Comparing Advanced Genetic Algorithms and Simple Genetic Algorithms for Groundwater Management

Rachel Arst[1] and Barbara Minsker[2]
David E. Goldberg[3]

Abstract

The optimal design of groundwater remediation systems (well locations and pumping rates) is an important problem in environmental engineering. Since this problem is non-linear and very complex, it can be computationally expensive to solve. In this paper, the Extended Compact Genetic Algorithm (ECGA) and the Bayesian Optimization Algorithm (BOA), two advanced genetic algorithms, are tested to determine whether they will decrease computational time.

The identification of the links between the building blocks (short, highly fit chromosome sequences), called linkage learning, is the main approach by which more effective genetic algorithms are created. Both BOA and ECGA use linkage learning, but in slightly different ways. The ECGA uses probabilistic modeling of the population to learn linkage. BOA improves on that by using Bayesian networks to estimate joint distributions that are used to generate new candidate solutions. The performances of these algorithms are compared with simple genetic algorithms to demonstrate their computational power for several different types of problems with different levels of complexities. This paper summarizes the methodology; results will be presented at the conference.

Introduction

Genetic algorithms (GAs) have been used for many water resources applications, including groundwater remediation design, optimal reservoir system operation, calibrating rainfall-runoff models, remediation policy selection, and solving multiple objective groundwater pollution contaminant problems (e.g., Wang 1991; Ritzel *et al*, 1994; Wagner, 1995; Wang and Zheng, 1997; Wardlaw and Sharif 1999; Aksoy *et al*, 2000; Smalley *et al*, 2000; Dandy and Engelhardt, 2001). Researchers have usually solved groundwater quality management problems by combining groundwater flow and solute transport models with mathematical optimization to model the subsurface behavior and determine the best management strategy. Groundwater remediation is usually very expensive and simulation/optimization can reduce costs substantially (Zheng *et al*, 2002). One of the reasons that genetic algorithms have been chosen to solve such problems is that they are able to solve the non-convex, discrete, discontinuous problems (Goldberg, 1989) that arise frequently in these applications. Simple GAs (SGAs) have been used for this purpose, but they can be very slow and don't always converge to the optimal solution. This study will investigate whether new advanced GAs can solve groundwater remediation design problems faster and more reliably than the simple GA.

Background

[1]Department of Civil and Environmental Engineering, University of Illinois, Urbana, IL 61820. arst@uiuc.edu.
[2]Department of Civil and Environmental Engineering, University of Illinois, Urbana, IL 61820. minsker@uiuc.edu.
[3]Department of General Engineering, Director of Illinois Genetic Algorithms Laboratory, University of Illinois, Urbana, IL 61820.

Simple Genetic Algorithm Basics

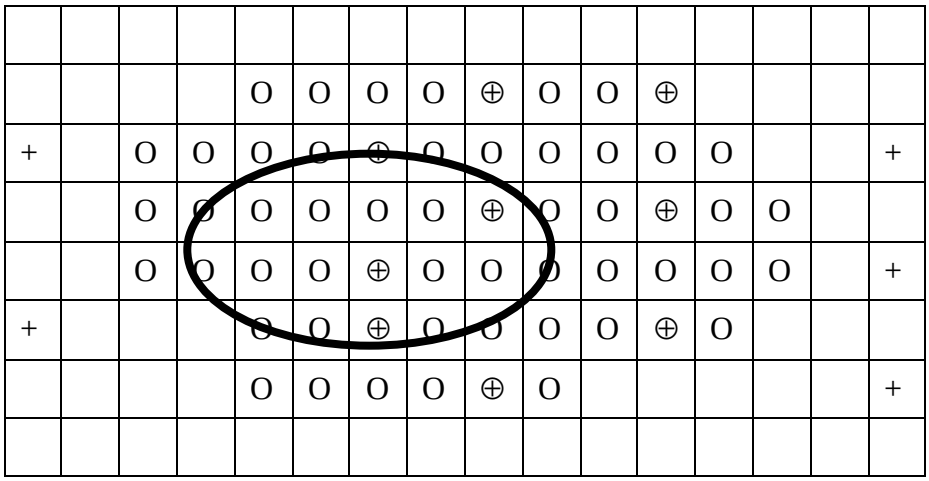
GAs are search algorithms based on the survival of the fittest theory. The search is usually done with a group or population of binary strings, which are encoded forms of the decision variables. Each string is analogous to a chromosome and each binary bit is analogous to a gene on that chromosome. There are three operators used in the simple GA: selection, crossover, and mutation. First, the objective function, also called the fitness function, is evaluated for each string. Selection occurs when the individuals with higher fitness values are assigned higher probabilities of producing offspring for the next generation. Therefore, the highly fit chromosomes will have a larger number of copies in the succeeding generation. There are several different types of selection used in GAs. We use tournament selection in this work, which allows only the fittest individual from a randomly selected group from the population to be put into a mating pool. After the mating pool is established, crossover takes place. Individuals in the mating pool are mated by randomly selecting place(s) to divide the two chromosomes and then exchanging the pieces. After crossover, mutation randomly switches a bit or bits in the chromosome with a user-specified probability. Using these operations, more fit members of the population are created over time and the population evolves to optimal or near-optimal solutions. The way the GA assembles optimal solutions can be described by building block theory. Building blocks are short, low order, and highly fit groups of binary digits in the chromosome (Goldberg, 1989). Certain combinations of building blocks are more fit than others and will ultimately take over through the survival-of-the-fittest operations.

