EVALUATION OF ADVANCED GENETIC ALGORITHMS APPLIED TO GROUNDWATER REMEDIATION DESIGN

BY

MARCIA SHANTA HAYES

B.S., University of Illinois at Urbana-Champaign, 1999

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, 2005

Urbana, Illinois

ABSTRACT

Genetic algorithms (GAs) have been used extensively for water resources optimization because of their flexibility and global search capabilities, but computational requirements for GAs can be substantial. This thesis evaluates a new GA, the hierarchical Bayesian Optimization Algorithm (hBOA), which is designed to reduce computational effort and improve accuracy. GAs operate by assembling highly fit segments of chromosomes (potential solutions), called building blocks. The hBOA enhances the efficiency of this process by using a Bayesian network to create models of the building blocks. The building blocks are nodes on the network, and the algorithm uses the network to generate new solutions, retaining the best building blocks of the parents. This work compares the performance of hBOA to a simple genetic algorithm (SGA) in solving a hypothetical case study and a field-scale remediation design to determine if any benefit can be gained through the use of this approach. The hBOA gave more reliable results than the SGA on the hypothetical case, particularly at lower population sizes. This reduced the population size required to reliably obtain the optimal solution by 50%. For the field-scale case, however, both algorithms gave similar reliability at the same population size. Further research is needed to explore whether implementing an elitist strategy for hBOA would increase its reliability on this case.

To My Family

ACKNOWLEDGEMENTS

There are many people I need to thank for helping me in this endeavor. First, I thank my advisor, Dr. Barbara Minsker, for all of her support during this project. Her guidance helped me to understand what it takes to do good research. Her words of encouragement motivated me when nothing was working.

I also thank the members of my research group for all of the advice and support over the last two years. I appreciate the time that you took to answer my questions, to look at my code, and to listen to my frustrations. I especially thank Abhishek Singh, Shanquan Yan, and Eva Sinha for their guidance.

I extend my gratitude to my friends and family for their support during the last two years. Thank you for listening to all of the updates and complaints, and for watching the victory dances. Mom, Dad, and Chris, thanks for everything. Kathryn, Brooke, and Earl, I wouldn't have made it without you. My thanks to the Peoria Jaycees for your support, encouragement, and much needed diversions from grad school.

And I also thank Dr. David Workman for starting me on the path to engineering.

This work was supported by the U. S. Army Research Office under Grant No. DAAD19-001-1-0025.

TABLE OF CONTENTS:

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

In January 2002, the U.S. EPA concluded that pump-and-treat is the most common form of remediation at sites on the National Priorities List with groundwater contamination; there are 700 such sites (USEPA, 2003). Pump-and-treat involves substantial capital cost for installing pumping wells, plus ongoing expenses to treat extracted water aboveground. Because these two factors drive the cost of this technology, it is critical to determine the optimal placement and pumping rates of all wells in a timely and cost effective manner.

There is a large body of work that implements optimization techniques for groundwater remediation design (see, e.g., *Wang et al* (2002), *Papadopoulou et al* (2003), or *Zhi et al* (2003) for examples), and genetic algorithms (GAs) in particular (see, e.g., *Aly et al* (1999)*, Bayer et al* (2004), or *Guan et al* (2004) for examples). This work compares three GAs: the simple GA (SGA) that was used in most previous studies, the Bayesian Optimization Algorithm (BOA), and the hierarchical Bayesian Optimization Algorithm (hBOA). This study builds on the efforts of Arst (2002), who found the BOA to be a promising optimization technique for pump-and-treat design on a simple case study. The hBOA is the next generation of BOA, modifying the Bayesian network so that it is more reliable at solving difficult problems. This work investigates the possible gains of using these advanced GAs for pump-and-treat design. The three GAs are first evaluated on a hypothetical case study to develop an initial understanding of their behavior. Then, the best methods are also tested on a field-scale site.

The organization of this paper is as follows. Chapter Two details the characteristics of the hypothetical case study. Chapter Three describes the objective function and the remediation design of the field-scale site, the Umatilla Army Depot. Chapter Four describes the methodology, and Chapter Five contains the results of the research. Finally, Chapter Six contains conclusions and summary of the work.

2. CASE STUDY APPLICATION

The GAs in this work are first evaluated on a hypothetical case study. Section 2.1 describes the case study, while Section 2.2 details the optimization formulation.

2.1 Hypothetical Case Study Description

The three GAs were first tested on a hypothetical site described by Smalley et al. (2000) and Gopalakrishnan et al. (2003). It is an isotropic, homogeneous aquifer of size 240 x 480 m contaminated with BTEX (benzene, toluene, ethylene, and xylene) at a peak concentration of 200 mg/L. The pumping system allows a maximum of three wells, chosen from 58 possibilities, each with a maximum pumping rate of 250 m^3/day. Of the 58 possible locations, 15 are observation wells. A grid of eight rows by 16 columns was superimposed on the site. The flow of contaminant was modeled by MODFLOW (McDonald and Harbaugh, 1988) while the contaminant levels were determined by RT3D (Clement 1997; Clement et al., 1998, 2000). Transport includes advection, dispersion, and linear adsorption. Figure 2.1 depicts this grid, as well as the locations of wells and plume concentrations.

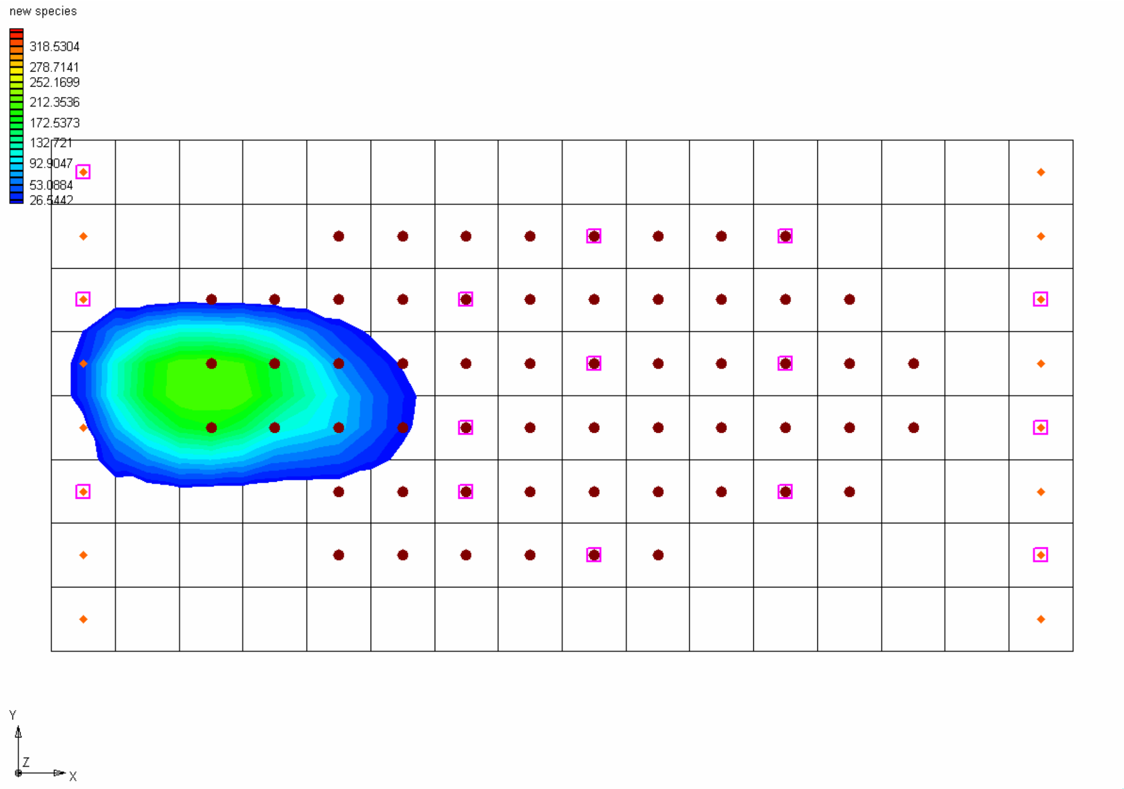

Figure 2.1: Grid, plume and well locations for hypothetical case (Source: *Yan*, 2003)

2.2 Hypothetical Case Study Optimization Formulation

The optimization formulation for the case study, developed by Smalley et al. 2000 and Gopalakrishnan et al. 2003, minimizes the cost of a pump-and-treat technology system subject to a human health risk target, pumping rate constraints, and hydraulic head constraints. The optimization problem is as follows:

Minimize $C_{TOT} = C_{REM} + C_{MON} + C_{SYS}$ (2.1)

$$Risk_{t,k}^{TOTAL} = Risk_{t,k}^{w} + Risk_{t,k}^{shw} + Risk_{t,k}^{nc} \leq TR \qquad \forall t, \forall k \tag{2.2}$$

$$u_{\min,i} \leq |u_i| \leq u_{\max,i} \qquad \forall i \tag{2.3}$$

$$h_{\min,l} \leq |h_{i,l}(u_i)| \leq h_{\max,l} \qquad \forall i, \forall l \tag{2.4}$$

Subject to

Where

C_{TOT} = Total cost

C_{REM} = Capital cost and operating expense of remediation wells (See Smalley et al. 2000)

C_{MON} = Cost of monitoring on site (See Smalley et al. 2000)

C_{SYS} = Cost of running ex situ remediation program (Evaluated from data from RACER (1999) as detailed by Gopalakrishnan et al. 2003)

