UNIVERSITY OF CALIFORNIA
Los Angeles

# BMP DECISION USING GENETIC ALGORITHMS FOR COST-EFFECTIVE POLLUTION CONTROL AT THE WATERSHED-LEVEL

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
in Civil Engineering

by

Min-mo Chung

2010

The dissertation of Min-mo Chung is approved.

Richard F. Ambrose

Keith D. Stolzenbach

William W-G Yeh

Michael K. Stenstrom, Committee Chair

University of California, Los Angeles

2010

**TABLE OF CONTENTS**

## LIST OF FIGURES

## LIST OF TABLES

## VITA

| | |
|---|---|
| April 09, 1971 | Born, Seoul, Republic of Korea |
| 1997 | B. A., Environmental Engineering<br>Kyung-hee University<br>Seoul, Korea |
| 1997- 1999 | M. S., Environmental Engineering<br>Kyung-hee University<br>Seoul, Korea |
| 1999- 2002 | Design Engineer, Kun-hwa engineering.<br>Seoul, Korea |
| 2004- 2005 | Research Assistant<br>University of California, Los Angeles |
| 2003- 2005 | M. S., Civil and Environmental Engineering<br>University of California, Los Angeles |

## ABSTRACT OF THE DISSERTATION


BMP DECISION USING GENETIC ALGORITHMS

FOR COST-EFFECTIVE POLLUTION CONTROL

AT THE WATERSHED-LEVEL

by



Min-mo Chung

Doctor of Philosophy in Civil Engineering

University of California, Los Angeles, 2010

Professor Michael K. Stenstrom, Chair



The main goal of this research was to demonstrate the use of an advanced optimization technique that is suitable for watershed-level best management practice (BMP) optimization. This kind of simulation requires finding the optimal solution from many numbers of feasible alternatives. In this study genetic algorithms (GAs) were selected for optimization in part because they are known to search the solution space globally. Most previous work in developing an optimization tool for this problem has used GAs for optimization by individually considering two objectives: minimizing cost and pollutant reduction. The disadvantage of this approach is that some good solutions might be lost

because the two objectives are considered separately.

In this study a BMP placement tool was developed that searched the best solutions from two objective functions simultaneously. For each GA population, the BMP placement tool calculated pollution reduction by combinations of five BMPs for the seventeen watersheds. At the same time, BMPs total cost (Construction and Operation, Maintenance and Repair cost) was computed. Final results were selected from the best combination of both objectives.

The input data including watershed area and pollutant loading for the optimization tool were adapted from the City of Los Angeles' Proposition O bond results. First generation of GA population was set with 100 chromosomes. Every chromosome was initialized by properties of seventeen watersheds. Each watershed DNA randomly contained properties of BMP type including pollutant removal rates, and total cost functions among five available set of BMPs.

A sensitivity analysis of GAs parameters was performed by comparing fitness values to determine better parameters for the best solution result. The tested GA operators were the population size, the number of generations, the crossover rates, the mutation rates and the overlapping rates. In this study, population size of 100, crossover rate of 60%, mutation rate of 5%, overlapping rate of 60% and a number of generations of 300 gave the best results in terms of fitness values.

Overall, the BMP placement optimization model performed well in reducing the each pollutant load and minimizing BMP total cost from the watershed.

# Chapter 1: Research Problem

## 1.1 Introduction

Non-point source (NPS) pollution from watersheds is a significant contributor to receiving water quality degradation. In the last few decades there has been increasing concern over water and storm water run-off pollutants that influence human or aquatic health.

Government regulations, such as the Clean Water Act and Phase II storm water regulations, are placing growing emphasis on NPS pollution control. One method of control is through implementation of best management practices (BMPs); BMPs are structural or non-structural methods by which NPS pollution is eliminated or reduced sufficiently to meet water quality criteria. Various approaches are being taken to mitigate storm-water impacts and these include individual actions, such as the industrial storm-water management permits as well as agency or city-wide actions, such as Los Angeles' recent passage of Proposition O, which provided $500 million for storm-water management. Proposition O was intended to fund projects to help achieve water quality requirements under the Federal Clean Water Act.

The problem of locating BMPs throughout a watershed for cost-effective pollution control can be stated as a combinatorial optimization problem (Lawler, 1976; Grötschel, 1982). A combinatorial optimization problem optimizes a set of categorical variables (a watershed-level BMP decision) based on an objective function that assigns an ordered value (cost-effectiveness of pollutant reduction) to that set. Two methods can be used to determine pollution reduction for each scenario: monitoring or modeling. However, modeling is frequently more beneficial in assessing long-term, watershed-level BMP effects.

Although mathematical models may be less accurate than field studies in quantifying pollutant levels, a model is useful in evaluating long-term relative changes in pollution levels due to implemented BMPs or other management practice changes. A computer model can complete a long-term simulation in a matter of minutes or seconds, allowing long-term stability and usefulness of BMPs to be analyzed. Also, computer models allow more control over parameters, enabling the researcher to vary model components as necessary. This manipulation aids discovery and understanding of parameter relationships.

The number of ways to allocate BMPs throughout a watershed exponentially increases with regard to the number of watersheds. For example, for 20 watersheds and 4 non-mutually exclusive (meaning that they can occur at the same time) BMPs could have $(2^4)^{20} = 1.2\text{x}10^{24}$ possible decision scenarios. Evaluation of all possible BMP scenarios becomes an intractable problem, one that is computationally difficult or impossible to solve for an exact solution in a finite time. The BMP placement problem has only been addressed through a limited number of procedures. An ideal procedure should choose cost-effective BMP scenarios based on each BMPs location-specific contribution to pollution reduction, as opposed to rule-based targeting criteria, such as a ranking system.

However, due to increases of computer CPU speed and decreases in computational costs, mathematical programming heuristics for solving intractable problems are becoming more widely used. Heuristics for large number problems have been tested in a variety of disciplines, including farm planning, molecular physics and chemistry, production and personnel scheduling, and factory design (Buick et al., 1992; Eglese, 1990; Swisher et al., 2000). Given search space and established criteria and relationships, these heuristics identify an optimal or near optimal solution set of one or more scenarios. Additionally, linear and nonlinear mathematical programming for environmental policy and resource management has been used since the 1960s. Greenberg (1995) provided an extensive survey of the use of optimization for controlling land, air, and water quality. Cooper et al. (1996) extended this review with regard to both deterministic and stochastic modeling of air pollution.

The success of locating the optimal BMP plan for a specific watershed depends on the ability to consider the complete range of possible scenarios within a watershed as well as the various BMP interactions throughout the watershed. Individual applications of targeted BMPs or ranking (manually selected BMPs) do not consider all possible watershed scenarios and may not provide the most cost-effective solution.

Theoretically, a comprehensive approach should find the optimal or a set of near-optimal solutions from among all possible scenarios. The relative benefits of the targeted plan should be evaluated and compared to those solutions. Computer technology enables the use of optimization techniques to evaluate a large number of scenarios. Srivastava (1999) has provided one of the few examples of an optimization heuristic that is beneficial in reducing pollution and BMP construction costs as compared to multiple random scenarios.

## 1.2 Objectives

The overall goal of this study is to increase the cost-effectiveness of BMPs within a watershed. Two specific objectives were necessary to realize this research goal:

1. To optimize BMP placement within watersheds based on cost and NPS pollution (In this study: Total Suspended Solids, Total Nitrogen and Total Phosphorous) reduction for the watersheds; and
2. To determine the cost effectiveness of BMPs using the optimization procedure as compared to a targeting strategy.

The research hypothesis was that each watershed has characteristics of area (acre) and annual load of each pollutant (kg). For this research a BMP was defined as a management practice or set of practices that result in reduced pollutant loading by removal of a fraction of the pollutants at the watershed outlet. In the cost-benefit function, the operational costs of each BMPs method were considered in addition to BMPs construction costs.

# Chapter 2: Literature Review

## 2.1 Introduction

A number of researchers have addressed the problem of BMP cost-effectiveness using NPS and/or economic models in conjunction with algorithms, heuristics, or decision criteria. Near-optimal placement of BMPs within a watershed can potentially be determined through heuristics that solve combinatorial optimization problems. Six such heuristics are described and compared in this chapter.

## 2.2 Optimization

Whereas watershed simulation models are numerous, optimization models are mostly limited to locating and sizing storage and detention facilities to meet water quantity or sediment removal objectives at least cost. Yeh and Labadie (1997) applied a successive reaching dynamic programming (SRDP) algorithm and a multiobjective genetic algorithm (MOGA) to watershed-level planning of storm water detention systems. The SRDP was used to locate and size the detention systems based on a single objective of water quantity. MOGA, a multiobjective evaluation, was used to develop trade-offs between system cost and detention effectiveness on water quality.

Predeep et al. (1999) used dynamic programming (DP) to identify least cost pond designs for both single catchment and multiple catchment systems. The DP was based on different levels of control at individual catchments while satisfying the specified levels of pollution and runoff control at the outfall. The DP was based on the integration of water quality and quantity through the use of isoquants (a contour line drawn through the set of points at which the same quantity of output is produced while changing the quantities of two or more inputs). The isoquants were developed for pollution control performance and runoff control based on two decision variables: the release rate from the pond and the active storage volume of the pond. These isoquants were then combined to identify the optimal release rate and used for the optimization with the objective of minimizing cost based on pond depth. Dorn et al (1995) used a genetic algorithm based optimization to develop a trade off curve between cost and sediment removal of detention pond systems. The trade off curve represented the level of sediment removal or maximum allowable cost specified by the decision maker.

## 2.3 Optimization for intractable problems

The intractable problems that can be solved but not fast enough for the solution to be useful are called *intractable*. Existing optimization heuristics for solving intractable problems include gradient and non-gradient search methods, which were developed from studies of natural systems. Many problems are addressed by creating a customized technique that incorporates multiple variations on basic heuristics. Such customization often improves solution efficiency and is suitable for a specific problem. Six basic heuristics were chosen and evaluated. The objective was to determine a basic heuristic well suited to the BMP placement problem. This section briefly describes the heuristics that were considered, with a more detailed discussion of the genetic algorithm (GA), which was selected after comparing the six heuristics.

### 2.3.1 Response surface methodology

As a line search heuristic, the response surface methodology (RSM) (Ibrahim and Liong, 1992; Jacobson and Schruben, 1989; Myers, 1971) generally uses regression to fit a first or second order polynomial to a part of the feasible region. The improving direction is then determined from the gradient of the polynomial. A line search is made along the improving direction until the polynomial no longer provides sufficient fit. The procedure is continued from each new point until the gradient of the fit polynomial is essentially zero. Response surface methodology uses statistical models, and therefore practitioners need to be aware that even the best statistical model is an approximation to reality.

### 2.3.2 Shuffled complex evolution

Duan et al. (1993) developed the shuffled complex evolution (SCE) method to address major characteristics of hydrologic model calibration problems. The SCE globally searches for the optimum by combining basic GA evolutionary concepts with a population grouping strategy. At each generation the SCE divides the search space into subsections, or complexes. Nelder and Mead's version of the simplex method (Bazaraa et al., 1990) is used within each subsection to generate offspring that drive the optimum in an improving direction. The SCE method then recombines the subsections by pooling all offspring into a single population, ranks the results, and starts over. Members of the population that rank higher than others in terms of fitness values have a larger probability of contributing to the next generation than do those members with lower fitness values. The SCE method continues in this manner until new searches do not improve on the optimum from the previous step.

The SCE method combines benefits of the GA and neighborhood search algorithms. As a result, the SCE method searches both globally and locally. When used with continuous data for calibration problems, it was shown to be more efficient than use of a basic GA (Cooper et al., 1997).

### 2.3.3 Simulated annealing

The simulated annealing heuristic (SA) (Eglese, 1990; Swisher et al., 2000) generalizes the annealing process used for crystalline solids. In this process the solid is heated to a high temperature and then cooled very slowly in an attempt to reach the lowest energy state possible. While at a high temperature, the crystalline structure of the solid is unstable and the solid is malleable. However, as the solid cools down, the crystalline structure becomes fixed. By cooling the solid very slowly, the annealing process attempts to reach the lowest energy state possible, thus, achieve the most structurally sound crystalline formation for the solid.

The current state of the thermodynamic systems is analogous to the current solution to the combinatorial problem. The energy equation for the thermodynamic system is analogous to the objective function, and ground state is analogous to the global minimum. The major difficulty in implementing the algorithm is that there is no obvious analogy for the temperature T with respect to a free parameter in the combinatorial problem. Furthermore, avoidance of entrainment in local minima (quenching) is dependent on the "annealing schedule", the choice of initial temperature, how many iterations are performed at each temperature, and how much the temperature is decremented at each step as cooling proceeds.

In simulated annealing, the process begins at a high 'temperature,' in order to allow the search to range widely over the response surface. As the 'temperature' drops, the search range narrows until the SA heuristic is focused on a single region of the search space that appears to contain the optimum, i.e., the minimum of the objective function. The region is then explored to determine a near optimal solution for the problem. Coding for the SA heuristic is minimal.

### 2.3.4 Tabu search

A memory-based heuristic, the tabu searches (TS) (Bettinger et al., 1998; Glover et al., 1993; Swisher et al., 2000) were invented from artificial intelligence concepts. The essential idea is to 'forbid' search moves to points already visited in the (usually discrete) search space. Starting with a single scenario, the basic form of the heuristic uses gradient or neighborhood search techniques to evaluate and compare scenarios. The process narrows the search space by maintaining a dynamic tabu list of unsuccessful, or forbidden, scenarios. The tabu list helps prevent moves in non-improving directions so that successive scenarios become increasingly optimal. However, in creating an efficient TS heuristic for a particular problem type, the structure of the tabu list must be designed carefully to prevent premature elimination of potential solutions. Ideally, the memory process used by the search should not only remember recent moves (short-term memory) but also have some way of looking back into longer-term memory and determining which patterns are working and which are not.

### 2.3.5 Genetic algorithm

Genetic algorithms (GAs) are widely used stochastic search methods originally developed by Holland (1975) and later refined by many others (Goldberg, 1989; Chambers, 1995; Srivastava et al., 1999). The genetic algorithms (GAs) use computer-based iterative procedures that employ the mechanics of natural selection and natural genetics to select the optimum solution for a given problem. Chromosomes that are judged to be the better fit are the most likely to survive into the next generation, and all chromosomes, regardless of fitness values are subjected to random mutations. As a random search algorithm, GA does not require continuity in the input variables. GAs search the optimum solution from a set of possible solutions at a time, rather than one solution at a time.

Nearly all GAs includes three basic components: a population of individuals, a function to score the fitness of an individual, and crossover, overlapping and mutation strategies for creating each successive population (Mitchell, 1999). A flow chart of the basic GA process is shown in Figure 2.1. The GA begins by creating an initial population of individuals. The probability of an individual's surviving to the next generation increases with increasing fitness. Individuals are introduced into the population in three ways: by direct reproduction with probability $p_r$, by mutation with probability $p_m$, or by crossover with probability $p_c$.

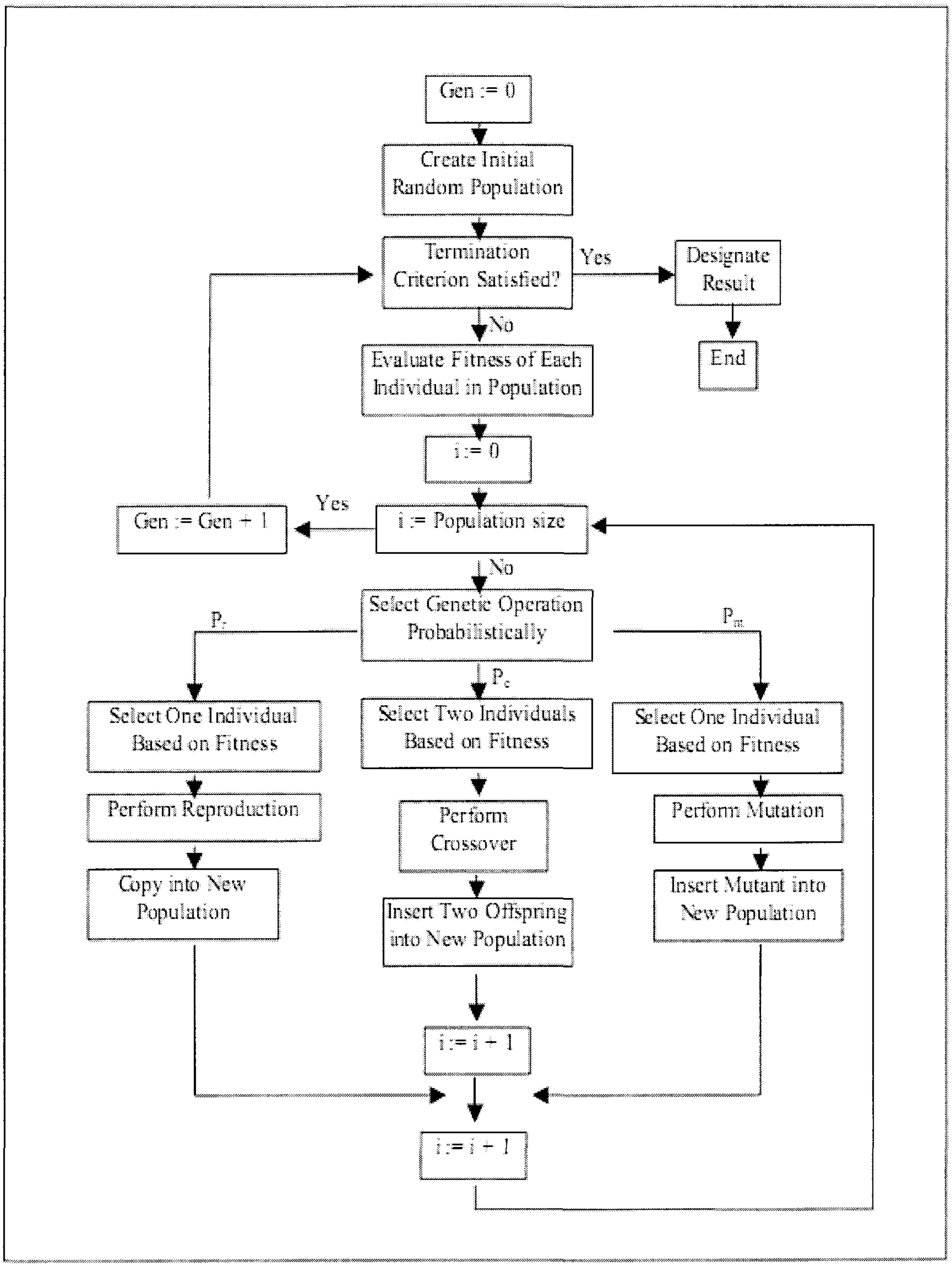


Figure 2.1: Flow chart of general GA (Koza, 1992)

Mutation changes one or more genes within an individual without regard to past or current fitness. Mutation is a purely random mechanism used to avoid local fitness maxima. Crossover combines two existing individuals to create two new individuals, each having values from both of the parents. Crossover helps redirect the search into new areas of the search space. Whether or not the parent individuals survive to the next generation depends on their fitness levels and on the replacement scheme of the GA.

The simple genetic algorithm uses non-overlapping populations. In each generation, the entire population is replaced with new individuals. Typically the best individual is carried over from one generation to the next (this is referred to as elitism) so that the algorithm does not inadvertently forget the best that it found. Maintaining the best individual also causes the algorithm to converge more quickly; in many selection algorithms, the best individual is more likely to be selected for mating. If the crossover accurately conveys good genetic material from parents to offspring, the population will improve. If the crossover operator does not maintain genetic material, the population will not improve and the genetic algorithm will perform no better than a random search. A crossover operator that generates children that are more often unlike their parents than like them leads the algorithm to do more exploration than exploitation of the search space. In search spaces with many infeasible solutions, such scattering will more often generate infeasible rather than feasible solutions.

The steady-state genetic algorithm uses overlapping populations. In each generation, a portion of the population is replaced by the newly generated individuals. At one extreme, only one or two individuals may be replaced during each generation (close to 100% overlap). At the other extreme, the steady-state algorithm becomes a simple genetic algorithm when the entire population is replaced (0% overlap). Since the algorithm only replaces a portion of the population of each generation, the best individuals are more likely to be selected and the population quickly converges to a solution. As a result, the steady-state algorithm often converges prematurely to a local optimum. Once again, the crossover and mutation operators are keys to the algorithm performance; a crossover operator that generates children unlike their parents and/or a high mutation rate can delay the convergence. In this study, the steady-state method is applied.

A GA ends upon reaching some termination criterion, which can be defined in a number of ways. For example, termination can be set to occur after a predetermined number of iterations of the optimization process. The termination criterion can also be defined as a minimal improvement in the maximum fitness score; that is, termination occurs either when the change in fitness score is less than a predetermined tolerance or when the score increase has remained below a tolerance for a predetermined number of generations.

