AGRICULTURAL BMP PLACEMENT FOR COST-EFFECTIVE POLLUTION CONTROL AT THE WATERSHED LEVEL

Tamie Lynne Veith

Dissertation submitted to the Faculty of Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Biological Systems Engineering

Mary Leigh Wolfe (Chair)
Darrell J. Bosch
Theo A. Dillaha III
Conrad D. Heatwole
Nicholas D. Stone

April 25, 2002
Blacksburg, Virginia

Keywords: genetic algorithm, spatial optimization, GIS, model, sediment delivery, targeting

AGRICULTURAL BMP PLACEMENT FOR COST-EFFECTIVE POLLUTION CONTROL AT THE WATERSHED LEVEL

Tamie Lynne Veith

Abstract

The overall goal of this research was to increase, relative to targeting recommendations, the cost-effectiveness of pollution reduction measures within a watershed. The goal was met through development of an optimization procedure for best management practice (BMP) placement at the watershed level. The procedure combines an optimization component, written in the C++ language, with spatially variable nonpoint source (NPS) prediction and economic analysis components, written in the ArcView geographic information system scripting language. The procedure is modular in design, allowing modifications or enhancements to the components while maintaining the overall theory.

The optimization component uses a genetic algorithm to optimize a lexicographic multi-objective function of pollution reduction and cost increase. The procedure first maximizes pollution reduction to meet a specified goal, or maximum allowable load, and then minimizes cost increase. For the NPS component, a sediment delivery technique was developed and combined with the Universal Soil Loss Equation to predict average annual sediment yield at the watershed outlet. Although this evaluation considered only erosion, the NPS pollutant fitness score allows for evaluation of multiple pollutants, based on prioritization of each pollutant. The economic component considers farm-level public and private costs, accounting for crop productivity levels by soil and for enterprise budgets by field. The economic fitness score assigns higher fitness scores to scenarios in which costs decrease or are distributed more evenly across farms. Additionally, the economic score considers the amounts of cropland, hay, and pasture needed to meet feed and manure/poultry litter spreading requirements.

Application to two watersheds demonstrated that the procedure optimized BMP placement, locating scenarios more cost-effective than a targeting strategy solution. The optimization procedure identified solutions with lower costs than the targeting strategy solution for the same level of pollution reduction. The benefit to cost ratio, including use of the procedure and implementation of resulting solutions, was demonstrated to be greater for the optimization procedure than for the targeting strategy. The optimization procedure identifies multiple near optimal solutions. Additionally, the procedure creates and evaluates scenarios in a repeated fashion without requiring human interaction. Thus, more scenarios can be evaluated than are feasible to evaluate manually.

Acknowledgements

I thank my committee for their guidance and support over the past four and a half years. In particular, I express my gratitude to Dr. Wolfe for encouraging and challenging me both academically and professionally throughout this degree. Additionally, my appreciation to Dr. Perumpral for funding a significant portion of my research work and providing departmental computer facilities necessary for completing this research.

I thank my family who has been supportive of my journey through graduate school since before I began. It is a treasure to be part of a family that has supported my endeavors without question, even when being a student starts to become a lifestyle.

I extend my gratitude to the Heatwoles for putting music back into my life. Additionally, I would like to thank Margaret for always having the teapot ready and Tone for endlessly listening to me brainstorm (even on the other side of that cell phone).

Table of Contents

List of Figures

List of Tables

Chapter 1: Research Problem

1.1 Introduction

Nonpoint source (NPS) pollution from agricultural lands is a significant contributor to water quality degradation. In the last few decades there has been increasing concern over water and soil-borne pollutants that influence human or aquatic health or that restrict human activities. Government regulations, such as the Clean Water Act, are placing growing emphasis on NPS pollution control. One method of control is through implementation of best management practices (BMPs); BMPs are structural, vegetative or cultural methods by which NPS pollution is eliminated or reduced sufficiently to meet water quality criteria (Novotny and Olem, 1994).

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 scenario) 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.

Field studies can be very useful in monitoring pollution reduction by BMPs for a particular site over time. Through monitoring, pollutant values before and after BMP implementation can be compared to determine if the cost and potential environmental disruption encountered in implementing, maintaining, and operating a BMP outweigh the long-term environmental impacts had that BMP not been installed. Monitoring is advantageous in its capability to reflect natural climatic variation and other variable environmental impacts, for example, those caused by wildlife or humans, over multiple seasons (short-term monitoring) or years (long-term monitoring).

However, because weather patterns can vary greatly from year to year, it is difficult to estimate long-term effects of management practices through short-term (e.g., 1-3 year) field studies. In addition, BMP impact can lag several years. A review of the Rural Clean Water Program estimated a five to fourteen year response time for significant water quality improvement after BMP implementation (Maas et al., 1988), indicating that field studies for BMP evaluation should be long-term. A long-term field study, encompassing pre- and post-BMP periods, is the most direct way to determine if a given BMP produced the desired results. However, when conclusions of such a field study are unfavorable, it is not possible to step back in time and try a different BMP. Thus, long-term field studies are not practical as an exploratory tool in selecting the most appropriate BMP for a particular site.

Additionally, field study replications are difficult due to the inability to control all aspects except study variables. Passage of time prevents exact replication of the studies. Likewise, simultaneously paired fields or watersheds always contribute some variation. Due to variations in topography, soil, climate, and previous land use, BMP effectiveness may vary from site to site (Dillaha, 1990).

Although mathematical models are less accurate than field studies in quantifying pollutant levels, models are 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, 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. Another valuable contribution of computer modeling is in the ability to evaluate multiple scenarios, including implausible and high risk ones, without risk to farmers, the economy, or the environment.

Government agencies often promote soil conservation and water quality improvement through BMP implementation by recommending cost-share programs for agricultural fields meeting certain criteria (USDA-NRCS, 2002; VA-DCR, 2002). Through site visits, on-site measurements, and field-level modeling, government personnel determine if fields meet cost-share criteria. Implementation of cost-share programs involves incorporation of BMPs such as vegetative filter strips, contour strip cropping, and waste storage facilities into current management practices. The decision to enroll a farm in one or more cost-share programs is generally based on whether a particular BMP or set of BMPs is predicted to improve water quality and control soil erosion (e.g., reduce NPS pollution leaving the field).

Cost-share programs may result in reduction of NPS pollution from the field or farm. However, the impact that implementing a particular BMP would have on watershed-level pollution may not be considered. By recommending management practices on a first-come, first-served basis through use of cost-share criteria instead of looking at the entire watershed, management practices may be recommended such that their watershed-level effects duplicate or overlap each other. As a result, management practice costs for the watershed may be unnecessarily inflated. For example, with a conservation tillage cost-share program, all farmers growing suitable crops may be equally encouraged to participate regardless of differing NPS pollution impacts to the watershed from different fields. Additionally, farmer participation in the program may depend heavily on economic or regulatory factors. For example, Royer (1987) found costs significantly impacted the likelihood of farmers to implement reforestation practices. Implementation was more likely when cost-share was available. Hersch (1993) found that farmers adopted cost-share practices in order to comply with strict NPS reduction regulations. These regulations established a maximum allowable phosphorous loading to streams. Instead, by focusing money and efforts on implementing BMPs on farms contributing significant quantities of NPS pollution, higher levels of NPS pollution may be controlled for a similar cost.

1.2 Problem description

A goal in the design of pollution reduction programs is to achieve the greatest possible reduction for a given cost (Heatwole et al., 1987a). Braden et al. (1989) discussed the economic advantage of selectively applying BMPs to reduce NPS pollution. Targeting, a form of selective BMP application, implements stricter pollution control in critical areas, i.e., those areas most contributing to NPS pollution. Braden et al. (1989) stressed that NPS pollution control through targeting is likely to be cheaper and less disruptive overall than applying the same control measures across the watershed, e.g., through cost-share.

The number of ways to allocate BMPs throughout a watershed is exponential with regard to the number of fields. For example, the search space for 50 fields and 4 non-mutually exclusive BMPs encompasses $(2^4)^{50}$ possible placement 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 amount of time.

The intractability of the BMP placement problem has limited the number of procedures developed for locating cost-effective BMP scenarios based on each BMP's location-specific contribution to pollution reduction instead of on satisfaction of rule-based targeting criteria. However, due to increases in computer speeds 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 the 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 problems in environmental policy and resource management have 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.

Previous research, reviewed in Chapter 2, has demonstrated the cost-effectiveness of targeting to control NPS pollution. However, the success of locating the optimal BMP scenario for a specific watershed depends on the ability to consider the complete range of possible scenarios within a watershed and to account for spatial variation and BMP interaction throughout the watershed. Individual applications of targeting do not consider all possible watershed scenarios and may not provide the most cost-effective solution. Additionally, a targeting strategy typically provides a single solution based on the targeting criteria used. Theoretically, a comprehensive approach would find the optimal or a set of near optimal solutions from among all possible scenarios and the relative benefits of the targeted plan could be evaluated compared to those solutions. Computer technology enables the use of optimization techniques to evaluate a large number of scenarios, using a high-level of spatial information. Previous nonlinear programming studies have minimized farmer cost increases based on pollution loading constraints and have used this work to demonstrate the tradeoff in pollution reduction and cost increase (Braden et al., 1989; Das and Haimes, 1979). An optimization heuristic has been shown beneficial in reducing pollution and farmer costs as compared to multiple random scenarios (Srivastava, 1999).

Previous optimization related to BMP placement has combined cost and pollution reduction into a single objective function. As a result, a given pollution reduction level may not be met by all near optimal solutions. Due to the importance of pollution reduction in this research problem, it may be more desirable that pollutant reduction criteria are met to the highest degree possible in all optimal or near optimal solutions. This restriction may be possible by creating a multi-objective function such that scenario costs are considered and optimized only after pollutant reduction criteria have been met.

1.3 Objectives

The overall goal of this research was to increase, relative to targeting recommendations, the cost-effectiveness of BMPs within a watershed. Two specific objectives were necessary to realize this research goal:

1. To optimize BMP placement within an agricultural watershed based on cost and NPS pollution reduction for the watershed; and
2. To determine the economic practicality of using the optimization procedure as compared to a targeting strategy.

The research hypothesis was that a computerized optimization procedure can improve current management practice selection strategies by incorporating explicit spatial information more complex than that used in current targeting methods. By identifying multiple near optimal solutions, as opposed to the single solution provided by the targeting strategy, there might be an increased probability that one of the near optimal solutions will be agreeable to an increased number of farmers. Thus, there may be an increased probability of successfully implementing a more cost-effective solution. The economic practicality of the optimization procedure was assessed through a cost-benefit analysis.

For this research a BMP was defined as a management practice or set of practices that results in reduced pollutant loading at the watershed outlet. Cost-effectiveness was defined, at the watershed level, as the ratio of average annual agricultural NPS pollution reduction to change in farm-level private and public costs as a result of adopting an alternate scenario in place of a baseline scenario. Private cost refers to the farmer's long-term average annual cost as a result of adopting a management practice or set of practices for an agricultural field. Public costs considered within the optimization procedure include governmental expenditures incurred as a result of adopting a management practice or set of practices. In the cost-benefit analysis, the public operational costs of each method were considered in addition to public and private scenario implementation costs.

Chapter 2: Literature Review

2.1 Introduction

A review of the literature demonstrates that targeting has been shown to improve effectiveness of NPS control measures by focusing on critical areas. 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. Five such heuristics are described and compared in this chapter.

2.2 Targeting to improve cost-effectiveness

Because economic and management resources are often limited, researchers and government personnel have suggested that NPS pollution control targeting schemes are necessary to maximize pollution reduction and improve water quality (Dickinson et al., 1990). Dickinson et al. developed and tested a targeting scheme on two watersheds with different spatial characteristics: a lowland and an upland watershed. They then determined the percentage of sediment yield reduction for each watershed achieved by applying four methods for assigning BMPs. These methods included assigning BMPs randomly to all farms, randomly to targeted farms, on a ranked order to targeted farms, and based only on slope steepness. The results clearly showed an environmental benefit in assigning BMPs based on ranked, targeted farms. Dickinson et al. stated that fewer than 30% of NPS remedial measures recommended by agricultural extension personnel are typically implemented. By using targeting to locate BMPs where they will be most beneficial, more of the implemented BMPs may be located in critical areas.

Case studies have shown that erosion and nitrogen (N) control plans reduce farmers' net incomes (White and Partenheimer, 1980; Prato and Wu, 1991; Huang et al., 1997). However, Carpentier et al. (1998) demonstrated that incorporating spatial information into NPS control policies can reduce NPS control costs. They applied BMPs to all farms such that each farm met a uniform criterion of 40% N reduction. They then presented a targeting framework that used farm-level spatial information to meet the N reduction policy while minimizing cost to farmers and taxpayers. The targeting framework applied BMPs to each field such that an average farm-level N reduction of 40% was maintained while total costs were reduced by 75% over the uniform policy costs. Based on a sensitivity analysis of various spatial attributes, Carpentier et al. (1998) concluded that consideration of spatial variability both within and across farms was necessary to accurately determine farm costs for reducing watershed-level N pollution.

Various targeting strategies for NPS pollution reduction have been analyzed. Willis et al. (1994) studied the Everglades Agricultural Area south of Lake Okeechobee. By considering baseline phosphorus (P) levels, hydraulic and water budget models, and various field experiments, Willis et al. (1994) evaluated effectiveness of several categories of BMPs in reducing P loadings. Implementation, operation, and maintenance costs of BMPs were also calculated. Cost and load reduction factors were combined for each BMP to estimate individual cost-effectiveness values. Finally, BMPs were combined by successively adding to the most cost-effective BMP the BMP with the next highest cost-effectiveness. This process was continued until the desired level of P load reduction was achieved. The resulting combination was termed the least-cost scenario for that reduction level. Three levels of P load reduction were considered: 25%, 35%, and 45%. The final choice of BMP scenario then depended only on the desired load reduction.

Heatwole et al. (1987a) evaluated cost-effectiveness of 15 different BMP scenarios for two Florida watersheds draining into Lake Okeechobee. Two water quality models were used for this analysis. A version of the Chemicals, Runoff and Erosion from Agricultural Management Systems (CREAMS) model (Knisel, 1980), was modified into CREAMS-WT (Heatwole et al., 1987b) for the southern Florida region and used to estimate N and P at the field level. A basin scale model, BASIN (Heatwole et al., 1986), was then used to integrate loadings from the fields with loadings from forests, rangeland, and non-agricultural land use areas to develop watershed-level predictions.

Cattle fencing, runoff detention basins, and dairy barn runoff impoundments were considered by Heatwole et al. (1987a) as BMPs. Spatial placement considerations of BMPs were made at the watershed level. For example, distance of pasture from streams and wetlands influenced the amount of fencing used for the pasture. Additional BMP placement considerations included land use, hydrologic soil group, and cattle density.

Heatwole et al. (1987a) calculated N and P load reductions in terms of kilograms per dollar of BMP cost for each scenario for each of the two watersheds. They then combined the results for the two watersheds and provided a ranking of the eight most effective BMP scenarios. The researchers found that BMP cost-effectiveness decreased as the number of BMPs applied to a region increased.

Das and Haimes (1979) presented a multi-objective technique for controlling point source and NPS pollution with respect to environmental quality and economic development. They demonstrated their technique on a river basin connecting three states and covering five government-planning regions. Allowable point source loads for all wastewater treatment facilities in the basin were determined by nonlinear optimization. The objective of the optimization program was to minimize expansion and operation costs over all treatment facilities. Nonpoint source values were lumped within each planning region. Pollution reduction quantities of each management practice for each soil erosion category considered were user inputs to the model. Das and Haimes (1979) developed a reduced gradient optimization algorithm which integrated optimization of point source, NPS, and economic objectives to arrive at a set of final solutions. Their method provided a listing of trade-off values, in cost of pollution control per unit of pollution reduction, for each of the considered pollutants and planning regions. This provided decision makers for each planning region with information about relative costs of pollution reduction methods. Using this trade-off information in combination with public preference and the amount of pollution reduction desired for the region, decision makers were in a position to choose the desired pollution reduction solution for their region.

Braden et al. (1989) discussed a theoretical nonlinear model optimizing BMP selection on a watershed in terms of cost and pollution reduction. They described the SEDiment EConomics model (SEDEC) as an empirical analog to this model. The SEDEC model optimizes pollution reduction and cost by calculating farm profit, field erosion, and sediment transport for numerous watershed scenarios. To improve results the SEDEC model analyzes fields by sections so that each field section falls within a single drainage area and contains a single slope and management practice. The Universal Soil Loss Equation (USLE) is used to calculate gross erosion at the field level.

Four variations of SEDEC were tested, each having a different sediment transport component. Three used fixed sediment transport ratios. The fourth considered the cropping management (C), supporting practice (P), and slope (S) factors of the USLE for determining sediment loss from each field. Transport of this sediment depended on the management practices of downstream fields. The flow path from each field was based on a simple runoff surface in which flow travels from ridge to stream, perpendicular to the contours. Braden et al. cited four studies that concluded that the fourth transport model resulted in close approximations of actual sediment loads. For example, White (1988) found results to be within 10% of the measured loads for one watershed in Illinois.

Braden et al. (1989) demonstrated the SEDEC model on a 431-ha site within an Illinois watershed with mild slopes used mainly for row crops. Several crop rotations and management practices were included in the simulation. Braden et al. (1989) reported the curves of pollution reduction versus marginal cost for the four variations. They found that the results were as expected based on the characteristics of the study.

2.3 Optimization heuristics for intractable problems

Existing optimization heuristics for solving intractable problems include gradient and non-gradient based neighborhood searches as well as methods 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 effectiveness for a specific problem.

For this research, the range of possible combinatorial optimization techniques was considered. Five basic heuristics were chosen and evaluated. The objective was to determine a basic heuristic well suited to this 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 five 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 is founded on statistical theory and is generally easy to implement but works best for problems with continuous input parameters.

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 have a larger probability of contributing to the next generation than do those members with lower fitness scores. 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 (SA) (Eglese, 1990; Swisher et al., 2000) heuristic mimics 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, the crystalline structure becomes fixed. By cooling the solid very slowly, the annealing process attempts to reach the lowest energy state possible and, thus, achieve the most structurally sound crystalline formation for the solid.

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. Throughout the annealing process the heuristic accepts a new value if it reduces the objective function. Additionally, the process will accept, with a controlled probability, an increased value. This alteration to a plain gradient method algorithm helps prevent the process from stopping prematurely at a local optimum. Coding for the SA heuristic is minimal. However, performance of the algorithm is strongly dependent on the algorithm parameters and the problem structure.

2.3.4 Tabu search

A memory-based heuristic, the tabu search (TS) (Bettinger et al., 1998; Glover et al., 1993; Swisher et al., 2000) draws from artificial intelligence concepts. 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

A genetic algorithm (GA) (Goldberg, 1989; Chambers, 1995; Srivastava et al., 1999) is conceptually based on natural selection techniques seen in biological evolution. Basic GAs model individuals of a population as chromosomes, with genes on the chromosome defining relevant traits of the individual. Chromosomes that are judged to be the most fit are the most likely to survive into the next generation, and all chromosomes, regardless of fitness are subjected to random mutations. As a random search algorithm, GA does not require continuity in the input variables. At each generation the GA evaluates multiple solutions, often from different areas of the search space. This parallelism decreases susceptibility to becoming fixed at local minima (Buckles and Petry, 1992).

Nearly all GAs include three basic components: a population of individuals, a function to score the fitness of an individual, and crossover 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. 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. Depending on the nature of the problem, a GA may be set to replace, at each generation, nearly all of the population, only one or two individuals, or an intermediate number of individuals. 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 below a predetermined tolerance or when the score increase has remained below a tolerance for a predetermined number of generations.

2.3.6 Previous use of heuristics with NPS models

These five 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. Then, 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,

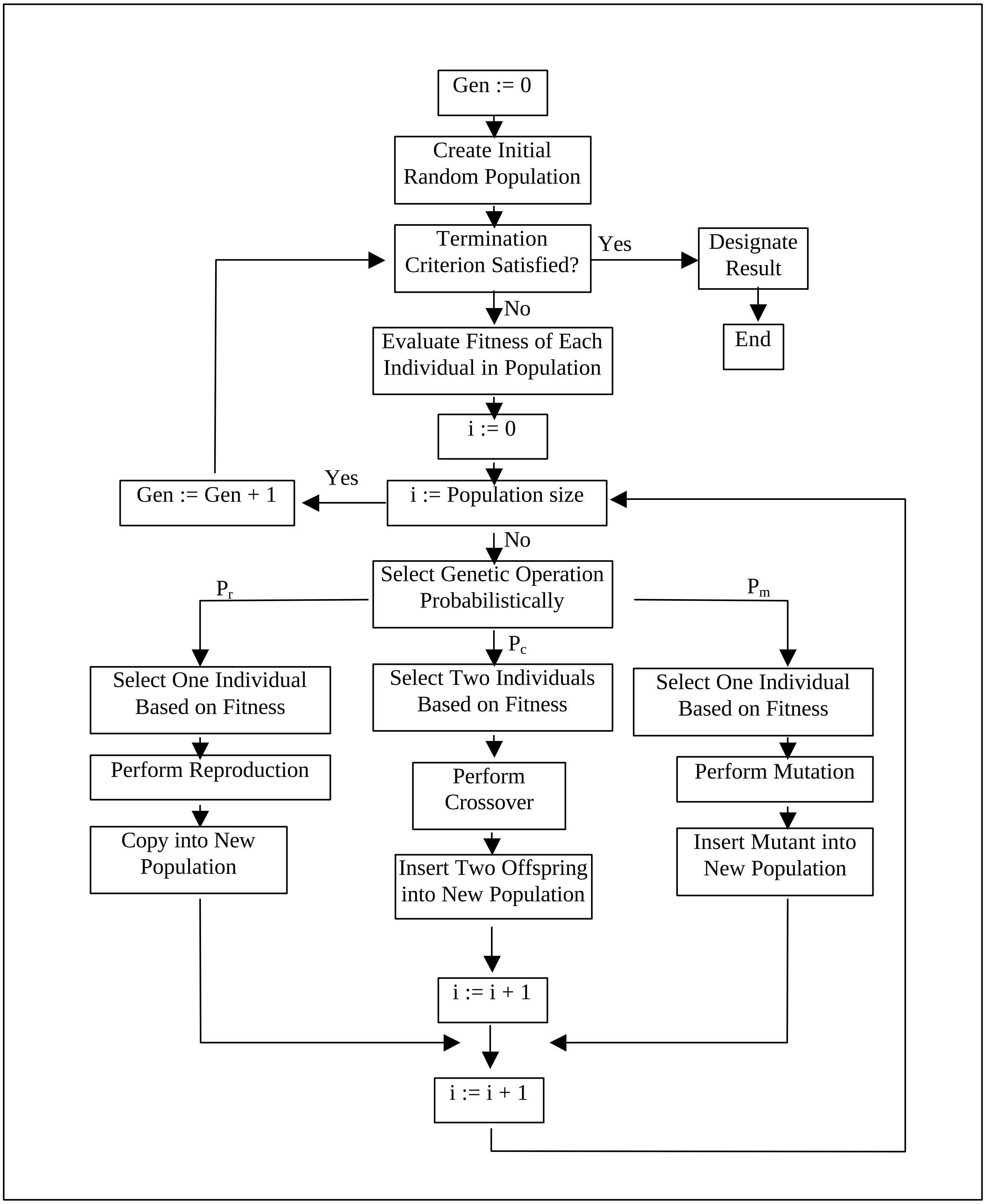

Figure 2.1: Flow chart of general GA (Koza, 1992)

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.

Optimization techniques for intractable problems 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 NonPoint Source pollution model (AnnAGNPS) (Theurer et al., 2001) 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.

2.4 Summary

Targeting focuses on critical areas within the watershed. As a result, targeting often reduces costs as compared to first-come, first-served approaches such as cost-share. A number of studies have developed targeting procedures to enable watershed-specific evaluation of NPS pollution control. Targeting methods incorporating both pollution prediction and economic models have been demonstrated. Additionally, spatial variability at the field and farm level has been shown to be an important aspect of effective targeting.

Two reviewed studies developed nonlinear optimization techniques for improving NPS control cost-effectiveness by decreasing watershed costs while meeting pollutant loading criteria. These techniques estimated NPS pollution using lumped parameter methods. Field-level private costs were determined, but within field variation of soil productivity was not considered, nor were public costs.

Five optimization heuristics for solving combinatorial optimization problems, such as this one, were reviewed. 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.

Chapter 3: Development of Optimization Procedure

3.1 Introduction

The solution set of the BMP placement problem can be characterized as a large number of variables, each with a small range of possible, categorical values. Solving the optimization problem relies on treating the agricultural fields, or management units, as individual variables, with the possible values of the management units being the identifiers for the available BMPs or sets of BMPs to be applied.

Development of the optimization procedure involved first determining which optimization heuristic to use. The five 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 from and economic impacts of each management scenario. These methods were then formulated into a multi-objective optimization function.

The resulting optimization procedure is comprised of three components: a GA, an NPS component for evaluating pollutant loading, and an economic component for assessing public and private costs. Finally, the optimization procedure was implemented as a computer program and tested.

Because of the computer time involved in running a detailed NPS model, using such a model to predict pollutant loadings within the optimization procedure may result in total runtimes of several days. 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 that considers within-field spatial variation and accounts for BMP placement effects. Additional opportunities to expand on existing work included incorporation of public costs and economic impacts of within-field soil productivity into the optimization procedure.

3.2 Choosing an optimization heuristic

The BMP placement problem was determined to be an intractable problem based on the large number of fields within even a small watershed, the exponential number of possible BMP combinations, and the computational complexity in quantitatively comparing watershed-level NPS loading relationships among watershed scenarios or resulting from individual BMPs placed within the watershed. 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 this problem.

Five heuristics for solving intractable problems were selected and considered in more detail 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 (high impact items in bold)

Factor	**Heuristic**				
	Response surface methodology	Shuffled complex evolution	Simulated annealing	Tabu search	Genetic algorithm
Demonstrated performance on BMP placement problems	No	No	No	No	**Yes**
Proven convergence	Yes	Uncertain	Yes	Uncertain	Yes
Formulation ease	Low	Low	**High**	Low	**High**
Continuity or differentiability required	Yes	Yes	**No**	**No**	**No**
Convergence rate	Uncertain	Uncertain	Uncertain	Uncertain	Uncertain
Relative efficiency	Uncertain	High	Low	Uncertain	Medium
Sensitivity to formulation	High	High	High	High	High
Number of initial points required	High	High	Low	Low	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 three heuristics were determined 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)

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 particularly unwieldy 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

The literature was reviewed for problems that specifically related the spatial distribution of agricultural characteristics within a watershed to NPS loadings from the watershed by using one of the five heuristics. Glover et al. (1995) listed a range of problems for which TS has provided high quality solutions. However, problems dealing with modeling of natural systems were not mentioned. Documented use of the SCE method has not extended beyond its development purpose of improving model calibration. Stone et al. (2002) used SA 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 scenarios with regard to NPS pollution control.

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. This portion of the technique is similar to the steepest descent algorithm (Bazaraa et al., 1993), which under the appropriate conditions has been proven to converge at the optimum.

The TS, SA, SCE, and GA are general search strategies for intractable problems and are not intended to enumerate over all possible solutions. Nor are they 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 located, 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 often involves sampling the search space through a factorial experimental design. The output (or response) 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.

Representing BMPs as categorical variables is not an issue for the other four heuristics. However, sets of BMPs cannot always be related to each other in terms of pollution reduction or cost. Thus, there is no easy way of assigning a neighborhood set for each BMP scenario, before calculating the objective function. Instead a NPS model must be used for each evaluation of the objective function. This would require customization of the SCE method, to replace the steepest descent algorithm used for evaluation of individuals within its population complexes. 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. 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 either technique for this research problem would require a mapping between the BMP on each field (a categorical representation) and some measure of fitness or impact of that BMP on the field (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, and GA do not require continuity or differentiability. They require only the capability of mapping each scenario to a fitness function. In this regard, these three heuristics are well suited to the BMP optimization research 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 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 at each 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 compares across the heuristics. Thus, this factor was determined to be “uncertain” for all heuristics (Table 3.1).

3.2.6 Relative efficiency

While convergence rate primarily considers the performance of individual heuristics, the relative efficiency factor was intended 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.

Because of the lack of literature comparing techniques for problems directly involving the correlation of BMP location to watershed NPS loadings, it was difficult to determine which technique was likely to be most efficient for this problem. Thus, the literature search was expanded to look at performance of the five methods in other applications of managing agricultural resources. Several of these techniques have been compared in the area of NPS model calibration. 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 12 000 evaluations. In the same study, SCE and GA techniques averaged 6000 and 9000 evaluations, respectively. Based on this study, these three heuristics were assigned relative rankings of “high”, “low”, and “medium”, respectively. Comparisons of RSM 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”.

3.2.7 Sensitivity to problem formulation

The accuracy and speed of the RSM in locating the optimum of the system modeled is dependent on how accurately the system is modeled by the response function. Choosing a well fitting function for a complex system can be very difficult without some prior knowledge of the characteristics of the system’s response surface. Jacobson and Schruben (1989) summarized a number of studies that modified the RSM in order to provide better fitting response functions. Additionally, Myers (1971) discussed a method for comparing efficiencies of two different response functions for a given problem being solved by RSM.

Duan et al. (1993) showed two variations of the SCE to perform consistently on eight different problems in terms of efficiency and effectiveness. However, in a following study, the same authors (Duan et al., 1994) reported that the SCE method is sensitive, in both efficiency and effectiveness, to the choice of parameter values used in the algorithm.

Unlike with the RSM, formulating the problem representation for the TS, SA, and GA heuristics requires knowledge of input characteristics of the system, not of the system’s response surface. Thus, a solid background understanding of the problem is helpful in formulating the representation.

Values of optimization parameters used by the TS, SA, and GA may vary depending on problem formulation. Tabu search is sensitive to selection rules established, both for moves to avoid and moves to encourage, by the problem formulation (Reeves, 1993). Determining optimal cooling schedule parameters for the SA often requires much experimentation for each problem type (Reeves, 1993). Similarly, success of the GA is dependent on an appropriate problem representation (Mitchell, 1999). Optimal choices of selection and genetic operators to be used for the GA depend on the problem and its representation.

Literature for all five heuristics indicates that their success is sensitive to problem formulation. Because of the lack of previous application of these heuristics to this type of problem, knowledge from previous formulations is available in only one case for the GA.

Thus, the sensitivity to problem formulation factor (Table 3.1) was ranked as “high” for all heuristics.