$Risk_{t,k}^{TOTAL}$ = Total lifetime cancer risk (See Smalley et al. 2000)

$Risk_{t,k}^{w}$ = Cancer risk from ingesting drinking water for exposure time t at exposure level k (See Smalley et al. 2000)

$Risk_{t,k}^{shw}$ = Cancer risk from inhalation of volatiles encountered while showering for exposure time t at exposure level k (See Smalley et al. 2000)

$Risk_{t,k}^{nc}$ = Cancer risk from volatiles encountered in non-consumptive uses for exposure time t at exposure level k (See Smalley et al. 2000)

TR = target level of exposure, set to 10^{-5}

$u_{min,i}$ = Minimum pumping rate for well i (m^3/day)

$u_{max,i}$ = Maximum pumping rate for well i (m^3/day)

$h_{i,l}(u_i)$ = Calculated hydraulic head from well i at location l(m)

$h_{min,l}$ = Minimum hydraulic head at location l (m)

$h_{max,l}$ = Maximum hydraulic head at location l (m)

The GA evaluates alternatives based on fitness, which incorporates the cost from Equation 2.1 as well as penalties for violation of constraints given by Equations 2.2 and 2.4. The pumping rate constraint, Equation 2.3, is automatically satisfied through the encoding of the pumping rates in the GA chromosome. A penalty for violating the risk constraint and the head constraint, Equations 2.2 and 2.4 respectively, is found by weighting the violation. The final fitness function is as follows:

$$\text{Fitness} = C_{TOT} + \omega_1{*}\text{Risk Violation} + \omega_2{*}\text{Head Violation} \tag{2.5}$$

Both ω_1 and ω_2 were set to 1000 as determined by Espinoza (2003). This algorithm seeks to minimize the fitness, so a lower fitness is better.

3. UMATILLA FIELD SCALE APPLICATION

The promising GAs are also evaluated on a field-scale case study at the Umatilla Chemical Depot. Section 3.1 describes the field site, while Section 3.2 details the optimization formulation.

3.1 Umatilla Field Site Description

The Umatilla Chemical Depot consists of 19,728 acres in northeastern Oregon that, in 1941, was established as a military reservation with the primary purpose of storage of chemical and conventional munitions. For approximately 15 years, water from a washout process was placed in two unlined lagoons, which were approximately 47 feet above the groundwater table. As the washout water infiltrated the soil, the contaminants, TNT (2,4,6-Trinitrotoluene) and RDX (Hexahydro-1,3,5- trinitro-1,3,5-triazine), were transported into the groundwater. The site was placed on the National Priorities List in 1984.

The US Army Corps of Engineers (USACE 1996 and 2000) developed a pump-and-treat system to remove the contaminants. This existing system consists of four extraction wells (one is inactive), and three infiltration basins that return water to the aquifer after treatment. USACE used a numerical model generated by MODFLOW 2000 (*Harbaugh et al*, 2000) and MT3DMS (*Zheng and Wang*, 1999) to predict fate and transport of the contaminant plumes. The site was divided into five layers, with the top representing the alluvial aquifer of concern; the other layers were added to more accurately predict concentrations in the top layer. The aquifer is divided into 12 homogonous hydraulic conductivity zones. A grid of 132 rows by 125 columns was superimposed on the site. Figure 3.1 depicts the wells, basins, plume, and grid for the Umatilla site.

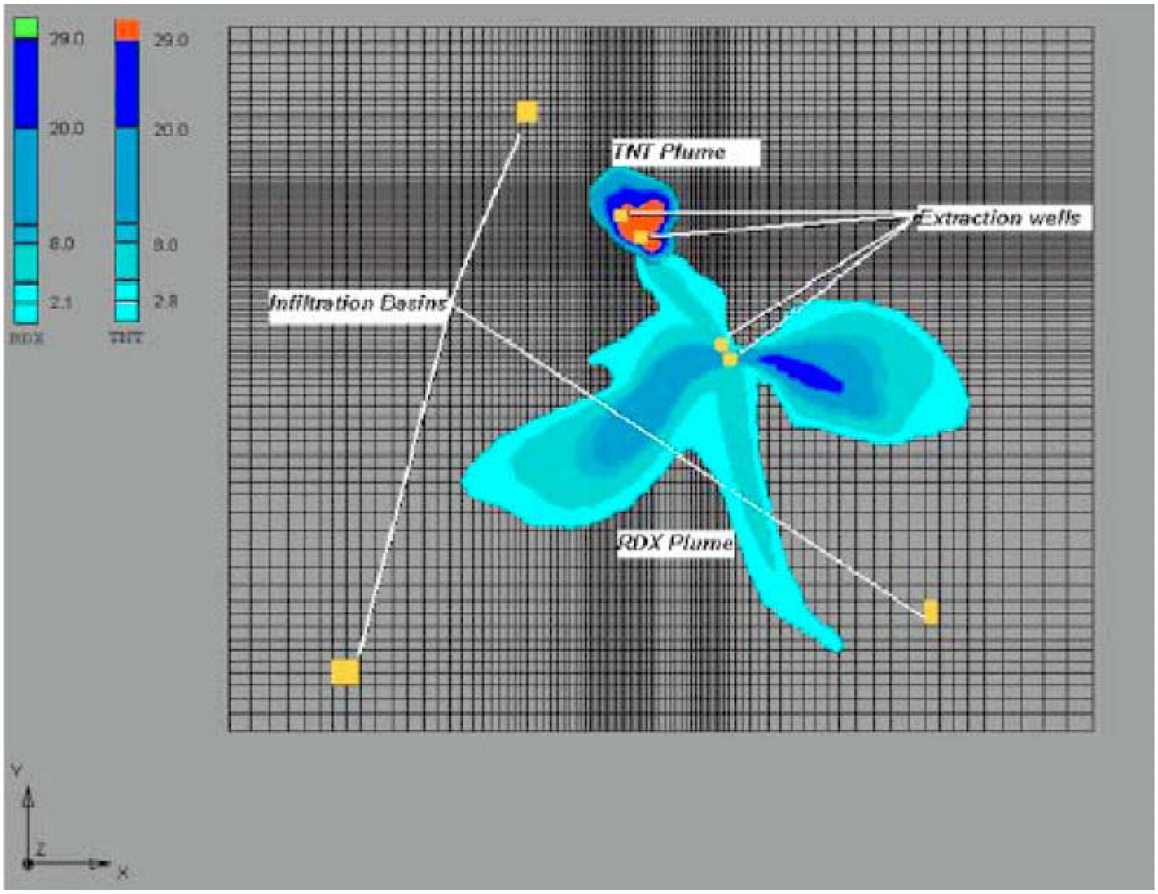

Figure 3.1: RDX and TNT plumes on grid at Umatilla

3.2 Umatilla Field Site Optimization Formulation

A recently completed Environmental Security Technology Certification Program (ESTCP) project selected this site for demonstration of pump-and-treat optimization with transport models (*Minsker et al.*, 2003, *Becker et al., in press*). As with the hypothetical case study, the goal for the optimization (one of three formulations developed for the ESTCP project) is to minimize cost. The formulation is as follows:

$$\text{Minimize } C_{TOT} = CCW + CCB + FCL + FCE + VCE + VCG + VCS \qquad (3.1)$$

Subject to

$$Q_1 \leq 360 \text{ gpm} \qquad (3.2)$$

$$Q_2 \leq 900 \text{ gpm} \qquad (3.3)$$

$Q_1 + Q_2 \leq 1170$ gpm (3.4)

$C_{RDX} \leq 2.1$ μg/l (3.5)

$C_{TNT} \leq 2.8$ μg/l (3.6)

$Q_{inj} = Q_{ext}$ (3.7)

$N_{inj} \leq 3$ (3.8)

$N_{ext} \leq 4$ (3.9)

Where

C_{TOT} = Total Cost

CCW = Capital costs of new wells

CCB = Capital costs of new recharge basins

FCL = Fixed cost of labor

FCE = Fixed costs of electricity

VCE = Variable electrical costs of operating wells

VCG = Variable costs of changing GAC units

VCS = Variable cost of sampling

Q_1 = Pumping rate on hydraulic zone 1

Q_2 = Pumping rate in hydraulic zone 2

C_{RDX} = Concentration of RDX

C_{TNT} = Concentration of TNT

Q_{inj} = Injection Flow

Q_{ext} = Extraction Flow

N_{inj} = Number of new injection basins

N_{ext} = Number new extraction wells

The first two constraints, Equations 3.2 and 3.3, are encoded directly into the chromosome, while the other constraints are handled by penalties in a similar manner to the hypothetical case study. For more on these equations, the details of which are too extensive to reproduce here, see Minsker et al. (2003).

4. METHODOLOGY

Sections 4.1 through 4.3 give overviews of the three genetic algorithms considered in this study. Section 4.4 and 4.5 give specific GA settings for the hypothetical case study and the Umatilla field scale study, respectively.

4.1 SGA Background

The simple genetic algorithm (SGA) is the standard binary GA that has been used in most previous water resource applications (see Goldberg 1989 for a description). The SGA was employed as a baseline for comparison with the other algorithms. In the case of groundwater remediation design, the SGA evaluates candidate solutions that are binary representations of possible configurations of well locations and pumping rates. Each configuration is known as a chromosome, and each bit in the binary chromosome is called a gene. It is important to remember that each chromosome is an alternative, by no means necessarily the best. The collection of all chromosomes is a population. The role of the GA is to iteratively change the population, mimicking the natural selection processes that occur as a population evolves generation after generation. Each generation is adapted over time to develop an optimal solution, just as a population becomes more fit via natural selection.