### 2.3.6 Artificial neural network.

An ANN (Artificial Neural Network) is a mathematical model that simulates the operation of the human brain. ANNs consist of many simple arithmetic computing elements corresponding to neurons, and the network as a whole corresponds to a collection of interconnected neurons. The connections have associated numeric weights. Weights are the primary means of long term storage in neural networks, and learning usually takes place by updating the weights. Function approximation by neural networks begins in a random state and learns using repeated processing of a training set. The training set is a set of inputs and target output is calculated and used to adjust the weights. This continues until errors are sufficiently small or until no more improvement is possible. The trained ANN can be used with new inputs for estimation and prediction. ANNs have recently gained significant popularity because it has been proven that an ANN can create non-linear mappings between input and output variables under certain conditions, More specifically, Scarselli and Tsoi (1998) proved that continuous functions can be approximated up to any degree of precision with a finite number of hidden nodes in a three-layered feedforward neural network. They also mentioned that four or more layer feed forward neural networks are rarely used in practice but they are universal approximations. Kralisch et. al.(2003) presented a new approach for the optimization of a given land use scenario of a catchment in order to obtain a specific nitrogen output from that catchment using neural network.

**Evaluation of ANN models**

Tools for fitting parametric models to data such as non-linear regression require a selection of a parametric model prior to function approximation. This selection can introduce bias that might have significant impact on the success of the function approximation. An ANN, which is a non-parametric model, avoids this problem at the potential cost of additional computational resources and increased variance. However, varying the number of hidden units effectively creates models that vary from parametric to nonparametric for feed-forward neural networks (Reich and Barai, 2000). Smoothing can be performed using a lesser number of hidden units, while overfitting can result from using too many hidden units. Overfitting is characterized by a continuous decrease in a training error while a testing error increases when both data sets are representative of a data set to be modeled. There is no direct way to estimate confidence intervals for real data analysis. However, sensitivity analysis can address the variability of model results. Error estimation methods influence tradeoffs between bias and variance. Brief descriptions about these methods are as follows:

a. Resubstitution test : A single data set is used for training and testing
b. Holdout test: A data set is randomly divided into disjoint training and testing sets. It is common to select 2/3 of the set for training and the remaining 1/3 for testing.
c. *k*-fold cross validation: A data set is divided into *k* subsets of roughly equal size. A network is trained *k* times, each time leaving out one of the subsets from training and using it for testing.
d. Leave-one-out cross validation: Similar to *k*-fold cross validation, except that *k* is equal to the number of elements in the data set. Therefore, the training data set contains *k*-1 elements and the testing data set contains only one element. Training and testing is repeated *k* times.

The detailed explanations of these methods are described by Reich and Barai (1999).

Tradeoffs between bias and variability of these methods are summarized in table 2-1. The number of internal iterations in the table denotes the number of executions performed over the entire data set. The resubstitution estimate is very optimistic, meaning that it is biased very high. Holdout test estimate is pessimistic, which means it is biased low.

Table 2-1 Properties of Error Evaluation Methods (Reich and Barai, 1999)

| Estimation method | Size of training set | Size of testing set | Number of internal iterations | Method variability | Method bias |
|---|---|---|---|---|---|
| Resubstitution | n | n | 1 | Very high | Very optimistic |
| Holdout test | (0.6-0.8)·n | (0.2-0.4)·n | 1 | High | Pessimistic |
| k-fold cross-validation | n(k-1)/k | n/k | k | Moderate-high | Nearly unbiased |
| Leave-one-out | n-1 | 1 | n | Moderate-high | Nearly unbiased |

Generally, 10-fold cross validation is used for sample size greater than a hundred and leave-one-out cross validation is used for smaller data sets. Some error evaluation methods are better than the others for specific problems, but no method is always superior (Reich and Barai, 1999). The authors also suggested the following procedure for choosing the best ANN for a data set and its operational parameters:

1. Divide the data set into training and testing subsets.
2. Select the best ANN and its operational parameters using the following method:
    a. Test each combination of ANN and/or operational parameters with a k-fold cross validations test.

b. Select the combination that leads to the best cross validation performance.

3. Assessment of the best ANN:

a. Create a model from all training data using the best ANN and its operational parameters.

b. Test the model on the testing data set.

It is common to compare developed ANN models with another ANN or regression models. Cannon and Whitfeild (2002) developed empirical downscaling models for stream flow at 21 stations in British Columbia. They showed that ensemble neural network models either outperformed or yielded the same performance as stepwise linear regression models at 19 out of 21 stations. Raid et al. (2004) used a multilayer perceptron neural network with a back-propagation algorithm to predict the basin's rainfall runoff. They demonstrated that the ANN is more suitable to predict runoff than a classical regression model even in arid and semiarid regions with a very irregular rainfall and runoff characteristics. The ability of ANNs to account for non-linear patterns and irregular seasonal variation in a data set makes them well suited for use in hydrological modeling applications (Marier and Dandy, 2000).

### 2.3.7 Previous use of heuristics with NPS models

These six optimization techniques have been used previously to calibrate NPS models. Calibration by an optimization technique requires observed data relating to the model input and output parameters. Additionally, model parameters to be calibrated must be identified. By multiple runs through the NPS model, the optimization technique determines values for model parameters such that model output values for observed input values match, as nearly as possible, the related observed output values.

For example, the RSM was used to calibrate the Soil and Water Management Model (SWMM) to better estimate peak flow rates for an urban watershed in Singapore (Ibrahim and Liong, 1992). Results from the calibrated model were found to compare well with measured results. The urban watershed in Singapore was later used to analyze the calibration of SWMM by a GA, again with good results (Liong et al., 1995). A GA was used to calibrate a water quality model for predicting dissolved oxygen in streams (Mulligan and Brown, 1998). For comparison, a practitioner also calibrated the model using field measurements, empirical relationships, and engineering judgment to sequentially determine the model parameters. Parameter estimates produced by the GA calibration were comparable with the practitioner's estimates along a 64-km river stretch. Sumner et al. (1997) used SA in combination with the Simplex algorithm to calibrate a conceptual rainfall-runoff model for 25 watersheds in Australia and found that the computer-calibrated model fit the measured data more closely than did the user-calibrated model. Bajwa et al. (2009) studied a spatially distributed ANN model for modeling watershed - scale rainfall-runoff process. Kralisch et al. (2003) applied ANN to optimization of watershed management and found that a suitably designed neural network learning procedure will find near-optimal solutions to the problem if the starting land use scenario is reasonable

Optimization techniques for intractable problems (can be solved but not fast enough for the solution to be useful) have not been widely used to assist in NPS pollution control. Bettinger et al. (1998) used the TS heuristic to help determine the optimal solution to a complex model for improving aquatic habitat conditions in a timber harvesting area.

A GA was used to optimize the placement of BMPs within a watershed with the goal of minimizing pollutant loadings at the outlet while maximizing total net returns (Srivastava, 1999; Srivastava et al., 1999). Srivastava et al. (1999) combined the Annual AGricultural Non-point Source pollution model (AnnAGNPS) with a GA. They demonstrated their method on a 725-ha agricultural watershed, comparing conventional, conservation, and no- tillage on rotated crops. All other land uses (forest, pasture, and urban) were unchanged. Because of the problem representation used and a feature of the GA that required fitness scores to remain non-negative, the baseline scenario was chosen as the maximum possible pollution-loading scenario for the watershed. Net returns were based on a simple economic model using The Pennsylvania State University extension crop budget guidelines for farmers.

The GA used by Srivastava et al. (1999) found the optimal scheme for either pollution reduction while holding net returns constant, or for net returns while holding pollutant loading constant. After about 3800 evaluations, the algorithm identified a solution better than those solutions resulting from 3000 random combinations of BMPs. When set to optimize pollution reduction, sediment was reduced 44% from the baseline. Additionally, the GA converged to an optimal fitness after about 100 generations. Thus, any solution scenario after the first 100 generations met or nearly met the optimization conditions.

These heuristics use different methods for formulating and solving the BMP placement problem. Use of these heuristics in the area of NPS pollution control has focused primarily on calibration of NPS models. The genetic algorithm (GA) is the only mathematical heuristic cited in the literature as having been used for determining optimal scenarios with regard to cost-effective NPS pollution control (Srivastava et al., 1999).

The research by Srivastava et al. (1999) provides a strong argument for the effectiveness of a GA in optimizing cost-effective BMP scenarios. Their work invites further exploration into use of optimization heuristics in solving BMP placement problems. In particular, Srivastava (1999) suggested a need to evaluate alternative GA formulations as well as to explore the use of other heuristics.

## 2.4 Characteristics of urban runoff

Early work was limited mostly to planning for non-point source management and identified agricultural runoff as being different and requiring alternative management strategies than point sources such as municipal and industrial wastewaters. Highways and streets (Sartor and Boyd, 1972) were among the first to be investigated as non-point source pollutants. Other early research included the Nationwide Urban Runoff Program (US EPA, 1983) which addressed different types of landuse. Stenstrom (1984) studied relationship about oil & grease with landuse at San Francisco bay area. This paper concluded that 90% reduction in discharge from commercial properties and parking lots, which represented only 9.6% of the total surface area of Richmond, California, would result in a 53% reduction total oil and grease discharge.

### 2.4.1. Pollutants sources in Urban Runoff

**Heavy metals.** Urban storm-water runoff is known to be the largest contribution of metals to the local receiving water bodies (Characklis and Wiesner, 1997). The metals from anthropogenic sources include As, Pb, Zn, Ba, Cd, Fe and Cr. Whereas Al, Ca, Mg, Sr, Hg and Mn are usually from natural sources (Zartman et al., 2001). The chemical nature and source of the individual metals lead to different partitioning between solid and liquid. Particulate metals in the urban runoff are typically associated with organic matter from the tire wear, pavement surface wear and dust from exhausted pipes as well as minerals from soil, pavement and sources in the watershed (Roger et al., 1998).

A pollutant's partition between solid and liquid phases is of concern because the particulate fraction plays an important role in determining BMP efficiency. Hunter et al. (1981) reported that approximately 50% of total metals in storm-water were associated with particles. Characklis and Wiesner (1997) concluded that Zn exists mainly in the dissolved phase up to 80% of total concentration, while Fe is usually combined with coarse materials. Roger et al. (1998) stated Pb and Zn were often found in the sediments from motorways and Zn was associated with the fine particles (<50 $\mu m$ ). Pb concentrations are declining in urban runoff because of elimination of leaded gasoline (Characklis and Wiesner, 1997).

**Organics and Nutrients**. Polynuclear aromatic hydrocarbons and microorganisms such as fecal coliforms and pathogens are of concern because of their potential toxicity to human health and ecosystems. PAHs are usually combined with the other organic matter (Schueler, 1987) and the origins of PAHs in urban runoff are usually pavement leaching, tire abrasion, automobile's combustion processes and lubricating oils (Latimer et al., 1990; Takada et al., 1991; Ngabe et al., 2000; Krein and Schorer, 2000; Kamalakkannan et al., 2004). Atmospheric deposition and regional air pollution emission are also a significant PAH source (Herricks, 1995).

Urban runoff also carries significant amount of nutrients such as nitrogen and phosphorous (Abustan et al., 1998). Nutrients sources include fertilizer applied to yards, roof runoff, various household chemicals and street runoff. Vase et al. (2002) performed an experimental study of pollutant accumulation on urban road surfaces and found that majority of total phosphorous and nitrogen in the solid samples was associated with particles less that 50 $\mu m$ in diameter.

Borst and Selvakumar (2003) found large concentrations of fecal coliforms and pathogens in urban runoff. Microorganisms are self-suspended or absorbed to suspended particles and prefer particles that are larger than 30 $\mu m$ in diameter (Schillinger and Gannon, 1985).

### 2.4.2. Storm-water Pollutant Loading Models

Urban land use information is important for storm-water modeling (Wong, 1997; Burian et al., 2002) as quantity and quality of storm-water runoff is related to land use.

The Runoff Coefficient (RC) is one of the main components for determining storm-water runoff volume and represents the fraction of rainfall that actually reaches the receiving water (Wong et al., 1997). RC is highly correlated to imperviousness of the surface and the following equation is used by the County of Los Angeles, Department of Public Works (LADPW, 2000b):

$$RC = 0.8 \times I + 0.1 \qquad (2.1)$$

where, RC is runoff coefficient, I is the impervious fraction.

However, RCs used in this report were developed using a Geographical Information System (GIS) as a function of hydrologic soil group, slope, and land use according to Browne relation (1990) in the Ballona Creek Watershed. They differ from the LA County estimates based on soil type as well as imperviousness.

Then the annual average storm runoff volume is calculated as follows:

$$RV = RC \times A \times RF \times CF \qquad (2.2)$$

where RV is annual storm-water runoff ($m^3/yr$), A is drainage area ($m^2$), RF is annual rainfall (mm), and CF is conversion factor. The imperviousness and runoff coefficients can be estimated as a function of land use types.

The Event Mean Concentration (EMC) is the average pollutant concentration during a storm event and is widely used for estimating storm-water runoff pollution. The mathematical definition of EMCs is total pollutant mass discharged during an event divided by total volume discharge of the storm event as follows (Huber, 1993):

$$EMC = \frac{M}{V} = \frac{\int C(t) \cdot Q(t)dt}{\int Q(t)dt} \qquad (2.3)$$

where M is total mass of pollutant during the storm event, V is total storm-water runoff volume, C(t) is pollutant concentration that is function of time, and Q(t) is storm-water flow rate over time. EMCs are also related to land uses although they are dependent on sites and storm events (Smullen et al., 1999).

Based on the information of runoff volume and EMCs for each pollutant type, wet-weather pollutant load can be estimated as follows:

$$PLi = \alpha \times RC \times RF \times A \times EMCi \qquad (2.4)$$

where PLi and EMCi are the annual pollutant load and the EMC for pollutant type i, respectively and $\alpha$ is a conversion factor for consistency of the units. These simplistic models are most appropriate for longer time periods, such as seasonal or annual periods, assuming average annual rainfall.

This simple modeling approach has been described as the “simple method” or the “spread sheet method.” The term “volume-concentration” method is used in this dissertation. Park et al. (2009) have performed a sensitivity study of several applications of this method to the upper Ballona Creek Watershed, and show the impact of the variability in assumptions, such as land use definitions.

## 2.5 Best Management Practices (BMP)

Reductions in non-point source (NPS) pollution can be attained by reducing activities that produce NPS pollutants, reducing the amount of pollutants generated by an existing activity and reducing the negative effects these pollutants can have by controlling their dispersal. To that end, NPS best management practices (BMPs) are important tools in controlling NPS pollution and environmental contamination. Also urban storm-water runoff can be controlled by the use of various best management practices. BMPs are either non-structural or structural, varying from small, site-specific practices to large-scale regional practices.

### 2.5.1 Structural storm-water BMPs

An urban storm-water BMP is believed to be a "best" way of treating or limiting pollutants in storm-water runoff. The storm-water treatment practices investigated here include wet ponds, bioretention, bio-swale, infiltration system, sand filters, and bio-filter practices.

**A. Wet ponds**, also called wet detention ponds, have been used in many sites in USA longer than any other storm-water BMP. Wet ponds are runoff-holding facilities that have standing water in them constantly. Storm flows are held in the ponds temporarily and then released to minimize large scale flooding. The primary pollutant-removal mechanism is settling (sedimentation) while storm-water runoff resides in the pool. Nutrient uptake also occurs through biological activity. Wet ponds can be designed to look like natural lakes to enhance the value of surrounding property. They have been and can be used for commercial sites as small as 1 acre and for watersheds as large as 100 acres or more.

Wetlands, also called constructed wetlands, are comparable to wet ponds but are much shallower and more heavily vegetated with wetland plants. They serve as a natural filter for urban runoff, help slow the flow of water to receiving waters, and replenish groundwater. As storm-water runoff flows through the wetlands, sedimentation, adsorption, and biological processes achieve pollutant removal. In many sites, constructed storm-water wetlands have been located on watersheds as small as 4 to 5 acres, but they are more commonly used for larger drainage areas and typically serve watersheds ranging from 15 acres to more than 100 acres. Due to the vegetative cover, wetland effluent is typically cooler than that of wet ponds, minimizing the impact of thermal pollution. Wetlands consume a relatively large amount of space and thus have limited applicability in highly urbanized settings.

**B. Sand filters** are usually two-chambered storm-water treatment practices. Water enters the first chamber, where debris and large suspended solids settle out, and then moves to a second chamber, where a filter bed filled with sand or another filtering media remove other forms of pollution. At the bottom of the sand layer, an underdrain pipe typically connects the treated water with the existing drainage network. Sand filters are particularly well suited for treating storm-water runoff in urban areas because they can be designed to be walked over or driven on, thus preserving expensive land. Typically, the sand filter will treat a drainage catchment of only a few acres. This practice is designed for impervious watersheds in particular.

**C. Biofilters** pass storm-water slowly over a vegetated surface in the form of a swale or filter strip to filter pollutants and infiltrate the runoff. Biofilters are most effective when designed to receive sheet flow from paved areas and maximize water contact with the biofilter vegetation and the soil surface. They are often placed within vegetated setbacks, landscaped common areas and other required open areas in residential, commercial, industrial and institutional land uses. Biofilters can have tributary areas of up to 5 acres, which makes them appropriate for lawns and parking areas. There are several limitations as follows:

1. Irrigation may be necessary to maintain vegetative cover.
2. Not appropriate for steep unstable slopes.
3. Large area requirements may make this BMP infeasible for some sites.
4. Not appropriate for pollutants toxic to vegetation.

Biofilters are an effective means for removing storm-water pollutants, infiltrating runoff, stabilizing soil and controlling erosion. Biofilters accomplish this in several ways. Vegetative covers shield soil surfaces from the impact of falling rain. Vegetation, such as turf grass or other ground cover, disperses flow and provides a rough surface to reduce flow velocity, which promotes infiltration and sediment deposition. Plants also remove nutrients in storm-water and transpire moisture from the soil. Pollutant removal effectiveness for biofilters is a function of area, flow depth, travel time and the quality of the vegetative cover. Biofilters are relatively easy to design, install and maintain. Finally, maintaining a biofilter often requires little more than normal landscape maintenance activities such as irrigation and mowing. Compared with some other means for improving storm-water runoff quality, biofilters provide a relatively unobtrusive, attractive, long-term and inexpensive storm-water quality management technique.

**D. Bioretention** areas are landscaped and vegetated filters for storm-water runoff. Surface runoff is directed into shallow, landscaped depressions. Trees and shrubs are planted in bedding material consisting of a high percentage of sand and lesser amounts of silt, clay, and/or organic matter. During rain events, storm-water pools above the mulch and soil in the system. The remaining runoff filters through the mulch (layers of pervious material such as old leaves, small pieces of wood and sawdust) and prepared soil mix. Typically, the filtered runoff is collected in a perforated underdrain and is returned to the storm drain system. Bioretention systems are ideally suited to many ultra-urban areas as they will fit existing parking lot lands or other landscaped areas. Because bioretention potentially can fulfill two purposes—water quality control and landscaping requirements—their use is expected to increase. Bioretention areas typically serve very small watersheds, such as (portions of) parking lots or residential runoff areas.

**E. Infiltration System** is the process where water enters the ground and moves downward through the unsaturated soil zone. Infiltration is ideal for management and conservation of runoff because it filters pollutants through the soil and restores natural flows to groundwater and downstream water bodies. The slow flow of runoff allows pollutants to settle into the soil where they are naturally mitigated. The reduced volume of runoff that remains takes a long time to reach the outfall, and when it empties into a natural water body or storm sewer, its pollutant load is greatly reduced.

Infiltration basins can be either open or closed. Open infiltration basins, include ponds, swales and other landscape features, are usually vegetated to maintain the porosity of the soil structure and to reduce erosion. Closed infiltration basins can be constructed under the land surface with open graded crushed stone, leaving the surface to be used for parking or other uses.

Infiltration systems are often designed to capture the "first flush" storm event and used in combination with a detention basin to control peak hydraulic flows. They effectively remove suspended solids, particulates, bacteria, organics and soluble metals and nutrients through filtration process of the vehicle, absorption and microbial decomposition. Groundwater contamination should be considered as a potential adverse effect and should be considered where shallow groundwater is a source of drinking water. In cases where groundwater sources are deep, there is a very low chance of contamination from normal concentrations of typical urban runoff. In some cases infiltration basins may not be appropriate due to groundwater contamination concerns.

**f. Vegetated Swales** are open, shallow channels with vegetation covering the side slopes and bottom that collect and slowly convey runoff flow to downstream discharge points. They are designed to treat runoff through filtering by the vegetation in the channel, filtering through a subsoil matrix, and/or infiltration into the underlying soils. Swales can be natural or manmade. They trap particulate pollutants (suspended solids and trace metals), promote infiltration, and reduce the flow velocity of storm-water runoff. Vegetated swales can serve as part of a storm-water drainage system and can replace curbs, gutters and storm sewer systems.