3.2.8 Number of initial points required

Creators of the SCE (Duan et al., 1993) suggested that the number of points used to start the algorithm be equal to or greater than the dimension of the problem to satisfy the requirements of the Nelder and Mead search strategy. For the RSM also, the number of points needed to fit the polynomial at each step must be at least the dimension of the problem. The dimension of the current problem is one plus the number of management units in the watershed, which can number in the hundreds for small watersheds. The GA also requires multiple initial points to define the starting population. Thus these three heuristics were ranked as “high” with regard to the number of initial points needed. Because the TS and SA techniques each start with a single scenario, they were ranked as “low” with regard to the number of initial points needed.

Both the RSM and the SCE require continuity. Thus, knowledgeable selection of the initial set of points such that they reflect the characteristics of the search space may be helpful in ensuring that the RSM and SCE techniques converge towards the optimum efficiently. However, the initial points for these as well as the other three heuristics may be determined randomly.

3.2.9 Selected heuristic

Based on a subjective comparison of the heuristics with regard to the above factors, the GA was chosen as the optimization heuristic for this problem and used in development of the optimization procedure. Theoretically, the GA is intended to find one or more near optimal solutions within a reasonable amount of time. With regard to this research, a near optimal solution suggests a watershed scenario that meets the cost-effectiveness criteria. This scenario may then be fine-tuned by policy makers to meet individualized farmer needs or used as a starting point for more detailed predictive modeling.

Compared to the other heuristics, the SA and GA seemed more straightforward to formulate in a manner that could accommodate evaluation of different watersheds. Because the SA and GA do not require continuity or differentiability, they are well suited to this combinatorial optimization problem. Both the SA and GA have been proven to converge arbitrarily close to the optimum under certain assumptions. Additionally, unlike the TS, the SA and GA do not require problem-specific selection rules.

Convergence rate and relative efficiency of the GA, in comparison with the other heuristics, were not clear. Performance of the GA in these two areas appeared to be no better or worse than that of the majority of the other heuristics and to, perhaps, be dependent on the specific problem and/or problem formulation. The GA, like the other heuristics, was seen to be sensitive to problem formulation. The previous work with the GA in placement of management practices (Srivastava et al., 1999) was available to provide some insight into a possible problem formulation.

3.3 Development issues

Development of the optimization procedure involved representing the relationships of the physical system (i.e., the agricultural watershed) as a mathematical model. To meet the research goals for the physical system, the model had to provide a way to rank cost-effectiveness ratios through fitness scores and a single objective function. Additionally, it was considered desirable that the optimization procedure prefer farms meeting feed and nutrient management area requirements and that it divide costs as evenly as possible across farms.

The overall goal for the procedure was to determine the most cost-effective scenario where cost-effectiveness was defined as the ratio of pollution reduction to cost increase from a baseline scenario. The total watershed cost for each scenario, including the baseline scenario, is calculated relative to the profit-maximizing scenario as opportunity cost minus net return. Opportunity cost refers to the cost of not choosing the management practice with the highest net return. Thus, the cost increase from the baseline for a given scenario reflects decreased net return as a result of changing management practices.

Under the cost-effectiveness goal, the procedure should find increasingly cost-effective solutions as it progresses from the baseline towards the optimal scenario. However, for the optimization procedure to accurately represent the physical system, optimizing this ratio as a single function raised two issues: one related to the variables of the objective function (i.e., defining fitness scores for the GA) and one involved with the relationship between these variables (i.e., defining the objective function). These issues are addressed in the following two subsections.

After resolving these issues, the procedure was expanded to include consideration of farm area requirements and cost fairness to farmers in BMP allocation (Sections 3.3.3 and 3.3.4, respectively). The area requirement refers to the amount of land needed in cropland and hay/pasture for the farm to produce sufficient amounts of feed and to have sufficient land available for manure/litter spreading. The allocation amounts vary based on farm type. The consideration of cost fairness refers to the attempt to distribute costs evenly among multiple farmers in the watershed. In comparing two scenarios with nearly equivalent total cost for the watershed, the scenario for which that cost is divided most evenly among farmers is preferred by the procedure.

3.3.1 Fitness scores

The cost-effectiveness ratio, pollution reduction over cost increase, can be written as p/c where p and c are real numbers. Mathematically, however, $(-p)/(-c) = p/c$, where p and c are positive real numbers. In terms of this research problem this relationship implies, for example, that reducing 10 Mg/ha of sediment for a cost increase of $100 is equal in cost-effectiveness to increasing sediment by 10 Mg/ha but decreasing costs by $100. In the first situation both terms are positive with regard to the cost-effectiveness definition whereas in the second situation both terms are negative. Thus, both ratios equal 0.10. Additionally, it is mathematically implied that both situations are more cost-effective than reducing 10 Mg/ha of sediment while decreasing costs by $10 (0.10 Mg/ha/$ vs. –0.10 Mg/ha/$).

Consequently, formulating a single fitness score from this cost-effectiveness ratio poses a problem resolving the four potential numerical cases caused by the variables of the ratio. That is, either the numerator or denominator of the ratio may be mathematically positive or negative. For appropriate representation of the physical system by the model, it was crucial to consider which of the four theoretical cases were allowable. This resulted in development of two fitness scores: one for pollution reduction and one for costs.

In the interest of assigning BMPs to decrease NPS pollution from a baseline, pollution increase at the watershed outlet, as a result of altering a BMP assignment, was not an acceptable option; such a scenario would be no better than the baseline scenario in terms of pollution reduction. The modeling impact of this situation was to form a pollution fitness score that focused on positive pollution reduction. As a result, all scenarios that increased pollution as compared to the baseline were given a pollution fitness score of zero. Scenarios that reduced pollution were given a positive fitness score.

In addition, meeting environmental goals often requires pollution reduction to meet a pre-established maximum allowable level. Water quality standards, governmental regulations, or levels established by pollution control policies might set this level. Management practice changes that fail to reduce pollution sufficiently to meet criteria are not likely to be an acceptable allocation of time and expenses. Thus, the allowable level of reduction was incorporated into the pollution fitness score. Scenarios that met or exceeded the pollution reduction criteria were given the highest fitness score. Creating a pollution fitness score in this manner narrowed the possible configurations of the cost-effectiveness ratio by eliminating the possibility of the pollution reduction value being negative.

While it was anticipated that costs would increase from the baseline as BMPs were added, a scenario meeting the pollution reduction criterion and decreasing cost would certainly be acceptable. Thus, in modeling the scenario cost, the economic fitness score had to allow for both increase and decrease in cost as compared to the baseline. Because all cost calculations include opportunity costs, a scenario cost remains positive even if it decreases below the baseline scenario. Thus, the cost score, as formulated, always remains positive. However, cost increase, expressed as change in cost relative to the baseline, may be positive or negative.

The economic fitness score was developed further to account for the extent to which each farm meets any area requirements and to attempt to distribute costs among the farms as much as possible. The cost score has the added benefit of changing little for costs that are near the baseline but increasing rapidly as the cost decreases.

Restricting the pollution reduction score to nonnegative values reduced numerical confusion in evaluating the cost-effectiveness ratio. Also, because the computer program used for the GA portion of the procedure does not permit negative objective function values, restricting the fitness scores simplified transfer of the objective function into program code.

3.3.2 Objective function

As mentioned in the preceding section, using the cost-effectiveness ratio as a single objective function for the GA does not clearly define the response surface to the research problem. By use of a ratio as the sole determination of fitness, a scenario resulting in 10 Mg/ha per dollar is equivalent to a scenario resulting in 200 Mg/ha per $20 and neither scenario is preferred over the other. However, as target reduction criteria are introduced, the scenario preference becomes dependent on which of the scenarios, if any, meet the reduction criteria. A straightforward, realistic solution to this problem was to split the single cost-effectiveness ratio objective function into a multi-objective problem. That is, cost-effectiveness was separated into two objectives: a) meet or exceed the pollution reduction criterion and b) minimize cost increase.

The difficulty of finding a quantitative solution to a multi-objective problem, such as this research problem, is increased when the objectives deal with different quantities (i.e., pollutant load and cost). One method of reconciling multiple objectives into a single set of objective functions is to use a lexicographic method (Coello, 2000; Rentmeesters et al., 1996; Roumasset, 1976). In this method, the objective functions are prioritized in some manner, such as by desirability or importance. Then, the problem is solved by maximizing each objective function in turn.

The multi-objective optimization problem was solved by defining fitness scores that described the objective criteria of pollution reduction and minimal cost increase and by using a lexicographic ordering technique to optimize these fitness scores in sequence. Based on discussion in the previous section, modeling cost-effectiveness was done by first meeting the pollution reduction criteria and then minimizing the cost increase. Using this method, the optimization component finds improved solutions based on fitness scores, which follow, but are not identical to, the pollution loading and cost increase values.

3.3.3 Area requirements

Because the BMPs used in this study affect the crops and forages produced by farms in the watershed, it was important that the solutions not be chosen based on pollution reduction and cost alone. The solutions must also conform to reasonable farming practices. Therefore, the optimization problem was re-examined with regard to extent to which each farm configuration meets the land use requirements for feed and for manure/litter spreading. Three ways of incorporating the area requirements into the optimization problem were considered.

The first involved a separate optimization program to determine a population of scenarios that all met the area requirements for all farms. This would be used as the search space for the BMP placement optimization technique. The main difficulty with this method is making certain that all possible scenarios meeting the area requirements are included in the search space of the GA and that no other scenarios are included. This method would not be applicable to watersheds in which the area requirements cannot be met by any scenario, because the search space would be empty.

The second, very similar, way of incorporation involved enforcing the area requirements lexicographically, before the pollution reduction requirement. Thus, each scenario would first be assessed for meeting the area requirements. If the requirements were not met, an extremely low fitness score would be assigned. If the requirements were met, scenario evaluation would continue with assessment of the pollution reduction requirement. The main drawback with this method was that it evaluates area, pollution, and cost objectives separately, without acknowledging interactions among the variables in these three areas. That is, if management practices are assigned by considering only area requirements and not other factors, such as soil type and topography, the method may not locate the most cost-effective scenario.

When allocating a farm’s production land among management practices, the area requirements impact farm economics with regard to sufficient feed production and nutrient management. However, the area requirement is not necessary for determining if a scenario reduces pollution. Thus, a third method was developed that incorporated the area requirements into the economic component but not into the pollution reduction component. This method was used in the procedure.

The area and economic considerations were combined through a fitness function that increases as the cost per farm for a suggested scenario decreases and the area requirements per farm are realized. When area requirements per farm are not met to the same extent as in the baseline scenario, the fitness score for that farm is reduced. The opportunity cost of the baseline scenario is used to scale the function to allow for different ranges of costs in each optimization run. The opportunity cost for the baseline scenario is calculated by summing, over all management units, the maximum possible net returns provided by the considered management practices.

3.3.4 Cost fairness

By using the lexicographic solution method, BMPs are first located throughout the watershed with respect to their pollution reduction capabilities. This may mean that certain farms are allocated numerous BMPs and incur large cost increases while other farms incur little or no increase in cost. It may be easier to work with only a few farmers to change their management practices, instead of convincing multiple farmers to change. However, if farmers are not willing to implement the selected BMPs, then the solution scenario is not beneficial. In order to improve the likelihood of watershed-wide acceptance and implementation of a solution scenario, it was considered more likely that farmers would be willing to absorb a little cost, particularly if multiple farmers were incurring the same cost, than to be one of a few to absorb the majority of the cost.

Spurlock and Clifton (1982) demonstrated that a NPS pollution control strategy based on the marginal cost of pollution reduction is economically more equitable to farmers than a strategy based on meeting a per acre pollution reduction level. In the optimization procedure, a simple approach to cost fairness at the farm level is taken. Scenarios dividing total farm-level implementation costs more evenly across farms are preferred. Cost distribution is not limited to farms meeting a certain level of pollution reduction. Specifically, a distance metric was included in the economic fitness function to introduce some measure of economic fairness in assigning BMPs throughout the watershed. For example, given two scenarios of equal total watershed cost and meeting area requirements equally, the economic fitness equation calculates a higher fitness score for the scenario in which the costs are divided more equally across the farms.

3.4 Overview of optimization procedure

The optimization procedure is comprised of three parts: an optimization component based on the GA heuristic, a NPS prediction component, and an economic analysis component (Figure 3.1). At each generation of the optimization procedure, the optimization component forms a number of scenarios to consider for addition into the GA population. Then the values (i.e., the management practice identifiers) for each new scenario are extracted from the optimization component and sent to the NPS and economic components. Based on this input, pollutant load and cost for the watershed are calculated. These values are then sent back to

Initialize population

Termination criterion met?

Y

Optimal scenarios identified

N

Generate next population

Calculate economic score

$$E = 1 + e = 1 + \frac{C_o}{\sqrt{\sum_i \frac{x_i^{\,2}}{a_i} + 1}} \qquad (3.6)$$

where
- E = economic fitness score
- e = economic fitness function,
- C_o = total opportunity cost for all farms in scenario [$],
- x_i = cost of working scenario for farm i [$],
- $x_i = (c_w + d_w)_i$,
- c_w = private cost of working scenario for farm i [$],
- d_w = public cost of working scenario for farm i [$],
- $a_i = \min\left\{\left(\frac{aw}{ab}\right)_i, 1\right\}$,
- aw = extent to which working scenario meets area requirements of farm i,
- ab = extent to which baseline scenario meets the area requirements of farm i, and

i indexes all farms in the watershed.

Calculate pollution score

$$P = \frac{\sum_i b_i p_i}{\sum_i b_i} \qquad (3.2)$$

where
- P = total pollutant fitness score,
- βi = weighting factor of pollutant i,
- p_i = score of pollutant i,
- $\sum_i b_i = 1$, and

i indexes all pollutants considered.

$$p_i = \begin{cases} 0 & \text{for } p_b \le p_w \\ \dfrac{p_b - p_w}{p_b - p_t} & \text{for } p_t < p_w < p_b \\ 1 & \text{for } p_w \le p_t \end{cases} \qquad (3.1)$$

where
- p_i = score of pollutant i,
- p_b = pollutant loading from baseline scenario [Mg],
- p_w = pollutant loading from working scenario [Mg], and
- p_t = target pollutant loading [Mg].

Calculate scenario fitness

$$F = \begin{cases} P & \text{for } P < 1 \\ E & \text{for } P = 1 \end{cases} \qquad (3.7)$$

where

F = combined fitness score.

Figure 3.1: Structure of optimization procedure, showing fitness scores used in each component

the optimization component where they are converted into fitness scores and evaluated. After the evaluation process determines which scenarios, both new and existing, to transfer to the next population, the entire process repeats. This continues until the procedure reaches the termination criterion.

Figure 3.1 summarizes the fitness equations associated with each component. The economic component (Section 3.6) performs the calculations needed for the economic fitness score (Equation 3.6). The individual and total pollutant fitness scores (Equations 3.1 and 3.2) are described in Section 3.7. The optimization component combines the economic and pollutant scores, as discussed in Section 3.5, to create the total fitness score (Equation 3.7).

3.5 Optimization component

The main part of the optimization component is the GA heuristic. The GA for this research uses a steady state replacement scheme, in which a given percentage or a set number of the population is replaced each generation. A tournament selection scheme selects two members of the population probabilistically based on the ratio of each individual’s fitness to the sum of all the fitness values. Of these two individuals, the one with the higher fitness score is chosen. The selection process is repeated and the two chosen individuals are used to create two new individuals by reproduction, crossover, and mutation, based on the assigned probabilities of these operations. New members are created and added to the previous generation until the replacement percentage is met. Then the least fit members of the temporarily expanded population are removed from the generation, resulting in a constant population size with each successive generation.

3.5.1 Problem representation

In using a GA, the optimization problem lends itself to a straightforward representation. Each watershed scenario can be thought of as an array of numbers or a chromosome. Thus, a possible solution to the problem is represented as a chromosome and each land use area is represented as a gene on that chromosome.

In nature the value of each gene along the chromosome is chosen from a set of possible values, or alleles, for that gene. In the watershed scenario representation, each member of the array acts as a placeholder for the member’s respective field or management unit; the value in that position represents the specific management practice on that field.

The baseline scenario is the scenario to which each new scenario is compared. This is not required to be the scenario used for initializing the GA. However, meeting this requirement does simplify the process because the allele set for each gene is defined independently based on the value of that gene in the initializing genome. Using the baseline scenario to define the array of allele sets allows the GA to be initialized with a random population where each individual is subject to the constraints of the allele array. Thus, any land use area not in production is assigned a single allele, representing the baseline value, and maintains a fixed set of management practices. Land use areas in production are assigned a set of alleles, corresponding to the set of acceptable BMPs. Baseline management practices can come from the current land uses and management practices in the watershed, from the profit-maximizing scenario (most profitable management practice for each field), or from any other scenario of choice.

3.5.2 Pollution reduction score

The pollution reduction score was developed to scale the pollutant loading of each scenario. Multiple pollutants can be considered in the optimization. The score for a single pollutant is calculated by Equation (3.1).

$$p_i = \begin{cases} 0 & \text{for} \quad p_b \le p_w \\ \dfrac{p_b - p_w}{p_b - p_t} & \text{for} \quad p_t < p_w < p_b \\ 1 & \text{for} \quad p_w \le p_t \end{cases} \tag{3.1}$$

where

p_i = score of single pollutant *i*,
p_b = pollutant loading from baseline scenario [Mg],
p_w = pollutant loading from working scenario [Mg], and
p_t = target pollutant loading [Mg].

Based on the goals of the optimization procedure, scenarios giving a pollutant loading less than the maximum allowable, or target, load are preferred. Scenarios with pollutant loads less than the target load but greater than the baseline load are also included in the optimization process. This is particularly important for case studies in which the target load is met by few or no scenarios. For pollutant loads between the baseline and target loads, the score increases linearly as pollutant load decreases (Figure 3.2). The fitness score of scenarios with pollutant loading larger than the baseline is set to zero, removing these scenarios from the optimization process. The baseline loading was chosen as an upper limit in order to prevent negative fitness scores, but retain flexibility in the use of the optimization procedure over a range of applications.

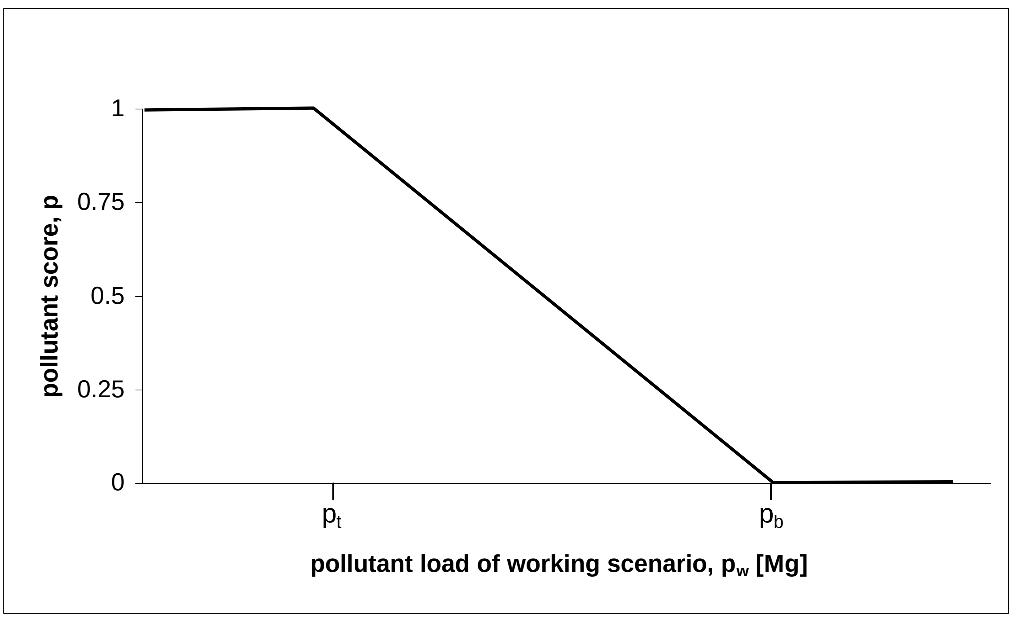

Figure 3.2: Individual pollutant score function

In the case of multiple pollutants of interest, a unique pollutant-targeting criterion can be set for each pollutant and the individual pollutant scores weighted against each other in terms of importance. The weighting factors should be fractions adding up to one. The weighted pollutant scores are combined by Equation (3.2) to create a single total pollutant score, which ranges from zero to one.

$$P = \frac{\sum_i b_i p_i}{\sum_i b_i} \tag{3.2}$$

where
P = total pollutant fitness score,
b_i = weighting factor of pollutant i,
p_i = score of pollutant i,
$\sum_i b_i = 1$, and
i indexes all pollutants being considered.

By incorporating targeting criteria as inputs, the optimization procedure maintains flexibility to numbers and types of pollutants, pollutant weightings, and targeting values. Although the NPS component currently considers only sediment, this process allows multiple pollutants to be considered simultaneously.

3.5.3 Economic score

The economic score was developed for this optimization procedure based on the development issues identified for this problem. It is structured to consider public and private costs as well as farm-level area requirements and cost fairness. The economic fitness function is given by Equation (3.3), using only the positive root of the right-hand side denominator.

$$e = \frac{C_o}{\sqrt{\sum_i \frac{x_i^{\,2}}{a_i} + 1}} \tag{3.3}$$

where
e = economic fitness function,
C_o = total opportunity cost for all farms in scenario [$],
x_i = cost of working scenario for farm i [$],
$x_i = (c_w + d_w)_i$,
c_w = private cost of working scenario for farm i [$],
d_w = public cost of working scenario for farm i [$],
$a_i = \min\left\{\left(\frac{aw}{ab}\right)_i, 1\right\}$,
aw = extent to which working scenario meets area requirements of farm i,

$$aw = \begin{cases} \frac{1}{n} \sum_r \left(\min\{\frac{a_o}{a_e}, 1\}\right)_r & \text{for } n > 0 \\ 1 & \text{for } n = 0 \end{cases} \tag{3.4}$$

n = the number of requirements for farm i,
a_o = area in working scenario contributing toward requirement for farm i [ha],
a_e = area required for farm i [ha],
r indexes all area requirements for farm i,
ab = extent to which baseline scenario meets area requirements of farm i as calculated by Equation (3.4) with $ab = aw$ and a_o = area in baseline scenario contributing toward requirement for farm i [ha], and
i indexes all farms in the watershed.

Scenario costs are used in the economic fitness function instead of using a cost increase quantity. This simplifies calculations by preventing the need to first calculate cost increase before solving the fitness function. The opportunity cost is then used in the numerator to scale the values of the economic fitness function relative to the watershed being evaluated.

The Euclidean distance metric was used in the economic fitness function to help distribute the impact of cost increase among farms. Using this metric instead of simply adding costs across farms results in a more preferable score when several farms each incur a little cost than when a single farm incurs the equivalent cost. For example, consider a three-farm scenario with an opportunity cost of $500 and zero area requirements. If the costs for farm 1 are $240 while the costs for farms 2 and 3 are $0, the value of the economic fitness function is 2.08. However, if the costs for each farm are $80, the economic fitness function value increases to 3.61.

However, the effect of the Euclidean distance metric is somewhat moderated by the public cost. Since each farm incurs the public cost once for one or more BMPs adopted, less total public cost is incurred if the management practice changes are distributed over as few farms as possible. The result is, roughly, that the economic fitness function will tend to prefer change in a minimum number of farms while preferring cost increases to be distributed as equally as possible among those farms. These two issues, working at cross-purposes, are weighted against each other, in a general sense, by the magnitude of the public and private costs, respectively.

A farm may or may not meet area requirements in the baseline or in working scenario, depending on the farm type and size and on the management of production land. For example, a dairy farm that requires all production land to be in corn to supply sufficient feed will no longer meet the area requirement if one of the fields is changed to hay in the working scenario. When the area requirement percentage met by a farm in the working scenario is less than that met by the baseline, a_i is less than one and farm i's contribution to the economic fitness function increases, resulting in a decrease in the economic fitness function. Conversely, when the area requirement percentage met by the working scenario is greater than that met by the baseline, a_i is set to one. Thus, the corresponding farm is not penalized within the economic fitness function.

In the event that $aw = 0$ or $ab = 0$ for a farm with nonzero area requirements, then that variable is set equal to 0.001. This imposes a penalty with regard to that farm, without allowing the denominator of the economic score to approach infinity.

An addend of one is included in the denominator of Equation (3.3), preventing the equation from becoming undefined as the farm-level cost metric, $\sum_i \frac{x_i^2}{a_i}$, goes to zero. Thus, the equation is limited to a maximum value of C_o. The function has a value of $\sqrt{C_o}$ when the farm-level cost metric equals C_o-*1* and continues to rise as the cost metric decreases further, as shown in Figure 3.3.

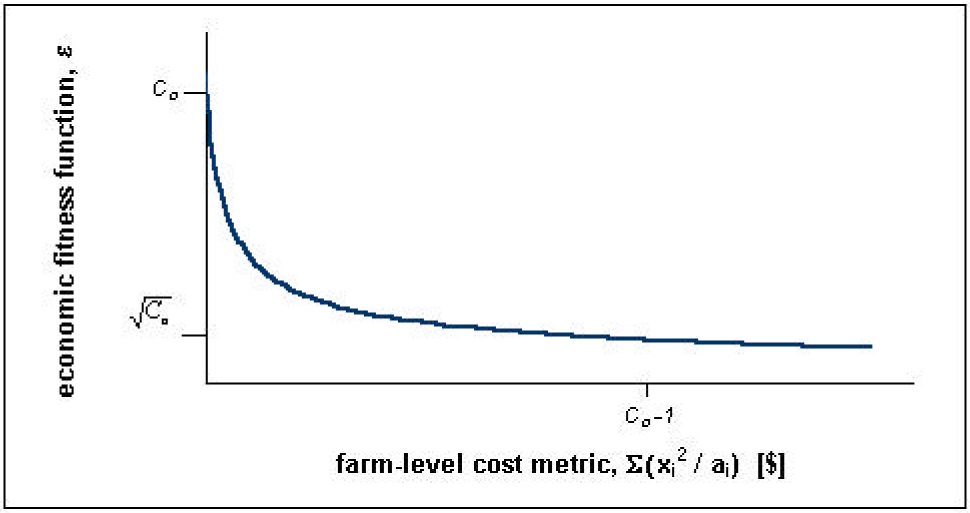

Figure 3.3: Economic fitness function

Finally, the economic fitness score (Equation 3.5) is created by increasing the economic fitness function by one so that the pollutant and economic scores can be combined without overlap. The genetic algorithm uses this feature when sorting scenarios for the next generation.

$$E = 1 + e = 1 + \frac{C_o}{\sqrt{\sum_i \frac{x_i^2}{a_i} + 1}} \tag{3.5}$$

where

E = economic fitness score.

3.5.4 Objective function

The pollutant and economic fitness scores are combined to create the objective function by which each scenario is evaluated (Equation 3.6).

$$F = \begin{cases} P & \text{for} \quad P < 1 \\ E & \text{for} \quad P = 1 \end{cases} \tag{3.6}$$

where

F = objective function (combined fitness score).

Each scenario is first examined to see if its pollutant load meets all pollutant-targeting criteria. Fitness scores are continuous and range from zero to (1+ C_o). All scenarios that meet the pollutant-targeting criteria (i.e., having a total pollutant score of one) are ranked based on their economic scores. Thus, their fitness scores equal their economic scores (ranging from one to 1+ C_o). All scenarios not meeting the pollutant-targeting criteria are ranked by their total pollutant scores so that their fitness scores equal their total pollutant scores (ranging from zero to one). Hence, for each population and for the GA as a whole the scenario that meets all pollutant-targeting criteria and farm area requirements for the least cost has the highest fitness score.

3.6 Economic component

The economic impact to the watershed for a given scenario consists of the sum of private costs, which reflect the farmers' compliance costs due to changing management practices, and public transaction costs, incurred by the government in ensuring that water quality goals are being met (Carpentier et al., 1998). The economic fitness score considers these costs while taking into account the extent to which crop-management practices for each farm meet the requirement of the farm type.

3.6.1 Private costs

Private costs, incurred by each farmer as a result of applying a management practice, are first determined at the field level as opportunity cost minus net return. Opportunity cost refers to the cost of not choosing the management practice with the highest net return. The private cost for each farm is the sum of field costs for all fields in the farm (Equation 3.7).

$$c_i = \sum_j \left(o_{ij} - \left[\sum_k \left(\sum_l y_{ijkl} \right) s_{ijk} - e_{ij} \right] a_{ij} \right) \tag{3.7}$$

where

c_i = private cost for farm i [$],
o_{ij} = opportunity cost for farm i and field j [$],
y_{ijkl} = yield for farm i, field j, crop k, and soil l [qty/ha],
s_{ijk} = selling price of crop k on farm i and field j [$/qty],
e_{ij} = enterprise production cost for farm i and field j [$/ha],
a_{ij} = field area for farm i and field j [ha],
i indexes all farms,
j indexes all fields per farm,
k indexes all crops per field, and
l indexes all soils per field.

For calculation purposes in the program, the gross return for each crop in a rotation must be represented as an average annual amount. Thus, selling prices for each year are annualized over the rotation length for each crop before multiplying by crop yield. The annualized gross returns (selling price * yield) are summed over all crops in the rotation to determine the average annual gross return for the rotation. Discrete compounding factors (Degarmo et al., 1997) are applied as necessary to bring all practices in the rotation to the present value and then to annualize that rotation's production cost over a five-year period. Then average annual net return for the rotation is calculated as average annual gross return minus average annual production cost. For one-year rotations, the annualized gross return per crop yield is identical to the cash crop selling price. An example calculation for a rotation with two years of conventional corn and three years of hay on a 1-ha field of a single soil type is shown in Table 3.2.

The method used for calculating net return in the optimization procedure was compared to two alternative methods for calculating net return and found to give identical results (Appendix A). In one method, the net return for each year was determined before discounting all rotation years to a present value and annualizing. In the second method, both the gross return and production costs for each year were discounted to a present value, annualized, and summed. Then the annual net return was calculated as the difference between the total annualized gross return and production cost.

3.6.2 Public costs

Public costs are calculated for each farm for which a BMP has been added to one or more fields (Equation 3.8).

$$d_i = c_i + e_i \tag{3.8}$$

where

d_i = public costs of a given scenario for farm i [$],
c_i = contracting costs of a given scenario for farm i [$], and
e_i = enforcement costs of a given scenario for farm i [$].