In the SGA, the initial population is generated randomly and will vary depending on the seed used for the random number generator. The fitness of each individual is first evaluated, which in this study is the objective function value plus a linear penalty for any violation of the constraints (hence, in this case, fitness is to be minimized). Next, parent chromosomes in the population are paired to mate in a process called selection. The specific implementation of the selection method varies, but in all cases, the process seeks to pair chromosomes to mate and create a child. The selection scheme for the SGA used in this study was tournament selection with replacement for the hypothetical case study and without replacement for the Umatilla field study. Tournament selection without replacement allows a given chromosome to mate only once. Once the chromosome mates, it is not allowed to do so again. If replacement is allowed, then the chromosome returns to the mating pool. In the next step, crossover, genetic material from the parents is exchanged, and children are created. Uniform crossover was used in this work. Each bit in the chromosome has a specific probability, 0.5 in this work, of being exchanged from one chromosome to the other. Next, mutation occurs. In an SGA, mutation is generally a low probability event in which the value of a single bit is changed from zero to one, or vice versa. In this study, the probability of mutation was set to 0.001.

Finally, in the replacement process, the members of the next generation are chosen. For most algorithms, chromosomes are placed in the next generation in proportion to their fitness. A chromosome with a better fitness will have more copies in the next generation. In this manner, the population of chromosomes will converge to the one with the best fitness.

Replacement strategies can be either elitist, or non-elitist. With elitism, the best chromosomes of the parent generation are guaranteed to reach the next generation. In non-elitism, there is no such guarantee. While non-elitism may seem counterproductive, it can help to maintain a more diverse population, as the best chromosome will not overtake the population as quickly. This more diverse population will allow the GA to search more of the decision space and not prematurely converge to a local minimum.

Two different replacement schemes were used for the SGA on the hypothetical case study. One is non-elitist, and the other elitist. For the SGA without elitism, the children replace all parents to form the next generation. For the SGA with elitism, the best parent is always included in the next generation. For the Umatilla Field Site, a replacement strategy called $\mu+\lambda$ was used instead, where μ represents the parents and λ the children. In this elitist implementation, the parents and children are combined, and the best of both are chosen for the next generation.

These processes then repeat until the SGA converges to a single solution or reaches the maximum number of generations allowed. SGAs work by combining segments of promising chromosomes until the best combinations are identified that lead to high fitness. Known as building blocks, and explained in detail by Goldberg (1989), these highly fit segments are exchanged in the crossover phase. However, the crossover process may break these building blocks when the chromosome is split randomly. The next step, mutation, may also have this effect. Advanced genetic algorithms are designed to overcome these difficulties, as described in the next sections.

4.2 BOA Background

The BOA operates similar to an SGA at the outset, randomly generating the initial population and then evaluating it using the objective function and constraints. The BOA employed in this work uses tournament selection without replacement as the selection scheme. Then, a Bayesian network (Pearl, 1998) replaces the crossover and mutation steps of a GA.

To illustrate how a Bayesian network works, consider as an example Figure 4.1, which depicts the variables necessary to create a great cup of coffee. These variables are binary – the water is filtered or not, both the brew method and the bean are good or bad. The output, a great cup of coffee, is measured by its selling price. Next imagine that many cups of coffee have been generated, and for each cup, the values of the variables are known, as well as the selling price.

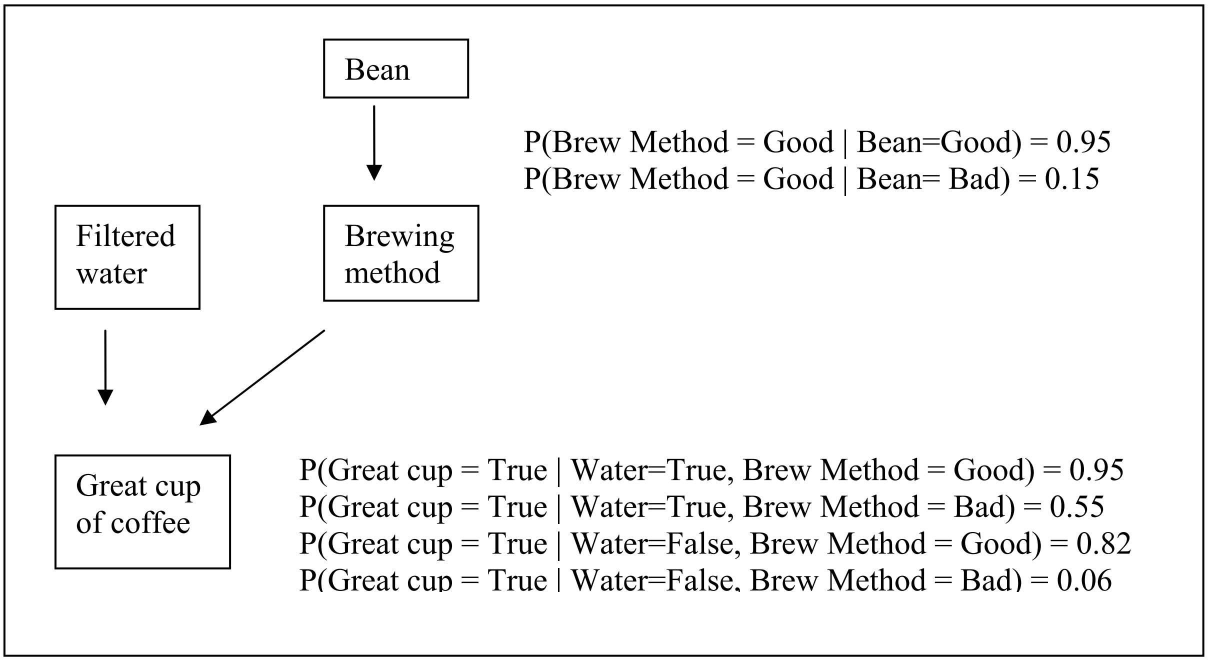
Bean
P(Brew Method = Good | Bean=Good) = 0.95
P(Brew Method = Good | Bean= Bad) = 0.15
Filtered water
Brewing method
Great cup of coffee
P(Great cup = True | Water=True, Brew Method = Good) = 0.95
P(Great cup = True | Water=True, Brew Method = Bad) = 0.55
P(Great cup = True | Water=False, Brew Method = Good) = 0.82
P(Great cup = True | Water=False, Brew Method = Bad) = 0.06

Figure 4.1: Example Bayesian Network

A Bayesian network is a directed acyclic graph that demonstrates the relationship (directional arcs) between the variables (nodes) in a problem. Nodes not connected with an arc are conditionally independent, while the direction of an arc indicates the conditional dependence of the node. In our example, a great cup of coffee is only dependent on the filtered water and the brewing method. The second part of the Bayesian network is the conditional probability table, also depicted in Figure 4.1. The specific values in the table are generated by data, which, in this example, is the specific value of the variable for each cup of coffee. The purpose of the Bayesian network is to determine the relationship between the variables, and the values of those variables that lead to a *promising* solution, a great cup of coffee.

In the BOA, the bits in the chromosome are the variables. The selling price is the chromosome fitness. Each cup of coffee is like a member of the population. The promising members of the population (i.e., those with highest fitness) provide the data used to generate the conditional probabilities. The dependencies among the bits are modeled in the network, with the best groups of bits (i.e., the building blocks) acting as the nodes of the network.

Once the network is created in each generation, promising solutions are then generated using the conditional probabilities from the Bayesian network. These conditional probabilities map which predecessor variables must be specified before a value can be assigned to a successor variable. The variables are ordered in the network so that predecessors are before successors. Then, these variables are assigned values in this order to create the new candidate solutions. In the example above, the algorithm would first order the variables so that all variables are assigned before they are needed. One such order is the bean, the water, and then the brew method. Next, these variables are given values: good, true, and good, for example. In effect, the process is using the building blocks present in promising solutions. The replacement strategy is not elite, as the entire parent population is replaced by the children.

4.3 hBOA Background

The hBOA method is similar to BOA but, as detailed by Pelikan (2002), several refinements were made to the approach. Building blocks still act as nodes on the Bayesian network. However, a hierarchical element is added to the code. Building blocks are grouped to form larger blocks with a known contribution to the total fitness. For example, the coffee bean is actually an aggregation of the variables freshness, quality and roasting time. The single value that represents several variables is computationally easier to process. As the building blocks are consolidated, there are fewer nodes on the network. It is computationally easier to span the paths of the network to build new candidate solutions because there are fewer paths.

hBOA also invokes niching to preserve diversity. Diversity is important because, for complex problems, building blocks with low fitness at a lower level may be needed at a higher level to achieve an optimal level of fitness. To address this issue, the algorithm includes niching, a process whereby only similar chromosomes compete in the selection phase, allowing a range of dissimilar chromosomes to be preserved to maintain diversity. Since dissimilar chromosomes do not compete, the hBOA will not converge to a single optimal solution as the SGA or BOA would.

The hBOA uses tournament selection with replacement as the selection scheme. Additionally, the replacement strategy in hBOA is unique, employing restricted tournament replacement. In this approach, a subset of the parent population is randomly chosen. The child chromosome is compared to the specific member of the subset to which it is most similar. If the child's fitness is better than the parent, then the parent is replaced by the child. This method is not elitist.