### 2.5.2 Non-structural BMPs

- Nonstructural BMPs focus on prevention and removal of storm-water volumes and constituent loads at their source. There are no physical structures associated with nonstructural BMPs

Examples of common nonstructural BMPs for industries are:

- Materials management practices that prevent either rainfall or storm-water from collecting and transporting water pollutants
- Storm drain maintenance practices such as street sweeping and catch basin cleaning
- Equipment maintenance
- Spill prevention, control, and cleanup
- Eliminating non-storm-water discharges

- Employee training
- Recordkeeping
- Good housekeeping

. Benefits of nonstructural BMPs include:

- Limiting the amount of pollutants that potentially enter storm-water runoff
- Reducing the need for the more expensive structural BMPs
- Improving overall BMP efficiency and helping to reduce maintenance requirements

### 2.5.3 Total BMP costs

The total cost of a storm-water BMP is made up construction costs plus maintenance and operation costs. When comparing storm-water BMP and deciding which practice to select, we also should consider the long-term maintenance cost.

Typical costs for each BMPs were retrieved from Chapter 6.0, "Costs and Benefits of Storm Water BMPs," of an EPA on-line document (EPA, 1999). Table 2-2, it states that each BMPs construction costs/acre. All values reported in the document need to be divided by an adjustment factor to account for regional differences. Using the average annual federal inflation rate (3%), the capital cost of wet ponds in 2005 is \$2,622.22/acre.

Capitalized at a 3% interest rate over a 25 year finance period, this value becomes $150.59/acre/year. It should be noted hat stormwater ponds can be expected to function for up to 50 years. To this value, the annual operation and maintenance costs must be added. Operation and maintenance costs for retention basins can range from 3-6% of the construction costs (EPA, 1999). This study used an average value of 4.5% and applied this to the regionally adjusted construction cost, to get $118.00/acre/year O&M cost. Thus, the final cost of wet ponds is $268.59/acre/year. Other BMPs' cost calculations are similar to this one.

Table 2-2. Storm-water BMP Costs

| Cost | Wet Ponds | Infiltration | Bioswale Bio-retention | Sandfilter | Biofilter |
|---|---|---|---|---|---|
| Construction | $2,622.22/ac | $7,866.67/ac | $10,890/ac | $14,670.17/ac | $15,733.33/ac |
| Finance period | 25years | 25years | 25years | 25years | 25years |
| Sub-total capitalized over lifespan | $150.59/ac/yr | $451.7/ac/yr | $630.1/ac/yr | $842.5/ac/yr | $903.5/ac/yr |
| OMR/year (% Const.) | 4.5% | 10.5% | 5% | 12% | 6% |
| Annual OMR | $118.00/ac/yr | $826.0/ac/yr | $544.5/ac/yr | $1,760/ac/yr | $944.0/ac/yr |
| Total Cost | $268.59/ac/yr | $1,278/ac/yr | $1,175/ac/yr | $2,603/ac/yr | $1,848/ac/yr |

Source : ASCE (2001), EPA (1999)

Any cost estimate needs to be adjusted for inflation and regional differences. All costs assume a 3 percent annual inflation rate. In addition, studies are adjusted to the 'twenty cities average' construction cost index, to adjust for regional biases, based on a methodology followed by the American Public Works Association. Using EPA's rainfall zones (Fig. 2-2), a cost adjustment factor is assigned to each zone.

Table 2-3 Regional Cost Adjustment Factors

| Rainfall zone | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Adjustment Factor | 1.12 | 0.9 | 0.67 | 0.92 | 0.67 | 1.24 | 1.04 | 1.04 | 0.76 |

Source : APWA, 1992

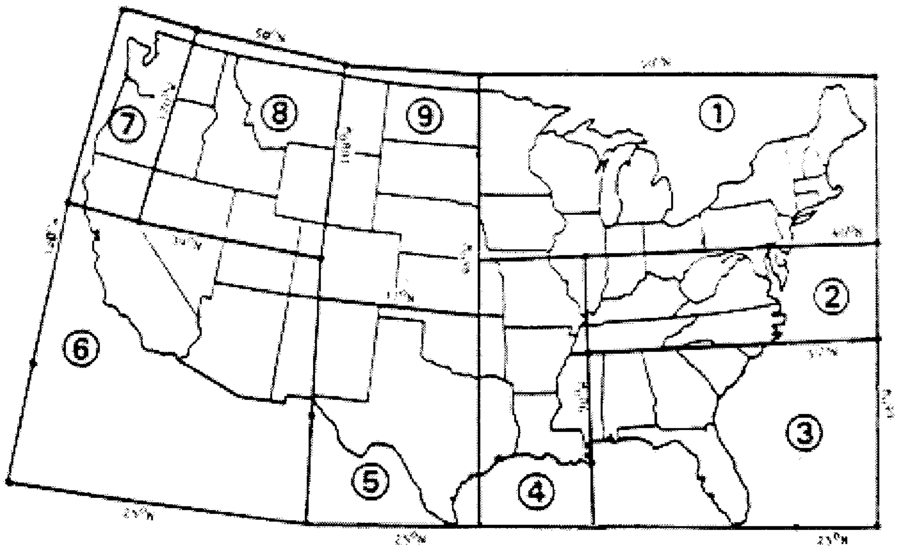


Figure 2.2 Rainfall zones of United States

Source: Methodology for Analysis of Detention Basins for Control of Urban Runoff Quality, prepared for U.S. EPA, Office of Water, Nonpoint Source Division, Washington, DC, 1986.
[55 FR 48073, Nov. 16, 1990]

### 2.5.4 Removal effectiveness

The performance of structural BMPs is highly dependent on site-specific factors including rainfall intensity, duration, and volume, pollutant concentrations, and climate patterns.

Table 2-4. Median removal rates for 5 BMPs

| | TSS (Removal %) | TN (Removal %) | TP (Removal %) |
|---|---|---|---|
| Wet ponds | 75 | 28 | 46 |
| Infiltration system | 80 | 60 | 55 |
| Bioswale | 80 | 70 | 35 |
| Biofilter | 82 | 45 | 59 |
| Sandfilter | 81 | 50 | 35 |

Source: Wossink and Hunt (2003) and BMP handbook (CASQA, 2003)

Table 2-4 shows median removal rates for five BMPs. While BMP performance by pollutant removal rate is now widely accepted, many researchers disagree over the use of removal rates (Strecker et al., 2001: Jones et al., 2008). Strecker et al. (2001) reported that effluent quality concentration can be a better way to characterize BMPs efficiency, and it is important to test whether the BMP had a statistically crucial effect on water quality. Jones et al. (2008) suggested that BMP analysis should be using an approach that focuses on the following:

- How much the BMP reduces runoff volumes
- How much runoff is treated (versus bypassed)
- Whether the BMP can demonstrate a statistical difference in effluent quality compared to influent quality
- What distribution of effluent quality is achieved.

Table 2-5. Median effluent concentrations for 5 BMPs

| | TSS (mg/L) | TN (mg/L) | TP (mg/L) |
|---|---|---|---|
| Wet ponds | 20 | 4.5 | 2 |
| Inflitration system | 27 | 10.5 | 3 |
| Bioswale | 22 | 5 | 2 |
| Biofilter | 32 | 6 | 4 |
| Sandfilter | 16 | 12 | 1.5 |

Source: International Stormwater Best Management Practices Database [1999-2008]

Thus, this GA tool simulated pollution reductions not only using by BMPs removal rates but also using by BMPs effluent concentration. However, this GA simulation does not include other potential benefits of BMPs. Energy conservation, creation of open space, habitat restoration, flood control, and water reuse benefits have not been quantified. There is a need to develop methods to quantify these benefits because they may improve water quality and reduce overall cost.

## 2.6 Proposition O

Federal mandates initiated in 1999 under the Clean Water Act established that over 60 water quality regulations would be adopted in Los Angeles. These regulations are developed for each specific body of water and/or watershed. The goals of Proposition O are

- Protect rivers, lakes, beaches and the ocean;

- Conserve and protect drinking water and other water sources;
- Reduce flooding and use neighborhood parks to decrease polluted runoff;
- Capture, clean up and reuse storm water.

It authorized the City of Los Angeles to issue $500 million in general bonds. After screening 52 submitted projects, 21 proposals were approved for concept development. The on-going project to comply with the Trash TMDLs(Total Maximum Daily Load) by installing catch basin inserts and covers in high trash generating areas were also approved for continued funding. The time of September, 2007, $462,432,662 has been allocated for those projects approved by City Council and $12,842,042 has been recommended by COAC (Citizens Oversight Advisory Committee) and AOC (Administrative Oversight Committee) and is pending City Council approval.

In this report (Stenstrom, 2007), storm-water pollutant loads of the project sites were estimated using the empirical volume-concentration approach that employs land use definitions for pollutant concentrations and runoff coefficients. Dry-weather pollutant load were estimated from dry weather runoff and concentration data collected by the Los Angeles County Department of Public Works' monitoring programs. The performance of proposed Best Management Practices (BMPs) was evaluated assuming all runoff from the drainage area passes through the series of proposed BMPs in a sequential way. The pollution reduction and the effect of a project on TMDL compliance were estimated at the watershed scale because it is required by the TMDL process. Three watersheds were considered: Greater Ballona Creek, Los Angeles River, and Dominguez Channel. The annual average stormwater runoff volume and pollutant loads were estimated for each watershed to accommodate watershed-scale decision making. Annual stormwater runoff volume from the watersheds were approximately $6.54*10^7$, $23.6*10^7$, and $4.32*10^7$ $m^3$/year for BC, LAR, and DC watersheds, respectively, assuming annual average rainfall of 305 mm, which is the 30 year average rainfall in the region. The average runoff coefficients for the entire watershed were 0.39, 0.36 and 0.49 for BC, LAR, and DC watersheds, respectively, which is proportional to the imperviousness of each watershed.

Table 2-6. Status of projects funding and calculating cost

| No | Project title | Total funding | total cost /mass load | total cost /drainage area |
|---|---|---|---|---|
| 11 | Cesar Chavez Complex | $9,540,000 | NA | $14,135 |
| 28 | Oros Green Street Trees | $972,651 | $188,696 | $115,806 |
| 41 | Inner Cabrillo Beach Water Quality Improvement Project | $8,811,353 | NA | NA |
| 52 | Catch Basin Inserts and Coverings Phase II | $10,000,000 | NA | NA |
| 51 | Santa Monica Bay Upgrades | $35,000,000 | $117,776 | $2,634 |
| 10 | Strathern Pit Multiuse Project | $22,505,000 | $22,543 | $16,403 |
| 12 | Cabrito Paseo Walkway/Bike Path | $4,463,009 | $500,292 | $232,110 |
| 16 | South Los Angeles Wetlands Park | $13,380,243 | $61,296 | $26,756 |
| 20a | Grand Avenue Storm-water BMPs | $1,075,927 | $165,261 | $68,618 |
| 20b | La Cienega/Fairfax Powerline Storm-water BMPs | $7,667,888 | $2,202 | $1,539 |
| 20c | Mar Vista Rec. Cntr BMPs | $4,556,186 | $2,201 | $18,693 |
| 22a | Imperial Highway BMPs | $2,723,403 | $170,686 | $164,695 |
| 22d | Westminster Dog Park BMPs | $1,438,755 | $13,745,865 | $517,166 |
| 23 | Aliso Wash - Limekiln Creek Confluence Restoration Project | $10,893,483 | $2,718 | NA |
| 29 | Echo Park Lake Project | $84,263,313 | $253,194 | $115,066 |
| 31 | Parking Grove in El Sereno | $3,984,635 | $457,031 | $1,191,577 |
| 35 | Rosecrans Recreational Center Storm-water Enhancement | $6,754,033 | $11,334,565 | $532,736 |
| 36 | Lake Machado Water Quality/Habitat Improvemen | $99,523,897 | $26,738 | $6,715 |
| 36a | Wilmington Drain Project | $17,942,534 | NA | NA |
| 40 | Peck Park Canyon Enhancement | $6,190,000 | $51,922,120 | $62,091 |
| 9 | The LA Zoo Parking Lot | $13,904,242 | $1,595,734 | $421,878 |
| 14 | Hansen Dam Recreational Area Restoration Project | $2,220,702 | $294,037 | $29,857 |

(Adapted from Proposition O Report, Stenstrom, 2007)

# Chapter 3: Development of Optimization Procedure

## 3.1 Introduction

Developments of the optimization procedure start from determining which optimization method to use. The six heuristics described in Chapter 2 were compared based on a number of factors, leading to selection of the genetic algorithm (GA) as the heuristic to incorporate into the optimization procedure. The next step was to develop methods for predicting the NPS pollution reduction and economic impacts of each BMPs placement scenario. These methods were then formulated into a multi-objective optimization function.

The resulting optimization procedure is comprised of three components: GA parameters, BMPs characteristics for evaluating each pollutant reduction, and an economic component for assessing construction and OMR (Operation, Maintenance and Repair) costs. Finally, the optimization procedure was implemented as a computer program using Microsoft C # 2.0 ("C sharp") languages and tested. C# is a new computer language intended to be a simple, modern, general-purpose, object-oriented programming language based on C++ and java.

Because of the computer time involved in running a detailed NPS model, using such a model to predict pollutant reduction or pollutant loading within the optimization procedure may result in total runtimes of several hours. Thus, efficient problem formulation that limits unnecessary evaluations of the objective function is desirable. In particular, the likelihood of the optimization procedure being used in the future may be improved by reducing runtime. Total runtime of the procedure may be reduced by development of a simplified NPS model.

## 3.2 Choosing an optimization heuristic

The BMPs placement problem was determined to be an intractable problem (can be solved but not fast enough for the solution to be useful) based on the large number of watershed, the exponential number of possible BMP combinations. Literature, particularly related to global optimization techniques (e.g., Swisher et al., 2000) and watershed-level NPS pollution control (e.g., Braden et al. 1989, Srivastava et al., 1999), was examined to determine potential methods for solving BMPs placement problem.

Six heuristics for solving intractable problems were selected and considered in order to determine potential characterizations of the problem. As part of this consideration, several factors were compared among the heuristics, including performance for similar types of problems in previous studies, proof of convergence, and ease of formulation. Next, each heuristic's continuity and differentiability requirements, convergence rate, and relative efficiency were considered, as were sensitivity of the heuristic to the problem formulation and the number of points needed as a starting requirement. Table 3-1 summarizes the heuristics in terms of these factors, which are discussed in more detail in the following subsections. Factors greatly impacting procedure development are shown in bold.

Table 3-1 Summary of heuristics in terms of each factor considered

| Factors \ Heuristics | Response surface method | Shuffled complex evolution | Simulated annealing | Tabu search | Artifial Neural Network | Genetic algorithm |
|---|---|---|---|---|---|---|
| Demonstrated performance on BMP placement problems | No | No | No | No | **Yes** | **Yes** |
| Proven convergence | Yes | Uncertain | Yes | Uncertain | Uncertain | Yes |
| Formulation ease | Low | Low | **High** | Low | Medium | **High** |
| Continuity or differentiability required | Yes | Yes | **No** | **No** | **No** | **No** |
| Convergence rate | Uncertain | Uncertain | Uncertain | Uncertain | Uncertain | Uncertain |
| Relative efficiency | Uncertain | High | Low | Uncertain | Medium | Medium |
| Sensitivity to formulation | High | High | High | High | High | High |
| Number of initial points required | High | High | Low | Low | **Very High** | High |

As a result of the overall process of choosing an optimization heuristic, it was determined that the problem was most simply suited to characterization as a combinatorial optimization problem. Thus, the response surface methodology (RSM) and shuffled complex evolution (SCE) heuristics, which require continuity in the input variables, could now be eliminated from consideration. The remaining four heuristics were determined to be suitable for solving the BMP placement problem. However, this problem appears easier to formulate for use with simulated annealing (SA) and the GA than for the tabu search (TS) and artificial neural networks (ANN).

In addition to the above heuristics, the use of a classical method, such as integer programming or nonlinear optimization, was considered briefly. For example, Braden et al. (1989) used nonlinear optimization to address this problem. They evaluated management practices by small hydrologic units instead of by fields. The hydrologic units were then grouped into catchments within the watershed and their order along the catchment flow path identified. This data preparation can become very difficult for large or topographically complex watersheds. A NPS model could be incorporated into a classical optimization method to overcome this difficulty. However, it seemed that use of a classical optimization technique would necessitate careful formulation with regard to the relationship between scenarios in order to implement an efficient optimization algorithm and prevent enumeration over all possible solutions. As compared to the simplicity of using a heuristic to solve this problem, efficient problem formulation using classical optimization techniques seemed less straightforward.

### 3.2.1 Demonstrated performance on BMP placement problems

Glover et al. (1995) listed a range of problems for which Tabu Search has provided high quality solutions. However, problems dealing with modeling of natural systems were not mentioned. SCE method has not extended beyond its development purpose of improving model calibration. Stone et al. (2002) used SA (Simulated Annealing) to assign fields and livestock attributes to farms based field characteristics, basic land use data, and watershed-level statistical information about farm types. Additionally, TS, SCE, SA, and RSM have been used to varying degrees in calibrating NPS models (Ibrahim and Liong, 1992; Liong et al., 1995; Cooper et al., 1997). However, they were not found in the literature to have been used in determining optimal BMPs placement scenarios with regard to NPS pollution control.

Kralisch et al. (2003) applied ANN to optimization of watershed management and found that a suitably designed neural network learning procedure will find near-optimal solutions to the problem if the starting land use scenario is reasonable

Genetic algorithms have been used to address biologically related questions such as biological arms races and symbiosis (Mitchell, 1999), but only one example was found in the literature dealing with the response of a watershed to NPS pollution reduction. As previously discussed, Srivastava et al. (1999) used a GA and AnnAGNPS to optimize BMP placement with regard to NPS loadings and to private costs. They found that the GA performed better than did scenarios consisting of random assignments of BMPs.

### 3.2.2 Proven convergence

Each heuristic was evaluated with regard to proven convergence to the optimum. The basic form of RSM is founded on statistical theory (Jacobson and Schruben, 1989; Myers, 1971), using least squares and experimental design to determine the response surface. For each surface suggested by the technique, gradients are used to determine the improving direction along the surface.

The TS, SA, SCE, and GA are general search strategies for intractable problems (can be solved but not fast enough for the solution to be useful) and are not intended to enumerate over all possible solutions. Nor they are guaranteed to find the optimum, regardless of how long they run. Instead, they search in a controlled, but often probabilistic, fashion for the best solution achievable within a finite amount of time. However, for a connected search space, the SA has been proven to converge arbitrarily close to the optimum (Lundy and Mees, 1986). The GA has also been proven to converge, with high probability, to the optimum for a problem involving allocation of documents within a computer (Siegelmann and Frieder, 1991).

By extension, the SCE, as a variation of the GA, can be expected to converge. However, since a documented convergence proof for SCE was not found, the SCE was ranked as “uncertain”. Likewise, literature on the proven convergence of the TS was not found, resulting in a relative ranking of “uncertain.”

### 3.2.3 Formulation ease

Solving a problem using the RSM (Response Surface Methodology) often involves sampling the search space through a factorial experimental design. The output of each point in the experimental design is calculated. Then the RSM is used to fit a response surface and determine an optimum. To use this method as an optimization procedure, one must relate a given response value with the associated input point. In this respect, the RSM is not well suited for a large number, combinatorial optimization problem. In this problem there are far more management units in a watershed than there are BMPs. The problem representation for this heuristic would necessitate analysis and manipulation of logistic regression equations with one dependent variable (pollutant loading at the outlet), hundreds of independent variables (each field being a separate variable), and few values for each independent variable (values consisting of a one-to-one mapping with each possible BMP set; that is, ten values for ten sets of BMPs). This configuration is beyond the capability of standard statistical software.

For the TA, SA, and GA techniques the problem could be formulated using a NPS model to calculate the objective function. However, the runtime for an NPS model is lengthy as compared to the other calculations within the optimization heuristics. Thus, efficient problem formulation that limits unnecessary evaluations of the objective function is desirable.

For the TS, the need for an efficient problem formulation increases the importance of designing a dynamic, problem-specific tabu list. In particular, to minimize the number of evaluations of the NPS model, it is preferable that scenarios can be checked against the tabu list without requiring evaluation by the NPS model. However, the categorical nature of this problem increases the complexity in creating an adaptive list based on BMP patterns within scenarios. Thus, TS was ranked as "low" with regard to formulation ease.

Both the SA and GA can be developed to use a NPS model in the objective function. Evaluation efficiencies for these heuristics are largely a function of optimization parameters, such as cooling rate and crossover rate or mutation rate. Effective values for these parameters are problem dependent, but can be determined through sensitivity analysis. Thus, the SA and GA were ranked as "high" in formulation ease as compared to the other methods.

### 3.2.4 Continuity and differentiability

The RSM requires continuous data (Myers, 1971), or a continuous representation of the data, in order to fit a surface to the optimization function using regression and directional search. Although it is possible to regress an equation in which some variables are ordinal, but not continuous, the remainder of the RSM requires that, based on this regression, the direction(s) of improvement can be ascertained. Because the relationships between groups of BMPS and the resulting watershed response cannot be precisely determined, it is not clear which BMP in a scenario should be changed to improve the watershed response.