Contracting costs are costs incurred by government agencies while forming agreements with those farmers who are required to change management practices. Enforcement costs include expenses incurred by the government agencies while ensuring contract agreements are met. Carpentier et al. (1998) discussed determination of these costs and provided estimated values. A farm without added BMPs has a public cost of zero. The total public costs for the watershed are the sum of all public costs calculated per farm.

Two additional types of public costs were considered when developing the optimization program: cost-share and information. In cost-share programs, the farmer implements an appropriate BMP and is reimbursed, in part, by a government incentive. Considered at a farm level, the impact of cost-share programs cancels out. That is, the total costs per farm equal the private costs (from which the cost-share amount is subtracted) plus the public costs (to which the cost-share amount is added). In the economic fitness score (Equation 3.3) private and public costs are added by farm before their sum is squared. Thus, cost-share amounts correctly cancel out of the total costs per farm as calculated by the economic score. Hence, the optimization program does not explicitly consider cost-share programs.

Information costs represent the costs involved in generating the optimal solution from the baseline scenario through development and use of the optimization procedure. Since, by this definition, information costs do not vary by run of the optimization scenario, they were not considered within the optimization procedure. Information costs of the procedure as compared to those of a targeting strategy are discussed in Section 4.7 with regard to research objective 2.

Table 3.2: Example net return calculation for a multi-year rotation on a 1-ha field of single soil type (mapping unit symbol 1B: Allegheny fine sandy loam)

Production cost calculation

Year	Crop	Production cost [$/ha]	Bring to end-of-year 1 discount factor[1]		Bring to end-of-year 1 discount * production cost
1	conventional-tillage corn silage	744.33	--	1	744.33
2	conventional-tillage corn silage	744.33	(P/F,9%,1) =	0.9174	682.85
3	grass hay establishment and harvest	527.86	(P/F,9%,2) =	0.8417	444.30
4	grass hay maintenance	359.14	(P/F,9%,3) =	0.7722	277.33
5	grass hay maintenance	359.14	(P/F,9%,4) =	0.7084	254.41
End-of-year 1 total					2 403.23

Production cost annualized over five years [$/ha]

End-of-year 1 * annualization factor ((A/F,9%,5) =0.1671) **401.58**

Gross return calculation

Year	Crop	Selling price [$/tons]	Bring to end-of-year 1 discount factor[1]		Bring to end-of-year 1 discount * selling price	Discounted selling price annualized over five years [$/tons] discounted price * annualization factor[2] ((A/F,9%,5) =0.1671)	Yield [tons/ha] for a 1-ha area	Annualized gross return [$] annualized selling price * yield
1	conventional-tillage corn silage	26.10	--	1	26.10	4.36	44.46	193.90
2	conventional-tillage corn silage	26.10	(P/F,9%,1) =	0.9174	23.94	4.00	44.46	177.89
3	grass hay	46.00	(P/F,9%,2) =	0.8417	38.72	6.47	9.26	59.91
4	grass hay	46.00	(P/F,9%,3) =	0.7722	35.52	5.94	9.26	54.96
5	grass hay	46.00	(P/F,9%,4) =	0.7084	32.59	5.45	9.26	50.42
Annualized total for all crops in rotation [$]:								**536.09**

Average annual net return = Annual gross return **-** Annual production cost = $536.09 **-** $401.58 **= $135.51/ha**

[1](P/F,9%,n) is the single payment, present worth factor for discounting a value *n* years in the future to a present value, using a 9% interest rate. (Degarmo et al., 1997; Table C.12)
[2](A/F,9%,n) is the uniform series, sinking fund factor for distributing a future value evenly over *n* years, using a 9% interest rate. (Degarmo et al., 1997; Table C.12)

3.7 NPS component

In developing the NPS component for this research, existing watershed-level NPS models were considered. A continuous NPS model would best account for the long-term impacts of BMPs by including climatic and growing season effects on water and NPS pollutant movement within and out of the watershed. Additionally, because spatial location of BMPs is an important design component of this optimization problem, the ability of the NPS model to represent variation between fields was essential. Soils often vary within fields, impacting both erodibility and crop yields within the field. Thus, consideration of variation within fields, if possible, was desirable. Since the genetic algorithm must calculate NPS pollution loading for each scenario, the NPS model simulation is necessary for each member of each generation. Thus, runtime for the NPS model was an important consideration.

Based on these factors, the applicability of commonly used, continuous NPS models to this research problem were reviewed. Four models, commonly used for NPS modeling, were eliminated from consideration because they are not distributed at a field level.

- WEPP95 (USDA-ARS. West Lafayette, Indiana. http://spc3.ecn.purdue.edu/weppdoc/WEPPUserSummaryCover.html Accessed 01 November 2001),
- HSPF (USGS Hydrologic Analysis Software Support Program. http://water.usgs.gov/software/hspf.html Accessed 01 November 2001),
- SWRRB-WQ (US Environmental Protection Agency. Washington, D. C. http://www.epa.gov/OST/SWRRB_WINDOWS/. Accessed 01 November 2001),
- SWAT2000 (USDA-ARS. Temple, Texas. http://www.brc.tamus.edu/swat/swatapp.html. Accessed 01 November 2001),

WEPP95 has discretization units of hillslopes and channels and is intended for use in watersheds less than 260 ha. Additionally, WEPP95 is limited to 75 hillslopes and 75 channel sections. HSPF is a large-scale, lumped parameter model for watersheds up to 100 000 ha in size. However, it is limited to 200 total operations, including the number of discretization units. SWRRB-WQ and SWAT are lumped parameter models intended for watersheds of 30 000 ha or larger and discretization units of subbasins.

The following two models can both account for variation within fields and, thus, were considered in more detail.

- AnnAGNPS Pollutant Loading Model (USDA-ARS. Oxford, Mississippi. http://www.sedlab.olemiss.edu/AGNPS.html Accessed 01 November 2001), and
- ANSWERS-2000 (Bouraoui and Dillaha, 1996).

AnnAGNPS and ANSWERS-2000 both allow for grid cells of less than one hectare, and can simulate watersheds up to 10 000 ha in size. However, because of the number of cells needed to adequately describe the spatial variation of a watershed, total runtimes for AnnAGNPS and, in particular, ANSWERS-2000 are extremely lengthy when multiple runs are needed. For example, a watershed divided into 6457 cells takes 45 minutes to complete a one-year simulation in ANSWERS-2000 when using a 1.2 GHz processor (R. W. Zeckoski, personal communication, Biological Systems Engineering Department, Virginia Tech, Blacksburg, Virginia, 17 January, 2002). A 7623 cell watershed takes about three hours to complete a five-year simulation in ANSWERS-2000 when using a 1.6 GHz processor. Additionally, both programs use historical or simulated climatic data, as opposed to an average annual climatic data set. Thus, multiple simulation years (e.g., 20 years or more) are necessary to adequately estimate average annual pollutant loadings (Heatwole et al., 1990).

A watershed of 143 cells takes 11 seconds to complete a one-year simulation in AnnAGNPS when using a 433 MHz processor (K. A. Flahive, personal communication, Virginia Tech, Blacksburg, Virginia, 10 January, 2002). Although 143 cells are much fewer than needed to adequately describe within field spatial variation for most watersheds, a GA requiring10 000 five-year, AnnAGNPS simulations of this watershed would take about seven days of computer runtime, assuming a linear increase in computation time for multiple years. The GA used by Srivastava et al. (1999) involved 3825 four-year AnnAGNPS evaluations of a 105-cell watershed. On a Pentium II 333 MHz computer, a single optimization run took a week of continuous run (P. Srivastava, personal communication, The Pennsylvania State University, University Park, Pennsylvania, 22 June, 2001).

For this research, discretization such that each field was simulated with one or more discretization units was needed to compare the impacts of BMPs with regard to their location in the watershed. Additionally, within field variation was desired to fully utilize the spatial data available. Consideration of soil and slope variation within fields can increase accuracy of sediment loss and economic predictions. However, current NPS models with adequate levels of discretization require prohibitive amounts of computer runtime for the number of evaluations needed by an optimization heuristic. It was desired that the optimization program run within a time frame reasonable for watershed planners. Since runtime is dependent on computer speed, the speeds of computers available to most users were a factor of consideration. A reasonable time was considered to be less than one day for small watersheds with few (<10) management alternatives per field when using a 1.6 Ghz computer. This allows optimization time of larger watersheds and/or large sets of management alternatives to remain feasible, perhaps overnight or over a weekend. Also, it maintains a more feasible runtime if slower computers are used.

To meet the within field discretization criterion, each watershed was discretized into cells smaller than most fields (0.09-ha cells). To meet the time criterion under this level of discretization, a NPS component was developed to determine cell-level gross erosion and route eroded sediment to the watershed outlet through downstream overland and channel cells. Use of a geographic information system (GIS) enabled the desired level of discretization and facilitated simultaneous cell-level calculations across the watershed. Also, it facilitated pollutant routing.

3.7.1 Gross erosion

The Universal Soil Loss Equation (USLE), shown in Equation (3.9) (Schwab et al., 1993), was used for the cell-level gross erosion model.

$$A = RKSLCP \tag{3.9}$$

where

A = average annual soil loss [Mg/ha],

R = combined rainfall and runoff erosivity $\left[\frac{MJ \cdot mm}{ha \cdot h \cdot y}\right]$,

K = soil erodibility $\left[\dfrac{Mg \cdot ha \cdot h}{ha \cdot MJ \cdot mm}\right]$,
S = slope steepness factor,
L = slope length factor,
C = cover-management factor, and
P = supporting practices factor.

The S and L factors are calculated based on Equations (3.10) and (3.11), given by Schwab et al. (1993).

$$\begin{aligned} S &= 10.8 \sin\Theta + 0.03 \quad \text{for } \Theta < 5.14 \text{ degrees} \\ S &= 16.8 \sin\Theta - 0.50 \quad \text{for } \Theta \geq 5.14 \text{ degrees} \end{aligned} \tag{3.10}$$

where
Θ = slope steepness in degrees.

$$L = \left(\frac{l}{22}\right)^m \tag{3.11}$$

where
l = slope length [*meters*], and
m = L-factor exponent = $\dfrac{\sin\Theta}{\sin\Theta + 0.269\,(\sin\Theta)^{0.8} + 0.05}$

The USLE is widely used and has been tested on over 10 000 plot-years of data (Foster, 1982). Unlike AnnAGNPS and ANSWERS-2000, the USLE predicts only erosion. However, it as been shown to be “acceptably accurate in many cases” for long-term average annual erosion (Foster, 1982; p. 98). The computer runtime of the USLE model is negligible, as compared to AnnAGNPS or ANSWERS-2000 for two reasons. First, the USLE predicts only average annual values. Second, using a GIS, cell-level or field-level USLE predictions can be done simultaneously within the watershed instead of one cell or field at a time. Although this prediction method does not allow for considering effects of single storms or seasonal variation, the results do estimate long-term impacts of established management practices.

A GIS-based model was developed to model gross erosion using the USLE. This provided the capability to model spatial variation systematically across a watershed. By using discretization units smaller than management units, spatial variability of soils and topography within management units is also considered. Required data for the gross erosion model include the USLE R and K factors, USGS 30-m digital elevation model (DEM), management unit boundaries, and land use and management practices for each unit. The *K* factor and land use data layers can be divided, or discretized, into cells to match the resolution of the DEM. Required data for the S and L factors include slope steepness, obtainable from a DEM, and characteristic field slope length, obtainable from a local resource conservationist or from field measurements. In cases where the impacts of specific fields have been predetermined to be of particular interest, or in watersheds where field slope lengths vary greatly, the increased accuracy provided by field measurements may be desirable. In such cases, a similar degree of detail and accuracy may be needed for other data inputs to produce the desired accuracy. For planning purposes in comparing the watershed response to alternate scenarios, characteristic slope length was judged to be at a similar level of detail and accuracy as the other data layers. Additionally, the C and P factors must be defined for each crop-management practice to be considered.

3.7.2 Sediment routing

The main goal of the sediment routing component was to account for downstream effects on sediment delivery, such as variations in land use, flow length, and slope steepness. To account for interactions between neighboring BMPs, it was desired to consider spatial variation in sediment delivery at the smallest available level (i.e., the GIS cell). To this end, a method was needed to calculate the delivery ratio for each GIS cell. This cell-level delivery function could then be applied both to gross erosion generated within a cell and to sediment that flows into a cell. Next, the delivery from each cell needed to be routed along the flow path. Using this process of routed delivery, the amount of net sediment yield from each area of the watershed that reaches the outlet can be determined.

Prediction of sediment routing and delivery is complicated by the stochasticity and number of factors involved in sediment transport and deposition processes (Novotny and Olem, 1994; Walling, 1983). One approach for sediment routing prediction involves the use of transport equations, which consider flow and sediment characteristics at a detailed level. A broader approach is to use delivery ratios, which compare sediment yield and sediment erosion over a given land area.

Guy et al. (1992) and Julien and Simons (1985) reviewed a number of sediment transport equations developed for channel flow, such as the Du Boys, Yalin, Yang, and Schoklitsch equations, and evaluated their performance with regard to overland flow. Transport capacity is a function of slope, flow rate or depth, shear stress, and particle size. Because the delivery function for this research is to be used in conjunction with the USLE, which predicts average annual soil loss, information is not available at a sufficient level of complexity to justify use of detailed transport equations. Estimation of parameters within the transport equations without sufficient detail, or generalization of the equations to predict annual sediment delivery, would introduce error into the model instead of estimate sediment yield more accurately.

Walling (1983) reported several delivery ratio equations, suggested by a number of researchers, for determining a single sediment delivery ratio for a given drainage area. Other researchers have used a ratio of measured soil loss to modeled gross erosion to determine a single delivery ratio for a watershed (Sheridan et al., 1982; Johnson et al., 1980). However, because the optimization technique of this research depends on BMP placement within the watershed, a single delivery ratio for the watershed is not adequate. Overland transport and deposition of sediment is a function of the BMPs within the flow path. Thus, a single watershed delivery ratio would not reflect varied BMP placements. Instead, the delivery of sediment from any given area of the watershed to the receiving stream is expected to vary based on BMP selection and placement within that flow path.

3.7.2.1 Cell-level delivery function

Sediment delivery through each cell is modeled by a set of equations that distinguish between overland and channel flow. Sediment delivery through cells under overland flow is represented by Equation (3.12). Development of Equation (3.12) is discussed below, followed by presentation and discussion of the channel flow delivery equation (Equation 3.15).

$$d = \max \left\{ a \sqrt{\frac{s}{l}}, \quad 1 \right\} \tag{3.12}$$

where

d = sediment delivery ratio through an overland cell,
α = land use coefficient [dimensionless],
s = slope steepness across cell [m/m], and
l = length of flow path across cell [m].

The slope steepness and length of flow path across each overland cell are determined by a GIS. The α-value was calculated for each management practice used in evaluation of the optimization procedure, as discussed below.

Novotny and Olem (1994) listed land cover and slope as key factors in affecting delivery rates. Additionally, they stated the importance of factors specific to storm events, such as rainfall impact, infiltration, ponding, and overland flow energy. However, because this research uses average annual erosion, consideration of storm specific factors was not feasible. Instead, sediment delivery was related more generally to overland flow velocity. This was done by basing the overland sediment delivery function (Equation 3.12) on the SCS flow velocity equation (Equation 3.13) (Haan et al., 1994).

$$v = as^{1/2} \tag{3.13}$$

where

v = velocity [m/s],
s = slope [m/m], and
a = land use coefficient.

This equation is applicable to overland and shallow channel flow. Also it considers the effects of land use and slope.

Watershed-level sediment delivery is a complex function of individual watershed characteristics. In particular, multiple studies, summarized by Walling (1983) have shown sediment yield at the watershed outlet to decrease as watershed area increases. Additionally, Walling (1983) summarized sediment delivery prediction equations developed for several regions of the United States. These equations proposed that sediment delivery ratios at the watershed level also decrease as watershed area increases. The prediction equations are functions of watershed area, relief, length, and slope.

The research summarized by Walling (1983) indicates that both slope and flow length are significant factors in predicting sediment delivery. Additionally, the inverse relationship between sediment delivery and watershed area suggests an inverse relationship between sediment delivery and overland flow length. Thus, to create a cell-level delivery function, the right side of Equation (3.13) was divided by the square root of the flow length on a per cell basis. Next, a new land use coefficient, α, appropriate for determining sediment delivery rates, was developed to replace the land use coefficient, *a*, from Equation (3.13), which is appropriate for determining velocity. The resulting equation (Equation 3.14) was used to calculate cell-level delivery ratios.

$$d = a\sqrt{\frac{s}{l}} \qquad (3.14)$$

As an empirical coefficient, α can be determined using two approaches. One approach includes use of measured sediment yield or delivery data along with slope and length. Another approach is to predict sediment delivery using an NPS model. After collecting data with either method, Equation (3.14) can then be solved for α.

Information available in the literature was insufficient to determine α values. Long-term or average annual yield data were found only for a few watersheds (Hrissanthou, 1990; Yitayew et al., 1999; USDA-ARS, 2002), several of which did not include published spatial land use data. One small watershed (Yitayew et al., 1999) was in a single land use of desert shrub. The watershed scenario in another study (Hrissanthou, 1990) consisted of only non-agricultural practices: forest, meadow, rock, and urban area.

Because use of measured data was not an option, the field-scale NPS model, RUSLE2 (University of Tennessee and USDA-ARS. http://bioengr.ag.utk.edu/rusle2/ Accessed 04 March 2002), was used to predict sediment delivery. In addition to being a field-scale model, RUSLE2 was chosen because it calculates and reports both sediment loss and gross erosion over different management practices along a hillslope. From this output sediment delivery ratios can be determined for varying management practices, slope steepnesses, and slope lengths. Additionally, RUSLE2 was designed to calculate average annual soil loss. This feature corresponded well with the NPS component, which also calculates on an average annual basis.

In order to predict sediment delivery from a given management practice, a slope profile in RUSLE2 was divided into two sections (Figure 3.4). The upper section was placed in fallow on a five percent slope in order to produce erosion. The length of the upper section was chosen to ensure sufficient yield entered the lower section to allow for a positive yield leaving the profile after deposition occurred on the lower section. To predict sediment delivery rates appropriate to the study area, a regional climate station (Staunton, Virginia) was used for simulation, as was a characteristic soil (silt loam) in the upper section of the slope profile.

In the lower section of the slope profile, varying slope steepnesses, slope lengths, and management practices were evaluated. The soil of the lower section was defined as non-eroding so that no gross erosion was simulated for the lower section. Thus, erosion leaving the upper section underwent deposition in the lower section. The amount of deposition was a function of the management practice, slope steepness, and slope length of the lower section. This allowed the delivery ratio for the lower section to be estimated as the ratio of the net soil loss from the slope profile to the net soil loss from the upper section. These two values of net soil loss are provided as output by RUSLE2.

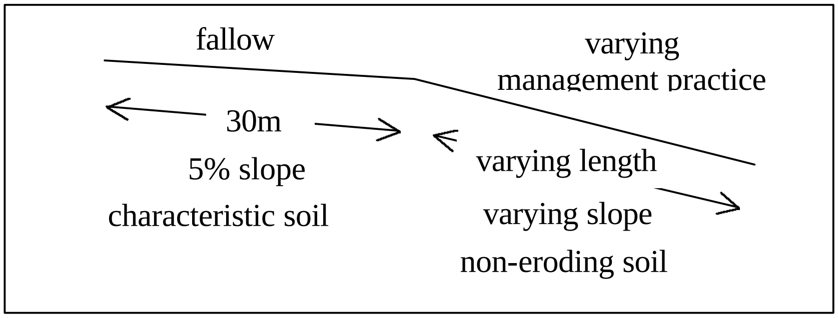

Figure 3.4: Slope profile used in RUSLE2 to estimate delivery ratios

To determine α values, three lengths (30m, 60m, and 90m) were evaluated for the lower section. Additionally, for each length and for each crop management practice, the slope of the lower section was varied from two to ten percent. For alfalfa, which can be grown on steeper slopes, the slope of the lower section was varied from two to 15 percent. Delivery ratios were calculated using the soil loss predictions from RUSLE2 for each combination of slope, length, and management practice.

As expected, delivery ratios increased for increasing slopes when the other two parameters were held constant. For longer slope lengths with constant slope and management practice, delivery ratios decreased. Also, for management practices with higher levels of land cover, delivery ratios were lower.

Equation (3.14) was solved for α for each delivery ratio calculated. The value of α was found to vary as a function of length. Since it was expected that most USGS DEMs used with the NPS component would have a resolution of 30-m, α was developed for a 30-m flow length. Reassessment of appropriate α-values should be made for alternate cell sizes. This can be done by using the described RUSLE2 procedure with a lower profile segment of length equal to the desired cell width. The α values for the range of slopes considered at the 30-m length are shown in Table 3.3.

Table 3.3: Values of α by management practice for low and high slopes, over a 30-m flow length

Slope	Conventional tillage corn silage	Conventional tillage corn silage with winter grain cover	Minimum-till corn silage	Minimum-till corn silage with winter grain cover	Alfalfa
2%	10.9	5.3	4.0	2.3	2.3
10%	8.4	7.0	5.7	3.9	
15%	--	--	--	--	4.2

The average α value was calculated over the low and high slopes and assigned to that crop-management practice (Table 3.4). Because both alfalfa and grass hay in the study region are reestablished on a rotational basis and consist of plants that are less sod forming than pasture (personal communication, Natural Resource Technician, NRCS, Harrisonburg, Virginia, 04 March 2002), alfalfa and grass hay were assigned the α value calculated for alfalfa. Because RUSLE2 is primarily for cropland use and is newly released, pasture, forest management, and farmstead scenarios were not available. Pasture in this area consists of sod forming grasses and is well maintained (personal communication, Natural Resource Technician, NRCS, Harrisonburg, Virginia, 04 March 2002). Thus, the pasture delivery ratio was assumed to be one-half the delivery ratio of alfalfa and the corresponding α value was calculated. Forest was judged to have a lower delivery ratio than pasture due to undergrowth and litter layer. Thus, the forest delivery ratio was assumed to be one-fourth the delivery ratio of alfalfa and the corresponding α value was calculated. The farmstead land use includes buildings and surrounding lawn, driveways, and parking areas. Based on the delivery ratios calculated for conventional-tillage corn silage and judging that farmstead land would be generally more impervious than cropland, delivery ratios of twice the conventional-tillage corn silage delivery ratios were used to solve for α. Using the described procedure with RUSLE2, additional land uses can be included as necessary.

Table 3.4: Sediment delivery α-factors by land use for a 30-m flow length

Description	α
Farmstead	19.2
Conventional tillage corn silage	9.7
Conventional tillage corn silage with winter grain cover	6.2
Minimum-till corn silage	4.9
Alfalfa / grass hay	3.3
Minimum-till corn silage with winter grain cover	3.1
Pasture	1.6
Forest	1.1

The relationships between independent and dependent variables in the sediment delivery equation are shown in Figure 3.5. Because of the square root relationship, the delivery factor for a given land use and flow length increases rapidly at first as slope steepness increases from zero percent. As the slope increases further, the rate of increase for the delivery function slows. This corresponds to the logic that once a critical slope steepness is reached for a given land use, the near maximum delivery level will be achieved. Similarly, for a given slope length, the delivery factor increases more quickly as the land use trapping efficiency decreases.

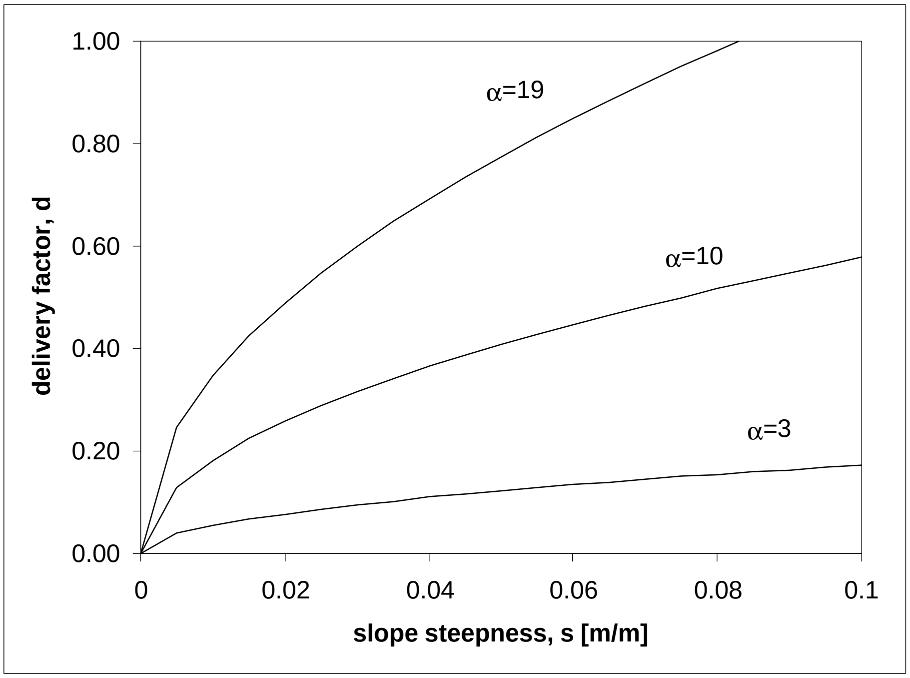

Figure 3.5: Delivery factor for varying slope steepnesses and α values; slope length = 30m

A much higher delivery is expected by channel than by overland flow. To model this distinction a separate sediment delivery equation was developed for two types of channels: shallow concentrated flow and stream flow through ephemeral and perennial streams. The distinctions are made in a GIS using a flow threshold for the number of upstream cells accumulating to create a channel cell.

The use of DEMs and a flow accumulation threshold to represent the stream network is widely used in GIS applications (Garbrecht and Martz, 2000). One method is to first determine the ephemeral and perennial stream network from USGS 7.5 minute series topographic quadrangles. This network can then be compared to the GIS flow accumulation layer created from the DEM in order to determine an accumulation threshold. For example, in this study, a flow accumulation threshold of 200 cells best matched the topographic stream network for the evaluation watershed, Muddy Creek (described in Section 4.2). An accurate depiction of the stream network using this method depends on the DEM resolution and watershed characteristics. A different threshold may be appropriate for different watersheds.

Overland sheet flow generally becomes shallow concentrated flow within 91m (300ft) (Akan, 1993). To determine a flow accumulation threshold for shallow concentrated flow, overland flow length GIS layers for a variety of flow accumulation thresholds less than 200 cells were considered. A 60-cell threshold resulted in a median overland flow length of 100-150m, which approximates the 90m length by which flow typically becomes concentrated. Thus, an accumulation threshold of 60 cells was used to distinguish shallow concentrated flow channels.

In small subwatersheds in a ridge and valley agroecosystem, it was estimated that little deposition would occur in main streams over the flow length of the watershed (T. A. Dillaha, personal communication, Biological Systems Engineering Department, Virginia Tech, Blacksburg, Virginia, 8 March, 2002). The maximum flow lengths of the two evaluation subwatersheds, Mini-Muddy Creek and Lola Run, are approximately 4243m (140 cells) and 6118m (205 cells), respectively. To facilitate a low level of deposition over these lengths, a cell-level delivery rate of 0.9998 was selected. Due to the multiplicative nature of the sediment routing function (Section 3.7.2), a 0.9998 delivery rate delivers 96% across a distance of 200 cells. This corresponds to a deposition of 4% of the sediment into the streams.

It was estimated that channels of shallow concentrated flow should have a delivery rate greater than the highest overland flow delivery rate but less than that of the main stream network. Using the RUSLE2 method previously described, the overland flow delivery rate estimated for farmstead land use was 56% for a 2% slope and 96% for a 10% slope, over a single 30-m cell. The median overland flow distance to reach the shallow concentrated flow network (60-cell threshold) is about four cells. Thus, the maximum overland delivery rate of 96% for a single cell results in a delivery rate of 85% over four cells. In contrast the main stream network delivers 99.98% per cell and about 96% over the watershed flow length.

A single cell delivery rate of 0.98 for shallow concentrated flow was chosen for two reasons. First, this delivery rate is between the single cell delivery rates for overland and stream flow (0.96 and 0.9998 respectively). Second, the difference in median overland flow distances between the 60- and 200-cell thresholds suggests a median shallow concentrated flow distance of about 100m, or 3 cells, before reaching a larger stream. Thus, the shallow concentrated flow delivery rate of 0.98 per cell results in a 94% delivery rate over 3 cells. This results in slightly more channel deposition than realized in the larger stream network (6% versus 4%).

Equation (3.15) describes the cell-level sediment delivery ratios developed for channel flow in the Muddy Creek watershed.

$$d = \begin{cases} 0.98 & \text{for shallow concentrated flow} \\ & \text{(flow accumulated from} \geq 60 \text{ and} < 200 \text{ 30-m GIS cells)} \\ 0.9998 & \text{for flow through ephemeral and perennial streams} \\ & \text{(flow accumulated from} \geq 200 \text{ 30-m GIS cells)} \end{cases} \tag{3.15}$$

where

d = sediment delivery through each channel cell.

Channel distinction was added since the overland equation does not account for flow depth or concentration within cells and, consequently, the higher velocities that exist in channel flow. As a result, the entire cell containing a stream is assigned the relevant stream delivery value; overland sediment moving to the channel is not treated separately for cells containing streams.

For different sized watersheds it may be desirable to adjust the cell sizes or to modify the channel definitions or delivery levels. For example, in large watersheds with long stream channels, the delivery factor of 0.9998 over 500 30-m channel cells results in a total delivery rate of 90% for that stream reach. The appropriateness of this overall delivery rate will depend on the size and flow characteristics of the stream.

3.7.2.2 Routing function

Sediment from each cell is routed through downstream cells to the outlet by Equation (3.16).

$$Y_i = A_i a_i \prod d_j \tag{3.16}$$

where

Y_i = sediment loss of cell i reaching the outlet [Mg],
A_i = gross erosion from cell i [Mg/ha],
a_i = area of cell i [ha],
d_j = sediment delivery ratio of cell j, and
j indexes all flow path cells between cell i and the outlet.

The routing process is illustrated for a single cell in Figure 3.6. The arrows show the flow path from cell 1, through cells 2 and 5 to the outlet (cell 9). Using Equation (3.16), the sediment delivery to the watershed outlet for cell 1 is calculated as:

$$Y_1 = A_1 a_1 d_1 d_2 d_5 d_9 \tag{3.17}$$

Summing the sediment loss reaching the outlet (i.e., the Y_i's) over all cells and dividing by the watershed area gives the sediment yield of the watershed in Mg/ha. This method is similar to that used by Kothyari and Jain (1997) for routing sediment in forested watersheds.

