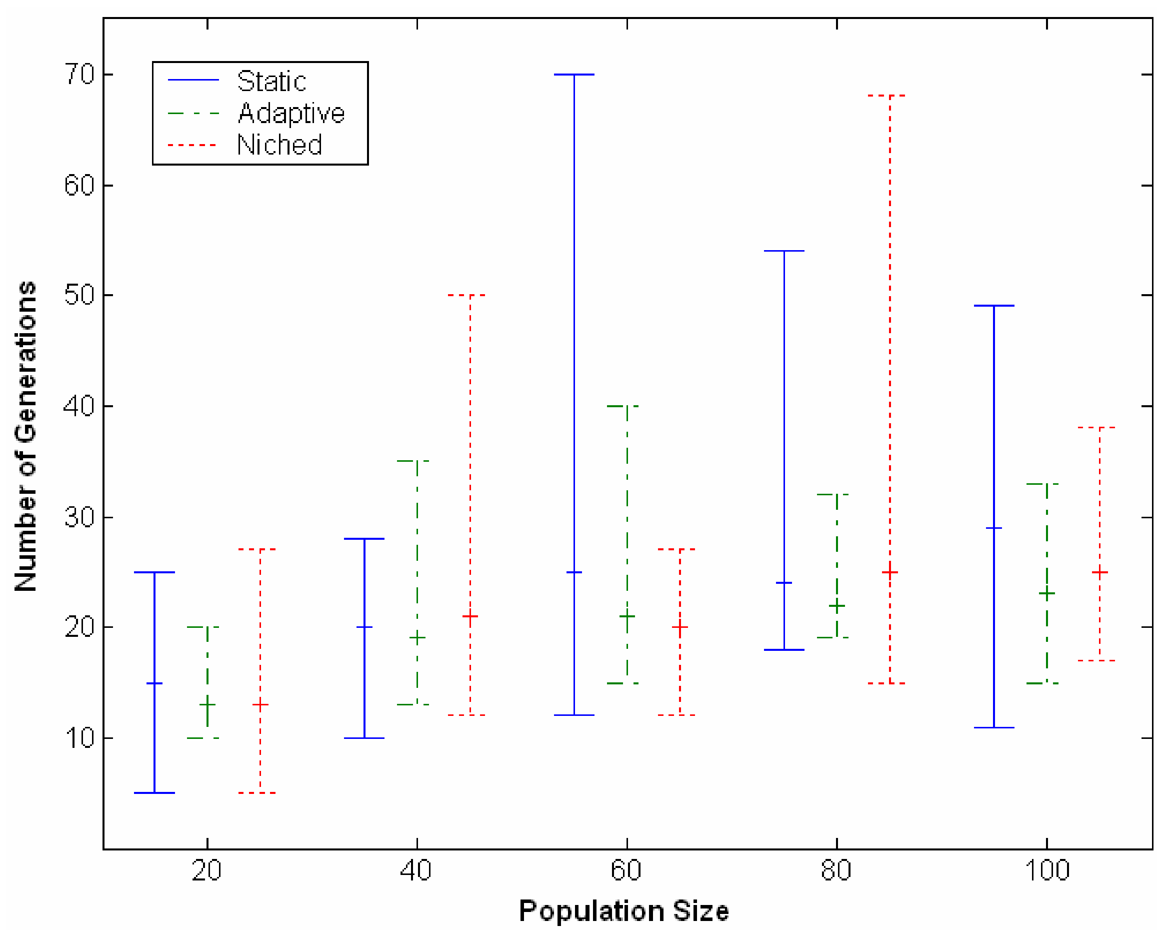

Figure 4. Time to convergence for the hypothetical case study. Bars show minimum, median, and maximum values.

Figure 4 shows the computational effort required to converge to the solutions shown in Figure 3. On average, across all population sizes and initial populations, the adaptive method has the fastest convergence time, about 11% faster than the traditional static penalty method. The niched method is close behind, performing about 7% better than the static method.