The SCE does not require continuity or differentiability in its objective function (Duan et al., 1993). However, use of the Nelder and Mead strategy within the SCE implies that the input variables are continuous (Bazaraa et al., 1990).

Using both techniques for this research problem would require a mapping between the BMP on each watershed and some measure of fitness or impact of that BMP on the watershed (a continuous, or at least ordinal, representation). Based on this mapping a gradient or path of improvement between BMPs could be determined. However, due to natural variation among and within field sites, it is extremely difficult to accurately measure interactions among BMPs or to determine an average pollutant reduction value for a particular BMP irrespective of its surroundings.

The TS, SA, ANN and GA do not require continuity or differentiability. They require only the capability of mapping each scenario to a fitness function. With this regard, these four heuristics are well suited to the BMP optimization replacement problem.

### 3.2.5 Convergence rate

Convergence rate can be defined (Bazaraa et al., 1993) as the ratio of the improvement of the objective function to the number of iterations, number of functional evaluations or amount of computational time. The number of objective function calculations required in each iteration of the optimization technique has a significant impact on the convergence rate. The RSM and SCE require repeated calculations of the objective function for neighboring points in order to determine the improving directions. The SCE algorithm requires the optimal function be calculated as many as three times for each new point (Duan et al., 1993).

The TS, SA, and GA methods calculate the objective function only once for each new point. However, like the RSM and SCE techniques, the GA requires numerous points, or population members, to be created and analyzed from every iteration. The TS may require repeated searching through previous iterations to determine if a new solution is not tabu. The SA requires multiple evaluations at each level of cooling and multiple levels of cooling.

Due to minimal previous literature on these heuristics for solving watershed response problems, neither the computation time nor number of iterations needed for convergence is clear. Also, it is not clear how the number of iterations or computational time needed for each iteration compare across the heuristics. Thus, this factor was determined to be "uncertain" for all heuristics.

### 3.2.6 Relative efficiency

While convergence rate primarily considers the performance of individual heuristics, the relative efficiency factor is used to compare heuristics. High relative efficiency refers to nearing an optimal solution in the least number of iterations or functional evaluations, as compared to other heuristics for the same problem. For example, while Lundy and Mees (1986) demonstrated a problem in which SA converges more quickly than a repeated descent algorithm, Eglese (1990) stated that for a problem with only a global optimum the descent algorithm will converge more quickly. Additionally, Eglese (1990) reviewed a number of modifications to SA that have been used to decrease run time.

Thyer et al. (1999) used calibration of a watershed runoff model for two watersheds to compare robustness and efficiency of the SA method used by Sumner et al. (1997) with the SCE method. Although results were heavily dependent on the watershed, the SCE method appeared to perform better. Cooper et al. (1997) found convergence under SA averaged 12000 evaluations. In the same study, SCE and GA techniques averaged 6000 and 9000 evaluations, respectively. Based on this study, SCE, SA and GA were assigned relative rankings of "high", "low", and "medium", respectively. Comparisons of RSM, of ANN and of TS to the other heuristics were not found in the literature. Thus, the relative efficiencies of RSM and TS were ranked as "uncertain."

## 3.3 Development considerations

Development of the optimization procedure involved representing the relationships of the physical system (i.e., the watershed) as a mathematical model. To meet the research goals for the BMPs placement, the model had to provide a way to rank pollutant reduction and cost-effectiveness of BMPs through fitness scores and objective functions.

### 3.3.1 Program structure

Assume

$B_A$ = BMP type A

$B_B$ = BMP type B

$B_C$ = BMP type C

⋮

$B_z$ = BMP type Z

The data structure for BMPs types array contains removal rate (R) and construction and OMR (Operation, maintenance and repair) money (M), which represent as: $R_A$, $R_B$, … , $R_z$ and $M_A$, $M_B$ … $M_z$

Each BMP type contains different removal rate depending on types of pollutant (TSS, TP and TN). Let represent the removal rate of each BMPs for pollutant type as $R^1{}_z$, $R^2{}_z$ and $R^3{}_z$.

The construction cost for each BMPs type is calculated by cost per watershed area (acre). So, the total cost of building and operating BMPs are wholly depending on the watershed area.

Let watershed property variables as $W^n$, where n is the index of watersheds. Other properties are represented as:

$W_A^n$ = Area of the watershed (acre).

$W_1^n$ = Total loading (kg) of pollutant type 1 for watershed index $n$.

$W_2^n$ = Total loading (kg) of pollutant type 2 for watershed index $n$.

$W_3^n$ = Total loading (kg) of pollutant type 3 for watershed index $n$.

**Representation of chromosome**:

Let's assume that the each element in a chromosome array is called DNA (also can be called gene). The number of genes (DNA) available in a chromosome array is based on the number of watersheds. Every DNA will contain the values of a randomly picked type of BMPs of the five available types. Refer to the chromosome figure below.

| $W_1$ | $W_2$ | $W_3$ | $W_4$ | $W_5$ | … | $W_{n-1}$ | $W_n$ |
|---|---|---|---|---|---|---|---|
| $B_p$ | $B_p$ | $B_p$ | $B_p$ | $B_p$ | | $B_p$ | $B_p$ |

Figure 3.1 Representation of single chromosome.

Where $p$ represents randomly picked BMPs type chosen from available set of provided BMPs, which is {A, B, C, … , Z } and $n$ is index number of watershed.

Therefore the chromosome can be written as

$$C_i = \{D_1^i,\ D_2^i,\ D_3^i, \ldots\ D_n^i\}$$

where $i$ = chromosome population index and $D^i$ is the DNA registered within the chromosome array.

### 3.3.2 Objective function

In any optimization problem, the formulation of the objective function is very crucial to the outcome of the optimization.

In this study, the fitness function is the same as the objective function. The objective functions used in this study were primarily of two kinds. One kind was designed to maximize pollutant reduction at the watershed outlet. The second function type was designed to minimize construction cost of BMPs.

Pollutant reduction performance of each BMPs can be simply estimated by multiplying by (1-PR) to pollutant load, where the PR (pollutant removal rate) are given in Table 2-3. The form of equation used in this study is

Objective function1: Total pollutant reduction

$$f(x) = \sum TP_i * (1 - (w_{1i} \cdot PR_i)) \qquad (3.1)$$

Where:

TP: Total Pollutant (kg).

PR: Pollutant Removal rate of BMP.

$w_{1i}$: Weighting penalty factor (Penalty of satisfying desired water quality)

Objective function2: Total cost (Construction cost + 1year maintenance cost)

$$g(x) = \sum w_{2i}(Cost_i * Area_i) \qquad (3.2)$$

Where:

Cost: Construction and OMR cost of BMPs

$w_{2i}$: Weighting penalty factor (Penalty of exceeding desired cost)

The ultimate objective function can be written as

$$Max\ f(x) + Min\ g(x) \quad (3.3)$$

This simulation model does not consider watershed soil type, watershed slope and sediment loads.

### 3.3.3 Fitness Value

Five kinds of BMPs are being considered in this study: wet ponds, infiltration, bio-filter, sand filter and bio swale. Each of them has a different construction cost function, OMR cost, and pollutant removal rate.

Let the fitness value for each chromosome be $F_j$. Hence, the fitness value for each chromosome can be calculated as:

$$F_j = \sum_{j=1}^{n} (Rm,_{j,pol1} + Rm,_{j,pol2} + Rm,_{j,pol3}) + \sum_{j=1}^{n} (Cost,_j) \quad (3.4)$$

Where, Rm : Remaining pollutant (kg), Cost : Total cost of construction and 1 year OMR

In other words, for this simulation, the lower the fitness value, the better solution set for the chromosome.

### 3.4 Program Implement

This optimization model was developed using the Microsoft® C# language. This model user interface consists of four parts. This model is easy to use and support all user input parameters.

1. Watershed input window.

   Users can open an existing Microsoft excel file to provide watershed data. Or users can type in data directly from user interface.

2. BMPs input window.

   Users can type in each BMPs properties (pollutant removal rate, construction and OMR cost) directly from user interface.

3. Graph and time window.

   Graph shows average and the best fitness value of current chromosome. Time window shows calculation time of simulation.

4. Genetic algorithm setting

   This window supports all GA operators. Users can type in all GA settings (population, crossover rate, mutation rate, overlapping rate and iteration number) directly from interface.

5. Result window.

   This window shows the selected BMP type for each watershed, removed pollutant loading and total cost of selected BMPs. The result can be saved to excel file.

### 3.4.1 Input data

The pollution loading and area of watersheds were adapted from Proposition O report (Stenstrom et al, 2007). In this report, Geographical Information System (GIS) was employed to identify the areas generating high pollutant loads. Storm-water pollutant loads generated from the drainage areas of the project sites were estimated using the empirical models described in the previous chapter for selected water quality parameters such as total suspended solids (TSS), total nitrogen (TN), and total phosphorous (TP). The input data are in Table 3-2, and Table 3-3. The output of the Proposition O report is the percentage of pollutant loads removed from the project site by the series of BMPs. The Proposition O projects were grouped for their location in each watershed: Ballona Creek, Los Angeles River, and Dominguez Channel. Finally, the cost-effectiveness of the projects was evaluated on both the cost per unit of drainage area treated and the cost per the unit mass of pollutant removed.

Table 3-2. Watershed area and pollutant loading.

| Name of watershed | Area (Acre) | TN (kg) | TP (kg) | TSS (kg) |
|---|---|---|---|---|
| La Cienega/Fairfax Powerline BMPs (#20b) | 5000 | 14070 | 1351 | 386368 |
| Mar Vista Recreation Center BMPs (#20c) | 243 | 542 | 63 | 14636 |
| Temescal Recreation Center BMPs (#22e) | 1600 | 1130 | 80 | 75215 |
| Westchester/LAX Storm-water BMPs (#22f) | 2080 | 3673 | 453 | 111937 |
| Penmar Water Quality Improvement (#22g) | 1470 | 2469 | 266 | 69057 |
| Los Angeles Zoo Parking Lot | 33 | 81 | 14 | 2308 |
| Strathern Pit Multiuse Project (#10) | 1370 | 4044 | 441 | 220659 |
| Cabrito Paseo Walkway/Bike Path (#12) | 19 | 45 | 5 | 1352 |
| Hansen Dam Recreational Area Parking Lot | 74 | 54 | 4 | 3891 |
| South Los Angeles Wetlands Park (#16) | 500 | 1442 | 124 | 34019 |
| Aliso Wash-Limekiln Creek Restoration (#23) | 11830 | 18858 | 1848 | 732153 |
| Oros Streetend Biofiltration Project (#28) | 8.4 | 17.4 | 2 | 767 |
| Echo Park Lake Rehabilitation Project (#29) | 732 | 1848 | 153 | 39289 |
| Parking Grove in El Sereno Project (#31) | 3.3 | 1.17 | 0.1 | 156 |
| Rosecrans Recreation Center Project (#35) | 13 | 8.19 | 0.52 | 580 |
| Machado Lake Rehabilitation (#36) | 14820 | 26764 | 2640 | 823275 |
| Peck Park Canyon Project (#40) | 100 | 149 | 15 | 4126 |

Adapted from Proposition O report (Stenstrom, 2007)

Table 3-3 The input data of BMPs property

| | Construction cost | O&M | TSS (Remov. %) | TN (Remov. %) | TP (Remov. %) |
|---|---|---|---|---|---|
| Wet ponds | $2,622/acre | 4.5% | 75 | 28 | 46 |
| Inflitration system | $7,866/acre | 10.5% | 80 | 60 | 55 |
| Biofilter | $15,733/acre | 6% | 79 | 41 | 59 |
| Bioswale | $10,890/acre | 5% | 80 | 70 | 35 |
| Sandfilter | $14,670/acre | 12% | 80 | 30 | 33 |

There are many different construction and OMR costs as well as pollutant removal rates for the BMPs. The code allows the user to input these parameters.

### 3.4.2 Implementation of the GA

The initial population consists of randomly generated chromosomes. Each chromosome represents a set of BMPs placement strategy. Because there are 17 sets of input watershed data, each chromosome has 17 DNAs. Each DNA holds characteristics (pollutant removal rate and cost) for a randomly picked single BMP. The initial size of the population is set at one hundred. The model will calculate minimum remaining pollutant and then calculate each population's minimum cost and then rank the entire chromosome by fitness values. The less the cost and the remaining pollutant, the less fitness value it will have. At each generation, new individuals are added to the population through crossover, mutation and overlapping, while parents individuals with low fitness values are discarded from the population.

**1). Crossover operation:**

Crossover operation is used to create new chromosomes for the next generation by randomly combining two selected chromosomes from the current generation through the selection process. This model uses an elitist selection strategy, which ensures the fittest chromosome from one generation is propagated into the next generation without any disturbance. The crossover rate is the user-defined probability that crossover reproduction will be performed. For example, a crossover rate of 0.6 means that on average 60% of the population undergoes the crossover operation. There are several crossover methods available, like single point, multi-point, uniform crossover, etc. This model used a uniform crossover method. In this model, this operation exchange the BMP type set in each chromosome.

- Assume $a$ and $b$ are 2 randomly generated numbers from the range (0, $n$), where $a<b$. $n$ represents number of watersheds.
- Assume $r$ and $s$ is randomly selected chromosome indices. In order to select a better fitness value of chromosome to perform crossover, $r$ and $s$ will be randomly selected from the range of (0, *population/3*). It can be written as $0 \le r \le P/3$ and $0 \le s \le P/3$.
- Let $C_r$ and $C_s$ be the identified chromosome, where $r$ and $s$ range from (0, P/3)
- When crossover is performed, the resulting chromosome becomes:

$$C_r = \{D_1^r, D_2^r, D_a^s, \cdots D_b^s, \cdots, D_{n-1}^r, D_n^r\}$$

$$C_s = \{D_1^s, D_2^s, D_a^r, \cdots D_b^r, \cdots, D_{n-1}^s, D_n^s\}$$

**2). Mutation operation:**

For each chromosome generated by the crossover operation, a mutation operation is applied using a predefined user's input. Mutation prevents the population from becoming saturated with chromosomes that are all similar and reduces the chance of premature convergence. High mutation rates increase the probability of destroying good chromosomes. If the population size is 500, the chromosome length is 17 and the mutation rate is 0.05, on average 425 DNA positions will alter in the whole population (500x17x0.05).

- Assume $a$ and $b$ be 2 randomly generated numbers from the range (0, $n$), where $a<b$. $n$ represents number of watersheds.

- Note that mutation on a certain DNA location in chromosome mean changing BMP in use. The mutation happen rate is based on simulation setting from GA property.
- Let $r$ and $s$ be randomly selected chromosome indices. In order to select a better fitness value of chromosome to perform crossover, $r$ and $s$ will be randomly selected from the range of (0, *population/3*). It can be written as $0 \le r \le P/3$ and $0 \le s \le P/3$.
- In order to beteter offspring chromosomes, select 2 chromosomes, where index number is from the range (0, P/3)
- After the chromosome set has been sorted, chromosome located at the end of the array list will contain larger fitness value.

**3). Overlapping operation**

- Overlapping operation is to produce a new chromosome by inheriting the same DNA from two parents' chromosome and randomly placed DNA for the unmatched DNA.
- The newly produced chromosome will replace its parent if the fitness value is better than its parent.
- In order to beteter offspring chromosomes, select 2 chromosomes, where index number is from the range (0, P/3)
- Let $C_r$ and $C_s$ is the identified chromosome, where $r$ and $s$ is from range (0, P/3)
- After overlapping is performed, the following chromosome resuls:

$$C_r = \{D_1^r, D_2^r, \cdots, D_{n-1}^r, D_n^r\}$$

$$C_s = \{D_1^s, D_2^s, \cdots, D_{n-1}^s, D_n^s\}$$

- Let $C_y$ be the child chromosome produced by overlapping from $C_r$ and $C_s$. $D_1^y$ is assigned with a new BMP type where BMP in used in $D_1^r$ is not same as $D_1^s$.

User interface looks like next page.

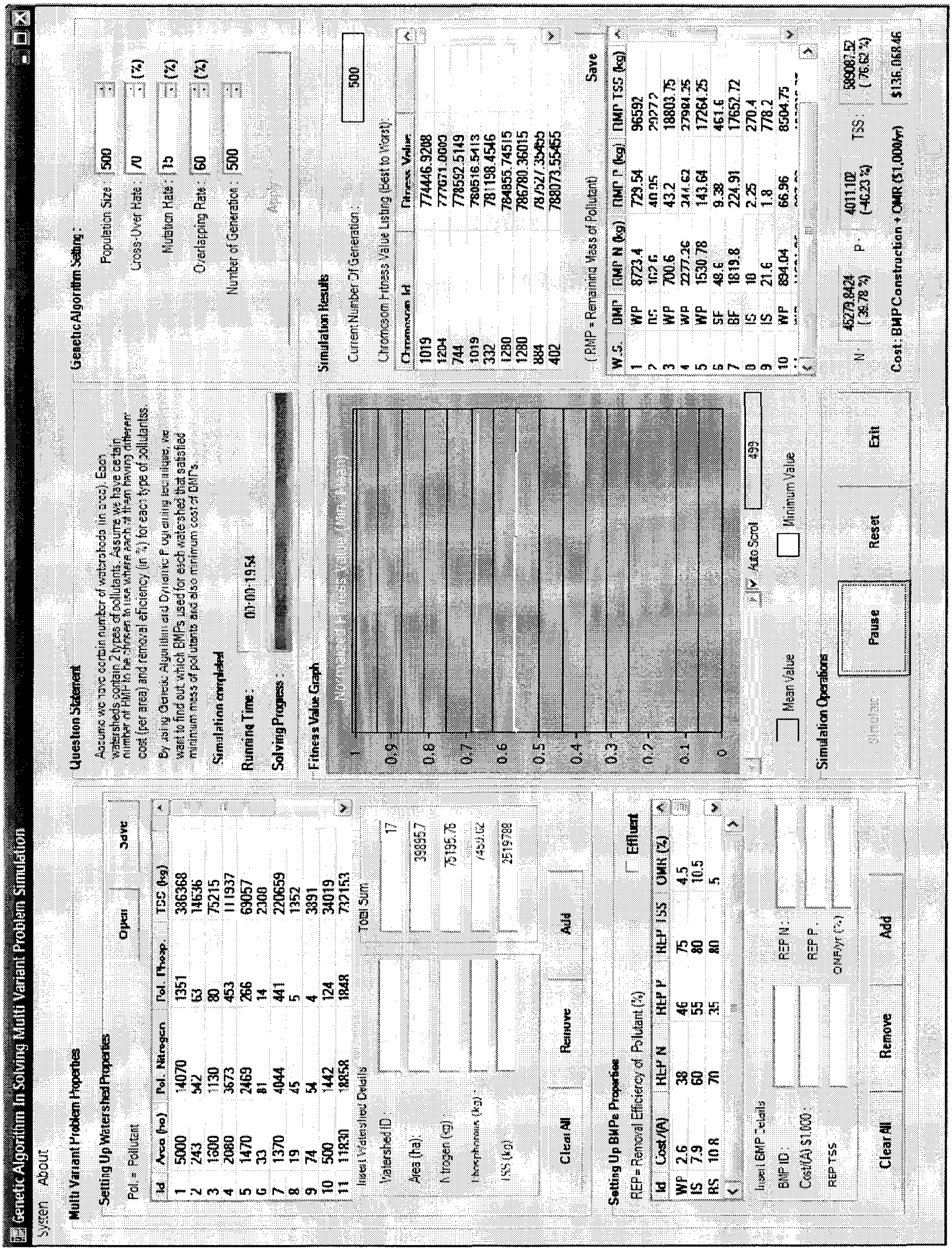


Figure 3.2 GA simulation user interface

# 4. Results

## 4-1 Parameter evaluation

In this part of the study, the effect of number of populations and generations were first investigated for the Prop. O GA simulation. The optimum population size and the number of generations were then selected. The population size and the number of generations are related by the total number of simulations in one GA run; therefore these two were studied together.

Several population sizes were considered to investigate improved convergence. Population sizes of 50, 100, 300, 500 and 1000 were initially investigated while keeping the number of generations at 300. The simulation was repeated 10 times for each parameter condition, and the results were averaged. Then optimum population size and the number of generations were selected from these simulation runs. The remaining GA parameters were as follows:

Crossover rate: 60%

Mutation rate: 5%

Overlapping rate: 60%

Number of generation: 300 times

The calculation time increased with an increase with increasing population size. Similar results were obtained for various population sizes in this study; however, better minimum fitness value and cost were obtained when the population was one hundred cases. The larger the population size, the longer the calculation time required for convergence. Table 4-1 and Figure 4.1 show the simulation results as a function of different number of population sizes. The large variability in fitness value at the beginning of each simulation occurs because the chromosome is random. After 300 generations, chromosomes are selected based on the fitness value, and the variability decreases. This pattern occurs in the other sensitivity graphs as well.