4.4 Hypothetical Case GA Parameter Settings

The GA parameter settings for the hypothetical case were based on work developed by Reed (2000) and implemented by Espinoza (2003). Different GA methods may require different minimum population sizes in order to generate reliable results. In order to determine the reliability of each method at different population sizes, three population sizes from the population sizing analysis (see Espinoza 2003 for details) were used: 60, 120, and 240. For all methods at all population sizes, the GA was allowed to run a maximum of 90 generations, but was allowed to stop if 90% of the population has a fitness that is within 0.001% of the best fitness. Since the hBOA does not converge, it was allowed to run for the maximum number of generations that the SGA with elitism ran, 90 generations. To rigorously test the results, 20 runs with different starting populations were completed for each GA method at each population size.

4.5 Umatilla Field GA Parameter Settings

The settings used for the Umatilla field site were also based on work by Espinoza (2003). The population size used was 160. Ten runs were performed using only two of the

GA methods, the SGA with elitism and the hBOA, as these were the two most promising methods from the hypothetical case.

The SGA runs were completed first. These runs were allowed to execute until either 90% of the fitness functions were within 0.1% of the minimum, or the maximum number of generations, 262, had been reached. This figure of 262 is based on Reed's (2000) work, which states that a GA can be expected to converge within the number of generations equal to twice the chromosome length. No SGA runs reached the maximum number of generations; the most generations needed until convergence was 52. Since the hBOA does not converge to a single solution as the SGA does, allowing the hBOA to progress 262 generations would be computationally intensive and futile. To maintain parity in the computational effort of the two algorithms, all runs of hBOA were allowed to execute for 52 generations.

5. RESULTS

In order to evaluate the results of each method, a fair method of comparison must be employed. Since the hBOA will not converge, as the other methods do, generations until convergence can not be used. The best fitness, which in these cases is the minimum fitness, found in a given generation was the method of comparison chosen. Since each generation requires approximately the same computational effort for each method, this approach is not biased against any of the GA methods.

5.1 Test Case Results

The GA methods varied in the actual number of generations that were executed. This is because some of the methods converged, and some executed for the maximum number of generations. The BOA, at all population levels, converged. The SGA without elitism converged at population levels of 60 and 120, but not 240. The SGA with elitism and hBOA did not converge in 90 generations. Table 5.1 lists the maximum final generation for each method at the given population level.

	Population:		
Algorithm	60	120	240
SGA, not elite	55	81	90
BOA	23	30	39
hBOA	90	90	90
SGA, elite	90	90	90

Table 5.1: Maximum Number of Generations Run on Test Case

Figure 5.1 provides a comparison between GA methods at different population sizes. It shows the best final fitness found by any random seed vs. the number of fitness evaluations required to find that solution. At each population size, there is a relatively small difference in fitness between the SGA with elitism and the hBOA for the same number of fitness executions. With the exception of the BOA, the best fitness value found by each method improves with an increase in population size. Since the BOA always converged, the total number of fitness evaluations for a given population size is always less than that for the other methods. However, the final fitness value is always worse than that for the other methods. This method, with the particular settings used, does not reliably produce an optimal result.

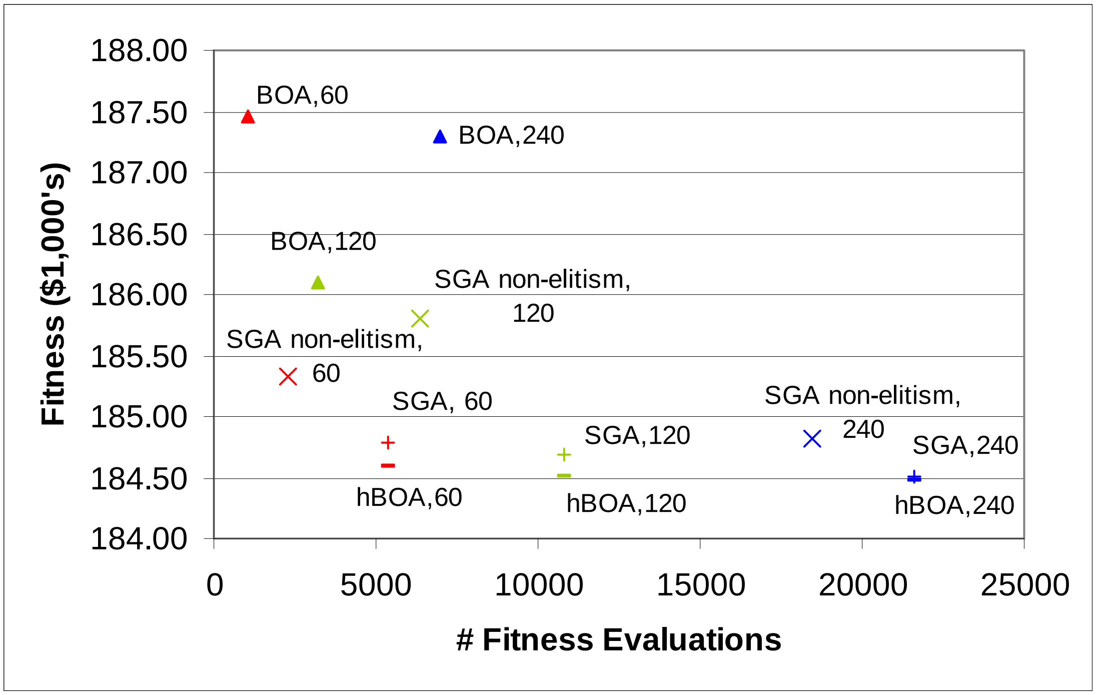

Figure 5.1: Best Fitness Found with Different Numbers of Fitness Evaluations, For Each Method and Population Size

The range in fitness of the final generation across random number seeds can be used as a measure of reliability of the algorithm. By this measure, the hBOA method is shown to be more reliable over all population sizes (see Figure 5.2), and particularly more reliable at the smallest population size. The bars represent the average final fitness (from all 20 random seeds) found by each GA method at a given population size. The error bars on each represent the value of the minimum fitness from the best and worst performing random seeds. The hBOA consistently has a smaller range, making it more reliable. The average final fitness found by the hBOA and the SGA with elitism are relatively close at the two higher population sizes. However, the reliability of the hBOA at the population size of 60 is significantly greater than the SGA with elitism. The range of the SGA with elitism is approximately 6.5 times greater than that for the hBOA. It is also apparent from Figure 5.2 that neither the BOA nor the SGA without elitism approach the optimal fitness.

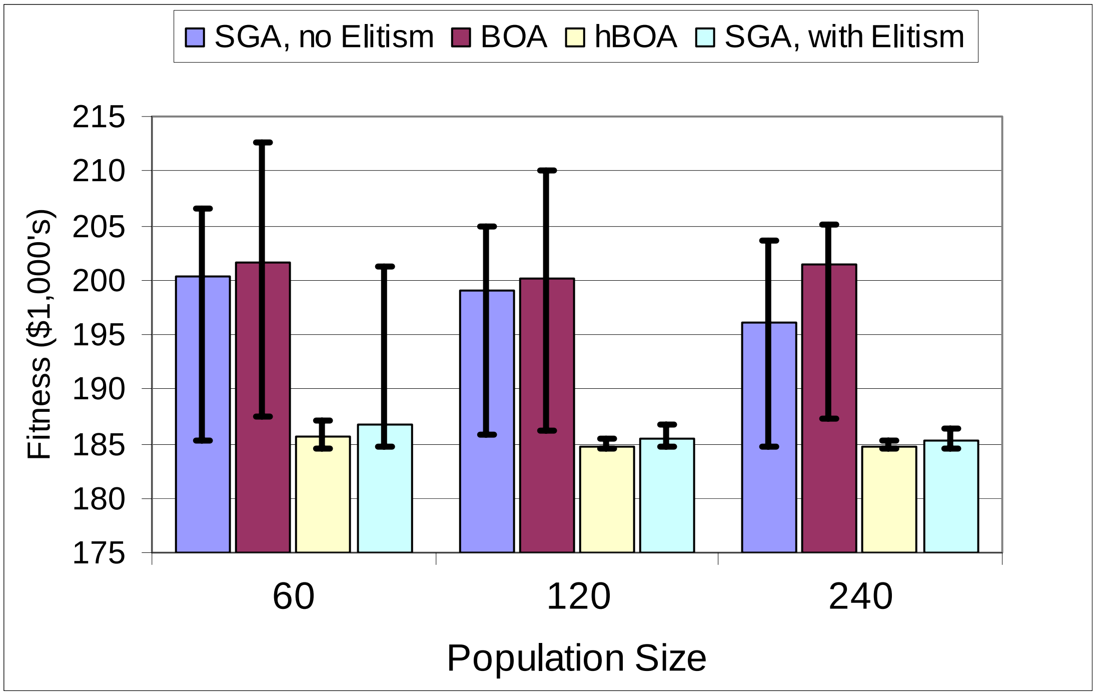

Figure 5.2: Best, Worst, and Average Fitness over All Seeds for Each Population Size and GA Method

Figures 5.3-5.5 show the average population fitness in each generation for three population sizes using one particular seed. The average fitness of the hBOA has significantly more fluctuations in fitness than the other methods because the hBOA uses niching to preserve diversity of the population. Diversity is necessary because, for so-called "deceptive" problems, good building blocks may not perform well when combined with other building blocks. The improved reliability of the hBOA algorithm indicates that this case study is, in fact, deceptive and the hBOA capabilities are beneficial.