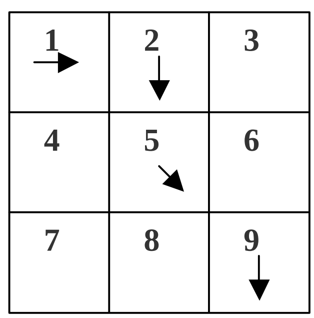

Figure 3.6: Routing of a single cell to the watershed outlet

The product of the delivery ratios can be rewritten into an additive exponential function (Equation 3.18).

$$\prod d_j = e^{\sum \ln(d_j)} \tag{3.18}$$

where

d_j = sediment delivery ratio of cell j, and
j indexes all cells in the downstream flow path of a given cell for both the product and the natural logarithmic function.

The summation in Equation (3.18) is very similar to a summation of travel distance along a path, which can be calculated by a function, *FlowLength*, within the ArcView GIS (Ver 3.2. Redlands, CA: Environmental Systems Research Institute). The *FlowLength* function

(Equation 3.19) determines the downstream flow path distance from a given cell to the outlet by weighting the length across each cell by an impedance value.

$$FlowLength_i = \sum \text{travel\_distance}_j \times \text{impedance}_j \tag{3.19}$$

where

j indexes all cells in the downstream flow path of cell i.

The *FlowLength* function closely approximates the summation of sediment delivery ratios (Equation 3.20).

$$\sum \ln(d_j) \approx -FlowLength\left(t_j \times \frac{-\ln(d_j)}{l_j}\right) \tag{3.20}$$

where

t_j = travel distance of flow between cell j and the next cell in the flow path,

$\left(\frac{-\ln(d_j)}{l_j}\right)$ = impedance value of cell j,

l_j = flow length assigned to cell j, and

j indexes all cells in the downstream flow path of a given cell for both the summation and the *FlowLength* function.

As a result, Equation (3.18) is closely approximated by Equation (3.21).

$$\prod d_j \approx e^{-FlowLength\left(t_j \times \frac{-\ln(d_j)}{l_j}\right)} \tag{3.21}$$

where

j indexes all cells in the downstream flow path of a given cell for both the product and the *FlowLength* function.

ArcView determines flow length of each cell as the length from the center of that cell to the center of the next cell. Correspondingly, the *FlowLength* function calculates weighted distances by multiplying travel distance from center to center of each grid cell with the average impedance from each pair of cells spanned. When the *FlowLength* function is used to calculate dimensionless weights, the travel distance and the flow length within the impedance factor do not always cancel out. For example, in square cells of Figure 3.6, the flow direction of cell 5 is diagonal while the flow direction of cell 9 is vertical. Thus, the flow length of cell 5 is a factor of $\sqrt{2}$ larger than the flow length of cell 9. Because the travel distance between these cells is equal to the flow length of cell 5 and the flow lengths assigned to each cell are unequal, the travel distance and flow length terms, as used in Equation (3.20), do not cancel completely. When flow direction is either parallel to the cell edge for both cells or diagonal for both cells, the function is equivalent to an arithmetic summation. However, for every two cells combining diagonal and non-diagonal flow, the sediment delivery weighting ($\ln(d_j)$) of one cell will by multiplied by $\sqrt{2}^{\pm 1}$. Because the multiplication term alternates depending on the flow directions of any two cells, it is estimated that the total error introduced over each flow path is small.

With Equations (3.16) and (3.18), a slope of zero percent will result in *ln*(0) = ∞. ArcView, which performs the *FlowLength* function in Equation (3.19) ignores any term of infinity and any associated calculations. The main impact of ignoring such a term is identical to adding an *ln*(1) = 0 term (or delivery function of 100%). This is the exact opposite of what should happen, since the problem is caused by delivery = 0% (i.e., a 0% slope). The other impact is that by ignoring the infinity cell, the *FlowLength* function also ignores half of the weighting of each adjacent cell.

This problem was addressed by replacing all delivery functions of zero with a value of 0.001. This magnitude was chosen because scientific precision in the delivery function is limited to two decimal places. By doing the replacement in the delivery function instead of the slope layer, the problem is solved without requiring additional manipulation to the input data.

With the modification, a previous delivery of zero becomes *ln*(0.001) = -6.908 in the summation of natural logarithmic terms. As a result, the *FlowLength* summation increases and the exponential of this summation decreases accordingly. With this modification, the *FlowLength* function calculates as expected and the delivery function weightings of all cells are considered as expected.

3.8 Program implementation

The optimization procedure was programmed using software that best addressed the needs of each component and maintained the modular design of the procedure. Code for correctly processing input data was included to reduce the complexity of input required. Starting with simple GIS data layers and tables of data, ArcView scripts were used to transform the data as necessary for input into the optimization procedure.

3.8.1 Program structure

The optimization procedure was programmed using the C++ language and the ArcView GIS. The optimization component was written as a dynamic link library (DLL) in C++, using the GALib genetic algorithm package (Ver 2.4.4. Matthew Wall, Massachusetts Institute of Technology, Cambridge, MA. http://lancet.mit.edu/ga/. Accessed 12 July 2001).

The pollution loading and cost components were processed in ArcView. A GIS was needed to provide a spatial structure within which the gross erosion and sediment transport sections of the NPS component could function. The GIS facilitated spatial calculations in addition to enabling within-cell calculations to be made simultaneously for all cells. Additionally, ArcView scripts were written to transform entered data into the formats needed for the three components of the optimization procedure. The ArcView scripts and C++ code are given in Appendix B.

The GIS is limited in its ability to interact with other programs, but is able to read a DLL provided that an ArcView script calls the DLL. Also, it is possible to call a script in ArcView from another program, provided that the calling program is being run through a script in ArcView. Hence for this optimization procedure, the main ArcView script calls the DLL that runs the GA (Figure 3.7). For each scenario evaluation, a function in the DLL calls a script in ArcView and passes the management practices for each field. The ArcView script determines the pollutant load at the outlet and the scenario cost. This information is then passed back to the DLL where the scenario fitness is assigned. When the DLL has met the termination criterion, control is returned to the main ArcView script.

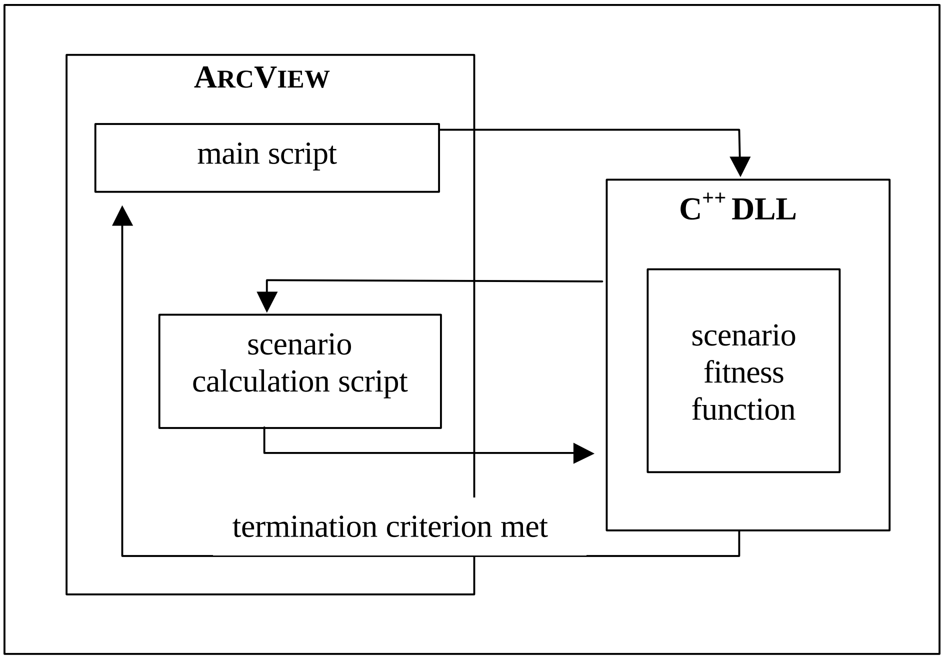

Figure 3.7: Interaction between ArcView and DLL in implementation of the optimization procedure

For the optimization component, each watershed scenario is represented as a chromosome or individual, a one-dimensional array with alleles, using the *GA1DarrayAlleleGenome* class from GALib. The size of the array is equal to the number of management units in the watershed. Each array element is associated with one management unit and consists of a single value identifying the crop-management practice on the management unit. Each array element is a member of an allele set, which is the set of all crop-management practices allowed on the element's corresponding management unit. The allele set for each non-changing management unit, such as forest, consists of a single number corresponding to that management unit. Each non-fixed management unit is assigned a set of alleles corresponding to the set of possible crop-management practices for that area. For example, for a management unit in conventional tillage corn silage, the set of possible crop-management practices might be conventional tillage corn silage, minimum-till corn silage, and conventional tillage corn silage with a winter cover crop. The GA keeps track of the allele sets for each entry in the array by defining a template. The template is created based on the initializing scenario using the *CreateGenomeTemplate* function. Thus, throughout the optimization, values of each array element are selected only from within the appropriate allele set.

The *BaselineInitializer* uses the initializing scenario and the chromosome template to create the first individual. The *PopInitializer* function populates the initial generation of the GA with random values based on the chromosome template. ArcView limits the number of spatial analysis calculations possible in a single ArcView session (that is, without exiting and restarting ArcView). When the number of spatial analysis calculations needed during an optimization run requires multiple ArcView sessions, the GA must be stopped before the ArcView calculation limit is reached and the final population must be saved. In this situation, the *ReadExistingPop* function is used to continue the GA by establishing the final population from the previous session as the initial population for the next session. The *PopEvaluator* function evaluates each population by calling the *Objective* function to evaluate each individual and then using the returned fitness scores to rank all individuals in the population. For each scenario to be evaluated, the *Objective* function sends the management practice identifiers, listed by field, to the NPS prediction and economic analysis components.

3.8.2 Required input

The optimization procedure requires GIS (Table 3.5) and tabular (Table 3.6) input in addition to the rainfall erosivity (USLE R-factor) of the watershed in SI units. Both vector and grid data layers are used by the GIS. Five text or database tables are used to supply additional data to the optimization procedure and to store input data. The structures of the database tables are described in this section. Data values are given in Section 4.2 for the two watersheds used in evaluating the optimization procedure.

The *Constraints.dbf* table identifies area requirements at the farm level. The *Pollut.dbf* table stores the weighting factors of each pollutant as well as the baseline and target loads. The data preparation scripts calculate the baseline loading from the baseline scenario. This table is used in calculating the pollutant score during the optimization run. The *Soilyld.txt* table lists soil productivity for each unique crop and soil combination. This information may be obtained from a soil survey (USDA-NRCS, 2001) or, in Virginia, a nutrient management handbook. The *Crops.txt* table identifies the crops within each management practice and stores crop selling prices, annualized over the rotation for multi-year rotations. The selling price is the dollar amount that the farmer receives for the crop. A crop is listed more than once in the table if it occurs in more than one management practice. For example, winter wheat cover will occur twice if the following management practices are listed: conventionally tilled corn with winter wheat cover and minimum-till corn with winter wheat cover. For each management practice considered, yields are estimated based on the productivity ratings of the soils within the field. The *Mp.txt* table stores descriptors of each management practice. In particular, the management practice type attributes the area in the management practice to its respective area requirement. Three types are currently available: “f” for fixed, “c” for cropland, and “h” for hay or pasture. Additional types can be easily added into the computer code.

The “f” management type category indicates that the management practice is fixed at the baseline management practice throughout the optimization; area in this category does not contribute towards meeting an area requirement. The “c” and “h” categories are used individually and in combination to assess their contribution towards meeting farm area requirements.

The management practice types are also used to set the allele sets for each gene of the GA genome. Currently, each fixed practice has its own allele set so that the management practice does not change. Fields initialized in “c” or “h” type practices may change to any “c” or “h” type practice. This coding is easily modified to further control of the allowable crop-management practices assigned to a management unit.

Every management practice and permanent land use of the watershed must be listed in the *MP.txt* table so that the optimization procedure can consider costs properly. However, in the *Crops.txt* table, only those crops that provide a monetary return for a management practice

Table 3.5: GIS input required for running the optimization procedure

GIS input data
Raster (grid) layers
DEM
Watershed
Soil erodibility (USLE K-factor) in English units
Characteristic slope length (for USLE L-factor)
Vector (shapefile) layers
Watershed outlet
Land use by field (management unit)
Attribute table
Farm number
Field (management unit) number
Farm type
Management practice identifier for baseline scenario
Soils
Attribute table
Soil number
Mapping unit symbol

Table 3.6: Tabular input required for running the optimization procedure

Database input tables

Constraints.dbf	*Crops.txt*	*Mp.txt*	*Pollut.dbf*	*Soilyld.txt*
Farm type	Crop name	Management practice identifier	Pollutant identifier	Mapping unit symbol
Area requirements for cropland, hay, and both [ha] and/or [%]	Unit selling price [$/qty]	Management practice description	Pollutant-weighting factor	Crop yield for each unique crop in *Crops.txt* [qty/ha]
	Management practice identifier	Management practice type	Target pollutant loading	
		Cropping management (USLE C-factor)	Baseline loading	
		Supporting practice (USLE P-factor)		
		Sediment delivery coefficient α		
		Unit production cost [$/ha]		

need to be listed. The optimization procedure first assesses a production cost to each management unit of the watershed based on the management practice. Secondly, it sorts through the crops and credits the private cost of the management unit with the selling prices of any crops associated with that management practice. Additionally, every unique crop listed in the *Crops.txt* table should be listed in the *Soilyld.txt* table. The optimization procedure matches these two tables to calculate the selling price of a crop in $/ha.

3.8.3 Data preparation

An ArcView script, *Prepdat.ave* (Appendix B.1), calculates flow direction and slope GIS raster layers from the DEM. These layers, along with the characteristic slope length layer, are used to calculate a flow-length-by-cell layer and the USLE S and L factors. The flow accumulation grid is also calculated and used along with the slope and flow-length-by-cell layers in the sediment delivery portion of the NPS component. The USLE R, S, L, and K factors are multiplied into a single GIS layer to be used in the gross erosion portion of the NPS component.

An ArcView script, *Prepdat1.ave* (Appendix B.2), was written to combine and translate the tabular and shapefile data into the information used by the optimization procedure. This script first adds two columns, “areaC” and “areaH,” to the land use attribute table. These columns are used to sort the area of each field into the applicable area requirement category. The script adds a column called “PubFlag” to flag, during the optimization run, whether or not the management practice of each field in the working scenario changes with respect to the baseline scenario. Additionally, the script adds the management practice identifier (MPID) column for the working scenario. The MPID column will change with each scenario evaluated during the running of the optimization procedure.

Next, *Mp.txt* is joined to the land use table by MPID and *soilyld.txt* is joined to the soils attribute table by mapping unit symbol. The values of the baseline management practice column are copied into the MPID column of the land use table and the join between the land use and *Mp.txt* tables is refreshed. This provides the C, P, and α factor values, which are used in calculating the pollution loading for the baseline scenario and for each working scenario.

The land use and soils shapefiles are intersected into a *SoilbyFLD.shp* shapefile, which divides each agricultural field into soil types. The area of each soil type within each field is calculated. To calculate the selling price in dollars of each crop for each soil within each field, the unit selling price is taken from *Crops.txt*. The unit selling price is multiplied by the crop yield (initially from *Soilyld.txt*) and by the area.

The selling prices and areas from *SoilbyFLD.shp* are summarized by field into a new table *Fldarea.dbf*. Columns for opportunity cost (“Oppcost”) and the respective MPID (“ProfmaxMP”) are added to *Fldarea.dbf*. Also, a column is added for the net return of each management practice (“Mpreti” where *i* is the MPID). The value of each net return column is calculated as total selling price minus the production cost. For calculating the total selling price of a management practice, the *Crops.txt* table is used to locate each crop in the management practice. The selling prices of these crops, given in *Fldarea.dbf*, are added together to determine the total selling price. The production cost is determined for each field by locating the management practice unit production cost from the *Mp.txt* and multiplying by the field area. The opportunity cost is the value of the maximum net return column. This column differs for each field. The *Fldarea.dbf* table is joined to the land use attribute table by field identifier. The joined information is used for calculations in the economic component of the procedure. Also, a “PrivC” column is added to the land use attribute table to hold private cost calculations during the running of the optimization procedure. The areas from *SoilbyFLD.shp* are summarized by farm into a new table *Farmarea.dbf.* The farm type is copied to this table from *SoilbyFLD.shp*. At this point the *Farmarea.dbf* table must be added to the ArcView project.

Next the *Constraints.dbf* table is joined to *Farmarea.dbf* by farm type. In the next ArcView Script, *Prepdat2.ave* (Appendix B.3), three columns for analyzing the area requirements are added: “cst_c,” “cst_h,” and “cst_b”. The value “cst_c” is the maximum of the cropland hectare requirement and of the cropland percentage requirement times farm area. The values of the remaining two columns are calculated similarly. Four columns for calculating the area weights, used in determining the economic score, are added to *Farmarea.dbf.* The first three columns hold, respectively, the area weights for cropland, hay or pasture, and both. The final column holds the combined area weight for the farm. Two columns for holding the private and public costs of the baseline scenario are added to *Farmarea.dbf.* The final step of the data preparation scripts runs the script for the economic component. The economic component creates a table, *TempSum1.dbf*, which holds the private and public costs of the working scenario by farm.

After preparing the data and adding *TempSum1.dbf* to the ArcView project, the script *Baseline.ave* (Appendix B.4) can be run. The *Baseline* script runs the script for the economic component, setting the working scenario equal to the baseline scenario. The *Baseline* script joins *TempSum1.dbf* to *Farmarea.dbf* by farm id and updates the baseline public and private cost columns with the public and private costs from *TempSum1.dbf.* Also, the *Baseline* script runs the NPS component to calculate the baseline pollutant loading for each pollutant. The *Pollut.dbf* table is updated with this information. A warning is given if the baseline loading is less than or equal to the target loading for any pollutant. In this case, one may choose not to continue with the optimization run or may choose to change the target loading. For example, the target loading may be specified as a percentage of the baseline loading. This script can be rerun as necessary for changes in the baseline scenario. The data are now ready for the optimization procedure.

3.9 Summary

Five optimization heuristics for intractable problems were compared based on factors considered important in solving this research problem. This problem was characterized as a combinatorial optimization problem. The SA, TS, and GA were found appropriate to solve this problem. The GA was selected based on formulation ease and on previous use in the problem of reducing NPS loadings through BMP placement.

Next, an optimization procedure was developed using the GA to optimize watershed scenarios. Scenarios were evaluated based on reduction in pollutant loading and cost increase from the baseline. Existing watershed-level NPS models were considered for calculating pollutant load reduction, but were found to be not sufficiently distributed or to be prohibitive in computer runtime. Thus, a NPS component was created that combined a GIS, the USLE, and a sediment routing technique to meet the needs of this problem. An economic component was created to consider public and private costs at the farm-level. Public costs included contracting and enforcement costs and were applied to all farms varying from the baseline. Private costs considered field-level net returns based on soil productivity and crop rotation enterprise budgets. Pollution and economic fitness scores were developed and combined into an overall fitness score, used for evaluating each scenario. Finally, the optimization procedure was coded into a computer program for implementation.

Because of the modular nature of the procedure, changes in the pollutant loading, cost components, or fitness scores can be made with minimal impact to the interaction among the components. Additional BMPs can be added, provided the NPS and economic components have information needed to quantify the impacts of each BMP on pollutant loading and cost, respectively. The optimization portion of the procedure is not inherently limited in its ability to evaluate management, vegetative, or structural BMPs.

Chapter 4: Evaluation of Optimization Procedure

4.1 Introduction

The components of the optimization procedure were tested to verify the computer coding. Specifically, the optimization component was tested to ensure that increasing fitness scores corresponded to minimizing pollution until the maximum allowable, or target, load was achieved and then minimizing cost increase while meeting the pollution target load. The economic portion of the procedure was examined to determine if, between scenarios of nearly equivalent cost, a higher fitness score was assigned to the scenario in which costs were divided more evenly across farms. Additionally, the impact of increasing area requirements was evaluated to determine if stricter area requirements resulted in a slower rate in fitness score increase across otherwise similar runs. A slower rate was expected due to the penalizing portion of the economic fitness function for not meeting area requirements to the same extent as the baseline. The NPS component was evaluated to determine the effectiveness of the sediment delivery function in modeling the impacts of BMP selection and placement.

The optimization procedure, as a whole, was evaluated to determine if the procedure found a more cost-effective solution than a targeting strategy. Differences in land allocation for four solution scenarios of the optimization procedure were compared with each other, with the baseline scenario, and with the targeting strategy. Finally, a cost-benefit analysis was performed to compare the optimization procedure with the targeting strategy.

4.2 Data used for evaluation

The evaluation tests were based on data from the Muddy Creek watershed in Rockingham County, Virginia (Figure 4.1). Two subwatersheds within the Muddy Creek watershed were used to strengthen the conclusions suggested by evaluation of the procedure and its components for agricultural watersheds in ridge and valley agroecosystems and to demonstrate the transferability of the optimization procedure among watersheds. For identification purposes, the following names were assigned to these subwatersheds: “Mini-Muddy Creek” and “Lola Run”.

The “Mini-Muddy Creek” subwatershed is 686 ha and covers the upper reach of Muddy Creek, a perennial stream. Fifty-seven percent of the subwatershed is in agricultural production (65 management units distributed among 10 farms) with 40% of the subwatershed in forest and the remaining 3% in residential use. The second subwatershed, “Lola Run”, is a 1014-ha watershed on an ephemeral stream that joins Muddy Creek downstream of the outlet point of Mini-Muddy Creek. The subwatershed consists of 125 management units in agricultural production (77% of the subwatershed), distributed among 18 farms, with 19% of the subwatershed in forest and 4% in residential use. The two adjacent subwatersheds are similar in land use and topography with Lola Run having a 2% lower average slope.

For the examples discussed in this work, fields were considered the basic management unit. The Muddy Creek land use and field boundary data layer was obtained from the Virginia Department of Conservation and Recreation.

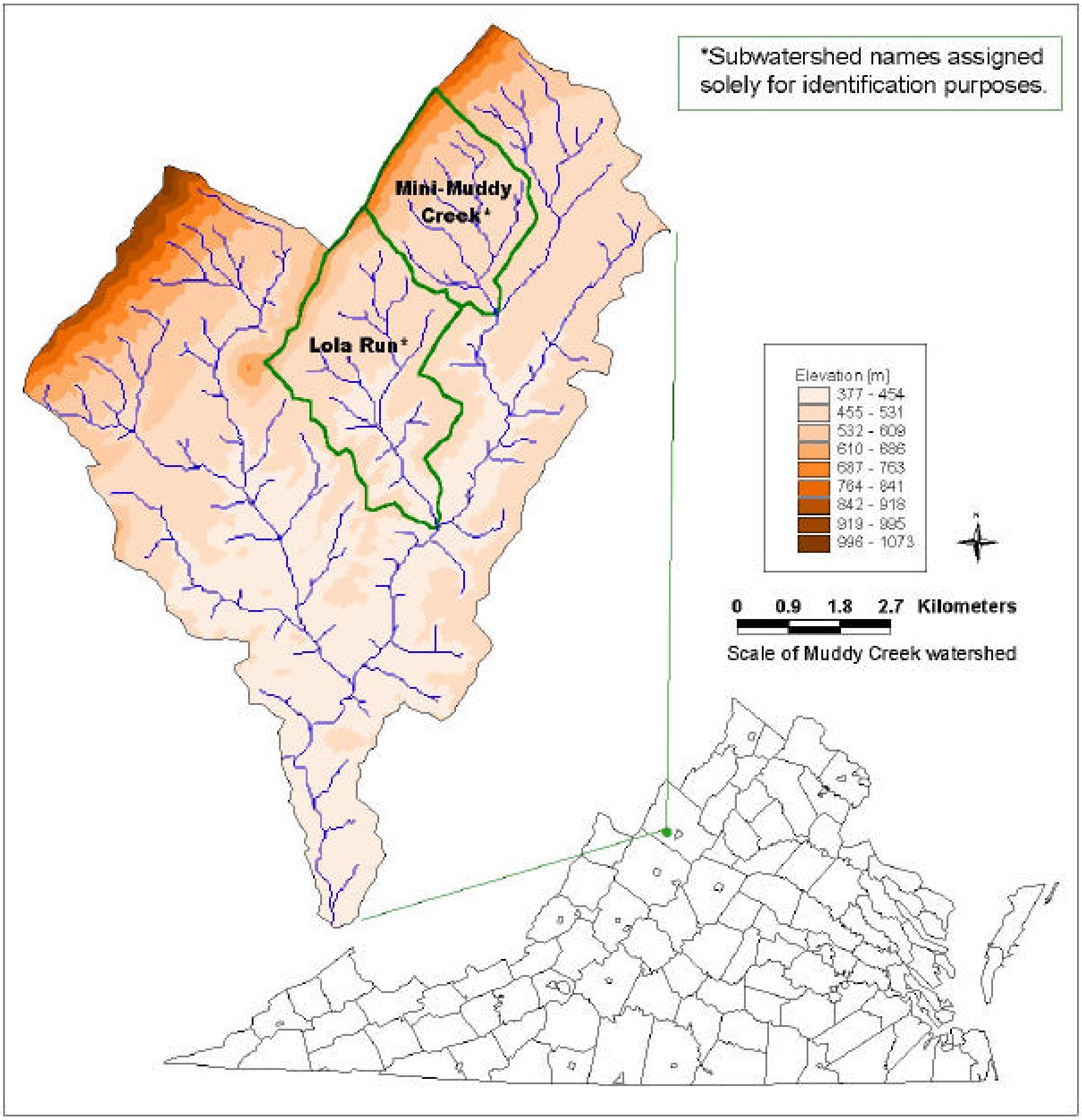

Figure 4.1: Location of Mini-Muddy Creek and Lola Run subwatersheds within the Muddy Creek watershed in Rockingham County, Virginia

The Muddy Creek land use layer was used in establishing the baseline scenario used in evaluating each subwatershed. These baseline scenarios placed all cropland in conventionally tilled corn silage. Forest/orchard, farmstead/residential, hay, and pasture were as identified in the land use layer.

Farm types and boundaries were estimated using a GIS clustering technique (Heatwole, 1999) and modified based on discussion with B. Patterson (personal communication, District Conservationist, NRCS, Harrisonburg, Virginia, 19 December 2000). Farm sizes were assigned based on farm area and capacity (J. W. Pease, personal communication, Agricultural and Applied Economics Department, Virginia Tech, Blacksburg, Virginia, 23 March 2000). Based on this information, the Mini-Muddy Creek subwatershed includes ten farms: seven beef, two dairy, and one poultry (Table 4.1). In contrast, the Lola Run subwatershed has 11 dairies and seven beef farms.

Table 4.1: Distribution of farm types used in defining the two study subwatersheds, based on characteristics of the Muddy Creek watershed

Farm type		Lola Run	Mini-Muddy
Dairy			
	small (60 cows)	4	--
	medium (100 cows)	5	1
	medium with poultry (100 cows, 2 houses)	1	--
	large (150 cows)	1	1
Beef			
	small (40 cows)	4	4
	small with poultry (40 cows, 3 houses)	1	--
	medium (70 cows)	--	2
	medium with poultry (70 cows, 2 houses)	1	--
	large (150 cows)	1	1
Poultry			
	5 houses	--	1

For the USLE, a single *R* value of 2800 MJ·mm/(ha·h·y) (Schwab et al., 1993; Figure 5.5) was used for both watersheds. The USLE erodibility *K* factor, taken from the SSURGO soil survey (USDA-NRCS, 2001) and converted to SI units, ranged from 0.0067 to 0.057 Mg·ha·h/(ha·MJ·mm). The *S* and *L* factors were calculated as described in Section 3.7.1. The *S* factor was calculated from the cell-to-cell slope determined by the GIS based on the DEM. The *L* factor used the flow length of 45m, which is a characteristic length of nonconcentrated flow for fields in this region (B. Cubbage, personal communication, Natural Resource Technician, NRCS, Harrisonburg, Virginia, 04 March 2002).

The management practice values used in the various evaluations are shown in Table 4.2. The purpose of the management practice type was explained in Section 3.8.2. The value of the management practice type for each management practice was set based on the definition of each optimization run. For example, in a run in which corn silage was allowed to be replaced by other crops, the management type for corn silage was set as “c”. In the same run, if forested land was considered to remain in forest, the management type for forest was “f”.

Table 4.2: Management practice values used to evaluate the optimization procedure

Management practice identifier	Management practice description[1]	Management practice type[2]	C-factor	P-factor flag[3]	α-factor	Production cost ($/ha)
0	farmstead	f	0.01	1	19.2	0.00
1	forest	f	0.003	1	1.1	0.00
2	CC	c	0.49	1	9.7	744.33
3	CC / WW	c	0.43	1	6.2	1220.48
4	MC	c	0.32	1	4.9	845.48
5	MC / WW	c	0.28	1	3.1	1321.62
6	CC (2 yrs) / H (3 yrs)	c	0.12	1	5.9	401.58
7	CC (1 yr) / MC (1 yr) / H (3 yrs)	c	0.11	1	4.9	417.09
8	alfalfa hay	h	0.02	1	3.3	899.03
9	grass hay	h	0.01	1	3.3	436.87
10	pasture	h	0.006	1	1.6	64.95
11	Contoured, CC	c	0.49	2	9.7	756.68
12	CC / WW, all contoured	c	0.43	2	6.2	1232.83
13	MC, contoured	c	0.32	2	4.9	857.83
14	MC / WW, all contoured	c	0.28	2	3.1	1333.97
15	CC (2 yrs) / H (3 yrs), all contoured	c	0.12	2	5.9	410.33
16	CC (1 yr) / MC (1 yr) / H (3 yrs), all contoured	c	0.11	2	4.9	425.83

[1]CC = conventional corn silage; WW = winter wheat; MC = minimum till corn silage; H = grass hay
[2]f = fixed allele set, containing the single corresponding management practice; c = cropland allele set, containing all “c” and “h” practices; h = hay allele set, containing only “h” practices. These values can vary according to simulation conditions.
[3]1 = up and down tillage; 2 = contour tillage. Corresponding P-factors are assigned at runtime by slope based on Novotny and Olem (1994; Table 5.6).