Table 4-1 Model parameters comparison with different population sizes

| **Population** | Cost ($1,000) | Pollutant reduction l(%) | Mean Fitness Value | Minimum Fitness Value | Calculation Time (s) |
|---|---|---|---|---|---|
| 50 | 135 | 54.1 | 883883 | 771131 | 8 |
| **100** | 128 | 53.9 | 894855 | 770014 | 8.7 |
| 300 | 155 | 54.5 | 926974 | 780511 | 9.7 |
| 500 | 132 | 54.2 | 929897 | 771435 | 12 |
| 1000 | 222 | 57.4 | 941123 | 782632 | 22.1 |

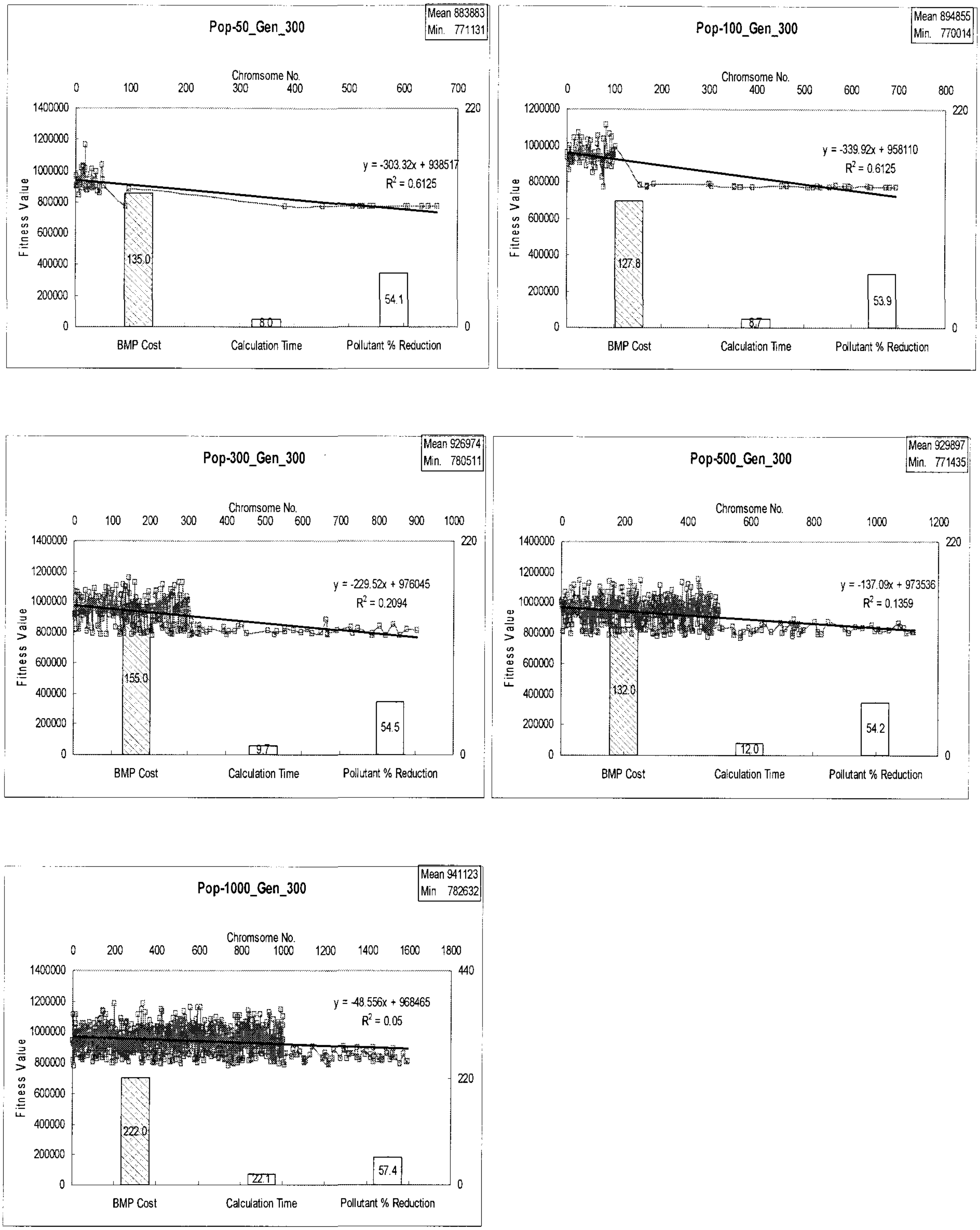


Figure 4.1 Number of population vs. parameter sets for Proposition O data

Different numbers of generation were considered to investigate better convergence. Generations of 50, 100, 300, 500, 1000 and 1500 were used. The following GA parameters were used:.

Population: 100

Crossover rate: 60%

Mutation rate: 5%

Overlapping rate: 60%

Table 4-2 and Figure 4.2 show the simulation results as a function of the generation number. Table shows a slightly better minimum fitness value occurred when applying 300 generation numbers.

Table 4-2 Model parameters comparison with different numbers of generation

| **Generation** | Cost ($1,000) | Pollutant reduction (%) | Mean Fitness Value | Minimum Fitness Value | Calculation Time (s) |
|---|---|---|---|---|---|
| 50 | 148 | 54.3 | 934166 | 778738 | 1.5 |
| 100 | 139 | 54.3 | 916740 | 772934 | 3.3 |
| **300** | 128 | 53.9 | 894855 | 770014 | 8.7 |
| 500 | 128 | 53.9 | 878357 | 769885 | 14.3 |
| 1000 | 128 | 53.9 | 866630 | 769841 | 27.8 |
| 2000 | 128 | 53.9 | 858641 | 769877 | 38.8 |

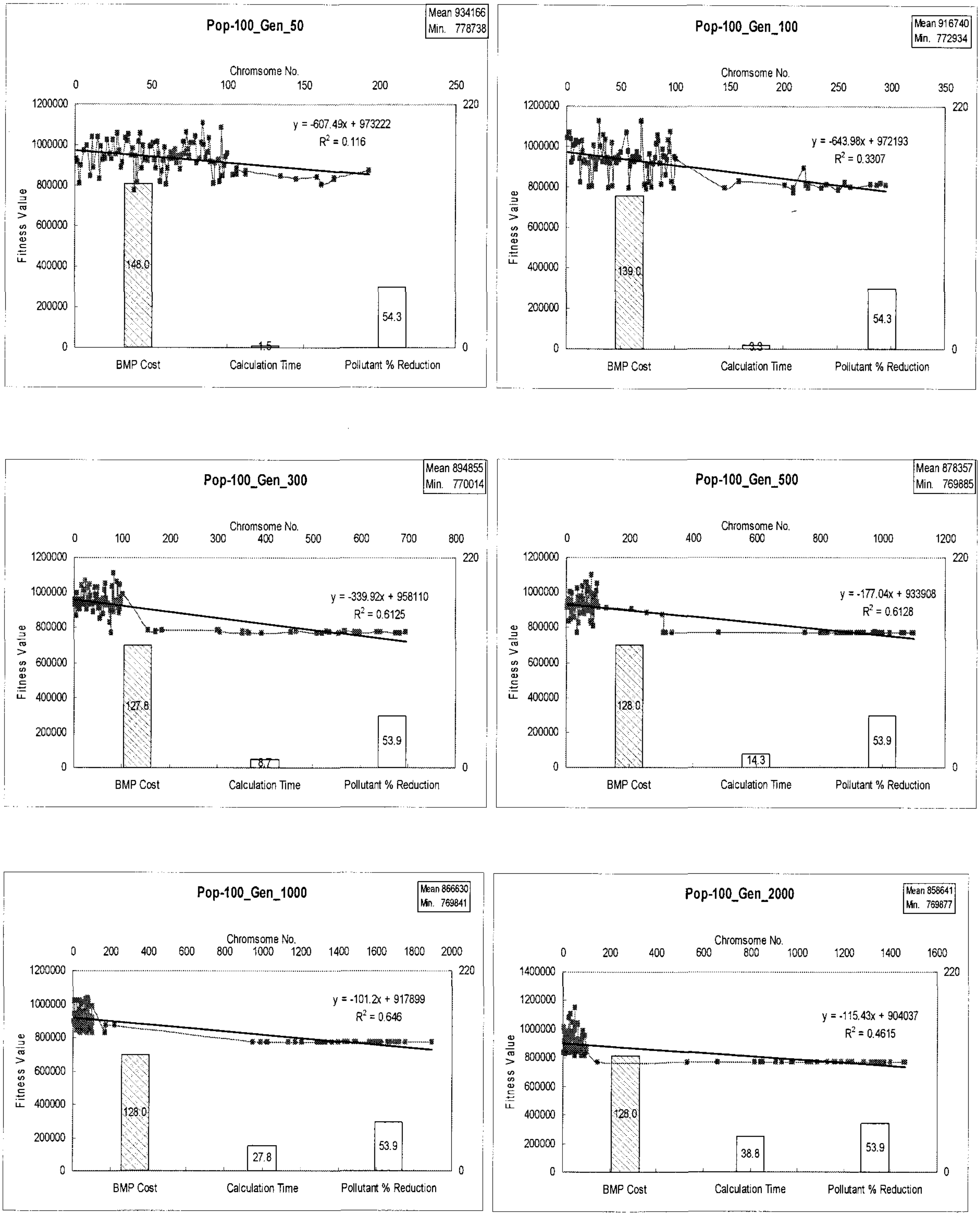


Figure 4.2 Number of generations vs. parameter sets for Proposition O

Based on the results, a population size of 100 and 300 generations were considered as the optimum population size and number of generations. These were used for the rest of the GA operator study.

The effects of crossover rate were investigated for this study and they ranged from 40% to 80%, in increments of 10%. The following GA parameters were maintained constant:

Population: 100

Mutation rate: 5%

Overlapping rate: 60%

Number of generation: 300 times.

Table 4-3 and Figure 4.3 show the simulation results as a function of different crossover rates. The poorest performance occurred with 40% crossover rate while crossover rates of 50, 60, 70 and 80% showed similar results. The crossover rate has less effect on results as compared to other GA parameters. The 60% crossover rate gave the best mean fitness value and the shortest calculation time. Therefore, 60% crossover rate was identified as the optimum rate for this simulation.

Table 4-3 Model parameters comparison with different crossover rates

| **Crossover (%)** | Cost ($1,000) | Pollutant reduction (%) | Mean Fitness Value | Minimum Fitness Value | Calculation Time (s) |
|---|---|---|---|---|---|
| 40 | 139 | 54.4 | 901304 | 773537 | 10.6 |
| 50 | 128 | 53.9 | 894640 | 770247 | 8.9 |
| **60** | 129 | 53.9 | 893778 | 770918 | 7.9 |
| 70 | 129 | 53.9 | 899220 | 770615 | 8.8 |
| 80 | 128 | 53.9 | 895022 | 769989 | 8.8 |

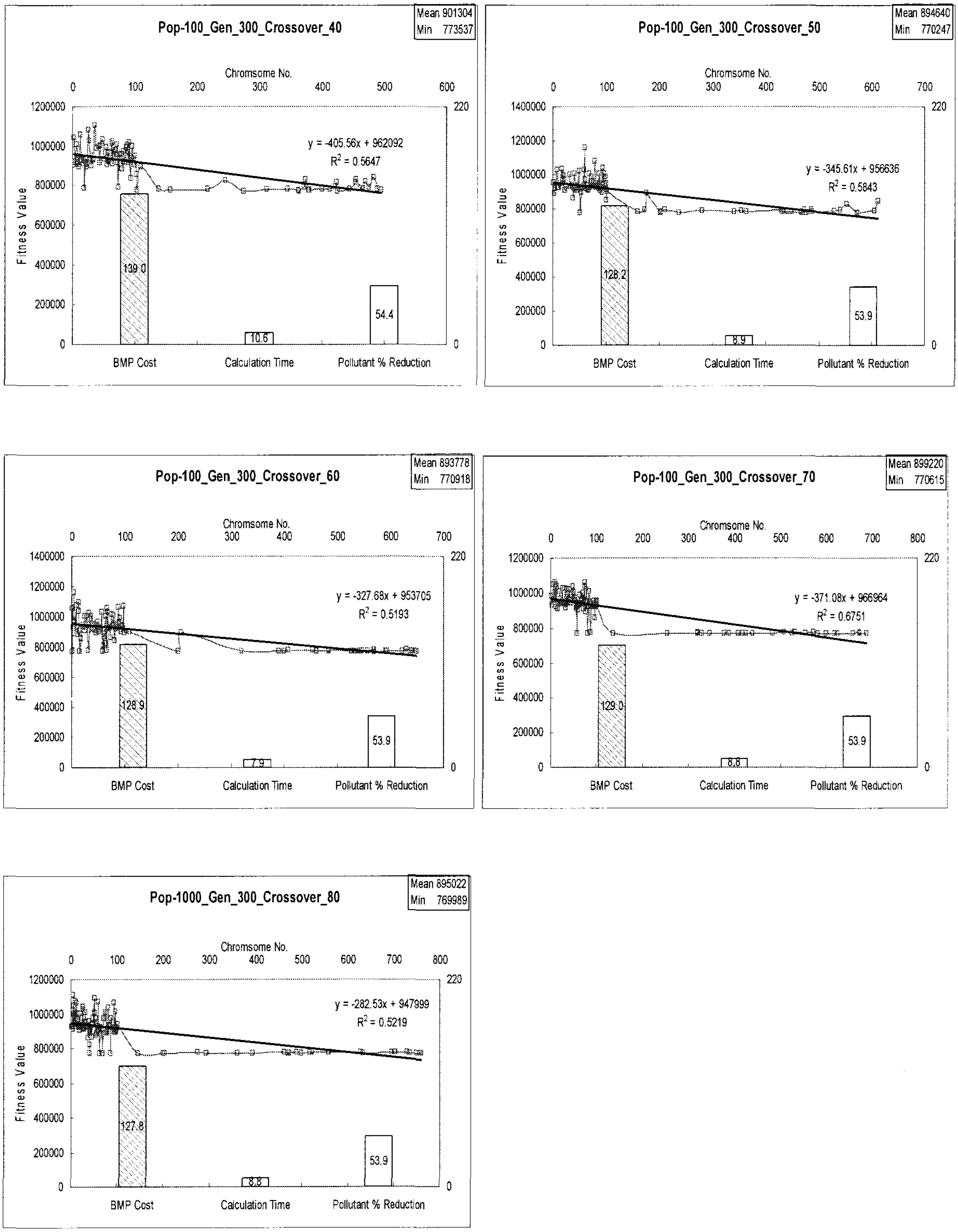


Figure 4.3 Different crossover rates vs. parameter sets for Proposition O

The effect of different mutation rates on convergence were investigated by varying the mutation rate from 1% to 15%. The other GA parameters were maintained constant as shown below.

Population: 100

Crossover rate: 60%

Overlapping rate: 60%

Number of generation: 300 times.

Table 4-4 and Figure 4.4 show the simulation results for different mutation rates.

Almost identical results were obtained for different mutation rates. The mutation rate has less effect on result compared to other GA parameters, and a mutation rate of 5% showed the best mean and minimum fitness values. Therefore, 5% mutation rate was identified as the optimum rate for this simulation in terms of cost.

Table 4-4 Model parameters comparison with different mutation rates

| **Mutation (%)** | Cost ($1,000) | Pollutant reduction (%) | Mean Fitness Value | Minimum Fitness Value | Calculation Time (s) |
|---|---|---|---|---|---|
| 1 | 128 | 53.9 | 914887 | 771008 | 8.8 |
| 3 | 128 | 53.9 | 917322 | 770400 | 8.7 |
| **5** | 129 | 53.9 | 909278 | 770213 | 8.5 |
| 10 | 128 | 53.8 | 919203 | 770376 | 9 |
| 15 | 128 | 53.9 | 917139 | 770879 | 8.8 |

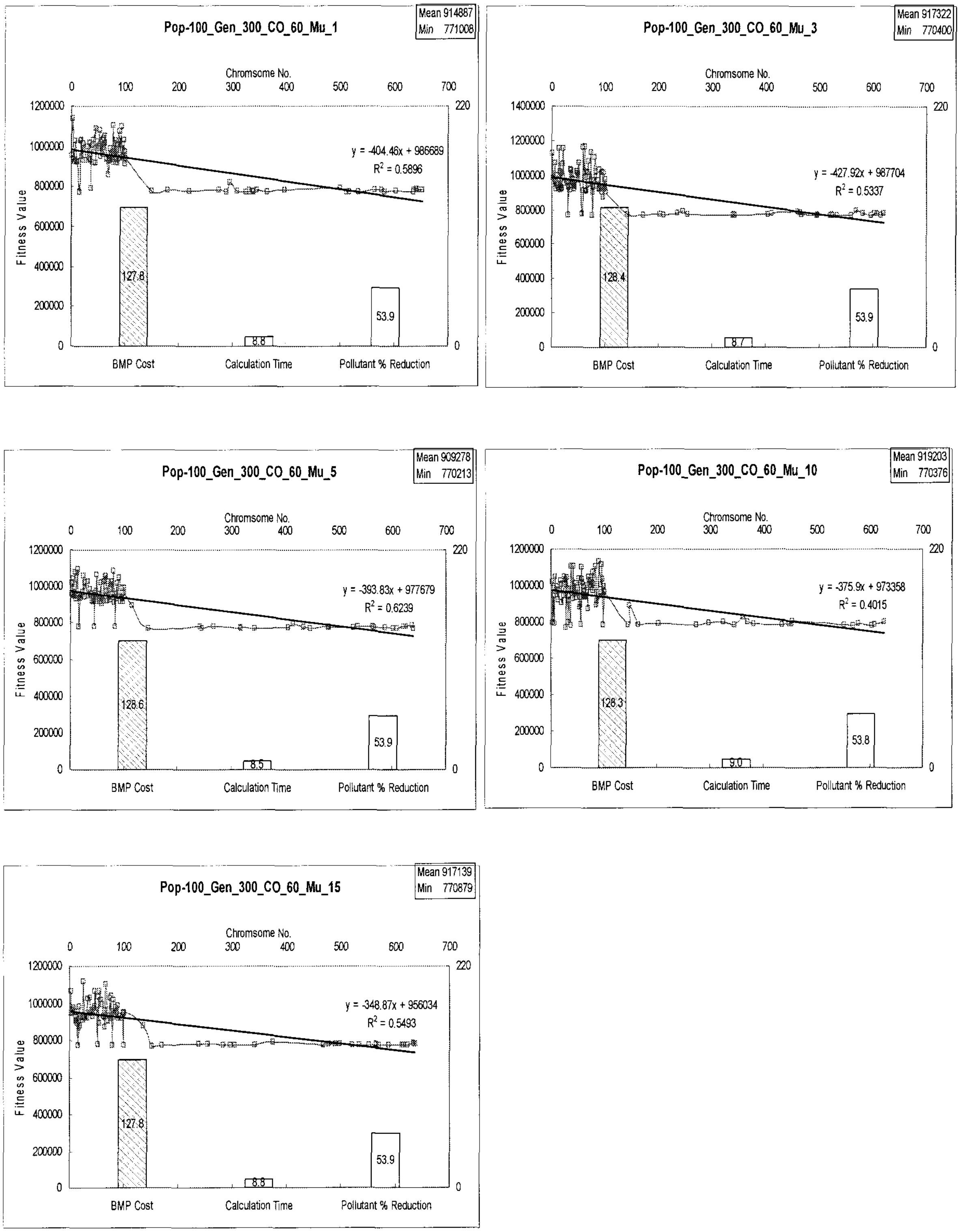


Figure 4.4 Different mutation rates vs. parameter sets for Proposition O

The effect of overlapping rates was investigated by varying the rate from 30% to 80% , in increments of 10%. While keeping the other GA parameters constant at the values shown below:

Population: 100

Crossover rate: 60%

Mutation rate: 5%

Number of generation: 300 times.

Table 4-5 and figure 4.5 show the simulation results against different overlapping rates. Convergence was nearly independent on overlapping rate. An overlapping rate of 60% had slightly lower mean fitness and was selected.