Probabilistic Model Building GAs (PMBGAs)

A class of GAs called competent GAs have recently been introduced (Goldberg 1999). These competent GAs are able to solve hard problems (ones that earlier GAs couldn't) quickly, accurately and reliably (Sastry and Xiao, 2001). Linkage learning GAs are competent GAs that identify the important building blocks to be conserved under crossover. Crossover can disrupt building blocks and impede the GA from finding good solutions, so models of the relationships among building blocks are needed to ensure that correlated genes stay together for better algorithm performance. Linkage learning GAs use building block identification to create efficient, faster, and more accurate GAs. These GAs use model-building instead of genetic recombination to guide the search (Harik,1999). These are different from the simple GA in that they use an explicit model of promising chromosomes as opposed to the implicit model that is created by genetic recombination (Pelikan and Goldberg, 2000). The ECGA and the BOA are linkage learning GAs that use probabilistic chromosome modeling techniques and are termed probabilistic model-building genetic algorithms (PMBGAs). This type of algorithm, specifically the ECGA, has been used successfully by Sastry and Goldberg (Sastry and Goldberg, 2000) to optimize a binary fluid power cycle. They formulated the problem as a non-linear constrained problem. These algorithms are described in detail in the following sections.

Case Study Application

A groundwater remediation system for a heterogeneous, isotropic, and confined hypothetical BTEX-contaminated aquifer was used to compare the efficiency of the ECGA and the BOA.



+ Observation wells (15)
O Possible Remediation wells (58)

Figure 1. The plan view of the case study aquifer.

Pump-and–treat technology was used with a goal of minimizing the pumping cost to meet the target risk level. There are 58 possible well locations for 3 remediation wells to be placed. There are 15 locations that are set aside for observation wells. The extracted water is treated by air stripping. The possible remediation well locations and the observation well locations are shown in Figure 1.

The dimensions of the area shown are 240 x 480 meters, which was modeled by using finite difference on a 16 x 8 meter grid. MODFLOW (McDonald and Harbaugh 1988) and RT3D (Clement 1997; Clement *et al* 1998) computer models were used to model flow and contaminant transport. The transport was modeled with advection, dispersion and linear adsorption. An analytical model (Smalley *et al*, 2000) was used to predict the concentrations at the observation wells in Figure 1. The aquifer characteristics are presented in Table 1.

Table 1. The case study aquifer characteristics.

Parameter	**Value**
Average hydraulic conductivity K (m/day)	6.12
Porosity n	0.3
Longitudinal dispersivity α_L (m)	15.00
Transversal dispersivity α_T (m)	3.00
Aquifer thickness b (m)	20.00
Retardation coefficient R	1.41

There are four decision variables in this problem: pumping well location, pumping rate(u_i), injection or extraction (Y_i), and installation (X_i) where:

$$X_i = \begin{cases} 1 & \textit{if a pumping well is installed at location } i \\ 0 & \textit{Otherwise} \end{cases}$$

$$Y_i = \begin{cases} 1 & \textit{for injection from well } i \\ 0 & \textit{for extraction from well } i \end{cases}$$

The objective function that was used to determine the cost is as follows:

$$\text{Min } C_{TOT} = C_{REM}(u_i, h_i) + C_{MON}(X_i) + C_{SYST}(u_i)$$

where
C_{REM} = the capital and operating costs for the wells
C_{MON} = the cost of on-site monitoring
C_{SYST} = the additional capital and operating costs for the ex-situ treatment system

The details of the monitoring and remediation costs are presented by Smalley *et al* (2000), while the details of the C_{SYST} evaluation, done using RACER (1999), a parametric cost modeling system, is presented by Gopalakrishnan *et al* (in press 2002).