4.2 Field-Scale Case Study

In contrast to the hypothetical case, the numerical model for the field-scale case is quite complex and requires significant time to evaluate each potential solution. Thus, only one population size (160 individuals) and 10 initial random populations were used to verify the observations found in the hypothetical case study on a real-world case.

Figure 5 shows the fitness results, which indicate that all methods have similar performance. The niched method has the best performance with an average cost of $1,662,886 and worst case cost of $1,663,026. The adaptive method performed slightly worse than niched, with an average cost of $1,662,934 and worst case cost of $1,663,168. The static method's performance is close behind with an average cost of $1,662,946 and worst case cost of $1,663,097. The differences in fitness between all three methods is so insignificant (less than 1%) that they can be considered identical.

Figure 6 shows the convergence time for the different constraint-handling methods. The static method has the fastest convergence time, with an average of 35 generations and worst case performance of 52 generations. The adaptive approach performs somewhat worse than the traditional static penalty approach, with an average of 37 generations and worst-case of 59 generations. The niched approach takes the most time, with an average of 42 generations needed and a worst case of 65 generations needed. When considering the trial-and-error experimentation required to set the penalty weights for the static penalty approach, however, the overall computational effort for the niched approach and adaptive approach should still be significantly lower than the static penalty approach.

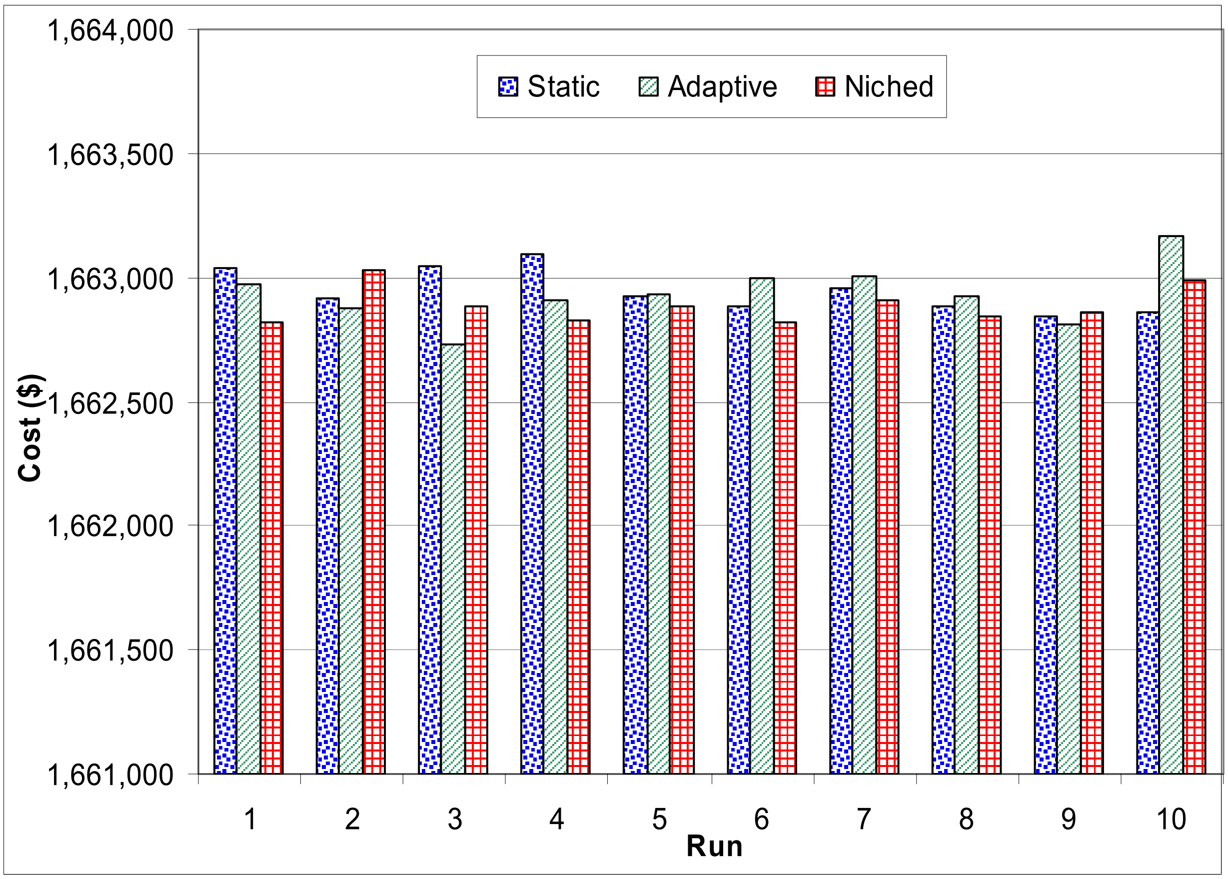

Figure 5. Fitness results for field-scale case study.

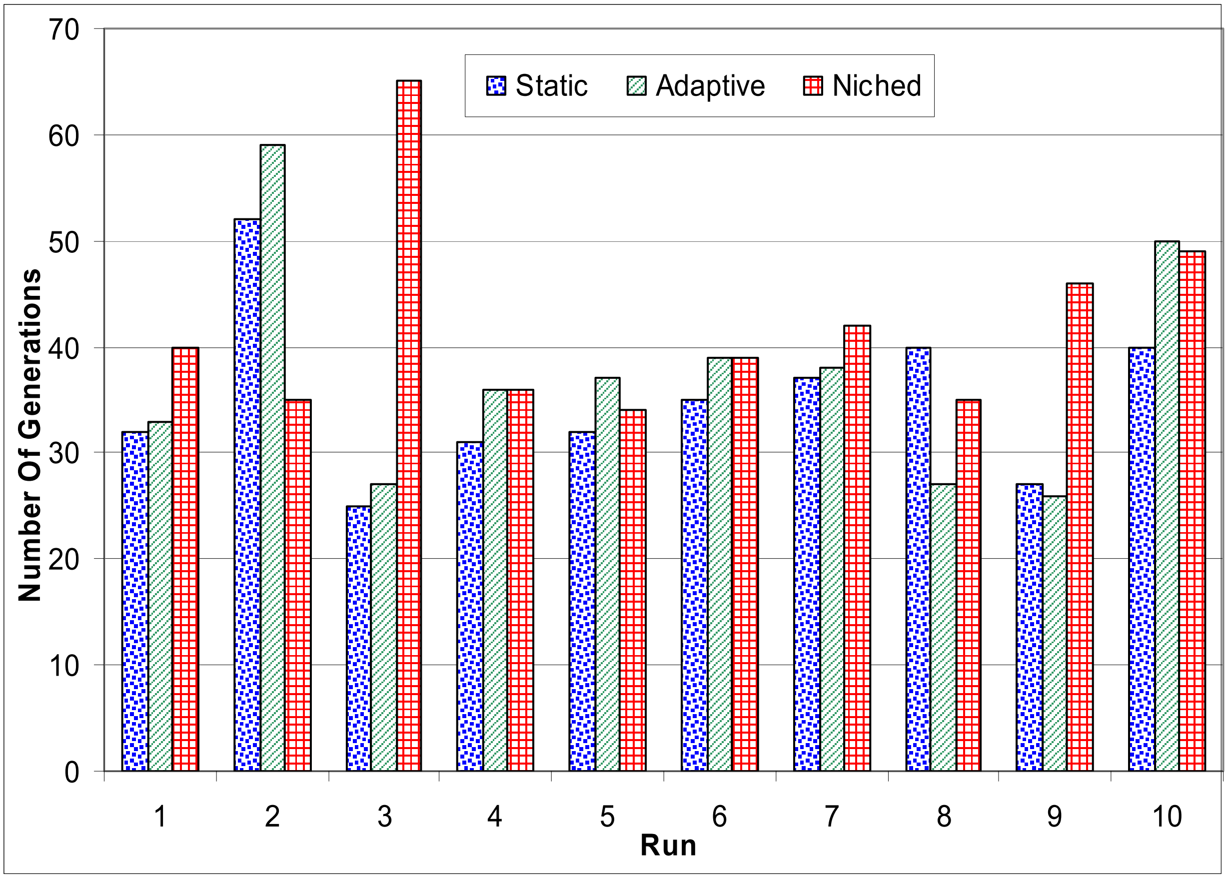

Figure 6. Time to convergence for field-scale case study.