Table 4-5 Model parameters comparison with different overlapping rates

| **Overlapping (%)** | Cost ($1,000) | Pollutant reduction (%) | Mean Fitness Value | Minimum Fitness Value | Calculation Time (s) |
|---|---|---|---|---|---|
| 30 | 137 | 54.4 | 911140 | 773914 | 8.4 |
| 40 | 130 | 54 | 909666 | 771097 | 8.7 |
| 50 | 129 | 53.9 | 910813 | 771413 | 8.5 |
| **60** | 129 | 53.9 | 909278 | 770213 | 8.5 |
| 70 | 131 | 54 | 911634 | 771294 | 8.2 |
| 80 | 129 | 53.9 | 905526 | 770301 | 8.5 |

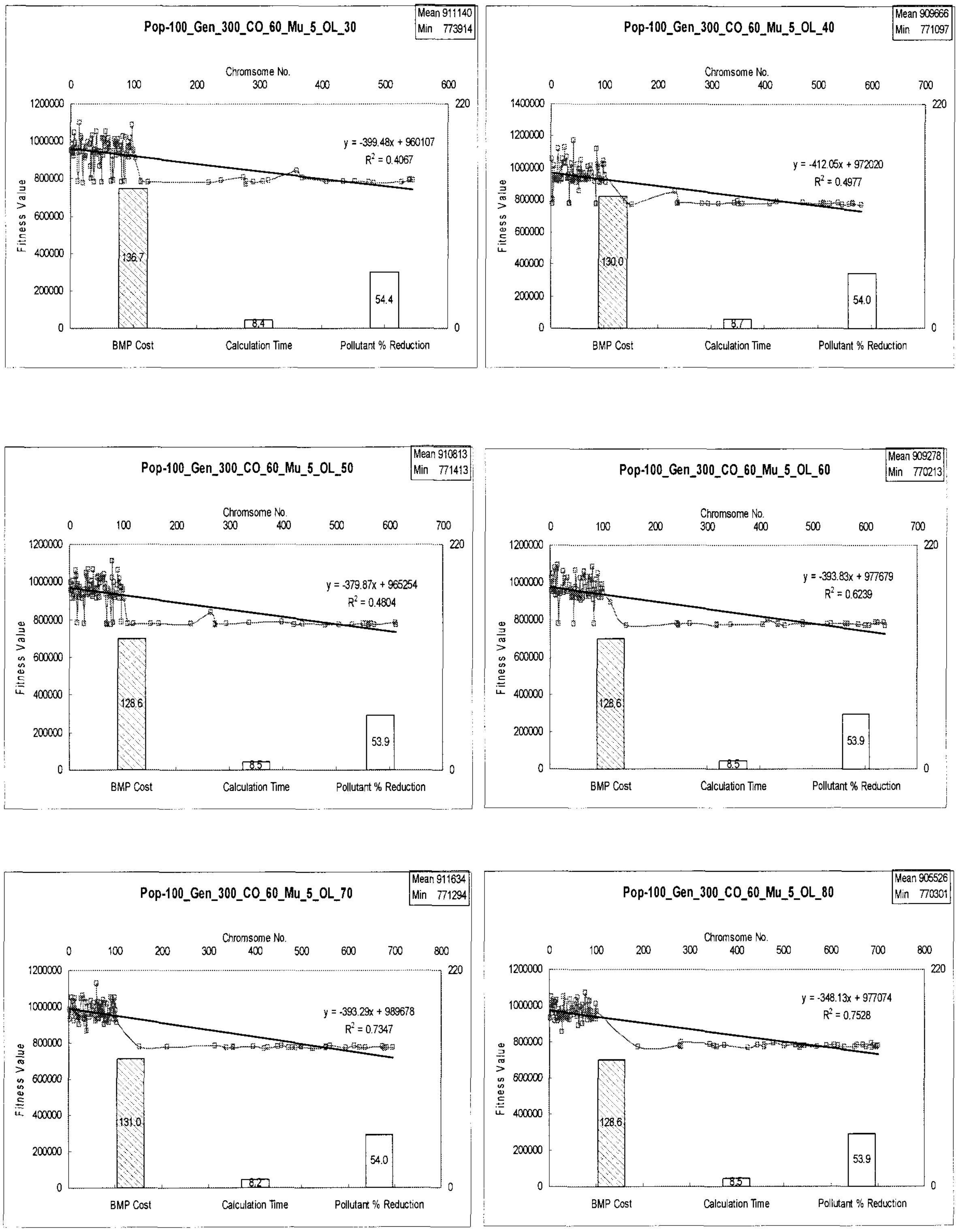


Figure 4.5 Different overlapping rates vs. parameter sets for Proposition O

Based on these results, it can be concluded that crossover, mutation and overlapping rates do not significantly affect the convergence for this simulation. A population size of 100, crossover rate of 60%, mutation rate of 5%, overlapping rate of 60% and 300 generations gave the best results in terms of minimum fitness value, and selected for the Proposition O simulation case. These values are typical of the values found by others (Goldberg, 1989; Holland, 1975).

## 4-2 Simulation results

Proposition O was passed to help the City of Los Angeles comply with the TMDL requirements of the Clean Water Act. Table 4-6 shows initial BMPs, total cost and pollutant reduction selected by Proposition O managers (Stenstrom, 2007). As expected, the total costs of the watersheds with the larger drainage areas were greater than the costs of those with smaller drainage areas. For example, the cost for the larger drainage areas (La Cienega, Aliso Wash and Machado) corresponded to 36% of cost of the entire BMPs cost. The smaller drainage area projects are often local and decentralized and their small scale prevents them from having a serious impact on receiving waters. They will need to

Table 4-6 Proposition O project result from report (Stenstrom et al., 2007)

| Name of watershed | BMP Applied | TSS reduction (%) | TN reduction (%) | TP reduction (%) | Total Cost/ watershed area | Total Cost($) |
|---|---|---|---|---|---|---|
| **La Cienega/Fairfax BMPs** | BF | 82 | 63 | 70 | $1,534 | $7,667,888 |
| Mar Vista Recreation BMPs | BF | 85 | 90 | 87 | $18,750 | $4,556,186 |
| Temescal Recreation BMPs | BF | 99 | 80 | 75 | $11,654 | $18,646,000 |
| Westchester/LAX BMPs | IS | NA | NA | NA | $692 | $1,438,755 |
| Penmar Water Improvement | BF | NA | NA | NA | $16,044 | $23,585,000 |
| Los Angeles Zoo Parking Lot | SF | 90 | 68 | 70 | $421,341 | $13,904,242 |
| Strathern Pit Multiuse Project | WP | 65 | 37 | 29 | $16,427 | $22,505,000 |
| Cabrito Paseo Walkway | BS | 64 | 25 | 47 | $234,895 | $4,463,009 |
| Hansen Dam Parking Lot | BS | 82 | 50 | 58 | $30,009 | $2,220,702 |
| South LA Wetlands Park | WP | 65 | 35 | 29 | $26,760 | $13,380,243 |
| **Aliso Wash-Limekiln Creek** | BS | 82 | 63 | 70 | $921 | $10,893,483 |
| Oros Streetend Biofiltration | BF | 77 | 63 | 53 | $121,581 | $972,651 |
| Echo Park Rehabilitation | IS | 92 | 50 | 60 | $115,114 | $84,263,313 |
| Parking Grove in El Sereno | BF | 87 | 86 | 53 | $1,328,212 | $3,984,635 |
| Rosecrans Recreation Center | BS | 91 | 68 | 77 | $519,541 | $6,754,033 |
| **Machado Rehabilitation** | BS | 93 | 26 | 60 | $6,716 | $99,523,897 |
| Peck Park Canyon Project | IS | 87 | 70 | 64 | $61,900 | $6,190,000 |
| **Total** | | **80** | **54** | **55** | | **$324,949,037** |

WP : Wet ponds, IS : Infiltration system, BF : Biofilter, SF : Sandfilter, BS : Bioswale
(Three largest drainage area projects in bold.)

be scaled-up to have the necessary impact on TMDL compliance.

Table 4-7 shows result from this GA simulation. The pollutant reduction rates for total suspended solids were quite similar with Proposition O's result, within the precision of BMP performance estimates. However, the pollutant reduction rates for total phosphorous and total nitrogen are lower than Proposition O's result. This is because the Proposition O selection process assumed the best reported TN and TP reduction rates among the various result, while, the GA simulation used average reported TN and TP reduction rates of each BMP properties. Total cost is only 158 million dollars compared to Prop. O's 325 million dollars.

The GA simulation results were similar to the manager-selected results with respect to costs among the different projects. The total costs of watersheds with larger drainage areas were greater than the total costs of smaller drainage areas. For example, the cost for the three largest drainage areas (La Cienega, Aliso Wash and Machado) corresponded to 54% of cost from the entire BMPs cost.

The GA optimization selected more wet ponds (11) than were selected by the Prop O managers (2). The GA technique selected wet ponds due to low cost and did not address land constraints as the Prop O managers addressed land constraints.

Table 4-7 Result from this GA simulation with removal rates

| Name of watershed | Selected BMP | TSS reduction (%) | TN reduction (%) | TP reduction (%) | Cost/ area | Total Cost ($1,000) |
|---|---|---|---|---|---|---|
| **La Cienega/Fairfax BMPs** | WP | 75 | 38 | 46 | $2,717 | $13,585 |
| Mar Vista Recreation BMPs | WP | 75 | 38 | 46 | $2,717 | $660 |
| Temescal Recreation BMPs | BS | 80 | 70 | 35 | $11,340 | $18,144 |
| Westchester/LAX BMPs | WP | 75 | 38 | 46 | $2,717 | $5,651 |
| Penmar Water Improvement | IS | 80 | 60 | 55 | $8,730 | $12,832 |
| Los Angeles Zoo Parking Lot | WP | 75 | 38 | 46 | $2,717 | $90 |
| Strathern Pit Multiuse Project | BF | 92 | 55 | 49 | $16,642 | $22,799 |
| Cabrito Paseo Walkway | WP | 75 | 38 | 46 | $2,717 | $52 |
| Hansen Dam Parking Lot | WP | 75 | 38 | 46 | $2,717 | $201 |
| South LA Wetlands Park | WP | 75 | 38 | 46 | $2,717 | $1,359 |
| **Aliso Wash-Limekiln Creek** | WP | 75 | 38 | 46 | $2,717 | $32,142 |
| Oros Streetend Biofiltration | WP | 75 | 38 | 46 | $2,850 | $23 |
| Echo Park Rehabilitation | BS | 80 | 70 | 35 | $11,340 | $8,301 |
| Parking Grove in El Sereno | WP | 75 | 38 | 46 | $2,989 | $9 |
| Rosecrans Recreation Center | BF | 92 | 55 | 49 | $16,642 | $216 |
| **Machado Rehabilitation** | WP | 75 | 38 | 46 | $2,717 | $40,266 |
| Peck Park Canyon Project | BF | 92 | 55 | 49 | $16,642 | $1,664 |
| **Total** | | 78.9 | 46.1 | 45.8 | | **$157,995** |

WP : Wet ponds, IS : Infiltration system, BF : Biofilter, SF : Sandfilter, BS : Bioswale (Three largest drainage area projects in bold.)

Table 4-8 shows the result of GA simulation when constrained to wet ponds as the only BMP. Total cost 109 million dollar is slightly lower than 157 million dollars. TSS and TP reductions are greater while TN reduction is 8% lower than the optimal result using five different BMPs (Table 4-7). If all watersheds had sufficient area and slope for wet ponds, using wet ponds for all watersheds could be a good scenario.

Table 4-8 Result from this GA simulation using only Wet Ponds

| **Name of watershed** | Selected BMP | Remaining TSS (kg | Remaining TN (kg) | Remaining TP (kg | **Total Cost ($1,000)** |
|---|---|---|---|---|---|
| **La Cienega/Fairfax BMPs** | WP | 96592 | 8723 | 729.5 | $ 13,715 |
| Mar Vista Recreation BMPs | WP | 3659 | 336 | 34.0 | $ 667 |
| Temescal Recreation BMPs | WP | 18803 | 700 | 43.2 | $ 4,389 |
| Westchester/LAX BMPs | WP | 27984 | 2277 | 244.6 | $ 5,705 |
| Penmar Water Improvement | WP | 17264 | 1530 | 143.6 | $ 4,032 |
| Los Angeles Zoo Parking Lot | WP | 577 | 50 | 7.5 | $ 91 |
| Strathern Pit Project | WP | 55164 | 2507 | 238.1 | $ 3,758 |
| Cabrito Paseo Walkway | WP | 338 | 27 | 2.7 | $ 52 |
| Hansen Dam Parking Lot | WP | 972 | 33 | 2.1 | $ 203 |
| South LA Wetlands Park | WP | 8504 | 894 | 66.9 | $ 1,372 |
| **Aliso Wash-Limekiln Creek** | WP | 183038 | 11691 | 997.9 | $ 32,450 |
| Oros Streetend Biofiltration | WP | 191 | 10 | 1.1 | $ 23 |
| Echo Park Rehabilitation | WP | 9822 | 1145 | 82.6 | $ 2,008 |
| Parking Grove in El Sereno | WP | 39 | 0.72 | 0.05 | $ 9 |
| Rosecrans Recreation Center | WP | 145 | 5.07 | 0.28 | $ 36 |
| **Machado Rehabilitation** | WP | 205818 | 16593 | 1425.6 | $ 40,651 |
| Peck Park Canyon Project | WP | 1031.5 | 92 | 8.1 | $ 274 |
| **Total** | | 75% | 38% | 46% | $109,430 |

Table 4-9 shows result of applying effluent concentrations of BMPs from Table 2-5, as opposed to pollutant removal rates of each BMP, it shows much approved pollutant reduction as well as total cost. With this simulation, only 9 wet ponds are selected for best solution. Unfortunately, only a limited number BMPs are sufficiently well documented to use this approach.

Table 4-9 Result from this GA simulation with effluent concentration

| **Name of watershed** | Selected BMP | Remaining TSS (kg) | Remaining TN (kg) | Remaining TP (kg | **Total Cost ($1,000)** |
|---|---|---|---|---|---|
| **La Cienega/Fairfax BMPs** | WP | 8.5 | 2 | 20 | 13,585 |
| Mar Vista Recreation BMPs | BF | 12 | 4 | 32 | 4,044 |
| Temescal Recreation BMPs | IS | 10 | 3 | 27 | 13,967 |
| Westchester/LAX BMPs | WP | 8.5 | 2 | 20 | 5,651 |
| Penmar Water Improvement | WP | 8.5 | 2 | 20 | 3,994 |
| Los Angeles Zoo Parking Lot | SF | 7 | 1.5 | 16 | 540 |
| Strathern Pit Multiuse Project | WP | 8.5 | 2 | 20 | 3,722 |
| Cabrito Paseo Walkway | BF | 12 | 4 | 32 | 316 |
| Hansen Dam Parking Lot | IS | 10 | 3 | 27 | 646 |
| South LA Wetlands Park | WP | 8.5 | 2 | 20 | 1,359 |
| **Aliso Wash-Limekiln Creek** | WP | 8.5 | 2 | 20 | 32,142 |
| Oros Streetend Biofiltration | BF | 12 | 4 | 32 | 140 |
| Echo Park Lake Rehabilitation | WP | 8.5 | 2 | 20 | 1,989 |
| Parking Grove in El Sereno | IS | 10 | 3 | 27 | 29 |
| Rosecrans Recreation Center | SF | 7 | 1.5 | 16 | 213 |
| **Machado Lake Rehabilitation** | WP | 8.5 | 2 | 20 | 40,266 |
| Peck Park Canyon Project | WP | 8.5 | 2 | 20 | 272 |
| **Total** | | 99.99% | 99.81% | 99.52% | $122,873 |

(Three largest drainage area projects in bold.)

Table 4-10 Total cost comparison O project result with GA simulation

| Name of watershed | Proposition O | | GA simulation | |
|---|---|---|---|---|
| | BMP Applied | Total Cost ($1,000) | BMP Applied | Total Cost ($1,000) |
| La Cienega/Fairfax BMPs | BF | $7,668 | WP | $13,585 |
| Mar Vista Recreation BMPs | BF | $4,556 | WP | $660 |
| Temescal Recreation BMPs | BF | $18,646 | BS | $18,144 |
| Westchester/LAX BMPs | IS | $1,439 | WP | $5,651 |
| Penmar Water Improvement | BF | $23,585 | IS | $12,832 |
| Los Angeles Zoo Parking Lot | SF | $13,904 | WP | $90 |
| Strathern Pit Multiuse Project | WP | $22,505 | BF | $22,799 |
| Cabrito Paseo Walkway | BS | $4,463 | WP | $52 |
| Hansen Dam Parking Lot | BS | $2,221 | WP | $201 |
| South LA Wetlands Park | WP | $13,380 | WP | $1,359 |
| Aliso Wash-Limekiln Creek | BS | $10,893 | WP | $32,142 |
| Oros Streetend Biofiltration | BF | $972 | WP | $23 |
| Echo Park Lake Rehabilitation | IS | $84,263 | BS | $8,301 |
| Parking Grove in El Sereno | BF | $3,985 | WP | $9 |
| Rosecrans Recreation Center | BS | $6,754 | BF | $216 |
| Machado Lake Rehabilitation | BS | $99,524 | WP | $40,266 |
| Peck Park Canyon Project | IS | $6,190 | BF | $1,664 |
| **Total** | | **$324,949** | | **$157,995** |

WP : Wet ponds, IS : Infiltration system, BF : Biofilter, SF : Sandfilter, BS : Bioswale

The cost-effectiveness of projects was evaluated in two ways: 1) the total cost of a project for the drainage area it treats, and 2) total cost per the unit of pollutant load (kg) removed.

The pollutants used in the cost analysis were Total Suspended Solids, and nutrients (Total Nitrogen, Total Phosphorous). Table 4-11 shows cost effectiveness results from Proposition O project report. For TSS removal, the Aliso Wash-Limekiln Creek Confluence Restoration Project and the La Cienega/Fairfax Powerline Easement Stormwater BMPs are the most cost-effective projects that cost less than $25/TSS kg. Strathern Pit Multiuse Project, the Machado Lake Project, the Santa Monica Bay Beaches LFD upgrades, Westchester/LAX Stormwater BMP, Penmar Water Quality Improvement and Runoff Reuse Project, and South Los Angeles Wetlands Park cost less than $500/TSS kg. The Peck Park Canyon Enhancement Project is the most expensive project that costs $40,000/TSS kg. The LA Zoo Parking Lot Project, Rosecrans Recreational Center Storm Water Enhancements and Westminster Dog Park Stormwater BMPs are also expensive projects that cost more than $10,000/TSS kg.

For nutrient removal, the Aliso Wash-Limekiln Creek Confluence Restoration Project and the La Cienega/Fairfax Powerline Easement Stormwater BMPs are the most cost effective projects that cost approximately $1,000/TKN kg. Westchester/LAX Stormwater BMP, Penmar Water Quality Improvement and Runoff Reuse Project, Machado Lake Project, Santa Monica Bay Beaches LFD upgrades, and Strathern Pit Multiuse Project cost less than $25,000/TKN kg. The Peck Park Canyon Enhancement Project is the most expensive project because it costs approximately $8 million/TKN kg. The LA Zoo Parking Lot Project, Rosecrans Recreational Center Storm Water Enhancements and Westminster Dog Park Stormwater BMPs are also expensive projects that cost more than $1 million/TKN kg. The Prop. O result shows that the Aliso Wash-

Limekiln Creek Confluence Restoration Project and The Mar Vista BMP are the most cost-effective projects for total mass reduction criteria. Conversely, Rosecrans recreation center BMPs, Parking Grove in El Sereno, and La Cinega powerline BMP are the most expensive projects to reduce pollutant loading.

Table 4-12 shows the result from this GA simulation. These results are quite different than chosen by the Proposition O project results. Total cost is 158 million dollars compared to Prop. O's 325 million dollars. The result shows that the Oros Streetend Biofiltration and La Cienega/Fairfax BMPs and Cabrito Paseo Walkway are the most cost-effective projects on the basis of pollutant reduction. Conversely, Peck Park Canyon Project, Rosecrans Recreation, and Temescal Recreation BMPs are the most expensive on the basis of pollutant removal.

Both results show that the unit cost of TP reduction is much larger than unit costs of TSS and TN reduction.