The constraints, which Smalley *et al* (2000) present in more detail, are put on the human health risk, the pumping rates, and the hydraulic drawdowns. The constraints are as follows:

$$\text{Risk}_{t,k}^{TOTAL} = \text{Risk}_{t,k}^{w} + \text{Risk}_{t,k}^{shw} + \text{Risk}_{t,k}^{nc} \leq \quad TR \;\forall t, \forall k$$

where:
$\text{Risk}_{t,k}^{w}$ = the cancer risk from ingestion of contaminated drinking water
$\text{Risk}_{t,k}^{shw}$ = the cancer risk from inhalation of volatiles from contaminated water in the shower
$\text{Risk}_{t,k}^{nc}$ = the cancer risk from inhalation of volatiles from contaminated water during non-consumptive uses (except from showering)
$\text{Risk}_{t,k}^{TOTAL}$ = the total lifetime carcinogenic risk
TR = the target risk level
t =time
k = location

This human health risk constraint ensures that the lifetime risk that is evaluated at locations k and at times t is less than the target level, TR. The constraints on the pumping rates and hydraulic heads are as follows:

$$u_{min,i} \leq \left| u_i \right| \leq u_{max,i} \qquad \forall i$$

$$h_{min,l} \le h_{i,l}(u_i) \le h_{max,l} \qquad \forall i, \forall l$$

where:
$u_{min,i}$ = minimum pumping rate for remediation well i
$u_{max,i}$ = maximum pumping rate for remediation well i
$h_{i,l}$, = computed hydraulic head (meters) for remediation well i at location l
$h_{min,l}$ = minimum hydraulic head (meters) for remediation well i at location l
$h_{max,l}$ = maximum hydraulic head (meters) for remediation well i at location l

The fitness of the design can then be evaluated as follows:

$$Fitness = C_{TOT} + w_1 \cdot Risk\ violation + w_2 \cdot Head\ violation$$

where:
ω_1 = the penalty weigh for the risk constraint
ω_2 = the penalty weight for the head constraint

Both of these values are set to 1000 in this case study. A penalty was not needed for the pumping rate since the binary representation of this decision variable in the PMBGA will limit it directly.

Methodology

The Extended Compact Genetic Algorithm (ECGA)

Overview. The class of probability distributions used in the ECGA is marginal product models (MPMs). MPMs represent relationships among genes in the chromosomes through products of marginal distributions. The genes are clustered into building blocks and each cluster is treated as a group that cannot be disrupted by crossover. The MPM in the ECGA uses a MDL (minimum descriptions length) model. These models are built on the premise that when all other things are equal, the simpler distributions are better than the complex ones. In each generation of the ECGA, a greedy search is performed to identify a probability distribution that best models the correlations among chromosomes within good members of the population. The optimal probability distribution is found by assigning higher fitness values to the distributions that represent relationships among the genes in the most compact fashion . The ECGA is based on the fact that learning probability distributions over multi-variate spaces is equivalent to linkage learning (Sastry and Goldberg, 2000) and that linkage learning is equivalent to building block identification (Harik, 1999). In the ECGA, the initial random population is generated, the fitness of the chromosomes is evaluated, and tournament selection is performed in the same manner as in the SGA. The MPM model is then built from the mating population using the MDL. Crossover is then performed as in the SGA, but is only allowed to occur between clusters in the MPM. This modified form of crossover preserves the critical relationships among the genes of highly fit chromosomes. This procedure is followed until a specified percent convergence is reached.

Steps to Ensure High Quality Solutions.

Setting GA parameters properly can be a time-consuming trial and error process. The following 3-step method, an extension of the 3-step method developed by Reed *et al* (2000), may provide an efficient method for ensuring algorithm convergence to optimal solutions. The following method is currently being tested and final results will be presented at the conference and in the thesis Arst *et al* (2002).

Step 1: Preliminary Problem Analysis

Population Sizing

In this step, basic calculations will be done to find a range of population sizes. The following equation that was developed by Goldberg, Deb, and Clark (1992) is tested. This equation gives a more conservative result than the one used by Reed *et al* (2000) for the SGA. Both equations are used to identify population sizes that allow the GA to decide correctly between the best and second-best individuals in an environment that contains noise between the building blocks. The equation Reed *et al* used is based on a random walk model, which assumes that errors can be made as long as the correct building blocks are eventually found. Since ECGA and BOA are creating models of building block correlations, this study will test whether the more conservative equation below is needed, which assumes that correct decisions between the best individuals must be made in the first generation. If the algorithm makes the wrong decisions in the first generation, then it won't be able to recover and will converge sub-optimally. Sastry and Goldberg (2000) derived an empirical equation for population sizing in the ECGA, but our initial experiments indicated that this equation is far too conservative for our problems.

$$n = 2 \cdot c(a) \cdot 2^{k} \cdot m' \cdot \frac{s_M^2}{d^2} \qquad (1)$$

where:

$c(\alpha)$ = the square of the ordinate of a unit normal distribution with a probability of α.

k = the order of the building block

m' = is the number of building blocks in the chromosome minus 1.