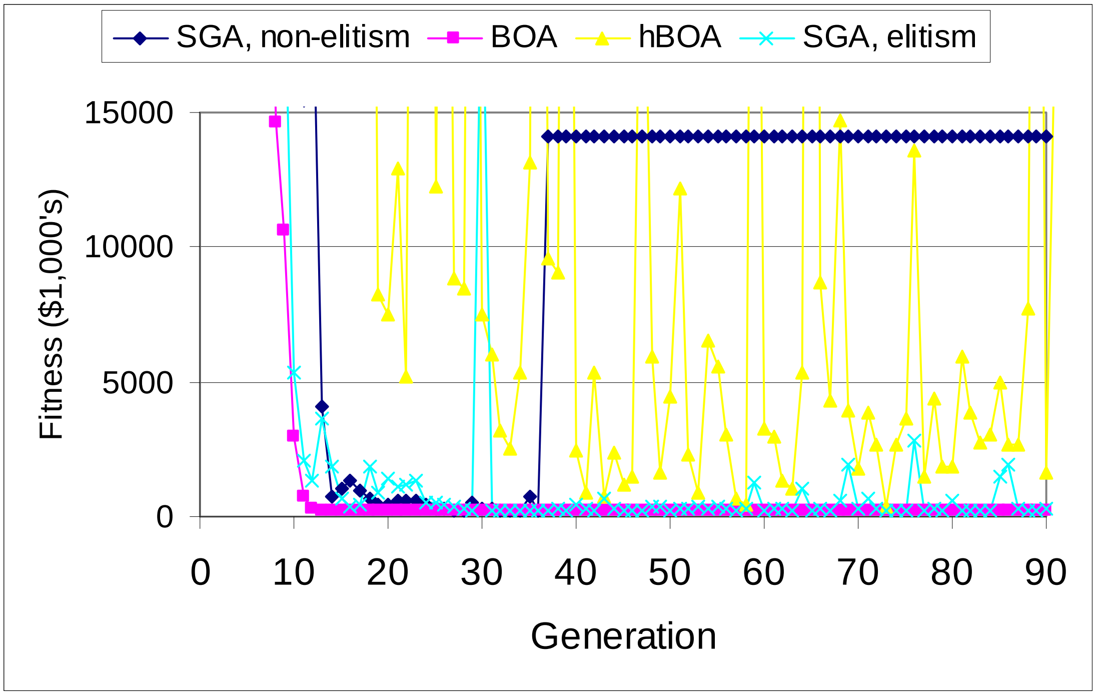

Figure 5.3: Average Fitness of Each Algorithm, for One Seed with Population Size 60

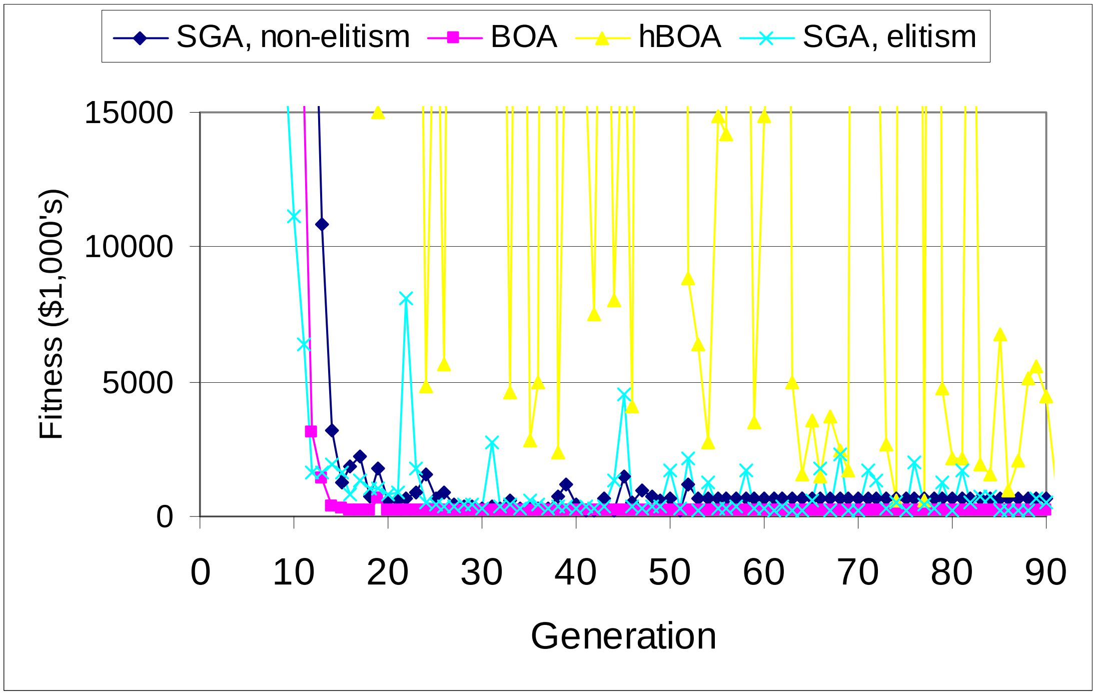

Figure 5.4: Average Fitness of Each Algorithm, for One Seed with Population Size 120

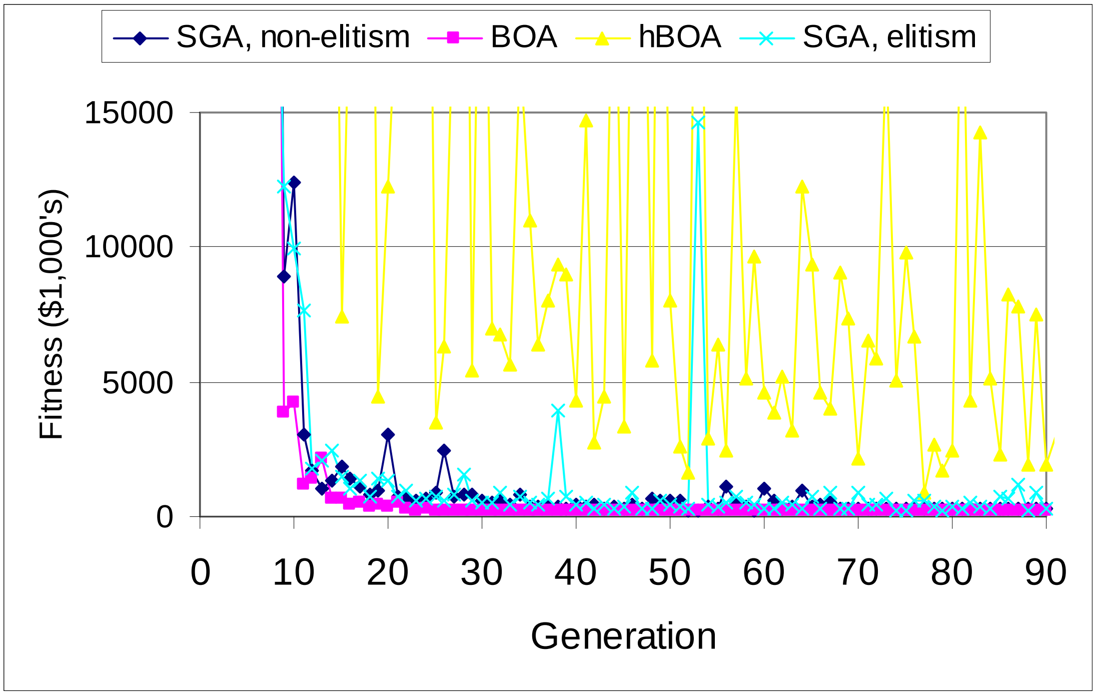

Figure 5.5: Average Fitness of Each Algorithm for One Seed with Population Size 240

Figure 5.2 depicted the range between the best fitness found in only the final generation for the best and worst seed. Also of interest is an examination, generation by generation, of the range in best fitness values between best and worst seeds. It is desirable to employ a GA method that narrows this range as rapidly as possible, which will give more reliable solutions, while still achieving acceptable accuracy. Figures 5.6-5.8 provide a visual display of the range that can be expected from a given method, generation by generation, at different population levels. The figures show the average of the best fitness over all seeds, along with the best seed and the worst seed.

At a population size of 60, Figure 5.6 shows that the SGA without elitism and BOA have the greatest amount of performance variation in each generation, followed by the SGA with Elitism. Only the hBOA demonstrates a narrow range.

At a population size of 120, Figure 5.7 shows that the SGA without elitism and the BOA continue to be unreliable. At this population, the SGA with elitism has a more narrow range in final fitness values, $2,100, in comparison with the range at a population of 60, $16,500. The hBOA continues to have a narrow range of fitness values, $890.

The SGA without elitism still does not show any improvement in reliability at a population size of 240 (Figure 5.8), but the BOA increased in reliability at this population level. The range at a population size of 120 was $23,900, and at a population of 240, the range is $17,850. However, the average fitness of the BOA in the final generation is $201,000, roughly 9% worse than the $184,000 at the optimum. Both the hBOA and the SGA with elitism continue to be reliable at later generations.