Table 4-11 Results of cost effectiveness in case of Proposition O from report.

| **Watershed** | Cost($) /TSS reduction (kg) | Cost($) /TN reduction (kg) | Cost($) /TP reduction (kg) | **Cost($)/ Total mass reduction (kg)** | Cost($) /mass loads (kg) | **Total Cost ($1,000)** |
|---|---|---|---|---|---|---|
| Mar Vista BMPs | 13 | 382 | 3,747 | 13 | 299 | 4,556,186 |
| Aliso Wash Creek | 17 | 722 | 6,551 | 16 | 14 | 10,893,483 |
| Strathern Pit Project | 127 | 15,837 | 87,910 | 126 | 100 | 22,505,000 |
| Penmar Improv. | 345 | 10,750 | 118,517 | 333 | 329 | 23,585,000 |
| South LA Park | 492 | 24,916 | 185,836 | 481 | 376 | 13,380,243 |
| Hansen Parking Lot | 573 | 50,992 | 590,612 | 566 | 562 | 2,220,702 |
| Oros Biofiltration | 1,323 | 66,483 | 593,079 | 1,295 | 1,236 | 972,651 |
| Machado Rehab. | 2,058 | 26,596 | 243,931 | 1,896 | 116 | 99,523,897 |
| Echo Park | 2,145 | 57,714 | 577,145 | 2,060 | 2,040 | 84,263,313 |
| Cabrito Walkway | 3,368 | 163,062 | 1,045,201 | 3,290 | 3,183 | 4,463,009 |
| LA Zoo Parking Lot | 6,024 | 197,672 | 1,016,391 | 5,813 | 5,786 | 13,904,242 |
| Peck Park Project | 15,246 | 77,501 | 1,238,000 | 12,610 | 1,442 | 6,190,000 |
| La Cienega BMPs | 15,428 | 254,747 | 2,839,958 | 14,473 | 19 | 7,667,888 |
| Parking El Sereno | 25,543 | 2,053,935 | 44,273,722 | 25,214 | 25,336 | 3,984,635 |
| Rosecrans Rec. | 129,885 | 714,712 | 11,256,721 | 108,848 | 11,472 | 6,754,033 |
| Temescal BMPs | NA | NA | NA | NA | 244 | 18,646,000 |
| LAX BMPs | NA | NA | NA | NA | 12 | 1,438,755 |
| **Total** | | | | | | $324,949,037 |

Table 4-12 Results of cost effectiveness from GA simulation.

| Name of watershed | Cost($) /TSS reduction (kg) | Cost($) /TN reduction (kg) | Cost($) /TP reduction (kg) | Cost($)/ Total mass reduction (kg) | Cost($) /mass loads (kg) | **Total Cost ($1,000)** |
|---|---|---|---|---|---|---|
| Oros Streetend Biofiltration | 40 | 3,448 | 24,783 | 39 | 29 | $23 |
| La Cienega/Fairfax BMPs | 47 | 2,541 | 21,860 | 46 | 34 | $13,585 |
| Cabrito Paseo Walkway | 51 | 3,019 | 22,445 | 50 | 37 | $52 |
| LA Zoo Parking Lot | 52 | 2,913 | 13,923 | 51 | 37 | $90 |
| South LA Wetlands Park | 53 | 2,479 | 23,817 | 52 | 38 | $1,359 |
| Aliso Wash-Limekiln Creek | 59 | 4,485 | 37,811 | 58 | 43 | $32,142 |
| Mar Vista Recreation BMPs | 60 | 3,206 | 22,782 | 59 | 43 | $660 |
| Machado Rehabilitation | 65 | 3,959 | 33,157 | 64 | 47 | $40,266 |
| Westchester/LAX BMPs | 67 | 4,049 | 27,120 | 66 | 49 | $5,651 |
| Hansen Dam Parking Lot | 69 | 9,798 | 109,271 | 68 | 51 | $201 |
| Parking Grove El Sereno | 77 | 20,167 | 194,915 | 76 | 57 | $9 |
| Strathern Pit Project | 112 | 10,251 | 105,509 | 111 | 101 | $22,800 |
| Penmar Water Improvement | 232 | 8,662 | 87,713 | 226 | 179 | $12,832 |
| Echo Park Rehabilitation | 264 | 6,417 | 155,012 | 253 | 201 | $8,301 |
| Temescal Recreation BMPs | 302 | 22,938 | 648,000 | 297 | 237 | $18,144 |
| Rosecrans Recreation | 405 | 48,029 | 849,082 | 402 | 368 | $216 |
| Peck Park Canyon Project | 438 | 20,308 | 226,422 | 428 | 388 | $1,664 |
| **Total** | | | | | | **$157,994** |

Table 4-13 shows the cost effectiveness comparison result of the Proposition O with GA simulation. Even cost/total mass loads of La Cienega power line, LAX stormwater BMP and Strathem pit project are larger than Prop. O's result. As a whole the results are much less costly than the BMPs selected by the Prop. O's mangers. The LA zoo parking lot and the Parking Grove in El Sereno' stand out as very expensive in Prop O mangers' selection, with the unit cost more than 100 times more than the optimal costs.

The simulation results as well as the Prop O managers' results do not consider BMP implementation plans and regional BMPs in an integrated fashion. The next generation of BMP optimization should include this integration.

Table 4-13. Comparison of cost-effectiveness

| Watershed | Proposition O report | | | GA simulation | | |
|---|---|---|---|---|---|---|
| | Cost($)/ Total mass reduction (kg) | Cost($)/ Total mass loads (kg) | Total Cost ($1,000) | Cost($)/ Total mass reduction (kg) | Cost($)/ Total mass loads (kg) | Total Cost ($1,000) |
| **La Cienega BMPs** | 14,473 | 19 | 7,668 | 46 | 34 | 13,585 |
| Mar Vista BMPs | 13 | 299 | 4,556 | 59 | 43 | 660 |
| Temescal BMPs | NA | 244 | 18,646 | 297 | 237 | 18,144 |
| **LAX BMPs** | NA | 12 | 1,439 | 66 | 49 | 5,651 |
| Penmar Improv. | 333 | 329 | 23,585 | 226 | 179 | 12,832 |
| LA Zoo Parking Lot | 5,813 | 5,786 | 13,904 | 51 | 37 | 89 |
| **Strathern Pit Project** | 126 | 100 | 22,505 | 111 | 101 | 22,799 |
| Cabrito Walkway | 3,290 | 3,183 | 4,463 | 50 | 37 | 52 |
| Hansen Parking Lot | 566 | 562 | 2,221 | 68 | 51 | 201 |
| South LA Park | 481 | 376 | 13,380 | 52 | 38 | 1,359 |
| Aliso Wash Creek | 16 | 14 | 10,893 | 58 | 43 | 32,142 |
| Oros Biofiltration | 1,295 | 1,236 | 973 | 39 | 29 | 23 |
| Echo Park | 2,061 | 2,040 | 84,263 | 253 | 201 | 8,301 |
| Parking El Sereno | 25,214 | 25,336 | 3,985 | 76 | 57 | 9 |
| Rosecrans Rec. | 108,848 | 11,472 | 6,754 | 402 | 368 | 216 |
| Machado Rehab. | 1,896 | 116 | 99,524 | 64 | 47 | 40,266 |
| Peck Park Project | 12,610 | 1,442 | 6,190 | 428 | 388 | 1,664 |
| **Total** | | | **$324,949** | | | **$157,994** |

(Larger value of cost($)/mass load(kg) than Proposition O in bold.)

# 5. Conclusions

The main goal of this research was to demonstrate the use of an advanced optimization technique that is suitable for watershed-level best management practice (BMP) optimization. Choosing and locating BMPs for a single watershed quickly become an intractable problem (can be solved but not fast enough for the solution to be useful). For example, selecting 5 non-mutually exclusive BMPs (i.e., they can occur at the same time) for 16 sub-watersheds produces $(2^5)^{16} = 1.2\mathrm{x}10^{24}$ possible decision scenarios. Current practice for agencies responsible for stormwater management is to rank BMPs based on effectiveness for at most one or two criteria, and choose BMPs based on ranking. Optimization of BMP selection is clearly beyond the capabilities of current stormwater management practice.

In this dissertation, six different optimization techniques were evaluated for the ability to optimize the BMP selection problem. The following techniques were reviewed for the applicability: Neural Networks, Simulated Annealing, Tabu Searches, Response Surface

Methods, Shuffled Complex Evolution and Genetic Algorithms. A Genetic Algorithm (GA) was selected for this study because the technique has proven convergence, does not require continuous input data, and has only a modest data requirement, especially when compared to a neural network. A code was written in C# to use a GA to optimize the BMP problem.

The input data including watershed area and pollutant total loadings for the optimization tool were adapted from the City of Los Angeles' recent Proposition O stormwater management project. Proposition O, approved by the voters in 2004, provided $500 million to improve stormwater management. At the time of this writing, 21 proposed projects had been approved for construction, and were described in sufficient detail to allow them to be studied for BMP optimization.

The GA chromosome was set up by initializing the DNA for each watershed. Every DNA contains the properties of randomly picked BMPs of the five available types. The properties include pollutant removal rates, BMP construction cost by watershed area and maintenance cost.

A sensitivity analysis of GAs parameters was performed to find better parameters to improve solutions. Based on this study, population size of 100, crossover rate of 60%, mutation rate of 5%, overlapping rate of 60% and a number of generations of 300 gave the best results in terms of fitness values. These operation values were the boundaries of the optimum GA operator ranges defined in the various literatures.

Overall, the BMP placement optimization model performed well in reducing the pollutant load and minimizing BMP total cost from the watershed.

Among many optimization heuristics, the genetic algorithm appeared to be the most suited to the BMP placement problem type. Its implementation was successful the optimal selections and placements were about half the cost of the manually selected BMPs ($324 vs. 157 million). The pollutant removal rates for total suspended solids were quite similar, within the precision of BMP performance estimates. The manually selected BMPs had better total nitrogen and total phosphorus removal rates (55% versus 45%).

The GA code was very efficient and took, on average, 9 seconds to execute on 2 GHz

Pentium Ⅳ processor running under MS Windows.

The example provided in this dissertation demonstrates the applicability of the GA technique to BMP selection and suggest that the GA technique can be useful to a broad range of intractable problems.

# Appendix

# (Program codes, C#)

```
private void Form1_Load(object sender, System.EventArgs e)
        {
            //Inialize operation UI.
            InitializeUI();

            //Load starting up problem property values.
            LoadStartUpLandProperties();

            //Update system info.
            lblInfoDisp.Text = "Click \"Simuate\" button to start ...";
        }


        public void LoadStartUpLandProperties()
        {

            string[] area = { "78", "111", "123", "82", "75", "101", "160", "108", "120", "185", "75", "80", "155",
"220", "125", "101", "84", "34", "45", "54", "88", "121", "133", "89", "79", "121", "150", "118", "110", "195", "78",
"88", "135", "240", "145", "100", "84", "31", "43", "52", "75", "82", "150", "210", "121", "111", "86", "30", "42",
"56" };
            string[] no_pol_1 = { "12", "19", "23", "11", "14", "20", "21", "15", "17", "25", "15", "12", "23", "21",
"14", "11", "12", "11", "15", "14", "12", "19", "23", "10", "14", "20", "27", "15", "18", "25", "15", "12", "29", "21", "17",
"11", "15", "11", "15", "14", "15", "11", "23", "41", "34", "31", "22", "11", "11", "12" };
            string[] no_pol_2 = { "2.1", "3.1", "5.2", "5.3", "4.5", "3", "3.2", "6.1", "4.5", "4", "3", "2.4", "1.1", "2.1",
"5.1", "1", "3", "4.5", "4.5", "5.2", "4.1", "6.1", "5.9", "5.3", "4.5", "3.9", "3.2", "6.1", "4.5", "4.2", "3.5", "2.4", "1.1",
"2.7", "5.1", "1.7", "3", "4.5", "4.2", "5.2", "3.3", "2.4", "1.1", "3.1", "6.1", "4", "3", "4.5", "4", "5.2" };
            string[] no_pol_3 = { "177", "265", "188", "140", "174", "160", "185", "195", "108", "145", "145",
"152", "183", "251", "194", "110", "100", "43", "130", "88", "187", "295", "198", "143", "179", "169", "195", "195",
"118", "145", "149", "152", "183", "153", "194", "110", "111", "143", "139", "88", "175", "152", "283", "251", "234",
"110", "100", "143", "139", "90" };


            string[] BMP_type = { "WP", "IS", "BS", "SF", "BF" };
            string[] cost = { "2.6", "7.9", "10.8", "14.6", "15.7" };            //Cost of BMP per area.
            string[] rep_1 = { "38", "60", "70", "40", "55" };                   //Removal(%) of TN.
            string[] rep_2 = { "46", "55", "35", "33", "49" };                   //Removal (%) of TP.
            string[] rep_3 = { "75", "80", "80", "80", "92" };                   //Removal (%) of TSS.
            string[] OMR = { "4.5", "10.5", "5", "12", "6" };             //OMR cost of BMP (% of consturcion)
```

```
            for (int id = 0; id < area.Length; id++)
            {
                ListViewItem item = new ListViewItem(Convert.ToString(id + 1));
                item.SubItems.Add(area[id]);                    //Area.
                item.SubItems.Add(no_pol_1[id]);                //pollutant type 1.
                item.SubItems.Add(no_pol_2[id]);                //polllutant type 2.
                item.SubItems.Add(no_pol_3[id]);             //polllutant type 3.
                lvWatershedProp.Items.Add(item);

            }

            for (int id = 0; id < BMP_type.Length; id++)
            {
                ListViewItem item = new ListViewItem(BMP_type[id]);
                item.SubItems.Add(cost[id]);
                item.SubItems.Add(rep_1[id]);
                item.SubItems.Add(rep_2[id]);
                item.SubItems.Add(rep_3[id]);
                item.SubItems.Add(OMR[id]);
                lvBMP.Items.Add(item);
            }
        }

        private void butSimulate_Click(object sender, System.EventArgs e)
        {
            //Clear previous data object.
            m_arlLandPropData.Clear();
            m_arlBMPPropData.Clear();

            //Load watershed property into data structure from UI.
            if (lvWatershedProp.Items.Count > 0)
            {
                //Generate watershed property data object from UI value and store in List.
                for (int i = 0; i < lvWatershedProp.Items.Count; i++)
                {
                    LandProperty lp_data = new LandProperty();
                    lp_data.land_id = lvWatershedProp.Items[i].SubItems[0].Text;
                    lp_data.area = Convert.ToDouble(lvWatershedProp.Items[i].SubItems[1].Text);
                    lp_data.in_pollutant_type1 = Convert.ToDouble
( lvWatershedProp.Items[i].SubItems[2].Text);
                    lp_data.in_pollutant_type2 = Convert.ToDouble
( lvWatershedProp.Items[i].SubItems[3].Text);
                    lp_data.in_pollutant_type3 = Convert.ToDouble
( lvWatershedProp.Items[i].SubItems[4].Text);
                    m_arlLandPropData.Add(lp_data);
                }
```

```
                Sum_id.Text = Convert.ToString(lvWatershedProp.Items.Count);

                double sum1 = 0, sum2 = 0, sum3 = 0, sum4 = 0;

                for (int id = 0; id < lvWatershedProp.Items.Count; id++)
                {
                    sum1 += Convert.ToDouble(lvWatershedProp.Items[id].SubItems[1].Text);
                    sum2 += Convert.ToDouble(lvWatershedProp.Items[id].SubItems[2].Text);
                    sum3 += Convert.ToDouble(lvWatershedProp.Items[id].SubItems[3].Text);
                    sum4 += Convert.ToDouble(lvWatershedProp.Items[id].SubItems[4].Text);

                }

                Sum_Area.Text = Convert.ToString(sum1);
                Sum_N.Text = Convert.ToString(sum2);
                Sum_P.Text = Convert.ToString(sum3);
                Sum_SS.Text = Convert.ToString(sum4);

            }
            else
            {
                string msg = "Please input watersheds for simulation.";
                MessageBox.Show(this, "Error", msg, MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
                return;
            }

            //Load BMP property into data structure from UI.
            if (lvBMP.Items.Count > 0)
            {
                //Generate BMP property data object from UI value and store in arraylist.
                for (int i = 0; i < lvBMP.Items.Count; i++)
                {
                    BMPProperty b_data = new BMPProperty();
                    b_data.BMP_id = lvBMP.Items[i].SubItems[0].Text;
                    b_data.cost = Convert.ToDouble(lvBMP.Items[i].SubItems[1].Text);
                    b_data.rep_type_1 = Convert.ToDouble(lvBMP.Items[i].SubItems[2].Text);
                    b_data.rep_type_2 = Convert.ToDouble(lvBMP.Items[i].SubItems[3].Text);
                    b_data.rep_type_3 = Convert.ToDouble(lvBMP.Items[i].SubItems[4].Text);
                    b_data.OMR = Convert.ToDouble(lvBMP.Items[i].SubItems[5].Text);
                    m_arlBMPPropData.Add(b_data);
                }
            }
            else
            {
                string msg = "Please input BMP data for simulation.";
```

```
                MessageBox.Show(this, "Error", msg, MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
                return;
            }

            //Load GA Engine property data.
            m_GAGeneration = Convert.ToInt32(udGANumGeneration.Value);
            m_GAPopulation = Convert.ToInt32(udGAPopulation.Value);
            m_GaEngine.RegisterOperationValues(
                Convert.ToInt32(udGACrossOverRate.Value),
                Convert.ToInt32(udGAMutationRate.Value),
                Convert.ToInt32(udGAOverlapRate.Value)
            );

            //Perform solution simulation.
            GenerateChromosom();

            //Supply data to GA Engine.
            m_GaEngine.SupplyData(m_arlLandPropData.Count, m_arlChromosom, m_arlBMPPropData,
m_arlLandPropData);

            //Start simulation thread.
            Thread sim = new Thread(new ThreadStart(PerformSimulation));
            sim.Start();
            m_ResetFlag = false;
            m_SimRunning = true;
            butSimulate.Enabled = false;
        }

        private void PauseChecking()
        {
            if (m_PauseFlag)
            {
                lblInfoDisp.Text = "Simulation paused...";
                while (m_PauseFlag && !m_ResetFlag)
                {
                    //Loop still when pause flag still on.
                    Thread.Sleep(250);
                }

                if (!m_ResetFlag)
                    lblInfoDisp.Text = "Simulation resumed...";
            }
        }

private void PerformSimulation()
```

```
{
    //Simulation variables.
    int curr_gen = 0;            //Variable hold current generation.
    InitializeUI();
    pbSolve.MaxValue = m_GAGeneration;
    lblInfoDisp.Text = "Simulation started...";
    Stopwatch stopWatch = new Stopwatch();
    stopWatch.Start();

    while (curr_gen < m_GAGeneration && !m_ResetFlag)
    {
        //Execute this when pause flag detected.
        PauseChecking();

        //Perform GA cross over operation.
        m_GaEngine.PerformCrossOver();

        //Perform overlapping.
        m_GaEngine.PerformOverlappping();

        //Perform mutation operation.
        m_GaEngine.PerformMutation();

        //Sort chromosom set.
        BubbleSortChromosom();

        //Update chromosom result list.
        DisplayBestChromosomResult();

        //Update result UI and graph.
        UpdateFitnessValueToGraph();

        //Sleep the thread for a minimum time.
        Thread.Sleep(10);

        //Increase generation counter.
        curr_gen++;

        //Update progress bar UI.
        pbSolve.Value = curr_gen;
        lblGenDisp.Text = curr_gen.ToString();

    }//End simulation loop.

    stopWatch.Stop();
    // Get the elapsed time as a TimeSpan value.
```

```
                TimeSpan ts = stopWatch.Elapsed;

                // Format and display the TimeSpan value.


                string elapsedTime = String.Format("{0:00}:{1:00}:{2:00}.{3:00}", ts.Hours, ts.Minutes, ts.Seconds,
ts.Milliseconds / 10);
                RunTime.Text = elapsedTime;

                if (m_ResetFlag)
                {
                    //No need to do result UI update if it is a reset case.
                    InitializeUI();
                }
                else
                {
                    //Update complete UI.
                    m_SimRunning = false;
                    pbSolve.Value = pbSolve.MaxValue;
                    lblInfoDisp.Text = "Simulation completed.";

                    //Display simulation result.
                    DisplayAllChromosomsResult();

                    //Calculate display result.
                    CalculateOptimalChromosomResult();
                }
            }

            private void DisplayBestChromosomResult()
            {
                int count = 0;
                lvResultChr.Items.Clear();
                foreach (Chromosom c in m_arlChromosom)
                {
                    ListViewItem item = new ListViewItem(c.Curr_id.ToString());

                    //Display fitness value in this chromosom.
                    item.SubItems.Add(c.ObtainFitnessValue().ToString());

                    lvResultChr.Items.Add(item);

                    if (count > 6) break;
                    count++;
                }
```

```
}

private void DisplayAllChromosomsResult()
{
    lvResultChr.Items.Clear();
    foreach (Chromosom c in m_arlChromosom)
    {
        ListViewItem item = new ListViewItem(c.Curr_id.ToString());

        //Display fitness value in this chromosom.
        item.SubItems.Add(c.ObtainFitnessValue().ToString());

        lvResultChr.Items.Add(item);
    }
}

private void CalculateOptimalChromosomResult()
{
    Chromosom optimal_chr = m_arlChromosom[0];
    foreach (DNA d in optimal_chr.Dna)
    {
        ListViewItem item = new ListViewItem(d.reg_watershed_prop.land_id);                //Display
Watershed ID.
        item.SubItems.Add(d.reg_BMP_prop.BMP_id);        //Display BMP ID.
        item.SubItems.Add(d.Obtain_RMP_Type1().ToString());//Remain Mass of Pollutant Type 1.
        item.SubItems.Add(d.Obtain_RMP_Type2().ToString());//Remain Mass of Pollutant Type 2.
        item.SubItems.Add(d.Obtain_RMP_Type3().ToString());//Remain Mass of Pollutant Type 3.
        item.SubItems.Add(d.Obtain_BMP_Cost().ToString());                 //Display Cost
        lvResultLand.Items.Add(item);
    }

    //Get a percentage of removal efficiency
    double sum1 = 0, sum2 = 0, sum3 = 0;
    sum1 = Convert.ToDouble(Sum_N.Text);
    sum2 = Convert.ToDouble(Sum_P.Text);
    sum3 = Convert.ToDouble(Sum_SS.Text);

    string rem1 = "", rem2 = "", rem3 = "";
    rem1 = optimal_chr.ObtainTotalRMP_Type1().ToString();
    rem2 = optimal_chr.ObtainTotalRMP_Type2().ToString();
    rem3 = optimal_chr.ObtainTotalRMP_Type3().ToString();

    double remN = 0, remP = 0, remSS = 0;
    remN = Convert.ToDouble(rem1);
    remP = Convert.ToDouble(rem2);
    remSS = Convert.ToDouble(rem3);
```

```
            //Until here for removal efficiency, Nasty code..

            lblTotalPollutant1.Text = string.Format("{0:N2}   \n   ({1:P})",
optimal_chr.ObtainTotalRMP_Type1().ToString(),((remN - sum1) / sum1));
            lblTotalPollutant2.Text = string.Format("{0:N2}   \n   ({1:P})",
optimal_chr.ObtainTotalRMP_Type2().ToString(), ((remP - sum2) / sum2));
            lblTotalPollutant3.Text = string.Format("{0:N2}   \n   ({1:P})",
optimal_chr.ObtainTotalRMP_Type3().ToString(), ((remSS - sum3) / sum3));
            lblTotalBMPCost.Text = string.Format("{0:C2}",
Convert.ToDecimal(optimal_chr.ObtainTotalBMPCost().ToString()));
        }

        private void InitializeUI()
        {
            Control.CheckForIllegalCrossThreadCalls = false;
            //Clear off the UI progress bar.
            pbSolve.Value = 0;

            //Clear result display UI.
            lvResultChr.Items.Clear();
            lvResultLand.Items.Clear();
            lblGenDisp.Text = "---";
           /* lblTotalPollutant1.Text = "---";
            lblTotalPollutant2.Text = "---";
            lblTotalPollutant3.Text = "---";
            lblTotalBMPCost.Text = "---";   */
            lblInfoDisp.Text = "---";

            //Clear off graph display data.
            exMovGraph.ClearGraphData();
        }

        private void BubbleSortChromosom()
        {
            int m, n;
            Object tmp_chr;
            for (m = 0; m < (m_arlChromosom.Count - 1); m++)
            {
                for (n = (m + 1); n < m_arlChromosom.Count; n++)
                {
                    if (
                        (m_arlChromosom[m]).ObtainFitnessValue() >
                        (m_arlChromosom[n]).ObtainFitnessValue()
                        )
                    {
                        tmp_chr = m_arlChromosom[m];
```

```
                    m_arlChromosom[m] = m_arlChromosom[n];
                    m_arlChromosom[n] = (Chromosom)tmp_chr;
                }
            }
        }
    }

    private void GenerateChromosom()
        m_arlChromosom.Clear();             //Clear previous generated chromosom.

        m_GAPopulation = Convert.ToInt32(udGAPopulation.Value);
        for (int i = 0; i < m_GAPopulation; i++)
        {
            Chromosom chr = new Chromosom();
            for (int m = 0; m < m_arlLandPropData.Count; m++)
            {
                DNA dna = new DNA();
                dna.reg_watershed_prop = m_arlLandPropData[m];
                dna.reg_BMP_prop =
m_arlBMPPropData[m_RandomEngine.Next(m_arlBMPPropData.Count)];

                //Register generated dna data object into chromosom.
                chr.Register_Dna(dna);
            }

            //Store complete generated chromosom in arraylist data structure.
            m_arlChromosom.Add(chr);
        }
    }

}

namespace GAMultiVarSim
{
    //This is Genetic Algorithm Engine, that take user GA setting from UI and perform GA calculation.
    //The chromosome data structure defined in datastructure.cs.
    public class GaEngine
    {
            private int cross_over_rate;
            private int mutation_rate;
            private int overlapping_rate;

            private double m_crossover_percent;
            private double m_mutation_percent;
            private double m_overlapping_percent;
```

```
		private int m_dna_size;
		private List<Chromosom> m_arlChromosom = null;
		private List<BMPProperty> m_arlBMPPropData = null;
		private List<LandProperty> m_arlLandPropData = null;

		private Random m_RandomEngine = null;

		public GaEngine()
		{
			//Initialize internal data.
			cross_over_rate = 1;
			mutation_rate = 1;
			overlapping_rate = 1;

			m_crossover_percent = 0.01;
			m_mutation_percent = 0.01;
			m_overlapping_percent = 0.01;

			m_RandomEngine = new Random(System.DateTime.Now.Millisecond);
			m_dna_size = 0;
		}

		public void SupplyData(int chr_size, List<Chromosom> arlChromosom, List<BMPProperty>
arlBMPPropData, List<LandProperty> arlLandPropData)
		{
			m_arlChromosom = arlChromosom;
			m_arlLandPropData = arlLandPropData;
			m_arlBMPPropData = arlBMPPropData;
			m_dna_size = chr_size;
		}

		public void RegisterOperationValues(int cross_over_rate, int mutation_rate, int
overlapping_rate)
		{
			this.cross_over_rate = cross_over_rate;
			this.mutation_rate = mutation_rate;
			this.overlapping_rate = overlapping_rate;

			m_crossover_percent = Convert.ToDouble(this.cross_over_rate) / 100;
			m_mutation_percent = Convert.ToDouble(this.mutation_rate) / 100;
			m_overlapping_percent = Convert.ToDouble(this.overlapping_rate) / 100;
		}

		public void PerformCrossOver()
		{
			int a, b, tmp;
```

```
//Randomly find 2 points.
a = m_RandomEngine.Next(m_dna_size);
b = m_RandomEngine.Next(m_dna_size);
if(a>b)
{
        tmp = b;
        b = a;
        a = tmp;
}


//Perform random 2 fix points cross over.
if(m_RandomEngine.NextDouble() < m_crossover_percent)
{
        Chromosom c1 = SelectChromosom();
        Chromosom c2 = SelectChromosom();
        Chromosom tmp_c1 = c1.Clone();
        Chromosom tmp_c2 = c2.Clone();
        DNA tmp_dna;
        for(int m=0; m<m_dna_size; m++)
        {
                if(m>=a && m<=b)
                {
                        tmp_dna = tmp_c1.GetDna(m);
                        tmp_c1.SetDna(m, tmp_c2.GetDna(m));
                        tmp_c2.SetDna(m, tmp_dna);
                }
        }


if( tmp_c1.ObtainFitnessValue() < c1.ObtainFitnessValue() ) c1.Copy( tmp_c1 );
if( tmp_c2.ObtainFitnessValue() < c2.ObtainFitnessValue() ) c2.Copy( tmp_c2 );
        }
}

public void PerformMutation()
{

        int chr_idx, fer_idx;

        //Continue to do mutation when the chance is given.
        double d = m_RandomEngine.NextDouble();
        if( d < m_mutation_percent)
        {
                chr_idx =
```

```
m_RandomEngine.Next( Convert.ToInt32( (2*m_arlChromosom.Count)/3 ), m_arlChromosom.Count );
                                        Chromosom opr_c = m_arlChromosom[chr_idx];

                                        //Mutation happen to all the location of DNA.
                                        for(int p=0; p<m_dna_size; p++)
                                        {
                                                fer_idx =
m_RandomEngine.Next( m_arlBMPPropData.Count );
                                                opr_c.UpdateDna(p, m_arlBMPPropData[fer_idx]);
                                        }
                                }
                }

                public void PerformOverlappping()
                {
                        if(m_RandomEngine.NextDouble() < m_overlapping_percent)
                        {
                                Chromosom c1 = SelectChromosom();
                                Chromosom c2 = SelectChromosom();
                                Chromosom tmp = CreateChromosom();

                                for(int m=0; m<m_dna_size; m++)
                                {
                                        if( c1.GetDna(m).reg_BMP_prop.BMP_id ==
c2.GetDna(m).reg_BMP_prop.BMP_id )
                                        {
//Only copy the dna to new produce dna when both parents dna registered the used BMP are the same.
                                                tmp.SetDna(m, c1.GetDna(m));
                                        }
                                }

    //Check the new produced chromosom. Replace its parents when the fitness value is better.

                if(tmp.ObtainFitnessValue() < c1.ObtainFitnessValue()) c1.Copy( tmp );
                if(tmp.ObtainFitnessValue() < c2.ObtainFitnessValue()) c2.Copy( tmp );
                        }
                }

                #region Private Methods For GA Engine
                private Chromosom CreateChromosom()
                {
                        Chromosom c = new Chromosom();
                        for(int m=0; m<m_dna_size; m++)
                        {
                                DNA dna = new DNA();
                                dna.reg_watershed_prop = m_arlLandPropData[m];
```

```
                        dna.reg_BMP_prop =
m_arlBMPPropData[m_RandomEngine.Next(m_arlBMPPropData.Count)];

                        //Register generated dna data object into chromosom.
                        c.Register_Dna(dna);
                    }
                    return c;
                }
                private Chromosom SelectChromosom()
                {
                    int idx;

                    idx = m_RandomEngine.Next( Convert.ToInt32(m_arlChromosom.Count/3 ) );

                    return m_arlChromosom[idx];
                }
                #endregion
        }
}
```