σ_M^2 = is the average fitness variation of the partition

d = the signal difference between the best and second best building blocks

To estimate σ_M, we use the following relationship from (Reed *et al* 2000), assuming that σ_M^2 is approximately equal to σ_{BB}^2, the standard deviation of the building blocks.

$$s_{fitness} \approx s_M^2 \sqrt{p(m-1)}$$

Rearranging, we obtain

$$s_M^2 \approx \frac{s_{fitness}^2}{p(m-1)}$$

and substitute into Equation (1) to get

$$n = 2 \cdot c(a) \cdot 2^k \cdot \frac{s_{fitness}^2}{pd^2}$$

The possibilities for n can be calculated using integer values of k between 1 and 5.

Time to Convergence
Currently, runs are being done to test several candidate methods for calculating the number of generations needed for the ECGA to converge. Results from these runs will be presented at the conference.

Is the ECGA right for you?
It is possible that the ECGA might be too computationally complex for some problems, so it is important to calculate the number of function evaluations for your problem to determine if the necessary computing time is available. The computational complexity can be calculated as follows,

computational complexity = $n*t_{conv}$

where computational complexity is the estimated number of fitness function evaluations required to solve the problem and t_{conv} is the number of generations the ECGA must perform to reach convergence. Given an estimate of the time for each fitness function evaluation, the overall computing time to solve the problem can then be approximated.

Step 2: Parameter Settings

In this study, we will test whether the probability of crossover must be set using this guideline from Reed *et al* (2000), which was developed for tournament selection in the simple GA (Thierens, 1995).

$$P_c \leq \left(\frac{s-1}{s} \right)$$

In the SGA, the probability of crossover should remain below this limit because higher levels can cause too many of the important building blocks to be destroyed. However, higher crossovers may be possible with the ECGA because it preserves important linkages. We started with this guideline to obtain a conservative estimate of the crossover and will do testing to determine if the crossover probability can be higher for the ECGA.

Step 3: Trial Runs

Trial runs need to be done to determine the exact population size from the sizes calculated in Step 1. Test runs are done for k equal to 1 and 2. If the fitness remains the same from the first trail run to the second, then no more runs are necessary. If k = 2 has a better fitness than k = 1, then run k = 3 and continue until there isn't a change in the fitness. Ideally this will happen at k

no large than 5. It is also important to make sure that a large majority (~ 80%) of the individuals in the population have the same fitness when the final population size is chosen.

The Bayesian Optimization Algorithm (BOA)

Overview. The BOA uses Bayesian networks as its probabilistic model (Pelikan, Goldberg, Cantú-Paz, 2000 and Pelikan, Goldberg, and Sastry, 2000). A Bayesian network is a directed acyclic graph, where each node represents a gene in the chromosome and the edges of the graphs represent the critical correlations among the genes that create highly fit chromosomes. Like the simple GA, the initial population is generated randomly. Any selection method can be used to select the best chromosomes from the initial population, but in this work we use tournament selection. A Bayesian network is then created using a greedy search in which edges between the most highly correlated genes are added to the network first. New chromosomes are generated using the joint distribution represented by the network. The new chromosomes are then added to the old population, replacing a specified percentage of the old population. This process continues until convergence is reached. The BOA is able to work independently of the building block length and can identify the building blocks as it goes along by using the information from the selected chromosomes. For more details on the algorithm, see Pelikan, Goldberg, and Cantú-Paz (1999).

Steps to Ensure High Quality Solutions.
The approach for BOA is still under development, but we expect that the steps will be similar to that described previously for the ECGA. Final results will be presented at the conference and in the thesis Arst *et al* (2002).

Results

Currently, runs are in progress and results will be presented at the conference.

Summary and Conclusions

The ECGA and the BOA are competent GAs that use linkage learning to improve GA performance and computational efficiency. These algorithms, which are part of a class of GAs known as probabilistic model building GAs, build different types of models of the correlations among the genes in current solutions that perform well. These models are then used to generate new solutions with similar statistical properties to the best current solutions, enabling a more informed search than would be possible with the traditional operations of crossover and mutation in the SGA. The results of the comparison between ECGA, BOA, and the SGA will be presented at the conference.