Figure 5.6: Average Fitness for All Seeds, with Best and Worst Seeds for Population=60

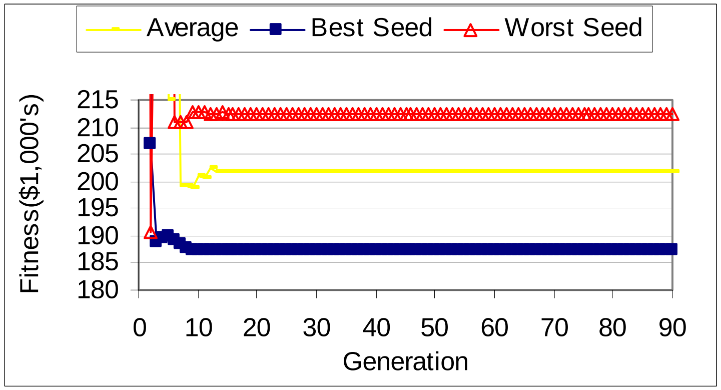

Figure 5.6a: Average Fitness for All Seeds, with Best and Worst Seeds for SGA Without Elitism, Population=60

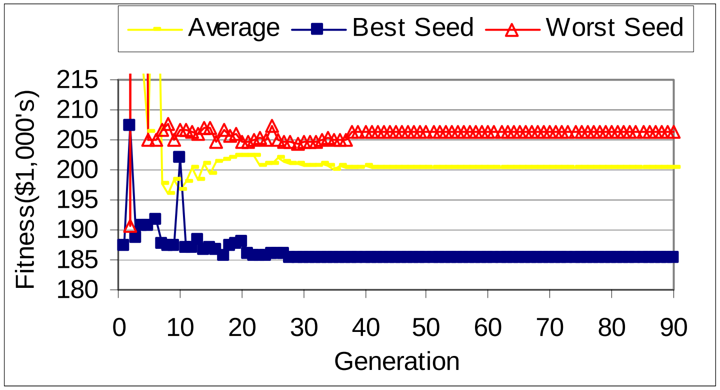

Figure 5.6b: Average Fitness for All Seeds, with Best and Worst Seeds for BOA, Population=60

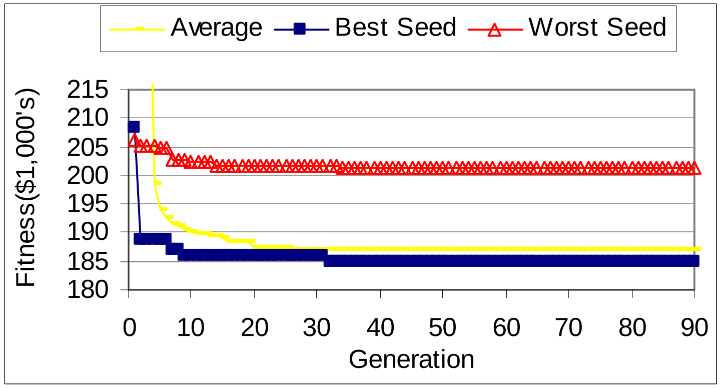

Figure 5.6c: Average Fitness for All Seeds, with Best and Worst Seeds for SGA with Elitism, Population=60

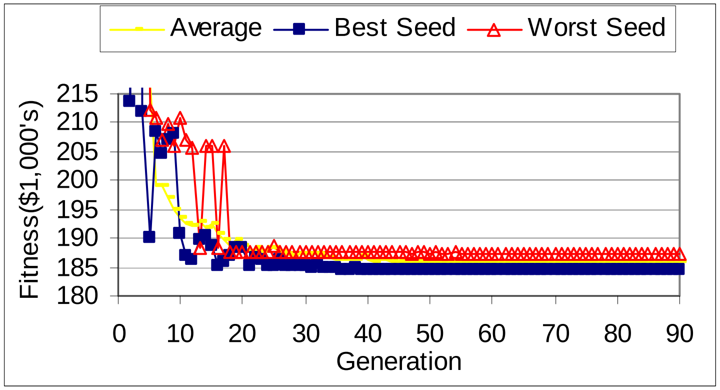

Figure 5.6d: Average Fitness for All Seeds, with Best and Worst Seeds for hBOA, Population=60

Figure 5.7: Average Fitness for All Seeds, with Best and Worst Seeds for Population=120

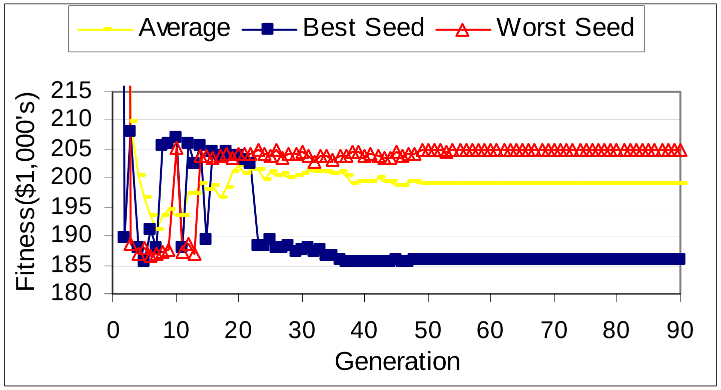

Figure 5.7a: Average Fitness for All Seeds, with Best and Worst Seeds for SGA Without Elitism, Population=120

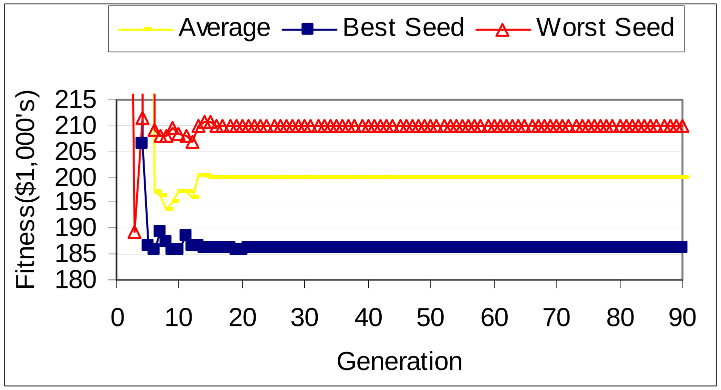

Figure 5.7b: Average Fitness for All Seeds, with Best and Worst Seeds for BOA, Population=120

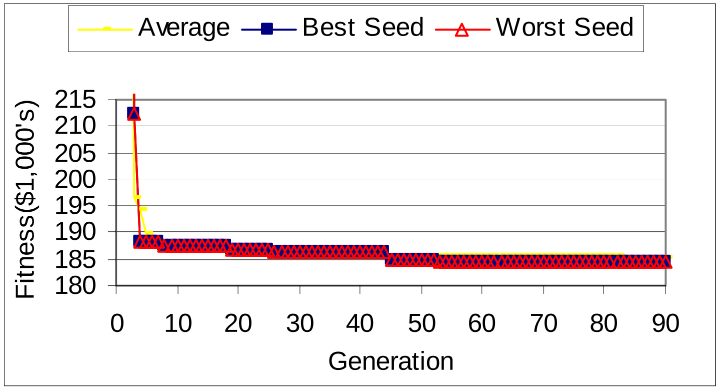

Figure 5.7c: Average Fitness for All Seeds, with Best and Worst Seeds for SGA with Elitism, Population=120

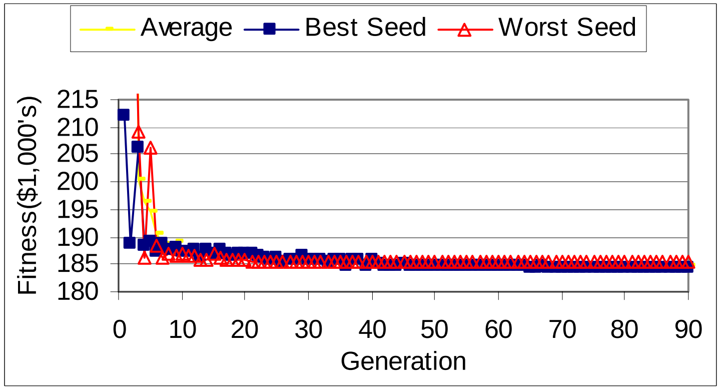

Figure 5.7d: Average Fitness for All Seeds, with Best and Worst Seeds for hBOA, Population=120

Figure 5.8: Average Fitness for All Seeds, with Best and Worst Seeds for Population=240

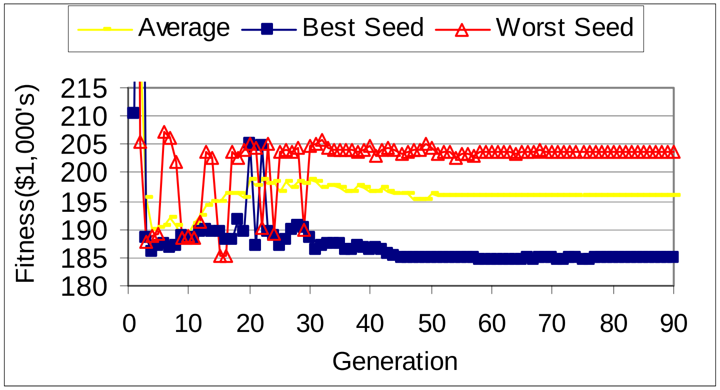

Figure 5..8a: Average Fitness for All Seeds, with Best and Worst Seeds for SGA Without Elitism, Population=240