The C-factors were obtained from USDA-NRCS (1988) and B. Cubbage (personal communication, Natural Resource Technician, NRCS, Harrisonburg, Virginia, 04 March 2002). For the forage crops, a higher value for alfalfa and grass hay than pasture is used in this region for two reasons. First, soil loss is expected during establishment periods for alfalfa and grass hay. Second, the alfalfa and the legumes in these hays generally cover the soil less well than the sod forming grasses in the pastures. The pastures in this region are well managed and fertilized, resulting in a relatively low C-factor. The P-factor flag (1 = up and down tillage, 2 = contour tillage) is converted by the optimization procedure into the appropriate values (Novotny and Olem, 1994; Table 5.6) at runtime. The α-factors were taken from Table 3.4. For a rotation the time-weighted average of the individual α-factors was used.

Production costs (Table 4.2) were determined in $/ac from the Virginia Farm Management Crop and Livestock Enterprise Budgets (VCE, 1999) and converted to $/ha. For contour tillage practices, a cost of $12.35/ha was added (VCE, 2001). Production costs included seed, fertilizer, machinery, and labor costs for the enterprise. Land ownership and tax costs were excluded, as were farm planning and management costs. These values do not change with respect to the management practice adopted.

Private costs were calculated using enterprise production costs, historical selling prices, and soil productivity. Selling prices for each cash crop are shown in Table 4.3. Buying prices were taken from the Virginia Agricultural Statistics Service (VASS, 2001) and used to calculate selling prices by subtracting a combined cost for marketing and transportation (D. J. Bosch, personal communication, Agricultural and Applied Economics Department, Virginia Tech, Blacksburg, Virginia, 07 March 2002). The pasture rent was taken from the Virginia Cooperative Extension (VCE, 2001).

Table 4.3: Crop prices for all management practices

Crop[1]	Buying price[2]	Market and transportation costs[3]	Selling price
Alfalfa	$111.00/ton	$30/ton	$81.00/ton
Grass hay	$76.00/ton	$30/ton	$46.00/ton
Pasture	$16.19/ha (rent)	--	$16.19/ha (rent)
Corn silage	$29.10/ton	$3/ton	$26.10/ton
Winter wheat grain	$2.00/bu	$0.40/bu	$1.60/bu

[1]Crop quantities given in English units.
[2](VASS, 2001); pasture rent from VCE (2001).
[3](D. J. Bosch, personal communication, Agricultural and Applied Economics Department, Virginia Tech, Blacksburg, Virginia, 07 March 2002).

Soil productivity values are listed by mapping unit symbol (USDA-NRCS, 2001) in Table 4.4. These values were taken as quantity per acre from the NutLite computer program, which is an interface to NutMan (V2, Virginia Tech Dept. of Entomology, ISIS Lab, Blacksburg, Virginia. http://www.isis.vt.edu/dss/nutman. Accessed 18 April 2002; Stone, 1995); the

Table 4.4: Soil productivity of Rockingham county soils

Mapping unit symbol	Alfalfa[1] [tons/ha]	Conventional corn silage[1] [tons/ha]	Minimum-tillage corn silage[1] [tons/ha]	Winter wheat grain[1] [bu/ha]	Grass hay[1] [tons/ha]
1B	7.41	44.46	44.46	158.08	9.26
22F	0.00	0.00	0.00	0.00	0.00
24B2	0.00	37.05	37.05	118.56	8.03
28A	0.00	0.00	0.00	0.00	0.00
29B2	17.29	44.46	44.46	158.08	9.26
29C2	16.25	44.46	44.46	158.08	9.26
29D2	13.83	44.46	44.46	158.08	9.26
2B	7.41	37.79	37.79	134.37	9.26
30C3	12.97	44.46	44.46	158.08	9.26
31C2	15.56	44.46	44.46	158.08	9.26
31D2	13.83	44.46	44.46	158.08	9.26
32C	13.83	44.46	44.46	158.08	9.26
33B2	17.29	44.46	44.46	158.08	9.26
33C2	16.25	41.79	41.79	148.60	9.26
33D2	16.25	35.57	37.79	126.46	9.26
33E2	16.25	35.57	37.79	126.46	9.26
34C	16.25	41.79	41.79	148.60	9.26
34E	16.25	41.79	41.79	148.60	9.26
35B	17.29	44.46	44.46	158.08	9.26
39B	0.00	33.35	33.35	83.98	5.25
39C	0.00	33.35	33.35	83.98	5.25
40B	0.00	31.49	31.49	83.98	5.25
40C	0.00	31.49	31.49	83.98	5.25
40D	0.00	29.64	31.49	83.98	5.25
41D	0.00	29.64	31.49	83.98	5.25
41E	0.00	29.64	31.49	83.98	5.25
47B	0.00	37.05	37.05	98.80	5.25
4A	0.00	0.00	0.00	0.00	0.00
54A	0.00	24.70	24.70	59.28	0.00
56C	17.29	44.46	44.46	158.08	9.26
56D	17.29	44.46	44.46	158.08	9.26
57E	0.00	34.83	37.05	83.98	5.25
59C2	7.41	31.57	31.57	114.26	9.26
5C2	0.00	21.00	21.00	83.98	5.25
5D2	0.00	19.76	21.00	79.04	5.25
60D	7.41	26.87	30.23	138.32	9.26
63B	12.35	44.46	44.46	158.08	9.26
68B	12.35	46.93	46.93	138.32	11.12
74E2	0.00	19.76	22.23	83.98	5.25
75E2	0.00	0.00	0.00	0.00	0.00
75F2	0.00	0.00	0.00	0.00	0.00
76A	0.00	51.87	51.87	158.08	5.25
8B	0.00	27.29	27.29	100.78	8.03

[1]All crop quantities given in English units.

Virginia Nutrient Management Handbook (VA-DCR, 1993); and J. C. Baker (personal communication, Crop and Soil Environmental Sciences Department, Virginia Tech, Blacksburg, Virginia, 14 March 2002). The values were converted into quantity per hectare.

However, because both the selling prices and soil productivities are given in English units and are multiplied together by the optimization procedure to get price per area, the crop quantities in English units are used, eliminating unnecessary conversions. The soil types represented by mapping unit symbols 22F, 75E2, and 75F2 are steep, rocky soils not suited for agriculture and occur only within forested land in the headwaters of the study watersheds. The soil types represented by mapping unit symbols 4A and 28A are nonproductive fluvaquents occurring along stream networks.

Public costs were taken from Carpentier et al. (1998) and annualized over a period matching the five-year rotation length of hay set by the enterprise budgets. All other management practices were either single year or five-year rotations. As a result, the contracting costs were calculated as \$194.42 per year and enforcement costs as \$153.40 per year for each farm that changed from the baseline scenario for one or more fields. If rotations of varying lengths are used, all costs should be brought to the present value and annualized over a common time frame, as necessary.

For the evaluation runs that considered area requirements, a minimum number of hectares per farm were required in cropland, hay, or both depending on farm type (Table 4.5). These area requirements were based on suggestions by Stone et al. (2002) (Table 4.6). Farm size definitions for the Muddy Creek watershed were based on Stone et al. (2002). The dairy and beef requirements are for feed, whereas the poultry requirement is for litter spreading. The values shown in Table 4.5 were used as the high requirement situations in the evaluations. The medium requirement situations used one-half of the values shown in Table 4.5, while the zero requirement situations used a zero value for all entries in Table 4.5.

Instead of applying the area requirements by farm, they could be applied at a regional level, such as county or subwatershed. For example, an area requirement might be that at least 40% of the baseline cropland remains in cropland at a subwatershed level. Appropriate changes could be made to the cost calculation so that costs are summed by the relevant region instead of by farm.

Since a single pollutant (sediment) was considered for these evaluations, the input table for setting pollutant target loads consists of a single line (Table 4.7). The “Base” field was filled by the optimization procedure with the baseline load at runtime. The “Target” load, filled in at runtime, was set according to the target load for each run.

For all optimization runs, the GA was initialized from random scenarios based on the allele sets for each gene. The runs were performed on Pentium IV, 1.6 GHz machines with 264 Mb RAM and 35 Gb of disk space available for virtual memory and swap space. Computational methods in the Spatial Analyst extension of ArcView limit processes to less than 2^{15} grid calculations per session (Huber, 2000). As a result, the optimization procedure was run in several sessions when necessary, with the final population of each run becoming the initial population of each subsequent run.

Table 4.5: Area requirements used in evaluating the economic component of the optimization procedure

Farm type	Feed requirements Cropland [ha]	Feed requirements Hay and/or pasture [ha]	Litter spreading requirements Cropland, hay, and/or pasture [ha]
Small dairy (60 cows)	30	2	0
Medium dairy (100 cows)	50	2.8	0
Large dairy (150 cows)	75	4	0
Small dairy with poultry (60 cows, 2 houses)	30	2	8
Medium dairy with poultry (100 cows, 2 houses)	50	2.8	8
Large dairy with poultry (150 cows, 1 house)	75	4	6
Small beef (40 cows)	0	56	0
Medium beef (70 cows)	0	98	0
Large beef (150 cows)	0	210	0
Small beef with poultry (40 cows, 3 houses)	0	56	10
Medium beef with poultry (70 cows, 2 houses)	0	98	8
Large beef with poultry (150 cows, 1 houses)	0	210	6
Poultry (5 houses)	0	0	14

Table 4.6: Suggested area requirements for meeting feed and litter spreading needs for Rockingham County, Virginia (Stone et al., 2002)

Farm type	Cropland	Hay and/or pasture	Cropland, hay, and/or pasture
Dairies:			
Small operation (60 cows)	0.5 ha/cow	2 ha	
Medium operation (100 cows)	0.5 ha/cow	2.8 ha	
Large operation (150 cows)	0.5 ha/cow	4 ha	
Poultry:			
First house			6 ha
Additional houses			2 ha/each
Beef:		1.4 ha/head	

Table 4.7: Example base and target loadings for a single pollutant

Pollutant identifier	Weight	Base	Target
1	1	3.45	0.64

4.3 Evaluation of optimization component

The initial conditions of population size and replacement percentage were varied to determine the impact of these factors and of computer runtime on increase in fitness score and degree of convergence. Mutation and crossover rates were varied to determine the most efficient values for a given population size and replacement percentage. Convergence rates were examined to confirm that the GA was functioning correctly. The termination criterion for evaluation of the optimization procedure was determined based on the convergence of the initial runs and on the variables of pollution reduction and cost being represented by the fitness scores. Also, the variables related to the fitness scores were evaluated to confirm that changes in the fitness scores corresponded with the expected changes in the pollution reduction and cost variables.

4.3.1 Determining initial conditions

Population size and reproduction parameters for GAs impact the efficiency of the GA. A large population size provides a more thorough coverage of the search space but can take substantial computer time to evaluate each generation. Conversely, a small population size may sample less of the search space but allows for more generations to be evaluated in an equivalent amount of computer time.

De Jong (1975) found a population size of 50-100, with a crossover rate of about 0.6 and a mutation rate of 0.001 per gene, worked best for a particular set of problems. Mitchell (1999) stated that, lacking other suggestions, De Jong's parameter values were widely used until the late 1980's. At that time two separate studies (Grefenstette, 1986; Schaffer et al., 1989) systematically determined that a population size of 20-30 with a crossover rate of 0.75-0.95 and a mutation rate of 0.005-0.01 produced the best results for their test problem sets. Mulligan and Brown (1998) used population sizes of 25, 100, and 200, replacing all but the previous best individual at each generation. Additionally, they used a crossover rate of 0.6 based on the work by De Jong (1975) and a mutation rate of 0.03 based on initial testing. Liong et al. (1995) used a crossover rate of 0.6 and a mutation rate of 0.001. Reported values for parameter sets used in other NPS calibration studies during the 1990s are incomplete. The GA application by Srivastava et al. (1999), which is most related to this dissertation, used a population size of 100 with 25% replacement. This combination showed convergence after 150 generations (3825 individual evaluations).

Initial selection of the most efficient combination of parameter values for a given problem is not straightforward. Mitchell (1999) cautioned that the ideal parameter values are likely to vary for different problem types and applications as a result of problem formulations and performance criteria.

ArcView allows a limited number of calculations per session. Thus, the minimal number of total evaluations needed to solve the problem became a desired characteristic. If used for planning purposes, the procedure would more likely be used if it can be shown to solve the problem under minimal supervision.

ArcView performs about nine spatial analysis operations for each GA individual evaluated. Thus, the limit of 2^{15} operations per ArcView session is reached after evaluation of about 3500 individuals. Accordingly, GA performance was evaluated over 3500 individual evaluations, corresponding to one session in ArcView.

The research goal for this dissertation was to increase cost-effectiveness relative to targeting. In achieving this goal it was desired to minimize the number of individuals evaluated as much as possible to minimize the impact of the ArcView limitation and to maintain a reasonable total runtime. As a result low population sizes with high replacement levels were considered.

A large population allows more of the response surface to be evaluated as compared with a smaller population. However, with a large population, there are a large number of evaluations per generation, corresponding with population size. Thus, the impact of the ArcView limitation and the increase in total computer runtime could be substantial as compared with a small population. Fewer evaluations per generation are made when using a smaller population. Thus, solutions that meet the pollutant-targeting criterion and reduce cost can potentially be located in fewer computer evaluations. Also, because more generations have been evaluated, more information is available regarding the state of convergence.

High replacement levels were considered to maintain breadth in sampling across the search space under low population sizes. In a GA with a high replacement level, the best few individuals are carried over from one population to the next and then selected from the final population. Using a high replacement level increases the number of evaluations per generation, but results in a broader sampling of the search space at each generation than when using a low replacement level.

Alternatively, a high population size with a low replacement level may have also provided efficient results. This combination would potentially start the heuristic with a broader sample of the search space than would a low population using high replacement. However, due to the low level replacement levels, more unchanged individuals must be maintained and carried over to successive generations. Also, as a result of increased computations per generation for larger populations, an increased number of ArcView sessions is potentially needed before being able to determine convergence trends. This was considered a potential limitation in the future acceptance and use of the procedure. Thus, the combination of high population size with a low replacement was not evaluated in this research.

There is a lack of previous research on the problem formulation used by the optimization procedure. As a result, parameter evaluation was needed to determine a combination of population size and replacement level that solved the problem efficiently based on the criteria of limited runtime while maintaining a set of alternate solutions. The impact of population size and replacement on the convergence of the GA was evaluated using five population sizes (100, 50, 25, 15, and 10) and three replacement levels (90%, 70%, and 50%). The Lola Run watershed was used in this analysis. Replacement levels were chosen to correspond with population sizes such that at least three individuals were carried over to the next generation, resulting in ten tested combinations (Table 4.8).

Table 4.8: Tested combinations of population size and replacement level

Population size	Replacement level		
	90%	70%	50%
100	X	X	
50	X	X	
25		X	X
15		X	X
10		X	X

Crossover and mutation rates of 0.9 and 0.05, respectively, were used to determine the population size and replacement level. Although a 0.05 mutation rate is higher than suggested by the literature, it was chosen to add increased diversity in the smaller populations. A crossover rate of 0.9 means that every two individuals selected from the preceding generation have a 0.9 probability of being combined through crossover. A mutation rate of 0.05 means that each gene in each individual has a 0.05 probability of being mutated.

Computer performance is often variable among computers depending on configurations. Thus, runtime was measured by the number of individual evaluations performed by each run. Each tested combination of population size and replacement level was run for several thousand individual evaluations, corresponding to a computer runtime of approximately four and a half hours on a Pentium IV, 1.6 GHz computer. The number of generations evaluated for each run varied as a function of population size, replacement level, and the total number of individual evaluations.

Fitness scores for the runs with the larger populations showed the lowest rate of improvement with respect to number of evaluations (Figure 4.2). Two runs for some initial conditions, such as population size of 15 and replacement of 70%, are shown. Parameters for runs are identical for similar initial conditions. However, a different initial population is used because the GA uses random generation to create the initial population of individuals.

Over an equivalent number of evaluations, the lower population sizes of 10 and 15 increased in fitness at a higher rate than the remaining combinations. The population size of 15 with 70% replacement performed better than all other combinations after an initial period of about 2000 individuals evaluated. The larger population size of 15 with 70% replacement has an advantage over the population size of 10. Because more individuals in the population size of 15 are available in each generation for crossover and mutation, more of the search space will be sampled. Thus, the population size of 15 with 70% replacement was chosen for evaluating the optimization procedure against the targeting strategy.

Although this combination of initial conditions is not necessarily ideal for all watersheds, the arguments for a low population size and high replacement level still hold. Some watersheds may require fewer or more generations, and thus an altered runtime.

For the population size of 15 with 70% replacement, three crossover rates (0.85, 0.90, and 0.95) were compared (Figure 4.3). Based on the overall performance, a 0.90 crossover rate was selected for the remainder of the evaluations. For this combination, mutation rates were varied from 0.005 to 0.07 (Figure 4.4). The 0.01 mutation rate, which shows the most consistent rate of increase and achieves the highest fitness score, was selected for evaluation of the optimization procedure.

Population size, crossover rate, and mutation rate typically interact nonlinearly (Mitchell, 1999). Thus, determining these parameters in a different order may effect the combination selected. Population size impacts the number of evaluations performed and, thus, the runtime. As a result, population size was determined first for this research. Crossover was determined next for two reasons. First, crossover is often considered the main method in a GA of introducing variation and exploring the search space (Mitchell, 1999). In contrast, mutation is viewed primarily as a method of preventing given genes within each individual from becoming permanently fixed. As such, crossover performs a more crucial role than mutation. Second, the work suggested by Grefenstette (1986) and Schaffer et al. (1989) suggested that 0.90 was within a reasonable range for small populations. The mutation rate, considered the least important parameter in the overall efficiency of the GA, was determined last.

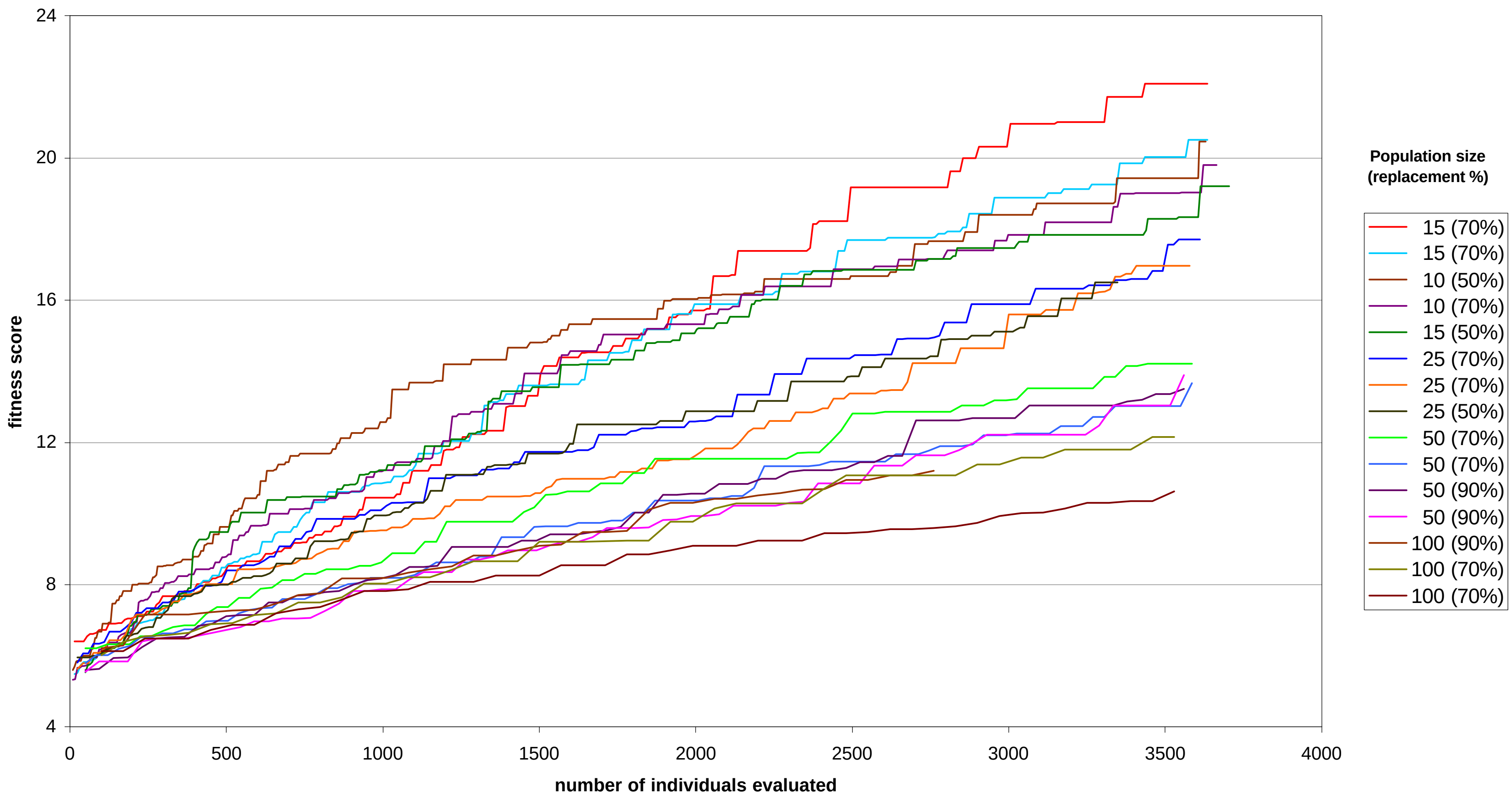

Figure 4.2: Convergence patterns over a similar runtime for varying initial conditions in Lola Run (for some initial conditions, multiple combinations are repeated with differing initial random populations)

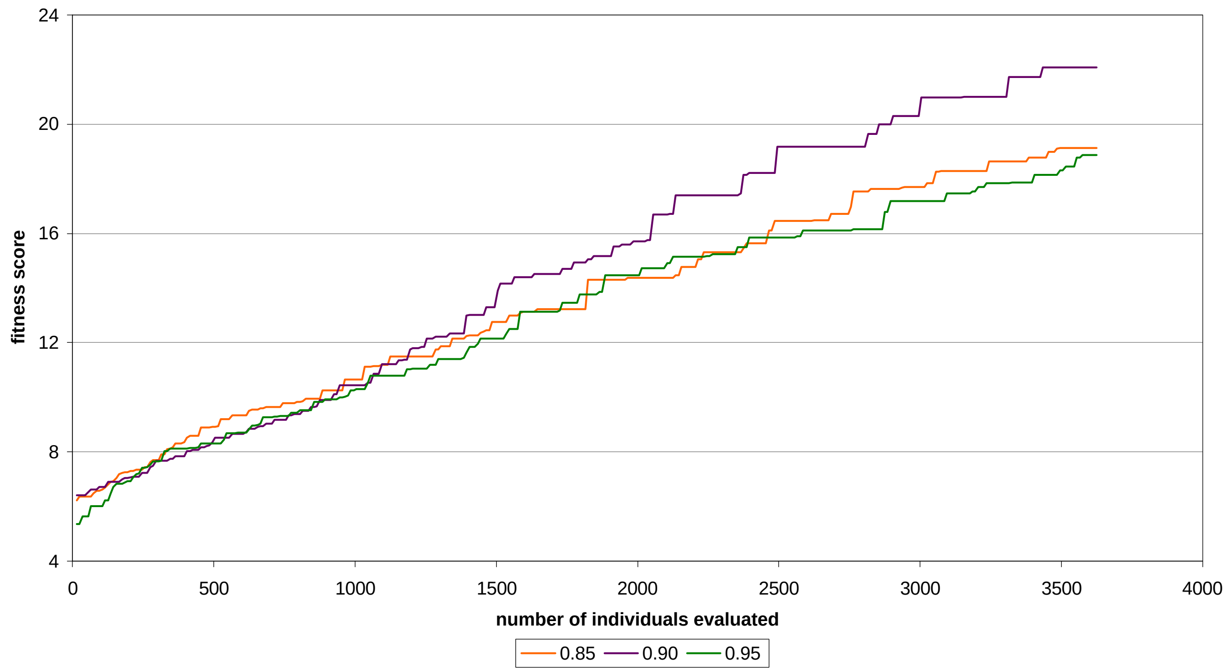

Figure 4.3: Comparison of crossover rates in Lola Run for population size 15 and 70% replacement with a 0.05 mutation rate

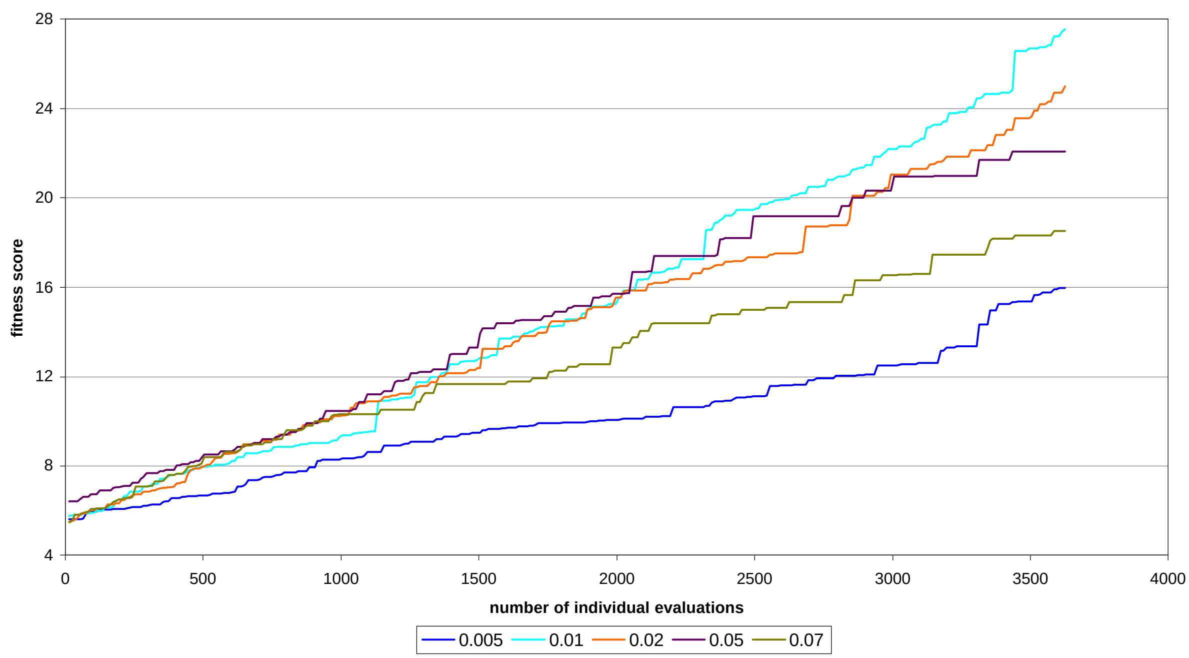

Figure 4.4: Comparison of mutation rates in Lola Run for population size 15 and 70% replacement with a 0.90 crossover rate

4.3.2 Determining termination criterion

Convergence rates shown in Figure 4.5 are based on a ratio of the best fitness score of the preceding 50th generation divided by the best fitness score of the current generation. The key feature of this graph is to confirm that the simulation has stabilized into a high level of convergence (95-100%).

The intent of this research was not to locate the true optimum, but to identify a set of solutions that are more cost-effective than a given targeting strategy. Thus, the termination criterion of the optimization depended more heavily on achieving an acceptable cost-effectiveness level in a reasonable runtime than on achieving a particular convergence level.

To determine how long a simulation should run before termination, a graph of the highest scoring scenario of each generation versus pollution target load, pollution reduction from baseline, and watershed cost (Figure 4.6) was found to be more helpful than a convergence graph. This graph can be easily updated and evaluated throughout the simulation. Using information from this graph, the simulation can be run until the pollutant targeting criteria have been met and the cost curve has begun to level out. Then a decision must be made as to the trade-off between running the simulation longer and achieving a lower cost solution. If the cost curve falls below the amount budgeted for watershed improvement, then simulation might be stopped and final population of the simulation chosen as the set of near optimal solutions.

By running the simulation longer than necessary to meet budget levels, perhaps 300-500 more generations, scenarios that reduce pollution for equal or lower cost may be identified. For example, in the run shown in Figure 4.6, a cost budget of $70 000 was met by the 700th generation with a scenario reducing sediment yield by 9.5 Mg/ha. However, by generation 975, the procedure identified a scenario with near equivalent cost but a sediment reduction of 11Mg/ha.

Upon termination, the best-found solution is reported. If replacement levels are such that only a single solution is carried over to successive generations or if the GA has converged to the optimum, then the last several generations of the GA before termination might require examination when alternative best-found solutions are desired. Otherwise a set of alternative best-found solutions is provided in the final generation of the GA. The preferred solution can then be chosen from among the alternative solutions based on favorable characteristics or trade-offs.

Trends in convergence rate for the Lola Run simulation corresponded to trends in watershed cost decrease, as expected (Figure 4.7). The procedure was not attempting to further reduce sediment after meeting the target reduction level. Despite this, sediment reduction did tend to improve noticeably at a few points in Figure 4.7 and then remain at each new level. In this run, crossover and mutation processes created a combination of field assignments that resulted in a large increase in sediment reduction at generation 970.