Acknowledgments

This material is based upon work supported by, or in part by, the U. S. Army Research Office under grant numbers DAAD19-00-1-0389 and DAAD19-001-1-0025.

List of References

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

Clement, T. P, Sun, Y., Hooker, B. S., and Petersen, J. N., (1998). "Modeling multi-species reactive transport in groundwater." *Ground Water Monitoring and Remediation*, 18(2), 79-92.

Clement, T. P., (1997). "RT3D - A modular computer code for simulating reactive multi-species transport in 3-Dimensional groundwater aquifers." Battelle Pacific Northwest National Laboratory Research Report, PNNL-SA-28967. http://bioprocesses.pnl.gov/rt3d.htm>.

Clement, T. P., Johnson, C. D., Sun, Y., Klecka, G. M., and Bartlett, C., (2000). "Natural attenuation of chlorinated solvent compounds: Model development and field-scale application." *Journal of Contaminant Hydrology*, 42, 113-140.

Dandy, G. C., and Engelhardt, M. (2001). "Optimal Scheduling of Water Pipe Replacement Using Genetic Algorithms." *Journal of Water Resources Planning and Management*, 127(4), 214-223.

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

Goldberg, D. E., Deb, K., Clark, J. H. (1992). "Genetic Algorithms, noise, and the sizing of populations." *Complex Systems*, 32(1) 10-16.

Harik, G. R., Cantú-Paz, E., Goldberg, D. E., and Miller, B. L. (1997), "The gambler's ruin problem, genetic algorithms, and the sizing of populations." *Proceedings of the 1997 IEEE Conference on Evolutionary Computation*, p. 7-12, IEEE Press, Piscataway, NJ.

Harik, G., (1999). "Linkage Learning via Probabilistic Modeling in the ECGA." Illinois Genetic Algorithms Laboratory, Department of General Engineering. http://www-illigal.ge.uiuc.edu>. Technical Report No. 99010, January.

McDonald, M.G., and Harbaugh, A.W. (1988). "A modular three-dimensional finite-difference ground-water flow model." Techniques of Water Resources Investigations 06-A1, United States Geological Survey.

Pelikan, M., Goldberg, D. E., Cantú-Paz, E. (1999). "BOA: The Bayesian Optimization Algorithm" Joint meeting; 8th – July: Orlando, FL, International conference on genetic algorithms, GECCO-99, p.525-532.

Pelikan, M., and Goldberg, D. E., (2000). "Research on the Bayesian Optimization Algorithm." Illinois Genetic Algorithms Laboratory, Department of General Engineering. http://www-illigal.ge.uiuc.edu>. Report No. 2000010, February.

Pelikan, M., Goldberg, D. E., and Cantú-Paz, E. (2000). "Bayesian Optimization Algorithm, Population Sizing, and Time to Convergence." Illinois Genetic Algorithms Laboratory, Department of General Engineering. http://www-illigal.ge.uiuc.edu>. Report No. 2000001, January.

Pelikan, M., and Goldberg, D. E., (2001). "Bayesian Optimization Algorithm, Decision Graphs, and Occam's Razor." Illinois Genetic Algorithms Laboratory, Department of General Engineering. http://www-illigal.ge.uiuc.edu>. Report No. 2000020, May.

Reed, P., Minsker, B., and Goldberg, D. E. (2000). "Designing a Competent Single Genetic Algorithm for Search and Optimization." *Water Resources Research*, 36(12), 3757-3761.

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

Sastry, K., and Goldberg, D. E. (2000). "On Extended Compact Genetic Algorithm." Illinois Genetic Algorithms Laboratory, Department of General Engineering. http://www-illigal.ge.uiuc.edu>. Report No., 2000026, April.

Sastry, K., and Xiao, G., (2001). "Cluster Optimization Using Extended Compact Genetic Algorithm." Illinois Genetic Algorithms Laboratory, Department of General Engineering. http://www-illigal.ge.uiuc.edu>. Technical Report No. 2001016, January.

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

Thierns, D. (1995). "Analysis and Design of Genetic Algorithms." Doctoral Disseration, Kathol. University Leuven, Leuven, Belgium.

Wagner, B. J. (1995). "Recent Advances in Simulation Optimization Groundwater Management Modeling." in *Reviews of Geophysics*, U.S. National Report to IGUU 1991-1994, American Geophysical Union, Washington DC, 1021-1028.

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

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

Wardlaw, R., and Sharif, M. (1999)."Evaluation of Genetic Algorithms for Optimal Reservoir System Operation." *Journal of Water Resources Planning and Management*, 125(1), 25-33.

Zheng, C., and Bennett, G. D. (2002). "Applied Contaminant Transport Modeling." Wiley-Interscience, New York.