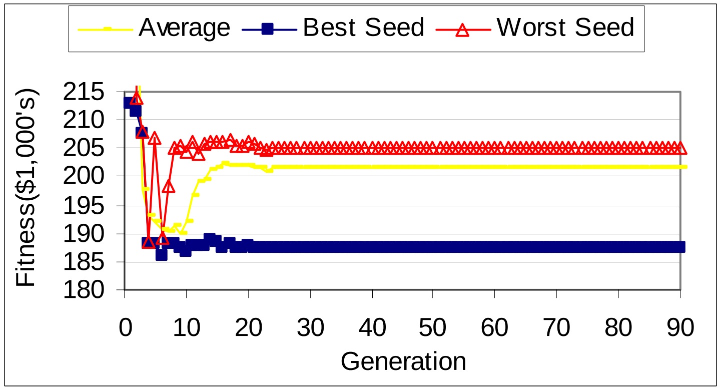

Figure 5.8b: Average Fitness for All Seeds, with Best and Worst Seeds for BOA, Population=240

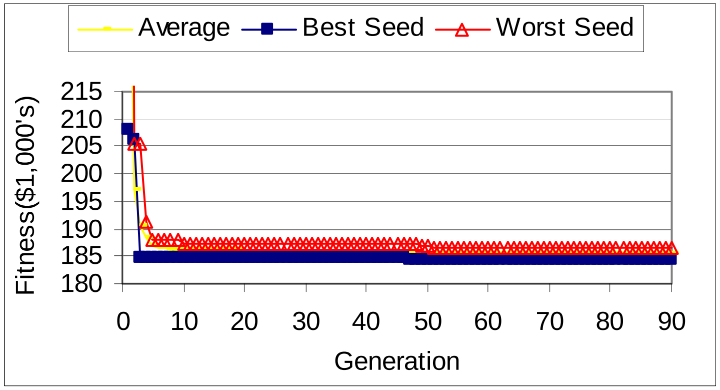

Figure 5.8c: Average Fitness for All Seeds, with Best and Worst Seeds for SGA with Elitism, Population=240

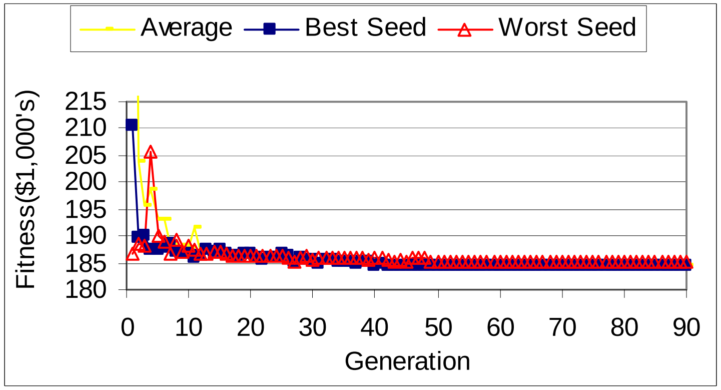

Figure 5.8d: Average Fitness for All Seeds, with Best and Worst Seeds for hBOA, Population=240

5.2 Umatilla Case Study Results

Figure 5.9 shows the performance of each method for the Umatilla Field Case Study. Here, the bars represent the average final fitness (from all 10 seeds) found by the GA method and the error bars represent the value of the minimum fitness from the best and worst seeds. The SGA with elitism has a slightly smaller range, but both algorithms achieve acceptable results for all seeds. The SGA with elitism has a fitness variation of 0.02% across all runs, while hBOA has a variation of 0.03%

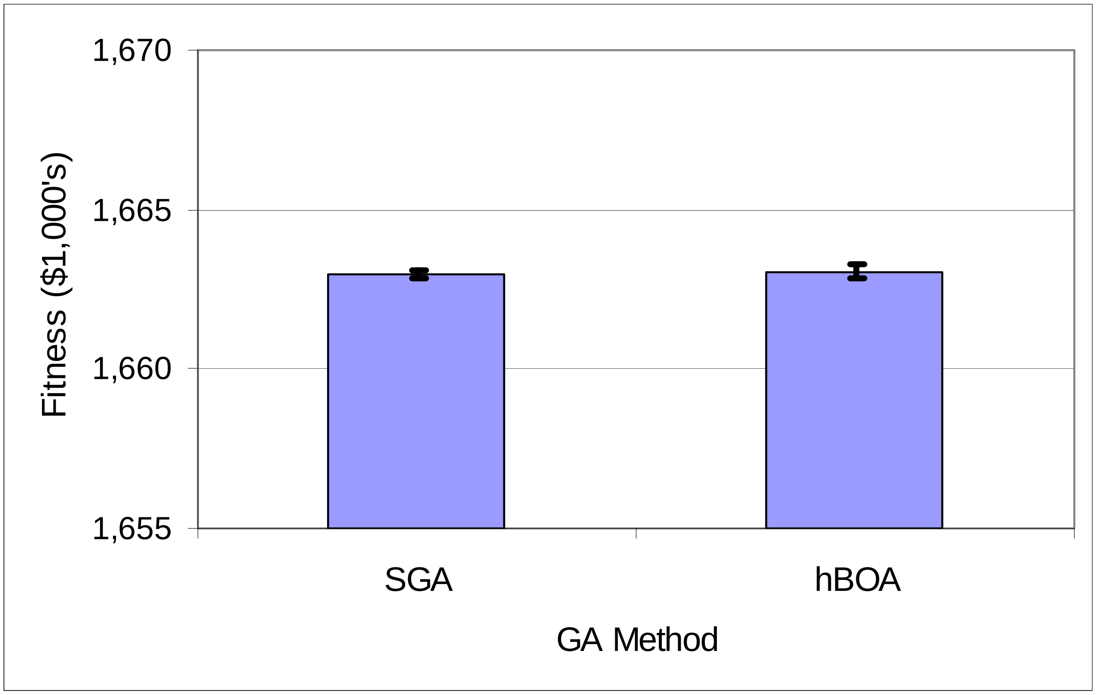

Figure 5.9: Best, Worst, and Average Fitness over all Seeds by GA Method for the Umatilla Case

Figure 5.10 shows the average population fitness in each generation for one particular seed. As with the previous case study, the average fitness of the hBOA has significantly more fluctuations in fitness than the other methods because the hBOA uses niching to preserve diversity of the population.

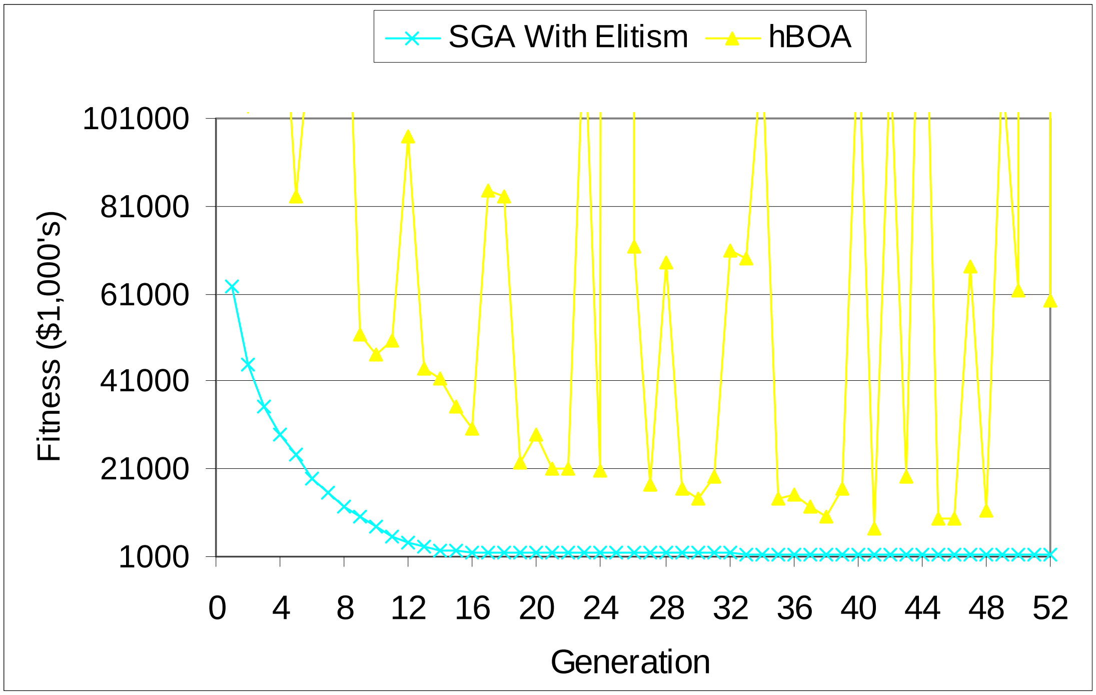

Figure 5.10: Average Fitness by Generation of Each Algorithm for One Seed

Figure 5.11 shows the average of the best fitness over all seeds, along with the best seed and the worst seed for the SGA with elitism. Figure 5.12 depicts the same information for the hBOA. Comparing these figures, it is apparent that the SGA attains a near-optimal solution much earlier than the hBOA. The SGA with elitism ensures that the best fitness found is passed to the next generation, so the SGA is able to find and retain a solution at such an early generation. The hBOA, however, is designed to preserve diversity through niching, with no elitism. Further research is needed to investigate whether the hBOA would also find the optimal solution more quickly and reliably if an elitist strategy were introduced into the algorithm, maintaining at least the best solution found in each generation.