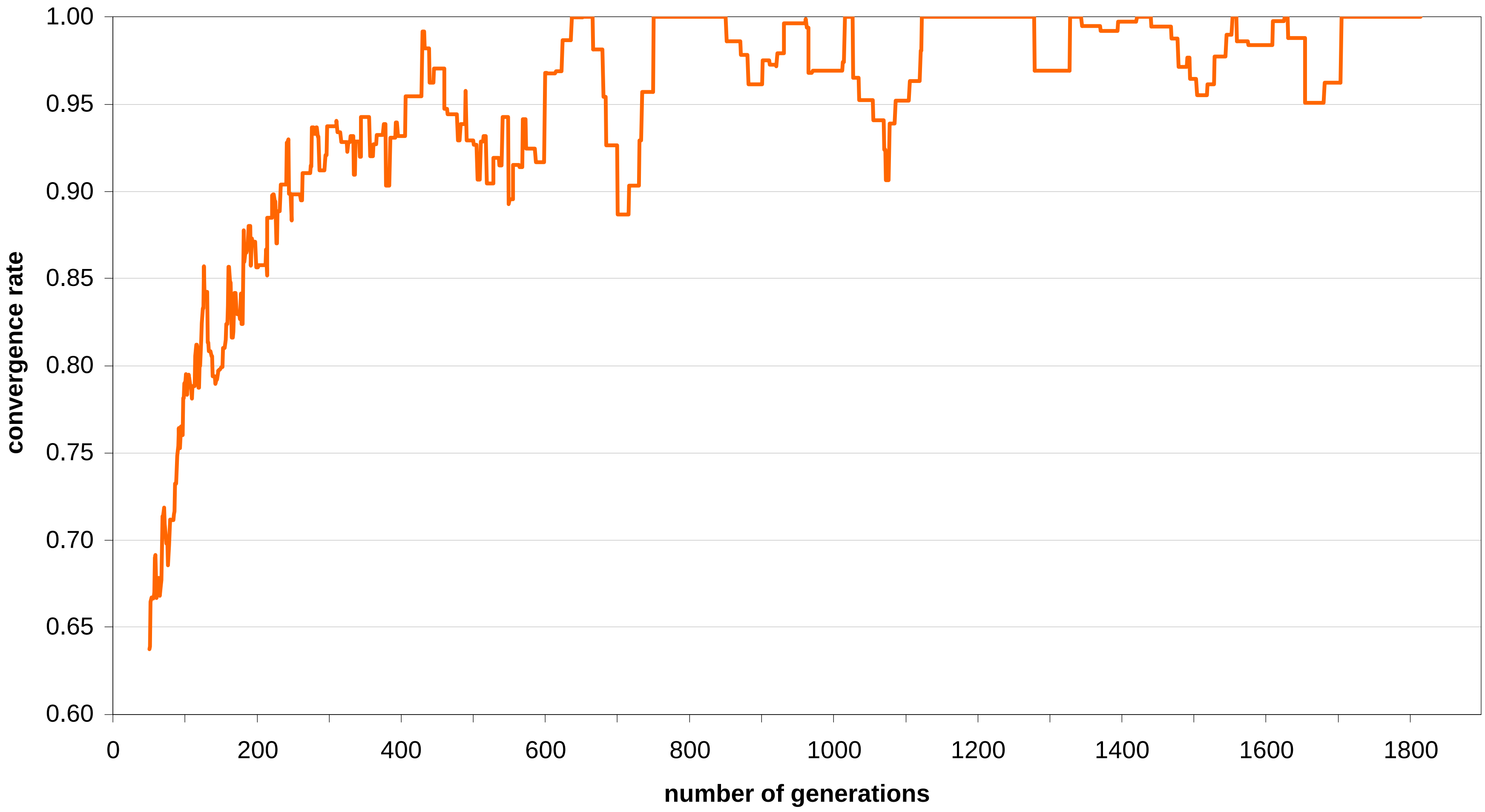

Figure 4.5: Convergence rate ((fitness score of generation (x-50)) / (fitness score of generation x)) for population size 15 and 70% replacement in Lola Run

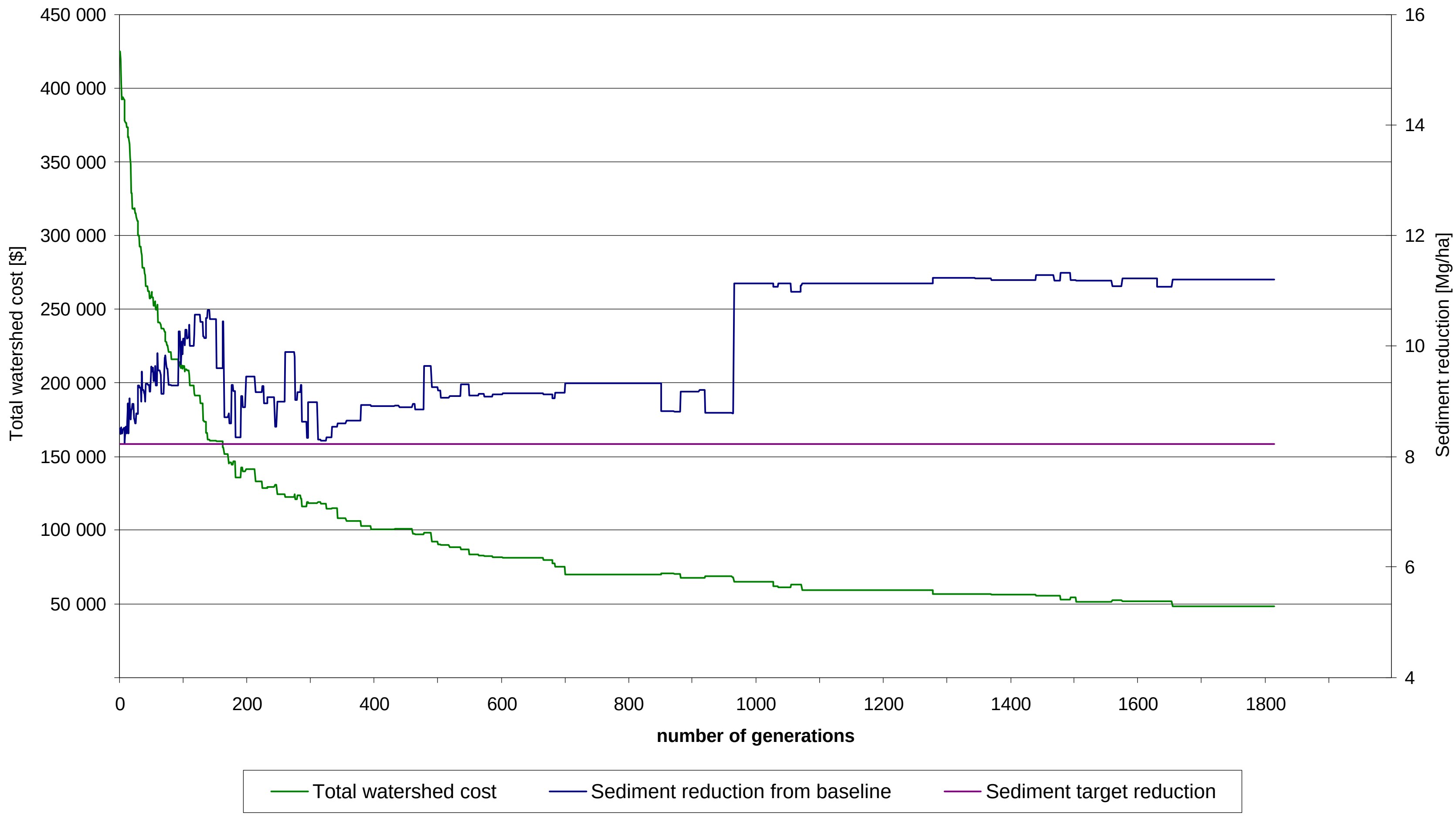

Figure 4.6: Best of generation watershed cost and pollutant reduction values for Lola Run (population size 15 and 70% replacement)

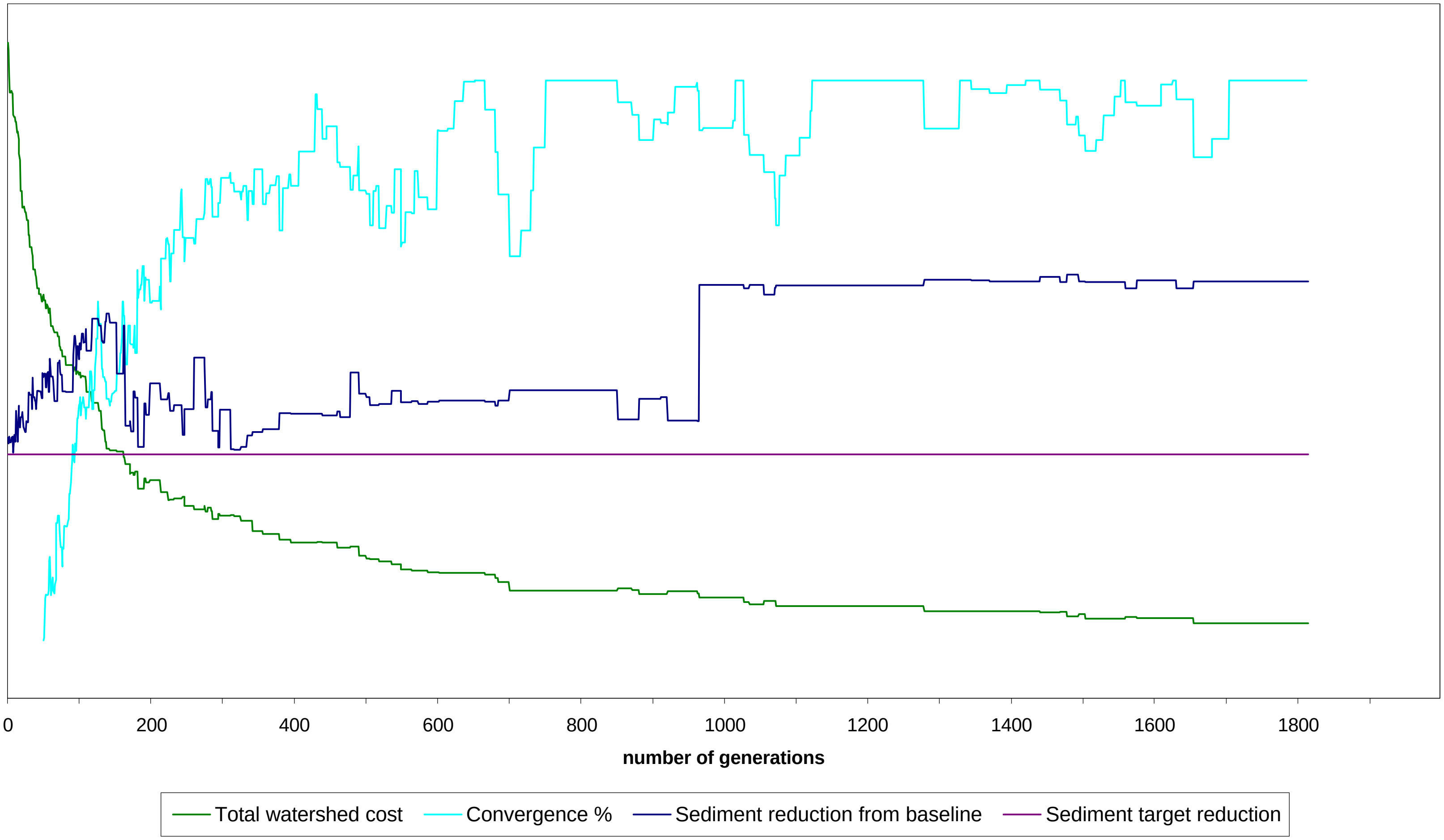

Figure 4.7: Generational comparison of watershed cost, convergence rate, and pollutant reduction values for Lola Run (population size 15 and 70% replacement) – y-scales are as in Figures 4.5 and 4.6

4.3.3 Fitness score performance

To evaluate the performance of the fitness scores with respect to the optimization variables, three hypotheses were formulated:

1) Pollutant loading decreases as the pollutant fitness score increases.
2) The farm-level cost measure (i.e., the square root of the sum of the square of each farmer's cost) decreases as the economic fitness score increases.
3) Total watershed cost (i.e., the sum over the watershed of each farmer's cost) decreases as the economic fitness score increases.

The hypotheses were evaluated for each watershed, without area requirements, for three optimization runs and one targeting strategy (Table 4.9). To facilitate comparisons, the maximum acceptable loading (i.e., target load) for all runs was set to the sediment loading achieved by the targeting strategy.

Optimization Runs 1 and 2 were considered to compare the optimization procedure and targeting strategy with only cropped fields allowed to vary. Optimization Run 3 was considered to evaluate the effect of allowing all agricultural land to vary and to demonstrate the optimization procedure over a variety of management practices.

The pollution fitness score (Equation 3.1) was developed to lead the optimization component to find solutions in which NPS component sediment yield predictions are below a maximum, or target, load. Thus, pollution fitness scores should increase as sediment yield decreases. Figures 4.8 to 4.13 confirm this relationship for the three different optimization runs, on each of the subwatersheds.

In these figures the unbounded pollution fitness score is shown. That is, the pollution score shown was not limited to a maximum value of one after meeting the target load. However, in all cases, there were few scenarios that increased in pollution reduction beyond the target load and such increases were very minimal.

The unbounded score is used for display and analysis only. The unbounded score allows analysis of pollution fitness after the target load has been met. The bounded version as given by Equation (3.1) was used in the optimization run.

In the runs shown, conventionally tilled corn silage generally had the highest net return but also the highest erosion. To meet the target load, BMPs were applied to the corn silage fields. This caused costs to increase. Because the optimization procedure was minimizing costs, the sediment yield was generally as high as possible while meeting the target sediment reduction. For example, Figure 4.12 shows the optimization program found a solution at generation 246 under Optimization Run 2 that slightly reduced sediment yield below the target load of 0.17 Mg/ha. However, the scenario cost increased by several hundred dollars. Subsequent solutions reduced costs by allowing the sediment yield to increase to the target load. If a crop such as alfalfa, which reduces erosion and increases net return, had been available, then the optimal scenarios may have reduced sediment yield further beyond the target load criterion.

Once the pollutant target load is met, the economic fitness score (Equation 3.5) drives the optimization process. As discussed in Section 3.5.3, the economic fitness score uses the

Table 4.9: BMP placement strategies used in evaluation of the optimization procedure for both subwatersheds

BMP Placement Strategy	Description
Baseline Scenario	All cropland was placed in conventionally tilled corn silage. Forest/orchard, farmstead/residential, hay, and pasture were as identified in the Muddy Creek land use layer.
Targeting	All fields in conventionally tilled corn silage in the baseline, with the majority of the field at greater than three percent slope, were converted to minimum-tillage corn silage with contour tillage and a winter wheat cover crop. All non-cropland remained as in the baseline.
Optimization Run 1	Two management variations of corn silage were allowed on all cropland: • conventional tillage and • minimum-tillage with a minimum-tillage winter wheat cover crop, on the contour. All non-cropland remained as in the baseline.
Optimization Run 2	Eight management variations of corn silage were allowed on all cropland: • conventional tillage, on the contour and not; • conventional tillage with a minimum-tillage winter wheat cover crop, on the contour and not; • minimum-tillage, on the contour and not; and • minimum-tillage with a minimum-tillage winter wheat cover crop, on the contour and not. All non-cropland remained as in the baseline.
Optimization Run 3	Three basic management practices were allowed on all land identified in the baseline as cropland, hay, or pasture: • conventional tillage corn, • hay, and • pasture. Four BMPs for the corn silage crop were considered, both individually and in combination: • minimum tillage, • minimum-tillage wheat grain as a winter cover, • contour farming, and • conversion of the row crop to forage (pasture or grass hay). All non-agricultural land remained as in the baseline.

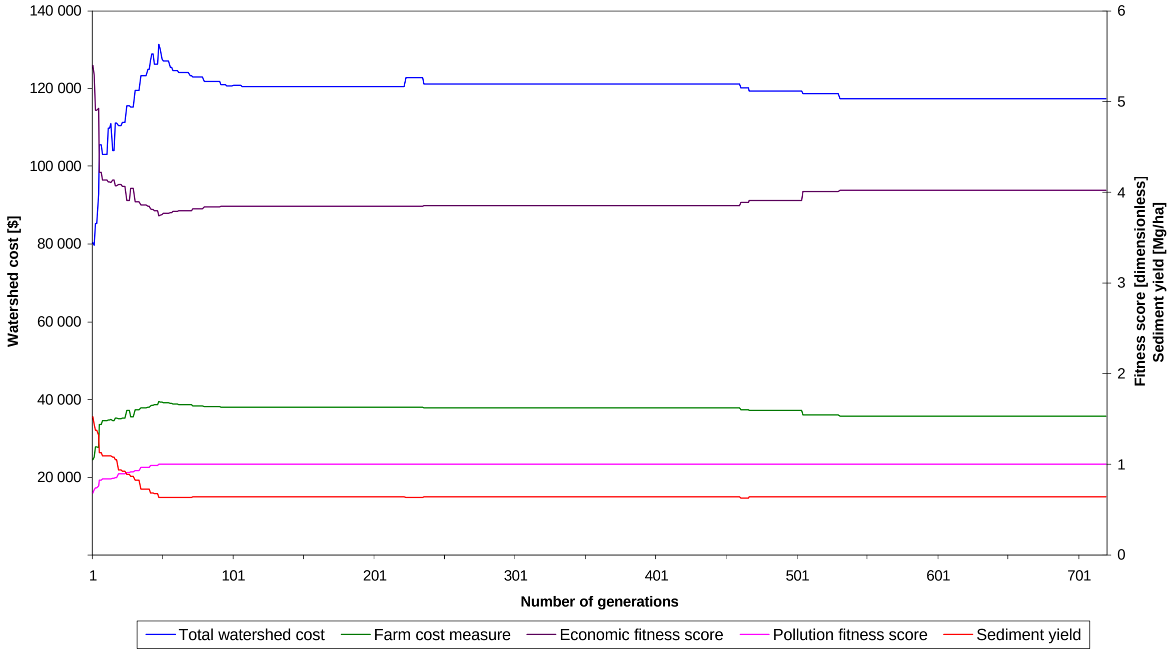

Figure 4.8: Comparison of cost and pollution variables with fitness scores for Lola Run under Optimization Run 1

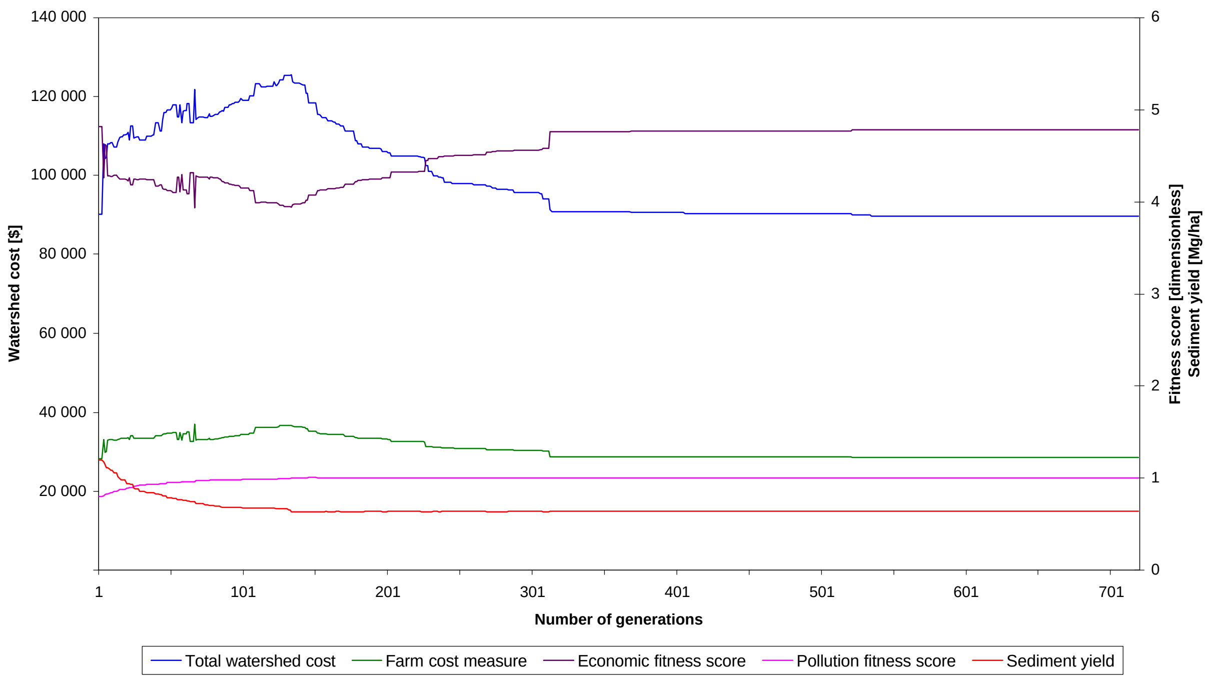

Figure 4.9: Comparison of cost and pollution variables with fitness scores for Lola Run under Optimization Run 2

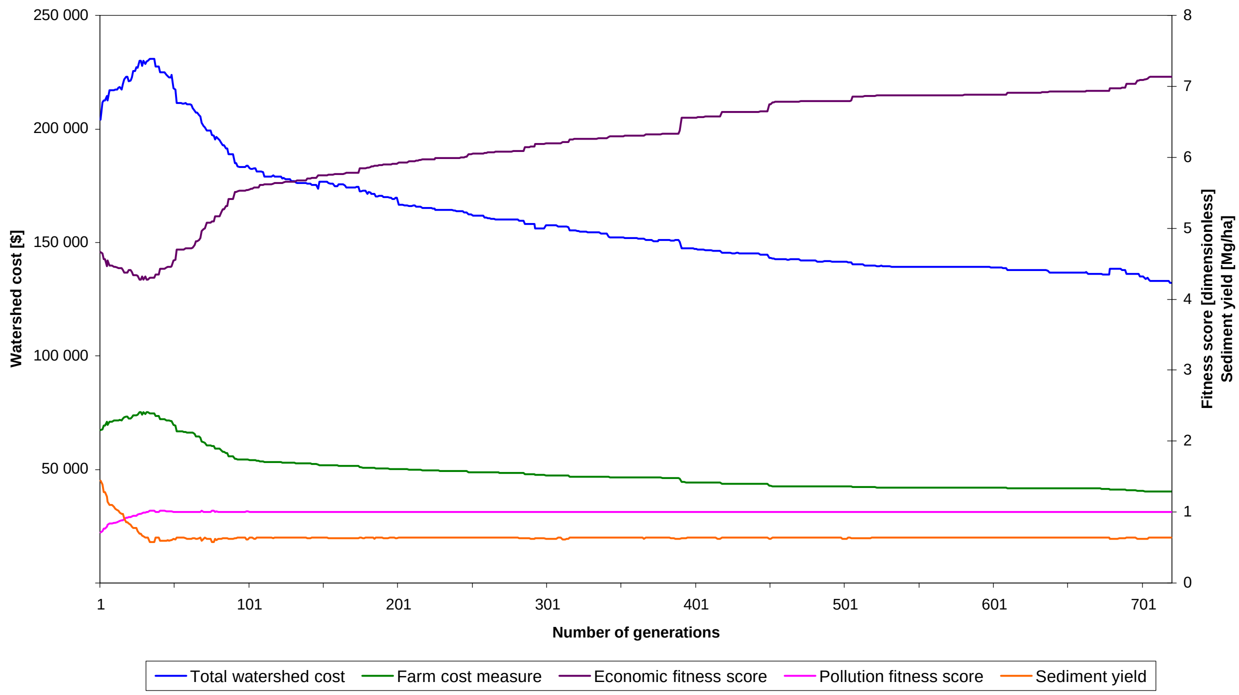

Figure 4.10: Comparison of cost and pollution variables with fitness scores for Lola Run under Optimization Run 3

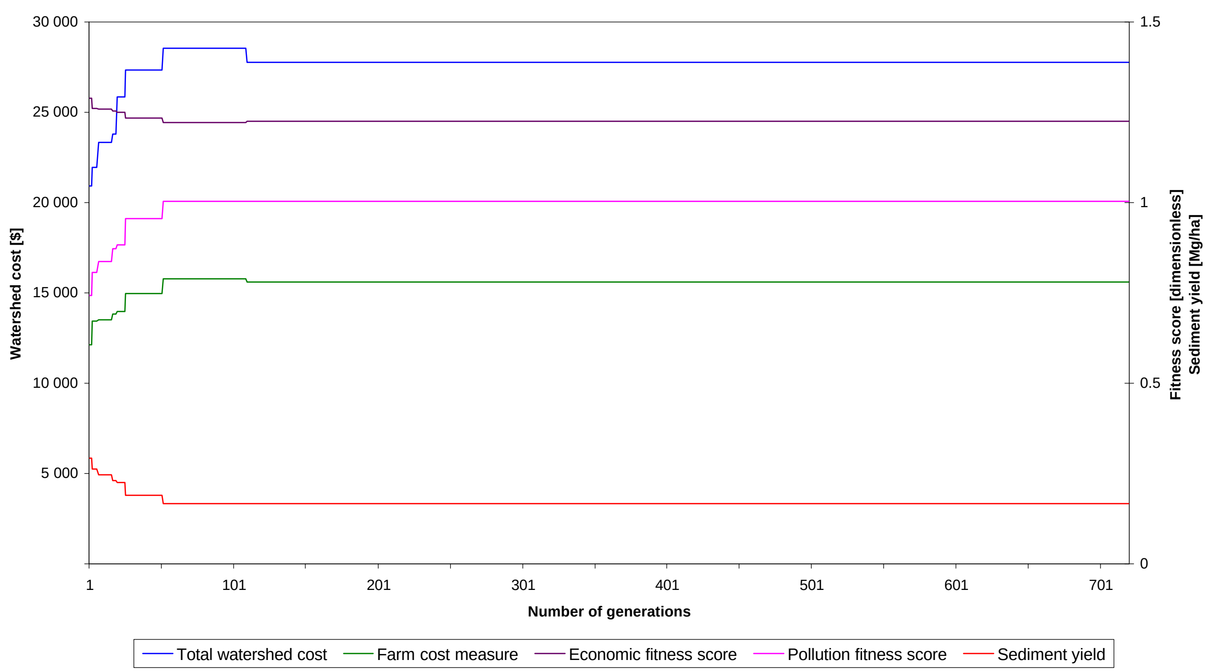

Figure 4.11: Comparison of cost and pollution variables with fitness scores for Mini-Muddy Creek under Optimization Run 1

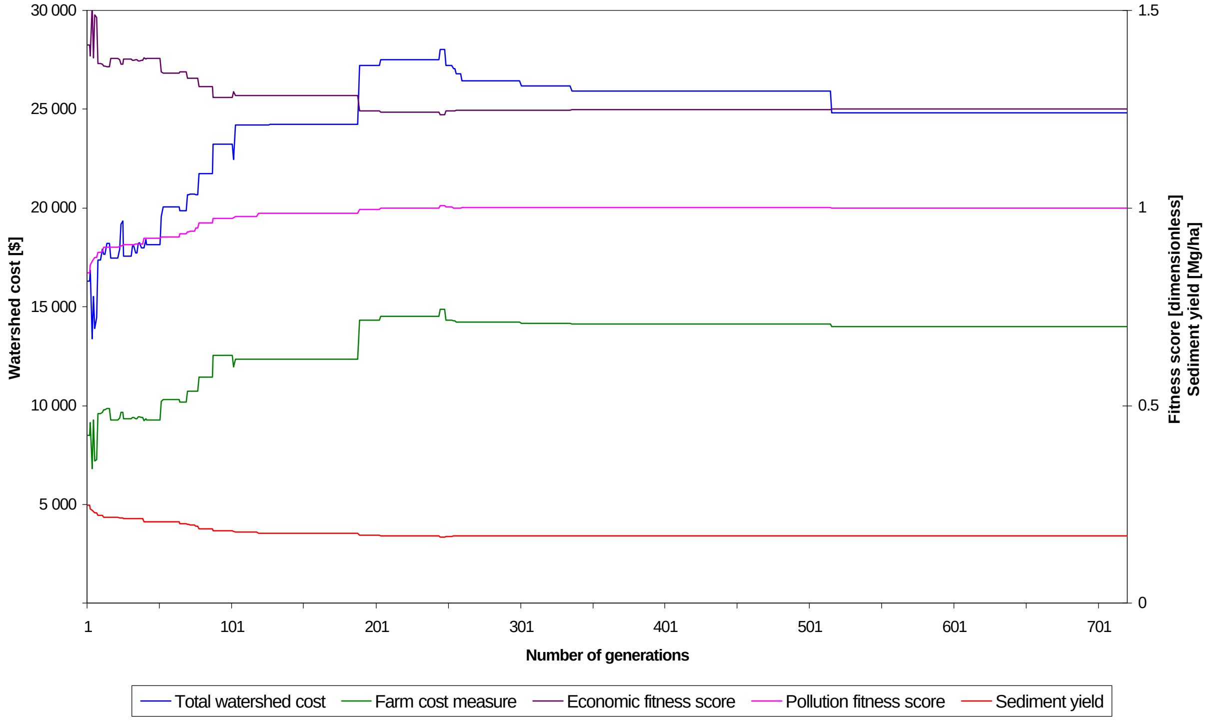

Figure 4.12: Comparison of cost and pollution variables with fitness scores for Mini-Muddy Creek under Optimization Run 2

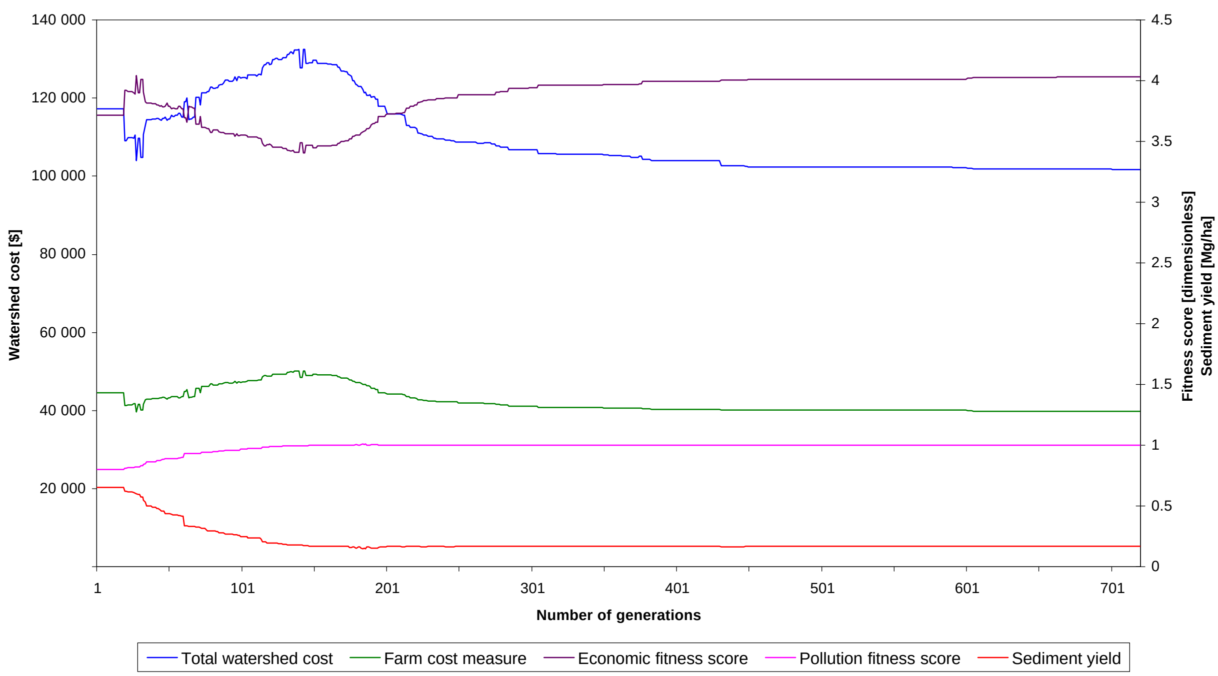

Figure 4.13: Comparison of cost and pollution variables with fitness scores for Mini-Muddy Creek under Optimization Run 3

farm-level cost measure to formulate scenarios of decreasing cost while preferring scenarios in which costs are divided evenly among farms. The economic fitness score does increase as the farm-level cost measure decreases (Figures 4.8 to 4.13).