# References

Abustan, I., Ball, J.E. 1998. Modelling the export of phosphorus from urban catchments. Proc. Hydrostorm '98, Adelaide, Austalia, 45-50.

ASCE, 2001. Guide for Best Management Practice (BMP) Selection in Urban Developed Areas. American Society of Civil Engineers, Reston, Virginia.

Bazaraa, Mokhtar S., John J. Jarvis, and Hanif D. Sherali. 1990. Linear Programming and Network Flows. New York, NY: John Wiley & Sons.

Bazaraa, Mokhtar S., Hanif D. Sherali, and C. M. Shetty. 1993. Nonlinear Programming: Theory and Algorithms. New York, NY: John Wiley & Sons.

Bajwa, S.G., Vibhava, V. 2009. A Distributed Artificial Neural Network Model Watershed - scale Rainfall - Runoff Modeling, American Society of Agricultural and Biological Engineers, Vol. 52(3): 813-823.

Bettinger, Pete, K. Norman Johnson, and John Sessions. 1998. Improving aquatic habitat conditions over time while producing wood products: an examination of options. Journal of the American Water Resources Association, 34(4): 891-907.

Borst, M. and Selvakumar, A. 2003. Particle associated microorganisms in storm-water runoff. Wat. Res., 37, 215-223.

Browne, F.X. (1990) Storm-water Management, in Standard Handbook of Environmental Engineering, ed. by R.A. Corbitt, McGraw-Hill Inc., New York

Buick, R. D., N. D. Stone, R. K. Scheckler, and J. W. Roach. 1992. CROPS: A whole-farm crop rotation planning system to implement sustainable agriculture. AI Applications, 6(3): 29-50.

Bureau of Engineering (BOE) (2007) Proposition O May 2007 Monthly Report

Burian, S.J., Brown, M.J., and McPherson, T.N. (2002) Evaluation of land use/land cover datasets for urban watershed modeling, Water Science and Technology, 45 (9), 269-276

Cannon, A.J., Whitfield, P.H. 2002. Downscaling Recent Stream-flow Conditions in British Columbia, Canada using Ensemble Neural Network Models. Journal of Hydrology, 259, 136-151.

Chambers, Lance (editor). 1995. Practical Handbook of Genetic Algorithms: Applications, Volume 1. Boca Raton, FL: CRC Press.

Characklis, W.G and Wiesner, M.R. 1997. Particle, metals and water quality in runoff from large urban watershed. Wat. Env. Res., 70(7), 1295-1302.

Cooper, W. W., H. Hemphill, Z. Huang, S. Li, V. Lelas, and D. W. Sullivan. 1996. Survey of mathematical programming models on air pollution management. European Journal of Operational Research 96(1): 1-35.

Cooper, V. A, V. T. V. Nguyen, and J. A. Nicell. 1997. Evaluation of global optimization methods for conceptual rainfall-runoff model calibration. Water Science and Technology 36(5): 53-60.

Dorn, Jason, S. Ranjithan, S. Liehr, and R.C. Borden 1995. A Genetic Algorithm Approach to Water Supply Watershed Management. In Computing in Civil Engineering 2 971-978. ASCE, New York, NY

Duan, Q. Y., V. K. Gupta, and S. Sorooshian. 1993. Shuffled complex evolution approach for effective and efficient global minimization. Journal of Optimization Theory and Applications 76(3): 501-521.

Eglese, R. W. 1990. Simulated annealing: a tool for operational research. European Journal of Operational Research 46(3): 271-281.

EPA. 1999. Preliminary Data Summary of Urban Storm Water Best Management

Practices, Chapter 6: Costs and Benefits of Storm Water BMPs. U.S. Environmental Protection Agency, Office of Water, Washington, DC.

Greenberg, Harvey J. (1995). Mathematical programming models for environmental quality control. Operations Research 43(4): 578-622.

Glover, Fred, Eric Taillard, and Dominique de Werra. 1993. A user's guide to tabu search. Annals of Operations Research 41(1-4): 3-28.

Glover, Fred, James P. Kelly, and Manuel Laguna. 1995. Genetic algorithms and tabu search: hybrids for optimization. Computers in Operations Research 22(1): 111-134.

Goldberg, D. E. 1989. Genetic Algorithms in Searching, Optimization, and Machine Learning. Reading, MA: Addison Wesley.

Greenberg, Harvey J. 1995. Mathematical programming models for environmental quality control. Operations Research 43(4): 578-622.

Grötschel, Martin. 1982. Approaches to hard combinatorial optimization problems. In Modern Applied Mathematics – Optimization and Operations Research, ed. B. Korte, part IV, ch. 2, 437-515. New York, NY: North-Holland Publishing Company.

Heatwole, C. D., A. B. Bottcher, and L. B. Baldwin. 1986. Basin scale model for evaluating Best Management Practice implementation programs. Transactions of the ASAE 29(2): 439-444.

Herricks, E.E. 1995. Storm-water runoff and receiving systems. CRC Press, Inc., Boca Raton, Florida, USA.

Holland, J. 1975. Adaptation in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor, Michigan.

Ibrahim, Yaacob and Shie-Yui Liong. 1992. Calibration strategy for urban catchment parameters. Journal of Hydraulic Engineering 118(11): 1550-1570.

Jacobson, Sheldon H. and Lee W. Schruben. 1989. Techniques for simulation response optimization. Operation Research Letters 8(1): 1-9.

Jones, J.E., Clary, J., Strecker, E., Quigley, M.M. 2008. 15 Reasons You Should Think Twice Before Using Percent Removal to Assess BMP Performance. http://www.stormh2o.com/january-february-2008/pollutants-research-BMP.aspx

Kamalakkannan, R., Zettel, V. G., Stead, K. and Ward, N.I. 2004. Chemical levels in contaminated storm-water and sediments from a motorway dry detention pond drainage system. J. Environ. Monit., 6, 175-181.

Kralisch, S., Fink, M., Flugel, W.A., Beckstein, C. 2003. A neural network approach for the optimisation of watershed management. Environmental Modeling & Software 18: 815-823

Krein, A., Schorer, M. 2000. Road runoff pollution by polycyclic aromatic hydrocarbons and contribution to river sediments. Water. Res., 34(16), 4110-4115.

Knisel, W. G. (editor). 1980. CREAMS: A field-scale model for chemicals, runoff and erosion from agricultural management systems. Conservation Research Report No. 26, USDA-SEA, Washington, DC.

Latimer, J.S., Hoffman, E.J, and Quinn, J.G. 1990. Sources of petroleum hydrocarbons in urban runoff. Wat. Air Soil Pollut., 52, 1-21

Liong, Shie-Yui, Weng Tat Chan, and Jaya ShreeRam. 1995. Peak-flow forecasting with genetic algorithm and SWMM. Journal of Hydraulic Engineering 121(8): 613-617.

Mitchell, Melanie. 1999. An Introduction to Genetic Algorithms. Cambridge, MA: The MIT Press.

Lundy M. and A. Mees. 1986. Convergence of an annealing algorithm. Mathematical Programming 34(1): 111-24.

Maier, H.R., Dandy, G.C. 2000. Neural networks for the Prediction and Forecasting of Water Resources Variables. Environmental Modeling and Software. 15, 101-124.

Myers, R. H. 1971. Response Surface Methodology. Boston, MA: Allyn and Bacon.

Ngabe, B., Bidleman, T.F. and Scott, G.I., 2000. Polycyclic aromatic hydrocarbons in storm runoff from urban and coastal South Carolina, Sci. of the Total Envir., 255, 1-9.

Park, M-H, Swamikannu, X., and M.K. Stenstrom, 2009. Accuracy and Precision of the Volume-Concentration Method for Urban Stormwater Modeling, *Wat. Research*, **43**(11), 2773-2786

Predeep, Kumar Behera, F. Papa, and B.J. Adams, 1999. Optimization of Regional Storm Water Management Systems. JWRPM, 125(2), 107-113

Raid, S., Mania, J., BBouchaou, L., And Najjar, Y. 2004. Rainfall-runoff Model using an Artificial Neural Network Approach, Mathematical and Computer Modeling, 40, 839-846

Reeves, C. R. 1993. Improving the efficiency of tabu search for machine sequencing problems. Journal of the Operational Research Society 44(4): 375-382.

Reich, Y., Barai, S.V., 2000. A methodology for Building Neural Networks Models from

Empirical Engineering Data, Engineering of Applications of Artificial Intelligence, 13(6): 685-694

Roger, S., Montrejaud-Vignoles, M., Herremans, L. and Fortune, J.P. 1998. Mineral, physical and chemical analysis of the solid matter carried by motorway runoff water. Wat. Res., 32(4), 1119-1125.

Sartor, J.D., and Boyd, G.B. 1972. Water pollution, aspects of street surface contaminants. EPA-R2-72-081, USEPA.

Scarselli, F. and Tsoi, A.C. 1998. Universal Approximation using Feed Forward Neural Network: A Survey of some existing methods, and some new results, Neural Networks, 11(1)

Schueler, T. R. 1987. Controlling urban runoff: A practical manual for planning and designing urban BMPs. Metropolitan Washingtom Council of Goverments, Washington, D.C., USA

Schillinger, J.E and Gannon, J.J. 1985. Bacterial adsorption and suspended particles in urban storm-water. J. Wat. Pollut. Control., 57, 384-389.

Siegelmann, Hava Tova and Ophir Frieder. 1991. The allocation of documents in multiprocessor information retrieval systems: an application of genetic algorithms. IEEE

International Conference on Systems, Man, and Cybernetics 1(1): 645-650.

Srivastava, P., P. D. Robillard, J. M. Hamlett, and R. L. Day. 1999. Watershed optimization of Best Management Practices using AnnAGNPS, GIS and genetic algorithm. ASAE Paper No. 99-3012. St. Joseph, MI: ASAE.

Stenstrom, M.K., Gary, S.S., and Taras A.B. 1982. OIL AND GREASE IN URBAN STORMWATERS. Journal of Environmental Engineering, Vol. 110, No. 1, February, 1984.

Stenstrom, M.K., Park, M. 2007. An Evaluation on the Effectiveness of Proposition O Projects for Water Quality Improvement. Institute of the Environment, University of California, Los Angeles

Stone, N. D., B. E. Cline, and J. Pease. 2002. Generating farm descriptions in a watershed from incomplete data using simulated annealing. Journal of the AWRA 38. In press.

Strecker, E.W., Quigley, M.M., Urbonas, B.R., Jones, J.E., Clary, J.K. 2001. Determining Urban Storm Water BMP Effectiveness, Journal of Water Resources Planning and Management, May/June, 144-149.

Sumner Neil R., P. Mick Fleming, and Bryson C. Bates. 1997. Calibration of a modified

SFB model for twenty-five Australian catchments using simulated annealing. Journal of Hydrology 197(1-4): 166-188.

Swisher, J. R., P. D. Hyden, S. H. Jacobson, and L. W. Schruben. 2000. A survey of simulation optimization techniques and procedures. In Winter Simulation Conference Proceedings, 1(1): 119-28. Piscataway, NJ: IEEE.

Takada, H., Harada, M. and Ogura, N. 1991. Distribution and sources of polycyclic aromatic hydrocarbons in street dust from the Tokyo metropolitan area, Sci. Total Env., 107, 45-69.

Thyer, Mark, George Kuczera, and Bryson C. Bates. 1999. Probabilistic optimization for conceptual rainfall-runoff models: a comparison of the shuffled complex evolution and simulated annealing algorithms. Water Resources Research 35(3): 767-773.

Vaze, J. and Chiew, F.H.S. 2002. Experimental study of pollutant accumulation on an urban road surface. Urban Wat., 4, 379-389.

Wong, K., Strecker, E.W., and Stenstrom, M.K. (1997) A geographic information system to estimate storm-water pollutant mass loadings, Journal of Environmental Engineering, ASCE, 123, 737-745

Wossink, Ada, and Bill Hunt. 2003. The Economics of Structural Storm-water BMPs in North Carolina, WRRI Research Report Number 344.

Yeh, Chao-Hsien, Labadie , J.W. 1997. Multiobjective watershed-level planning of storm water detention systems. Journal of water resources planning and management, 123(6), 336-343

Zartman, R.E, Ramsey III, R.H., and Huang, A. 2001. Variability of total and dissolved elements in storm-water runoff. J.Soil & Wat. Conservation, 56(3), 263.