5. CONCLUSION

Overall, both of the new methods show significant promise for use in water resources optimization problems. Results for both case studies were comparable or better than the traditional static penalty approach, with no trial-and-error experimentation required. The niched method is more general, because it can compare solutions based on criteria other than just cost (or another continuous fitness function), such as non-numerical or discrete constraints. The niched method does require modification of the GA selection operations, however, unlike the adaptive penalty method that, like the traditional penalty method, only requires changes to the fitness function. This could be an advantage for those using existing off-the-shelf GA solvers. The adaptive method sometimes encountered scaling difficulties with multiple constraints, but the modifications presented here appear to be robust enough to work without further modifications. Further testing is needed to verify these results on other water resources problems.

REFERENCES

Barbosa, HJC; Lemonge, ACC. 2003. A new adaptive penalty scheme for genetic algorithms. *Information Sciences* 156 (3-4): 215-251.

Barbosa, HJC; Lemonge, ACC. 2003. An adaptive penalty scheme for steady-state genetic algorithms. *Genetic And Evolutionary Computation - GECCO 2003, Pt I, Proceedings* 2723: 718-729.

Cieniawski, SE; Eheart, JW; Ranjithan, S. 1995. Using Genetic Algorithms To Solve A Multiobjective Groundwater Monitoring Problem. *Water Resources Research* 31 (2): 399-409.

Clement T. P., Sun, Y., Hooker, B. S. & Petersen, J. N. Modeling Multispecies Reactive Transport In Groundwater. Groundwater Monit. Remed. 18(2), 79–92, 1998.

Deb, K. 2000. An efficient constraint handling method for genetic algorithms. *Computer Methods In Applied Mechanics And Engineering* 186 (2-4): 311-338.

Goldberg, D. E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, New York, NY.

Gopalakrishnan, G., and B. Minsker, "Optimal sampling in a noisy genetic algorithm for risk-based remediation design," *Journal of Hydroinformatics*, 5(1), 11-25, 2003.

Harbaugh, A.W., E.R. Banta, M.C. Hill, and M.G. McDonald. 2000. *MODFLOW-2000, the U.S. Geological Survey modular ground-water model: User guide to modularization concepts and the ground-water flow processes*. U.S. Geological Survey Open-File Report 00–92.

Lemonge, ACC; Barbosa, HJC. 2004. An adaptive penalty scheme for genetic algorithms in structural optimization. *International Journal For Numerical Methods In Engineering* 59 (5): 703-736.

Minsker, B., Y. Zhang, R. Greenwald, R. Peralta, C. Zheng, K. Harre, D. Becker, L. Yeh, and K.Yager (2003). *Final Technical Report for Application of Flow and Transport Optimization Codes to Groundwater Pump and Treat Systems*, Environmental Security Technology Certification Program. Available at http://www.frtr.gov/estcp.

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

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

U.S. Army Corps of Engineering (USACE). (1996). Final remedial design submittal, contaminated groundwater remediation, explosives washout lagoons, Umatilla Depot Activity, Hermiston, OR.

U.S. Army Corps of Engineering (USACE) (2000) Explosives washout lagoons groundwater model revision (preliminary draft), Umatilla Chemical Depot, Hermiston, OR.

Wang, C., and Zheng, C. (1998). "Groundwater management optimization using genetic algorithms and simulated annealing: formulation and comparison." Journal of American Water Resources, 34(3), 519-530.