The economic fitness score was intended to correspond with reduction of total watershed cost, which is a more direct measure of the economic system being modeled than the farm-level cost measure is. All runs show the expected decrease in total watershed costs after the pollutant target load has been met.

The effect of using farm-level summation, as opposed to watershed-level summation, in the optimization function is shown in Figure 4.8 at generation 230. The watershed cost increases. However, the division of this cost across the farms is such that the farm-level cost, and hence the economic fitness score, do not increase. In contrast, Figure 4.12 shows increase in both farm-level cost and watershed cost at generation 190. In this case, the large increase in total watershed cost outweighed the effect of how costs were divided among farms.

4.4 *Evaluation of economic component*

The economic component was evaluated to determine how well it modeled fairness with respect to division of cost increases among farms. Three levels of area requirements were evaluated to determine the impact of increased area requirements on otherwise similar optimization runs. The impact, on total watershed cost, of using a farm-level cost summation versus a watershed-level cost summation in the economic fitness function of the optimization procedure was examined. This involved comparing the results of two optimization runs, where one optimized using a farm level cost measure and the other used a watershed-level measure.

4.4.1 Cost fairness

The extent to which the fitness scores promoted fairness of cost across farms was examined using the following hypothesis:

- Between two scenarios of near equivalent cost, the optimization procedure prefers the scenario for which costs are divided more evenly across the farms.

The hypothesis of cost fairness was tested on the Lola Run watershed with no area requirements. The optimization run for this hypothesis employed a data set used in comparing the impact of initial conditions on optimization efficiency. The division of costs more evenly across farms was measured in terms of decreased deviation from the median using Equation (4.1). The median was used as opposed to the mean because it was not clear that farm costs follow a normal distribution and the median is unbiased to extreme values.

$$s = \frac{\sum_i \left| X_i - \overline{X} \right|}{n} \tag{4.1}$$

where

s = median deviation of the scenario [$],
X_i = costs for farm i [$],
$\overline{X}$ = median farm cost of the scenario [$], and
n = number of farms in watershed.

Figure 4.14 shows results for the best of each generation throughout the optimization run. The fitness score remained constant or increased as the run progressed. In most cases the watershed cost decreased when the fitness score increased. In places of slight cost increase for the watershed, the higher scoring scenario often had a lower deviation from the median. For example, the scenario of lower watershed cost in generation 1203 was replaced in generation 1204 by a scenario of slightly higher watershed cost ($223 increase) but a $192 decrease in median deviation. Similar responses of the procedure are seen at generation 755 and 1602. In cases of substantial watershed cost increase, the fitness score generally decreased. This is because the change in watershed cost outweighed consideration of how that cost was distributed across the farms. Such cases are not seen in Figure 4.14 because it is a graph of the best scenario of each generation.

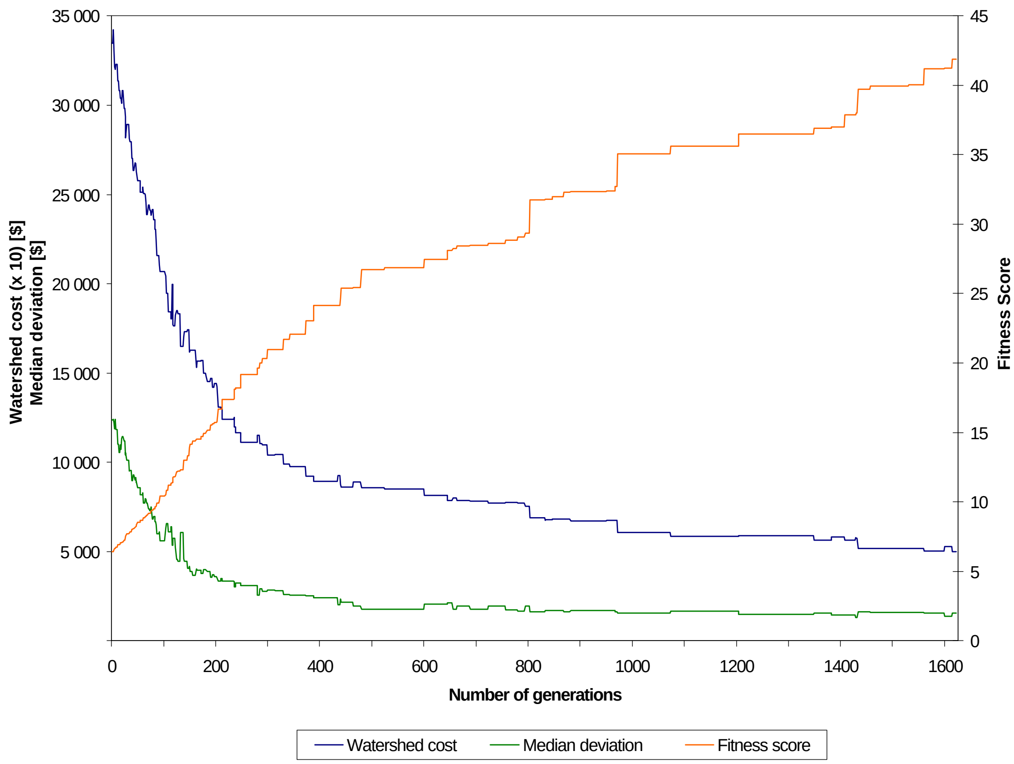

Figure 4.14: Comparison of watershed cost and median deviation for Lola Run as fitness score increased

The best scenario of generation 1203 has a median cost of $3136 and a median deviation of $1661 whereas the best scenario of generation 1204 has a median cost of $3156 and a median deviation of $1469. Figure 4.15 demonstrates the shift of farm costs towards the median for generation 1204, as compared to generation 1203. In particular, the number of farms with costs above $4500 decreased from four to two as the run progressed from generation 1203 to 1204. The same decrease was seen in the number of farms with costs less than or equal to $1500.

These results support the hypothesis that, when watershed costs between scenarios are near equivalent, the economic fitness score prefers the scenario in which the costs are more evenly divided across farms. This indicates that the economic fitness score responds as intended in terms of cost fairness.

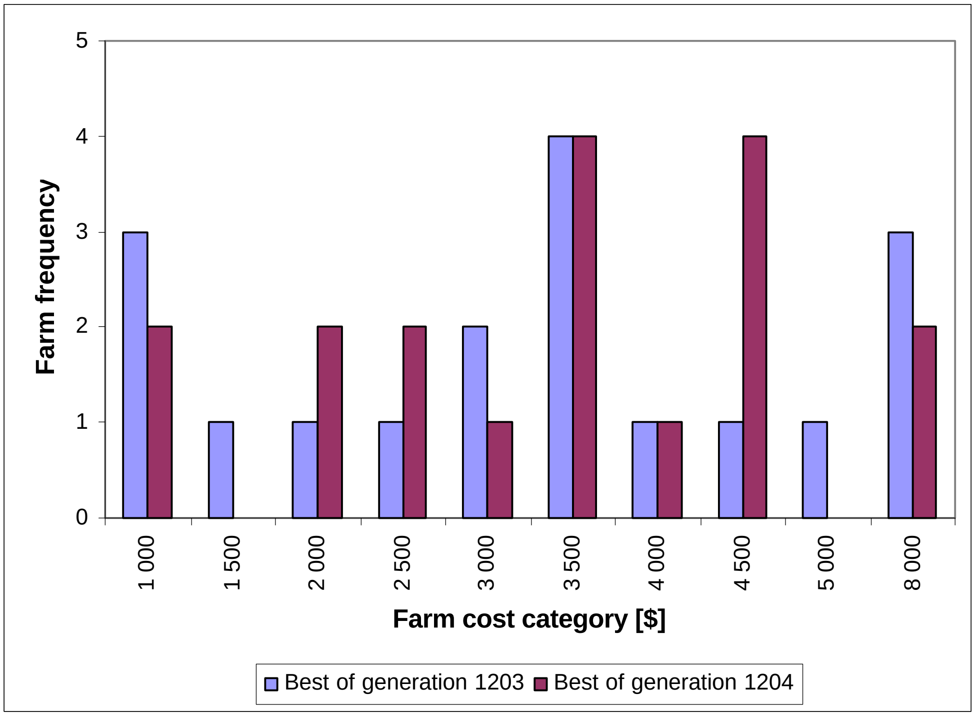

Figure 4.15: Comparison of farm-level costs for the best of generation scenarios for generations 1203 and 1204 of the run shown in Figure 4.14 (x-axis shows upper limit of each cost category)

4.4.2 Area requirements

The impact of area requirements on scenario fitness was evaluated by looking at the economic fitness score. The economic fitness score was formulated such that, in any given scenario, farms would be penalized for meeting the area requirements to a lesser extent than they did in the baseline scenario. Thus, the following hypothesis was tested:

- Meeting area requirements to a lesser extent than they are met in the baseline scenario results in lower fitness scores across otherwise similar runs.

A variety of crop and hay management practices were needed to fully evaluate the area requirement impact. Thus, Optimization Run 3 (Table 4.9), which included hay, pasture, and various management practices for corn silage, was run under no area requirements. The pollutant target load was set equal to the pollutant load of the targeting strategy to facilitate evaluation beyond addressing this hypothesis. The best-of-generation values from the optimization run were then used to test the area requirement hypothesis.

For each best-of-generation scenario, the economic component of the optimization procedure was used to calculate the percentage of the medium and high-level area requirements met by each farm in the working scenario. Additionally, the extent to which the area requirements were met in the best-of-generation scenario as compared to in the baseline scenario was calculated.

Tables 4.10 and 4.11 show the extent to which each farm met the medium and high-level area requirements under the baseline scenario and under one best-of-generation scenario (generation 359). No farms met the high-level area requirements under the baseline scenario for either watershed (column ab_i). Three farms in Lola Run and four farms in Mini-Muddy Creek met the medium-level requirements under the baseline scenario.

In order for a farm not to be penalized in the working scenario, the working scenario had to meet the area requirements at least as much as the baseline scenario did. For example, the baseline scenario for the first large beef farm listed in Table 4.10 met 20% of the high-level area requirement and 41% of the medium-level requirement. In order for this farm not to be penalized in the working scenario, the farm had to have 42 ha in hay and/or pasture under the high-level area requirement but 86.1 ha in hay and/or pasture under the medium-level area requirement. For all farms in each watershed, the high-level requirements were met to a lesser extent by the baseline than the medium-level requirements were met by the baseline. Thus, less acreage was required to meet the baseline percentage for the high-level condition as compared to the medium-level condition.

Additionally, in all except two farms in Lola Run and one farm in Mini-Muddy Creek, the extent to which the working scenario met the baseline percentage under the high-level requirements (column a_i) was larger or equal to the extent to which the working scenario met the baseline percentage under the medium-level requirements. As a result, it was expected that high-level requirements would produce equivalent or higher economic fitness scores as compared to medium-level requirements for each best-of-generation scenario.

Economic fitness scores for both the medium and high-level requirements were calculated by applying the economic component to each best-of-generation scenario. For each scenario, the economic fitness score for the high-level requirements was larger than or equal to that for the medium-level requirements, as expected (Figures 4.16 and 4.17). The fitness scores for the zero-level area requirements were equal to the scores of the other two requirement levels for the first 20 generations of Mini-Muddy Creek. This is because the best-of-generation scenario for these generations was the baseline scenario. Otherwise, fitness scores for the zero-level area requirements were consistently larger than those of the other two requirement levels for both watersheds. This was expected because no penalties are imposed under zero-level area requirements.

To further examine the impact of area requirements on economic fitness scores and on total watershed cost, Optimization Run 3 was run for each watershed under medium and high-level area requirements. These runs were compared with each other and with the zero-level run discussed previously. As expected, the zero-level run for Mini-Muddy Creek had the highest economic fitness score throughout the run, followed by the high-level run and then the medium-level run (Figure 4.18). Seventy percent of the farms in Mini-Muddy Creek are beef and have an area requirement of hay and/or pasture. In this watershed, the impact of area requirement on reducing fitness score despite decrease in watershed cost is seen from generation 125 to 175. The high-level watershed cost decreased below that of the zero-level. However, the high-level fitness score remained lower than the zero-level score.

Table 4.10: Proportion of area requirements met for a given Lola Run scenario, as compared to the baseline, when medium and high-level requirements are individually applied

Farm type	High-level requirement		Medium-level requirement	
	ab_i[1]	a_i[2]	ab_i[1]	a_i[2]
Large beef	0.20	1.00	0.41	1.00
Large dairy	0.92	1.00	1.00	1.00
Medium beef w/ poultry	0.72	1.00	0.93	1.00
Medium dairy	0.67	0.92	0.84	0.87
Medium dairy	0.68	1.00	0.85	1.00
Medium dairy	0.80	0.91	1.00	0.96
Medium dairy	0.73	1.00	0.97	1.00
Medium dairy	0.80	0.86	1.00	0.87
Medium dairy w/ poultry	0.73	0.98	0.95	0.80
Small beef	0.02	1.00	0.04	1.00
Small beef	0.09	1.00	0.19	1.00
Small beef	0.00	1.00	0.00	1.00
Small beef	0.13	0.81	0.25	0.81
Small beef w/ poultry	0.50	1.00	0.50	1.00
Small dairy	0.18	1.00	0.35	1.00
Small dairy	0.31	1.00	0.50	1.00
Small dairy	0.16	1.00	0.33	1.00
Small dairy	0.81	1.00	1.00	1.00

[1] ab_i = extent to which baseline scenario meets area requirements of farm *i* as calculated by Equation (3.4) with $ab = aw$ and a_o = area in baseline scenario contributing toward requirement for farm *i* [ha].
[2] $a_i = \min\left\{\left(\frac{aw}{ab}\right)_i, 1\right\}$ where *aw* = extent to which working scenario meets area requirements of farm *i*, as calculated by Equation (3.4), and *ab* as above.

Table 4.11: Proportion of area requirements met for a given Mini-Muddy Creek scenario, as compared to the baseline, when medium and high-level requirements are individually applied

Farm type	High-level requirement		Medium-level requirement	
	ab_i[1]	a_i[2]	ab_i[1]	a_i[2]
Large beef	0.39	0.77	0.78	0.77
Large dairy	0.70	0.79	0.90	0.67
Medium beef	0.24	1.00	0.47	0.99
Medium beef	0.37	1.00	0.74	1.00
Medium dairy	0.76	0.74	1.00	0.62
Poultry	0.67	1.00	1.00	1.00
Small beef	0.69	0.72	1.00	0.99
Small beef	0.30	1.00	0.59	1.00
Small beef	0.29	1.00	0.57	1.00
Small beef	0.50	0.52	1.00	0.52

[1] ab_i = extent to which baseline scenario meets area requirements of farm i as calculated by Equation (3.4) with $ab = aw$ and a_o = area in baseline scenario contributing toward requirement for farm i [ha].
[2] $a_i = \min\left\{\left(\frac{aw}{ab}\right)_i, 1\right\}$ where aw = extent to which working scenario meets area requirements of farm i, as calculated by Equation (3.4), and ab as above.

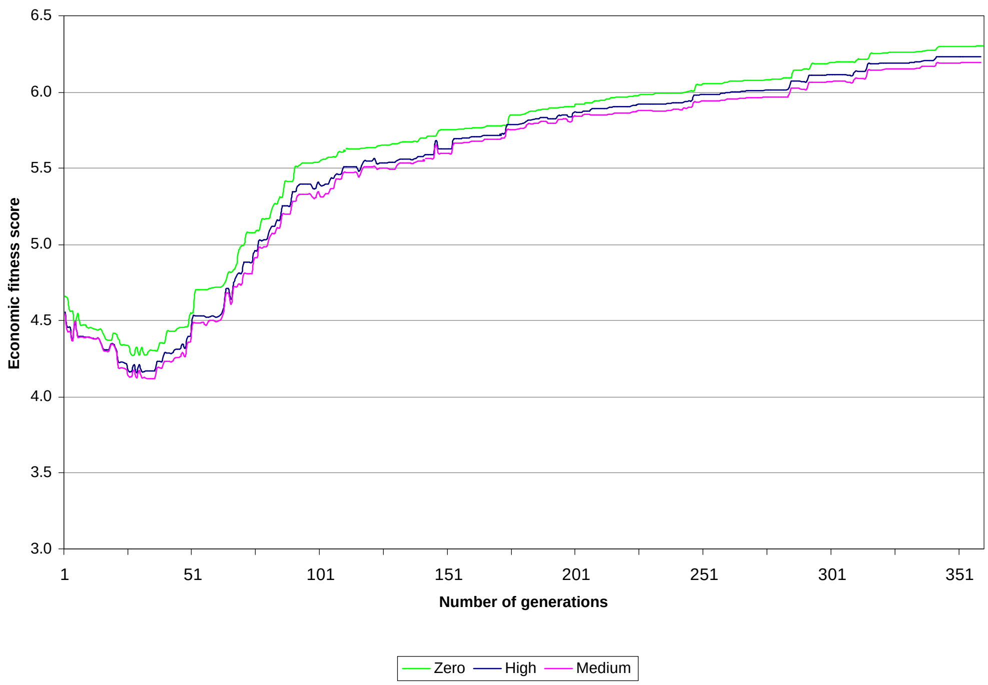

Figure 4.16: Economic fitness score comparison across three area requirement levels for Lola Run, where scores were determined by applying the economic component individually to each scenario in a set of best-of-generation scenarios

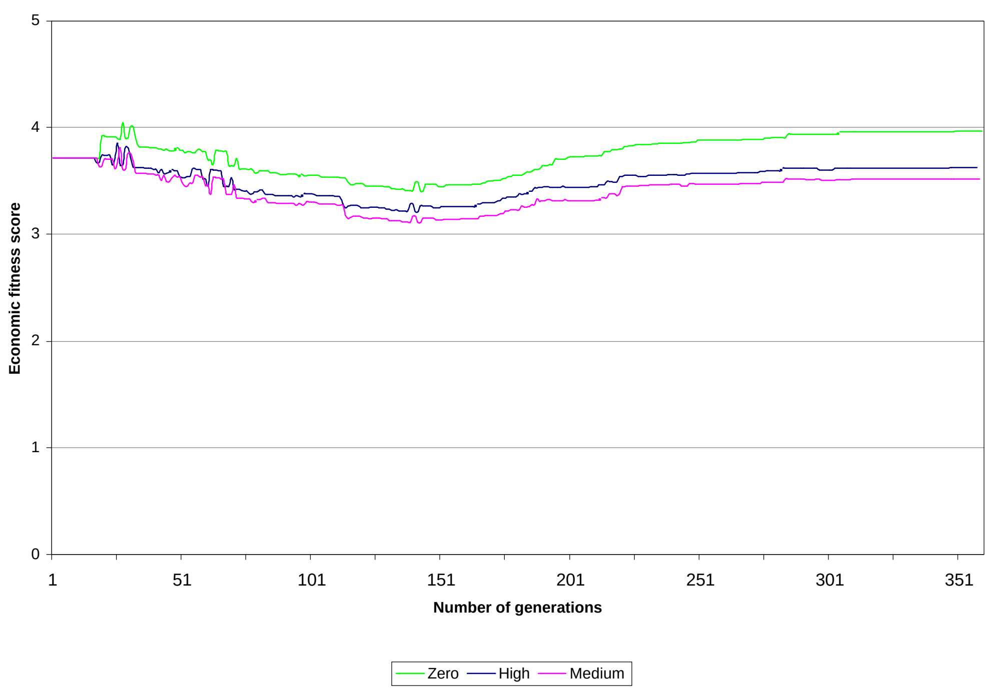

Figure 4.17: Economic fitness score comparison across three area requirement levels for Mini-Muddy Creek, where scores were determined by applying the economic component individually to each scenario in a set of best-of-generation scenarios

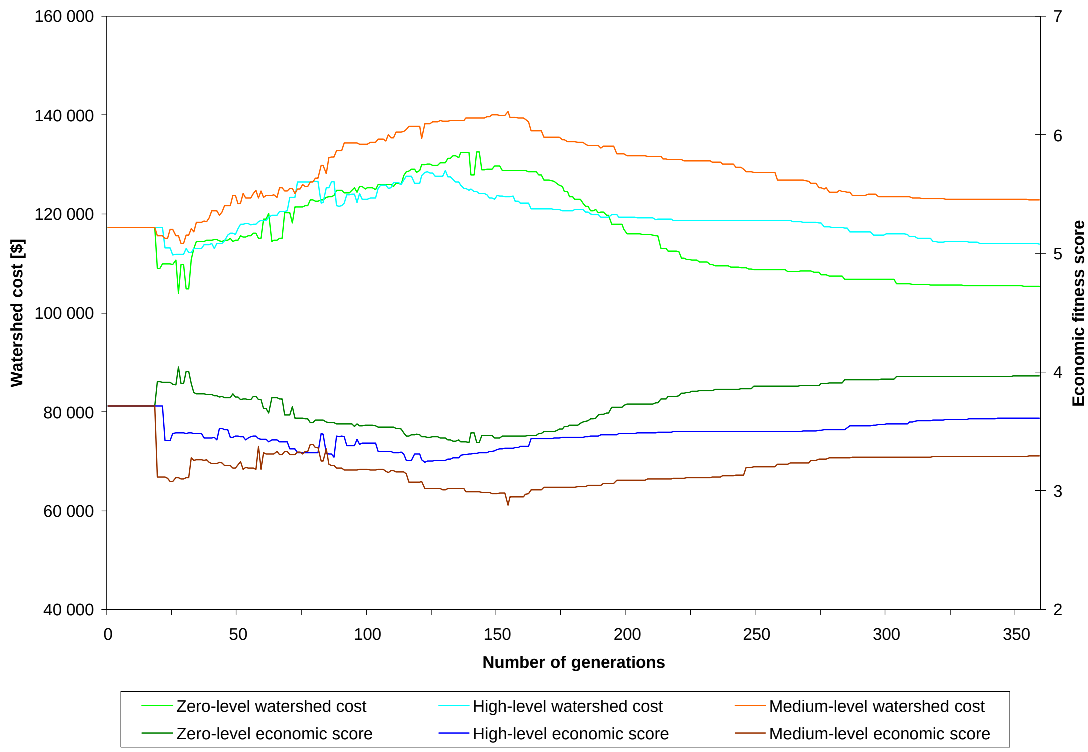

Figure 4.18: Fitness score comparison for three optimization runs under different area requirement levels for Mini-Muddy Creek

For Lola Run, the economic score for the zero-level run was consistently greater than or equal to that of the medium-level run (Figure 4.19). However, for the majority of the run, the high-level run scored higher than the zero-level run and had a lower watershed cost. From generation 90 to 135, when watershed costs for the zero-level and high-level runs were near equal, the high-level run had a lower fitness score than the zero-level run. This was due to the penalizing impact of the area-requirements on the fitness score. However, for the remainder of the run, the lower watershed cost for the high-level run resulted in the high-level fitness score being higher than that of the zero-level. Sixty percent of the farms in Lola Run are dairy, which have a cropland area requirement. The cropland used in this evaluation (corn silage) was more profitable than hay or pasture for most fields. Thus, as compared to the zero-level situation, the penalizing factor of not growing corn silage in the high-level run effectively increased the cost associated with switching to hay or pasture. The result was that the optimization moved towards an optimal solution in the high-level run more quickly than in the zero-level run for this watershed. In the high-level run a watershed cost of $146 638 was reached by generation 359 (Figure 4.19). In the zero-level run this cost was not achieved until generation 406 (Figure 4.10).

4.4.3 Farm-level versus watershed-level cost function

The economic fitness function (Equation 3.5) uses a farm-level summation of costs in the denominator. That is, through use of the Euclidean distance metric, total farm costs are calculated and weighted before being combined together. This formulation enables

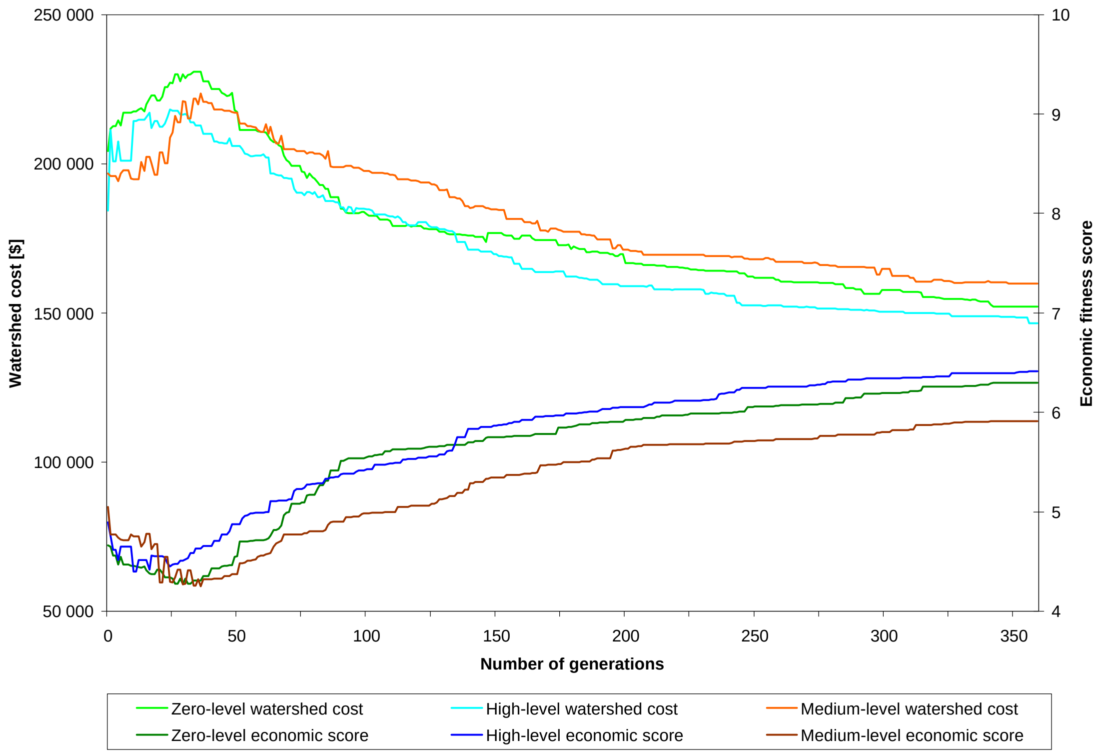

Figure 4.19: Fitness score comparison for three optimization runs under different area requirement levels for Lola Run

consideration of cost fairness (Section 3.3.4). However, it results in a less direct function for optimizing total watershed cost than does using a watershed-level summation in which costs for each field are simply added over the watershed. Optimization of total watershed cost is a main goal of the economic component of the optimization procedure. Thus, the following hypothesis was evaluated to test the impact of the farm-level formulation on the optimization process:

- The optimization procedure continues to locate solutions of decreasing total watershed cost when using an optimization function which sums costs by farm instead of by watershed.

This hypothesis was tested by comparing two versions of Optimization Run 3 (Table 4.9) on each watershed. The first version used Equation (3.5) as the economic fitness function. The second version used Equation (4.2), which sums costs at the watershed level without making a farm-level distinction.

$$e = \frac{C_o}{\sum_i \frac{x_i}{a_i} + 1} \tag{4.2}$$

where

e = economic fitness function,
C_o = total opportunity cost for all farms in scenario [$],
x_i = cost of working scenario for farm i [$],
$x_i = (c_w + d_w)_i$,
c_w = private cost of working scenario for farm i [$],
d_w = public cost of working scenario for a farm i [$],
$a_i = \min\left\{\left(\frac{aw}{ab}\right)_i, 1\right\}$,
aw = extent to which working scenario meets area requirements of farm i,
ab = extent to which baseline scenario meets area requirements of farm i, and
i indexes all farms in the watershed.

In both versions, no area requirements were used. Thus, $a_{i,}$ equaled 1 throughout each optimization run.

For Lola Run the total watershed cost curves by the two summation methods were shaped similarly (Figure 4.20). In comparison, the farm-level cost measure curve was shaped similarly, but had a reduced range of values. The total watershed cost, for both methods, was consistently approximately three times larger than the farm-level cost measure. Thus, the economic fitness function using total watershed cost in the denominator, instead of the farm-level cost measure, was one-third as large as the other economic fitness function. However, for both summation methods, the fitness functions increased steadily while the total watershed cost decreased. Additionally, total watershed costs between the two methods were similar throughout the runs. Similar results were seen for Mini-Muddy Creek (Figure 4.21).

For Mini-Muddy Creek, the solution for the farm-level optimization run resulted in a total watershed cost six percent less than that of the watershed-level optimization solution. In contrast, the solution for the farm-level optimization run for Lola Run resulted in a total watershed cost one percent greater than that of the watershed-level optimization solution. Total watershed cost differences between the two methods may be due to differences in watershed size and number of farms as well as in farm sizes, types, and productivities. Evaluation of multiple watersheds of varying characteristics may be instructive in determining characteristics most influencing cost differences between the two methods.

In both methods, increases in fitness corresponded with decreases in total watershed cost. These results indicate that the use of a farm-level cost measure in the optimization procedure does not adversely affect the overall economic goal of optimizing watershed costs. For Mini-Muddy Creek, the farm-level optimization version had a positive impact in reducing costs beyond solutions provided by the watershed-level version. Additionally, use of the farm-level measure has the benefit of incorporating a measure of fairness in cost increase.