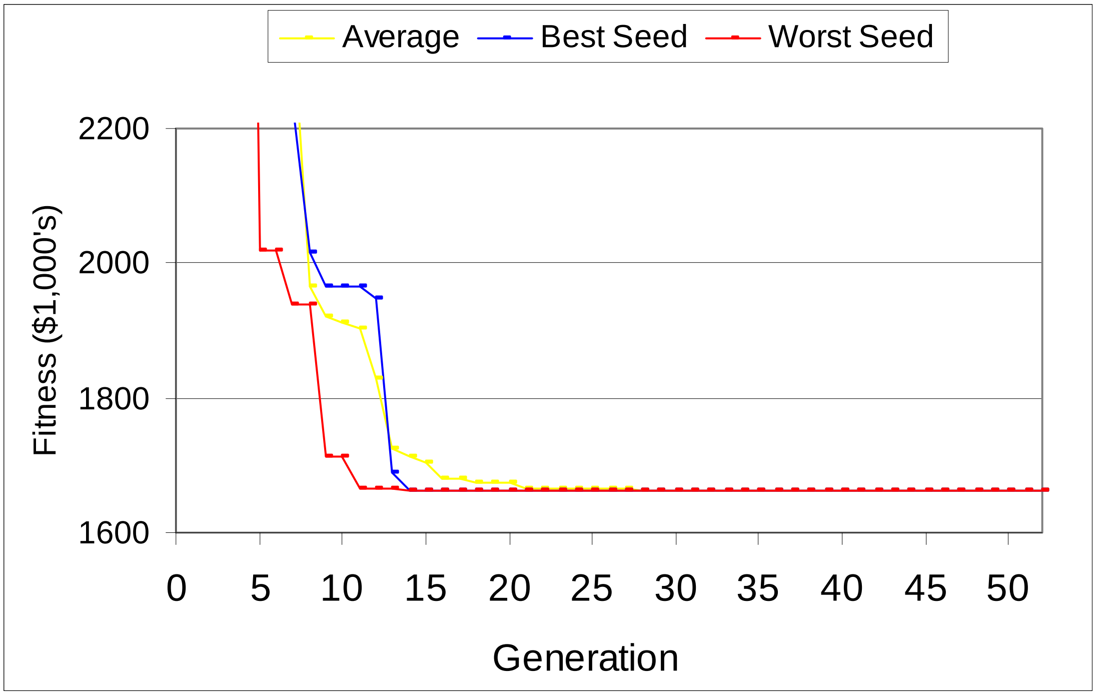

Figure 5.11: Average Fitness for all Seeds, with Best and Worst Seeds by Generation, SGA with Elitism

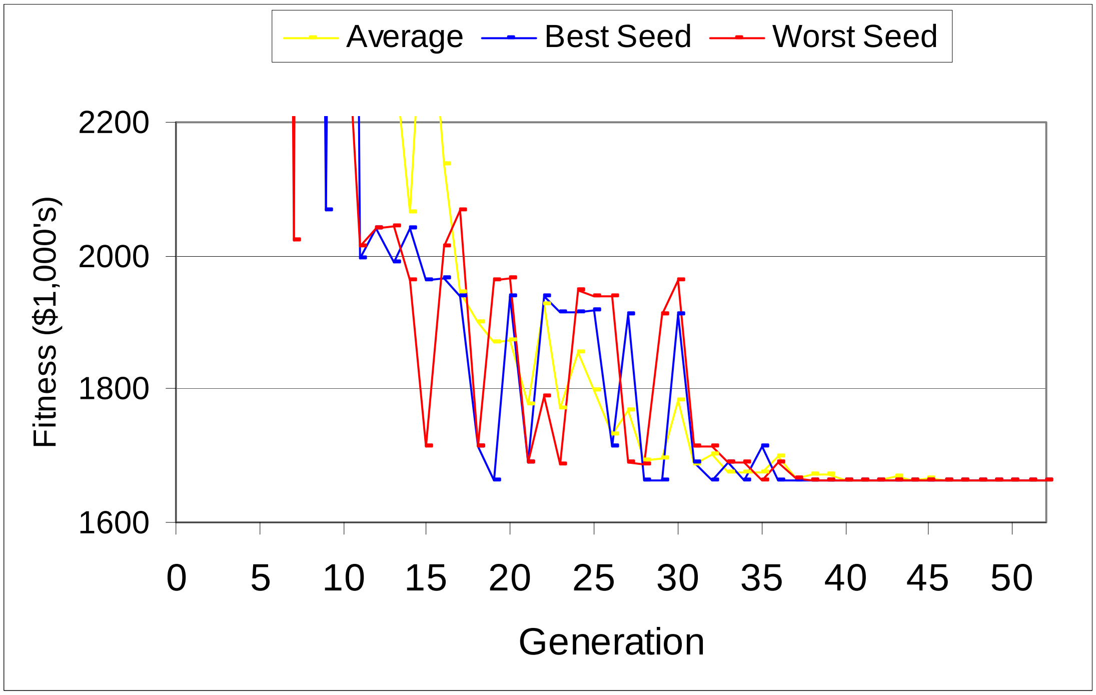

Figure 5.12: Fitness by Generation, hBOA

6. CONCLUSIONS

This work demonstrates that the hBOA more reliably identifies the optimal solution for the hypothetical groundwater remediation design case study. The BOA and the SGA without elitism were clearly ineffective in finding an optimal solution for the hypothetical case study with the settings used. The SGA with elitism generates reliable results for the hypothetical case study at higher population sizes, but is not able to perform as well as the hBOA at small population sizes. For the field-scale case study, both the SGA with elitism and the hBOA were able to find the optimal solution with acceptable reliability. These findings indicate that the performance of hBOA, which is designed to handle deceptive problems, may vary for different water resource problems. The dramatic difference between the SGA with elitism and the SGA without elitism on the hypothetical case suggests that introducing an elitist strategy into BOA and hBOA may further improve reliability, especially for the field-scale case.

REFERENCES

Aly, A.H., and R. C. Peralta (1999). Comparison of a genetic algorithm and mathematical programming to the design of groundwater cleanup systems, *Water Resources Research,* 35(8), 2415-2426.

Arst, R.E. (2002). "*Which are better, probabilistic Model-Building Genetic Algorithms (PMBGAS) or Simple Genetic Algorithms (SGAs)? A Comparison for an Optimal Groundwater Remediation Design Problem.*" Master's Thesis. Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign.

Bayer, P. and M. Finkel (2004). Evolutionary Algorithms for the optimization for advective control of contaminated aquifer zones, *Water Resour. Res., 40*, W06506, doi:10:1029/2003WR002675.

Becker, D., Minsker, B., Greenwald, R., Zhang, Y., Harre, K., Yager, K., Zheng, C., and Peralta, R., "Reducing Long-Term Remedial Costs by Transport Modeling Optimization," *Ground Water*, in press.

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, 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., 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.

Espinoza, F. (2003). "*A Self Adaptive Hybrid Genetic Algorithm for Optimal Groundwater Remediation.*" PhD Dissertation. Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign.

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.

Guan, J., and M. M. Aral (2004). "Optimal design of groundwater remediation systems using fuzzy set theory." *Water Resour. Res. 40,* W01518, doi: 10.1029/2003WR002121.

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

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.

McKinney, D.C. and Lin, M.D. (1994) Genetic Algorithms Solution of Groundwater Management Models, *Water Resource Research*, 31(3), 731-740.

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.

Papadopoulou, M. P., G.F. Pinder, and G. P. Karatzas (2003), Enhancement of the outer approximation method for the solution fo concentration-constrained optimal-design groundwater-remediation problems, *Water Resour. Res., 39*(7), 1185, doi:10.1029/2002WR001541.

Pearl, J. (1988). Probabilistic reasoning in intelligent systems: Networks of plausible inference.San Mateo, CA: Morgan Kaufmann.

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., 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.

Pelikan, M. (2002). "*Bayesian Optimization Algorithm: From Single level to Hierarchy*." PhD Dissertation. Department of Computer Science, University of Illinois at Urbana-Champaign.

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.

U.S. Environmental Protection Agency (USEPA) (2003) Improving Nationwide Effectiveness of Pump-And-Treat Remedies Requires Sustained and Focused Action to Realize Benefits. Available at http://www.epa.gov/superfund/action/postconstruction/oigptreport.pdf

Wang, T. A., W. F. McTernan (2002). “The development and application of a multilevel decision analysis model for the remediation of contaminated groundwater under uncertainty.” *Journal of Envionmental Management.* 64(3), 221-235.

Yan, S. and Minsker, B. S. (2003). “A Dynamic Meta-Model Approach to Genetic Algorithm Solution of a Risk-Based Groundwater Remediation Design Model,” American Society of Civil Engineers (ASCE) Environmental & Water Resources Institute (EWRI) World Water & Environmental Resources Congress 2003 & Related Symposia, Philadelphia, PA, 2003.

Zheng, C., and Wang, P. P. (1999). MT3DMS: A modular three-dimensional multispecies transport model for simulation of advection, dispersion and chemical reactions of contaminants in groundwater systems; documentation and user’s guide, *Contract Report SERDP-99-1*, U.S. Army Engineer Research and Development Center, Vicksburg, MS

Zhi Y., Guo H. Huang, and C. W. Chan (2003). “A fuzzy process controller for in situ groundwater bioremediation.” *Engineering Applications of Artificial Intelligence* 40(2), 131-147.