4.5 Evaluation of NPS component

The NPS component was evaluated with regard to sediment yield simulation both by changing the BMP applied over the majority of a watershed and by moving the location of a single land use within the watershed. The impact of BMP selection on sediment yield estimated by the NPS component was evaluated by comparison against the detailed ANSWERS-2000 NPS model. The purposes of this comparison were twofold. The first goal

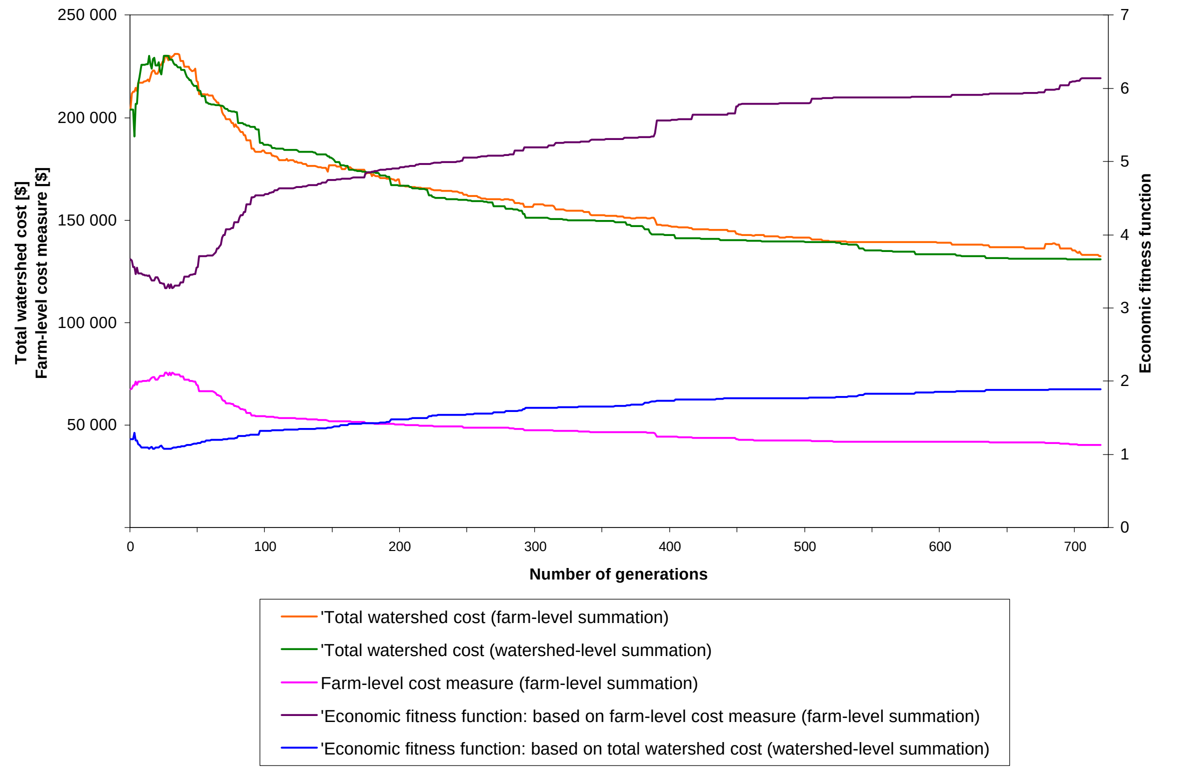

Figure 4.20: Comparison of the economic fitness function based on farm-level and watershed-level cost summations for Lola Run

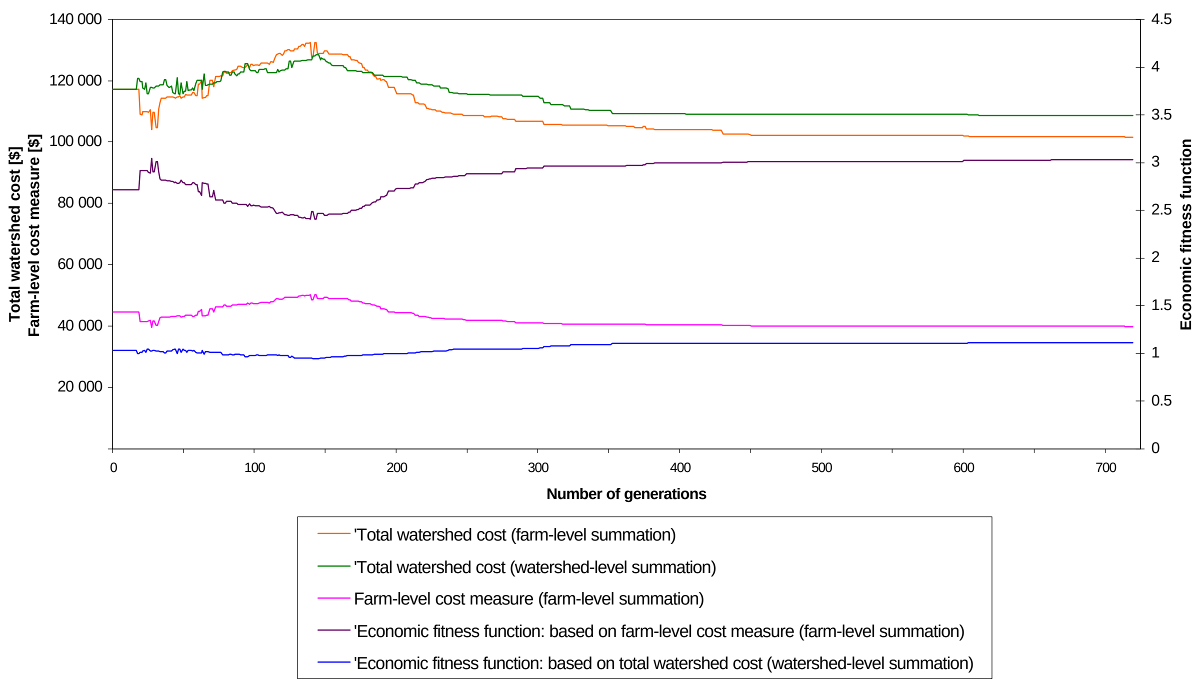

Figure 4.21: Comparison of the economic fitness function based on farm-level and watershed-level cost summations for Mini-Muddy Creek

was to assess the magnitude of difference in prediction levels between the NPS component and a more detailed model. The second goal was to compare relative differences in scenario yields between NPS prediction tools. The effect of land use placement within a watershed on sediment yield was tested to confirm accurate trends in the routing portion of the NPS component.

4.5.1 Impact of BMPs on sediment yield

To compare the impacts of different BMPs on sediment yield at the watershed outlet, all fields in cropland and hay in the baseline scenario were placed in one of five management practices (conventionally tilled corn silage, conventionally tilled corn silage with winter wheat grain, minimum-tillage corn silage, minimum-tillage corn silage with winter wheat grain, and grass hay) for a total of five scenarios for each watershed. Forest, pasture, and farmstead were kept constant for all scenarios.

The NPS component and ANSWERS-2000 were used to simulate sediment yield at the watershed outlet for each scenario and each watershed. Data used for the NPS component are discussed in Section 4.2. The ANSWERS-2000 input files (Appendix C) were created to reflect the same watershed characteristics as used for the NPS component. For all ANSWERS–2000 runs, a five-year set of weather data was simulated by the CLIGEN v5.017 weather simulation program (Nicks et al., 1995) using historical data collected at Dale Enterprises, Virginia.

The distributed, physically-based nature of ANSWERS-2000 makes it capable of simulating trends in sediment yield among different scenarios, as well as spatial variability within scenarios. Byne (2000) tested ANSWERS-2000 for a growing season in the Owl Run watershed (1153 ha) in Fauquier County, Virginia. Owl Run is a mixed land-use watershed with 66% agricultural, 26% forest, and 8% residential areas. Byne (2000) found that ANSWERS-2000 overpredicted total sediment yield at the watershed outlet for that growing season by nine percent.

Sediment yield at the watershed outlet is shown in Table 4.12 for each management practice scenario. Additionally, the ratio of the yield resulting from that scenario to the yield from the conventionally tilled corn silage scenario was calculated (Table 4.12).

For each scenario, results of ANSWERS-2000 were similar between the two watersheds, with slightly larger yields for Lola Run. The same trend was seen for the NPS component predictions. Yield differences are likely due to spatial variation of land use between watersheds, a smaller percentage of the watershed being covered by forest, and/or differences in watershed topography and flow patterns. The relative similarity in results across watersheds demonstrated consistency of the prediction tools across similar watersheds.

For the corn scenarios, the sediment yield predicted by the NPS component was up to an order of magnitude larger than that predicted by ANSWERS-2000. Scaling the results to the sediment yield of conventionally tilled corn silage shows that the treatment differences were fairly consistent. For each watershed, the ratio for each management practice to corn silage, except hay, was about 30% less for the NPS component than ANSWERS-2000. The NPS component ratio for hay was 40% of the ANSWERS-2000 ratio for the Mini-Muddy Creek watershed and 20% for the Lola Run

Table 4.12: Comparison between the NPS component and ANSWERS-2000 predictions of average annual sediment yield at the watershed outlet for different scenarios

	Conventionally tilled corn silage	Conventionally tilled corn silage with winter wheat grain	Minimum-tillage corn silage	Minimum - tillage corn silage with winter wheat grain	Grass hay
			Simulated sediment yield		
	[Mg/ha]	[Mg/ha]	[Mg/ha]	[Mg/ha]	[Mg/ha]
Mini-Muddy Creek					
NPS	4.715	2.743	1.767	1.255	0.076
ANSWERS-2000	0.456	0.394	0.234	0.191	0.023
Lola Run					
NPS	6.307	3.545	2.252	1.567	0.078
ANSWERS-2000	0.723	0.609	0.497	0.307	0.042
			Ratio to conventionally tilled corn silage		
	[-]	[-]	[-]	[-]	[-]
Mini-Muddy Creek					
NPS	--	0.58	0.37	0.27	0.02
ANSWERS-2000	--	0.86	0.51	0.42	0.05
Lola Run					
NPS	--	0.56	0.36	0.25	0.01
ANSWERS-2000	--	0.84	0.69	0.42	0.06

watershed. However, all predictions showed a similar trend in sediment yield across the five scenarios. Sediment yield was greatest for conventionally tilled corn silage and decreased as surface cover increased and surface soil roughness decreased.

The scenario ratios across watersheds, as predicted by the NPS component, were nearly equal. The ratios, as predicted by ANSWERS-2000, showed some variation for the minimum tillage corn silage scenario but were nearly equivalent for the other scenarios.

Without validation of the NPS component and/or ANSWERS-2000, the accuracy of the models' predictions of the response of the physical system cannot be determined. Comparison of these two prediction tools suggests that the NPS component may have overpredicted sediment yield, assuming that ANSWERS-2000 is the more accurate model. However, each model maintained constant ratios between scenarios across watersheds. The conclusions of this section impact the representation of target loads in the NPS component. The desired target load should be expressed relative to the baseline load as calculated by the NPS component. This will ensure that a consistent frame of reference for NPS calculations is maintained within the optimization procedure.

Although these results do not extend directly to watersheds in dissimilar agroecosystems, the models and theories used in both ANSWERS-2000 and NPS components are generally applicable. Thus, it is expected that, to the extent that the USLE is appropriate for a particular watershed (i.e., a watershed not dominated by gully or channel erosion), the NPS component would accurately reflect relationships in sediment yield among scenarios.

4.5.2 Impact of land use placement on sediment yield

The Mini-Muddy Creek watershed was used to test the sediment yield predictions of the NPS component with regard to land use or management practice placement within the watershed. It was expected that sediment yield at the watershed outlet would be increased when erodible land uses were located nearer to streams or to the watershed outlet. For this test a land use layer consisting of seven agricultural fields and two larger land use regions was created (Figure 4.22). Each agricultural field was created to be 3.6 ha in size. A constant USLE K-factor of 0.042 Mg·ha·h/(ha·MJ·mm) was assigned to eliminate variability in soil erodibility. Slopes in each field ranged from two to five percent. The C factor was set at 0.003 for forest, 0.01 for grass hay, and 0.49 for conventionally tilled corn silage areas. For all regions, a P factor of one was used. The α value was set at 1.1 for forest, 3.3 for grass hay, and 9.7 for corn silage.

Eight runs of the NPS component were performed. The upper region remained in forest for all test runs. For the reference run, the lower region of the watershed, including all seven agricultural fields, was placed in grass hay. For each of the seven test runs, a different agricultural field was placed in conventionally tilled corn silage. The remainder of the lower region was placed in grass hay.

For each test run the NPS component was used to calculate sediment loading from the watershed. Table 4.13 reports the gross erosion in the watershed by summing the megagrams of gross erosion per cell over all cells in the watershed and dividing the total by the watershed area. The gross erosion within the corn field was determined by summing the megagrams of gross erosion per cell over only the cells in the field and then dividing by the field area. The sediment yield at the watershed outlet is an accumulation over the entire watershed of the percentage of gross erosion (in Mg) delivered from each cell to the outlet. This quantity is divided by the area of the watershed to obtain Mg/ha. The results show that placement did affect sediment yield.

The differences in sediment yield did not vary consistently with the differences in gross erosion. For example, for field 1 sediment yield was one-half as much as gross erosion from the reference run. However, field 2 experienced a greater increase in sediment yield than field 1 but not in gross erosion. Test run 7 resulted in the least increase in gross erosion from the reference. The smallest increase in sediment yield was seen in test run 6 when field 6 was in corn.

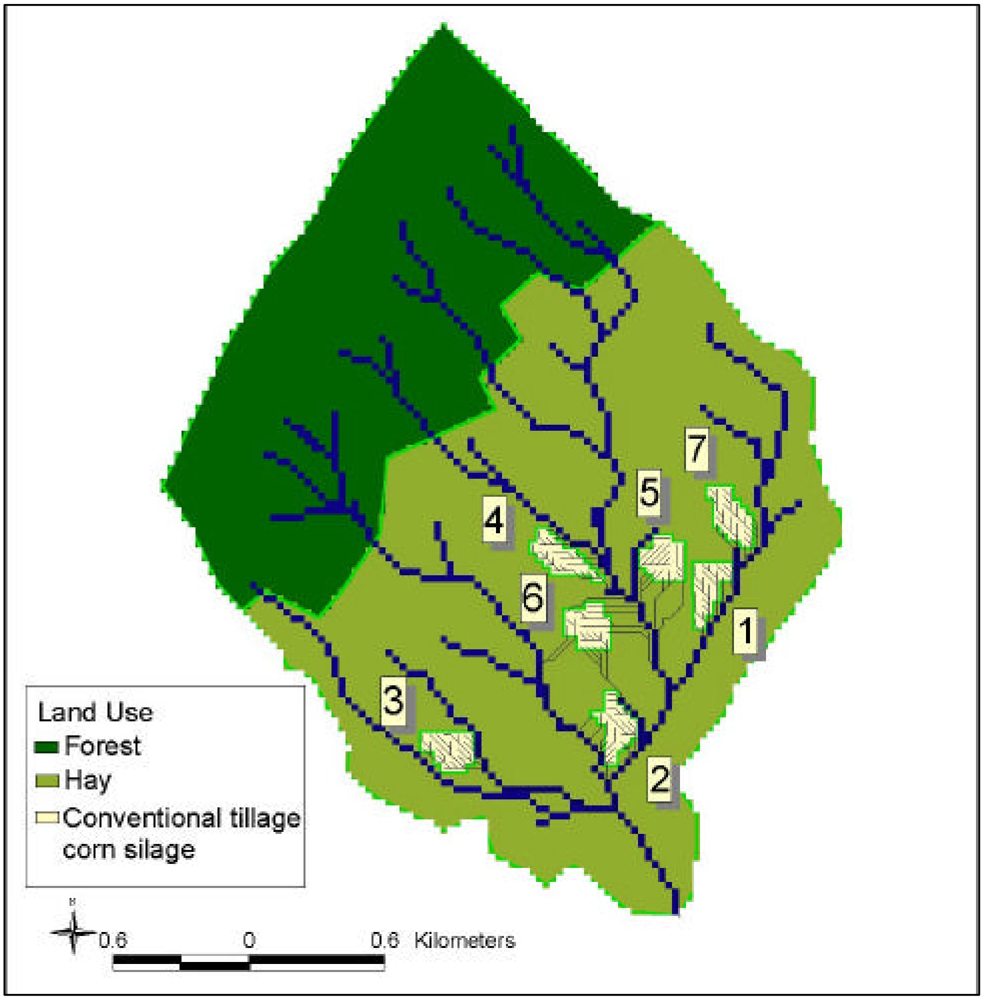

Figure 4.22: Fields and flow networks used to test sediment routing due to BMP placement

Table 4.13: Output from NPS model for placement test fields

Test run	Corn silage field	Gross erosion in watershed (Mg/ha)	Gross erosion within field (Mg/ha)	Increase in watershed gross erosion compared with reference run (%)	Sediment yield at watershed outlet (Mg/ha)	Increase in sediment yield compared with reference run (%)
Ref.	None	1.400	N/A	N/A	0.089	N/A
1	1	1.549	28.962	11	0.094	6
2	2	1.558	30.714	11	0.105	18
3	3	1.565	31.971	12	0.098	10
4	4	1.554	29.864	11	0.090	2
5	5	1.572	33.287	12	0.103	16
6	6	1.562	31.468	12	0.089	0
7	7	1.529	24.954	9	0.089	0

The differences in gross erosion among test runs are explained by the slope steepness and flow length characteristics of the cells in each field. These two factors contributed to variations in the S and L factors of the USLE, while all other factors of the USLE were controlled as discussed at the beginning of this section. Flow patterns and slopes for each field are shown in Figure 4.22 and Table 4.14, respectively.

Table 4.14: Distribution of slopes and flow directions for placement test fields

Field	Percent of diagonal flow cells	Percent of cells in each slope category		
		2-3 %	3-4 %	4-5 %
1	40	33	35	33
2	55	20	43	38
3	60	30	20	50
4	35	33	48	20
5	52	13	53	35
6	30	23	45	33
7	53	75	20	5

Eighty-eight percent of field 5, which has the highest level of gross erosion, has slopes of at least three percent. Also, 52% of its cells have a diagonal flow direction and thus a longer flow path than non-diagonally flowing cells. Fields 2 and 3, also among the highest in gross erosion, are composed mainly of at least three percent slopes, and have a majority of diagonally flowing cells. In contrast, field 7, which contributed the least gross erosion, has 75% slopes less than three percent and has a majority of non-diagonally flowing cells.

The increases in sediment yield relative to the reference run were as expected based on the placement of the fields. Field 2 in corn yielded the greatest amount of sediment at the watershed outlet; 6 of the 40 cells in the field flowed directly into a stream. Also field 2 is nearest the outlet. Portions of fields 3 and 5 flow directly into streams as well, but in less concentrated flow patterns than field 2. Additionally, fields 3 and 5 are farther from the outlet than field 2.

All outflow cells on the remaining fields (fields 1, 4, 6, and 7), except two cells of field 1, were buffered from streams by one or more cells of hay. Additionally, on these remaining fields, flow exiting the stream side of the field was distributed along the field edge instead of concentrating in one or two exit cells.

Relative differences in sediment yield due to differences in field locations within the watershed followed expected trends. For example, field 1 was located just downstream of field 7. Both fields were bordered by the same stream and had one to two cell widths of hay buffer along most of the stream edge. As expected, watershed sediment yield for the downstream field was greater than for the upstream field (0.094 Mg/ha vs. 0.089 Mg/ha).

4.6 Evaluation of optimization procedure

The optimization procedure, as a whole, was evaluated with respect to the first research objective of pollution reduction and cost decrease as well as the overall research goal of cost-effectiveness. The cost-effectivenesses of three optimization runs were compared with a targeting strategy. Land allocation and spatial placement of BMPs within the watershed were compared for the optimization and targeting strategy solutions. Additionally, differences in land allocation for two alternate solutions of a single optimization run were considered.

4.6.1 Cost-effectiveness

With an optimization heuristic such as a GA, it is not possible to guarantee that the heuristic will find the global optimum. However, for this research the goal was not to locate the true optimum, but to demonstrate the ability of the optimization procedure to find more cost-effective solutions than a targeting strategy. To assess performance of the optimization procedure with respect to the research goal, the following hypothesis was evaluated:

- The optimization procedure locates one or more scenarios that are more cost-effective than the targeting strategy solution scenario.

Three optimization runs and a targeting strategy, as described in Table 4.9, were applied to both evaluation watersheds (Lola Run and Mini-Muddy Creek). These four strategies were compared relative to the baseline scenario for each watershed.

For the targeting strategy and Optimization Runs 1 and 2, the same fields were allowed to vary with respect to management practices. Additionally, the highest return management practice for each field was identical across these three BMP placement strategies. Thus, all three strategies had the same opportunity cost. In Optimization Run 3 additional fields were allowed to vary and additional management practice alternatives for each field were considered. As a result, the opportunity cost for Run 3 was different than that of the other strategies (Table 4.15).

The targeting strategy was applied to both watersheds using the NPS and economic components of the optimization procedure. Sediment yield and watershed cost for the baseline scenario and the solution under the targeting strategy are shown in Table 4.15. Watershed cost for the baseline scenario used for the targeting strategy and Runs 1 and 2 was zero because conventionally tilled corn silage was the profit maximizing practice for all cropped fields. Sediment reduction and cost increase of the targeting strategy from the baseline were calculated. Sediment yield was reduced to 19% of the baseline for Lola Run and 26% of the baseline for Mini-Muddy Creek.

Next, the maximum acceptable, or target, pollutant load for the optimization procedure was set equal to the pollutant loading resulting from the targeting strategy. Because the targeting strategy did not consider area requirements, no area requirements were imposed in the optimization runs. For all cases the optimization procedure was run for 720 generations. At this point all runs had met their respective target-loading criterion and the rate of total watershed cost decrease for each run had become minimal (Figures 4.8 to 4.13).

The sediment yield of the best scoring individual of the final population was recorded and pollutant reduction from the baseline calculated. Total watershed cost was determined by summing public and private costs, which include opportunity costs, for each farm in the

Table 4.15: Comparison of three optimization procedure runs[1] with the targeting strategy[2]

	Opportunity cost [$]	**Sediment yield at watershed outlet [Mg/ha]**		**Total watershed cost [$]**	
		Baseline scenario	Final solution	Baseline scenario	Final solution
Lola					
Targeting	108 184	3.45	0.64	0	134 892
Optimization					
Run 1	108 184	3.45	0.64	0	117 426
Run 2	108 184	3.45	0.64	0	89 748
Run 3	247 107	3.45	0.64	138 923	132 284
Mini-Muddy					
Targeting	3515	0.65	0.17	0	29 434
Optimization					
Run 1	3515	0.65	0.17	0	27 773
Run 2	3515	0.65	0.17	0	24 803
Run 3	120 811	0.65	0.17	117 296	101 687

	Final sediment reduction from baseline [Mg/ha]	**Final increase in total cost (= decrease in net return) from baseline [$]**	**Final cost-effectiveness (x 10^{-5}) [Mg/ha/$]**
Lola			
Targeting	2.81	134 892	2.08
Optimization			
Run 1	2.81	117 426	2.39
Run 2	2.81	89 748	3.13
Run 3	2.81	-6639	-42.3
Mini-Muddy			
Targeting	0.48	29 434	1.63
Optimization			
Run 1	0.48	27 773	1.73
Run 2	0.48	24 803	1.94
Run 3	0.48	-15 609	-3.08

[1]The optimization procedure uses the pollution reduction from the targeting strategy as the target pollutant load.
Run 1: considers contoured, minimum tillage and a winter cover crop as a replacement for all areas in conventionally tilled corn silage.
Run 2: considers all combinations of conventional and minimum tillage, a winter cover crop, and contouring as replacements for all areas in conventionally tilled corn silage.
Run 3: includes hay and pasture to the combinations considered in Run 2 and allows replacement on all agricultural land.

[2]The targeting strategy replaces conventional tillage with contoured, minimum tillage and a winter cover crop for all fields in corn silage for which a majority of the field has greater than 3% slope.

solution scenario. Cost increase was determined by subtracting the baseline watershed cost from that of the solution scenario. This difference reflects the decrease in net return realized by the solution scenario as compared to the baseline scenario. The sediment reduction and cost increase results are tabulated in Table 4.15.

Cost-effectiveness, defined as pollution reduction divided by total watershed cost increase, was calculated for both the targeting strategy and optimization solutions (Table 4.15). Cost-effectiveness reflects the Mg/ha of pollution not leaving the watershed per dollar spent on pollution control. In both watersheds, total watershed cost increased under the targeting strategy, resulting in a positive cost-effectiveness ratio. Cost-effectiveness improved from the targeting strategy as more management practices were allowed in the optimization runs.

As discussed in Section 3.3, representing cost-effectiveness as a ratio does not define a clear response surface for this problem. Because the denominator of the ratio may be positive or negative, mathematical increase in the cost-effectiveness ratio does not necessarily correspond to increased pollution reduction per dollar spent. The correct relationship does hold as long as the sign of the denominator (and hence the ratio) remains constant. However, the cost-effectiveness ratio provides a quantitative measure of comparison across strategies when one factor of the ratio (pollution reduction or cost increase) is held constant.

For example, in this evaluation, with the single pollutant of sediment being considered, positive cost-effectiveness represents the Mg/ha of sediment not leaving the watershed per dollar spent. Sediment reduction is being held constant across the strategies by requiring that a target pollutant load be met. The goal is then on decreasing costs as much as possible. In this case, a more positive cost-effectiveness score is preferred among scenarios in which costs increase from the baseline. However, among scenarios in which the costs decrease from the baseline (e.g., Optimization Run 3), a more negative score is preferred. Comparing a scenario in which costs increase from the baseline with one in which costs decrease, the latter scenario, with the negative ratio, is preferred.

When choosing among scenarios of equivalent or nearly equivalent cost, the scenario with the higher sediment reduction level is preferred. When the cost is above the baseline, the cost-effectiveness ratio will be positive, whereas when the cost is below the baseline the ratio will be negative. In both cases increased sediment reduction corresponds to an increase in the absolute value of the ratio. Thus, in the first case, more positive ratios are preferred and, in the second case, more negative ratios are preferred.

In terms of pollution reduction and cost decrease, the results from all optimization procedure runs are better than those of the targeting strategy (Table 4.15). These results indicate that the optimization procedure works as intended. Additionally, these results were seen on two similar watersheds, supporting the repeatability of the optimization procedure and the benefits of the procedure for watersheds in a ridge and valley agroecosytem.

In all cases, after meeting the pollutant-targeting criterion, the optimization procedure decreased costs below targeting strategy costs. Thus, cost-effectiveness ratios show that the targeting strategy was less cost-effective than the optimization procedure. Optimization Run 3 was the most cost-effective of the three optimization runs, followed by Run 2 and then Run 1. The negative cost-effectiveness score for Run 3 was a result of reducing costs below the baseline. This cost reduction below the baseline occurred in both watersheds because agricultural practices were allowed to change from the baseline. For example, fields in hay or pasture were converted to corn silage whenever this change decreased costs and still met the pollution targeting criterion.

4.6.2 Land Allocation

Land allocation for non-fixed management units varies among solutions from the optimization procedure as well as between solutions for the optimization procedure and targeting strategy. Fixed management units include farmstead and forested areas. The highest scoring scenarios of Optimization Run 1, Run 2, and Run 3, as well as an alternative solution from Run 3, were compared to each other and to the targeting strategy solution (Table 4.16) for Lola Run. Spatial depictions of land use allocation for the baseline scenario, targeting strategy, and four optimization solutions are shown in Figures 4.23 through 4.28, respectively.

Of the 68 fields placed in conventionally tilled corn silage in the baseline scenario, only 10 were placed in that practice in the targeting strategy. The remainder were assigned a BMP set of contour, minimum-tillage corn silage with a winter wheat cover crop.

The first optimization run, Run 1, achieved the same level of pollution reduction by placing one-third fewer fields in contour, minimum-tillage corn silage with a winter wheat cover crop. As compared to the targeting strategy, Run 1 places more BMPs on fields along the streams. Fields further from the streams, particularly on the watershed edge, were less likely to be affected by this strategy. Several fields with a majority of the slope greater than three percent but not along the streams were affected by the targeting strategy. However, in the Run 1 solution these fields were left in conventional tillage corn silage.

Run 2, which considered all combinations of the targeting strategy BMP set, placed twice as many fields (19) in conventionally tilled corn silage as did the targeting strategy. Nineteen more fields were placed in contour, minimum-tillage corn silage with a winter wheat cover crop while the majority (23) of the remaining fields were placed in contoured, minimum-tillage corn silage. Run 2 mainly placed fields along the watershed edge in conventional tillage corn silage. Corn silage fields along the streams were primarily placed in contoured, minimum-tillage. However some fields, particularly those with steeper slopes were placed in contour, minimum-tillage corn silage with a winter wheat cover crop.

As expected, the two solutions from the third optimization procedure, Run 3, showed a more diverse set of management practices. These solutions used all possible management practices except up and down slope conventionally tilled corn silage with a winter wheat cover crop and minimum-tillage corn silage with a winter wheat cover crop, with and without contour tillage. Differences between the two solutions are shown as fields outlined in red on Figures 4.27 and 4.28. Under both solutions about 30% of the agricultural land was allocated to conventionally tilled corn silage, with or without contouring.

Variation among optimization solutions shown and additional variation present in other optimization solutions with high fitness scores demonstrate the potential for flexibility in BMPs within a scenario. Comparison of diversity in the optimization and targeting strategy solutions indicates that variation can be introduced into the watershed while still meeting the pollutant target load criterion. Examining and possibly combining different solutions

Table 4.16: Agricultural land allocation in Lola Run for the baseline scenario, targeting strategy, and four optimization solutions

Management Practice	Baseline	Targeted	Run 1	Run 2	Run 3 best solution	Run 3 alternate solution
	# of fields (ha)	# of fields (ha)	# of fields (ha)	# of fields (ha)	# of fields (ha)	# of fields (ha)
CC	68 (392.0)	10 (34.7)	28 (83.2)	19 (42.5)	24 (78.6)	25 (85.6)
CC / WW					2 (0.8)	2 (0.4)
MC				4 (18.6)	5 (15.3)	4 (4.0)
MC / WW				1 (6.6)		
CC (2 yrs) / H (3 yrs)					7 (42.2)	6 (32.5)
CC (1 yr) / MC (1 yr) / H (3 yrs)					5 (37.2)	3 (5.1)
H	44 (288.2)	44 (288.2)	44 (288.2)	44 (288.2)	3 (51.9)	3 (51.9)
pasture	13 (94.6)	13 (94.6)	13 (94.6)	13 (94.6)	15 (147.4)	18 (184.0)
CC, contoured				2 (3.3)	28 (145.2)	27 (141.8)
MC, contoured				23 (140.3)	17 (103.6)	19 (114.5)
MC / WW, all contoured		58 (357.4)	40 (308.9)	19 (180.8)		
CC (2 yrs) / H (3 yrs), all contoured					10 (112.9)	8 (95.4)
CC (1 yr) / MC (1 yr) / H (3 yrs), all contoured					9 (39.6)	10 (59.6)

[1]CC = conventional corn silage; WW = winter wheat; MC = minimum till corn silage; H = grass hay

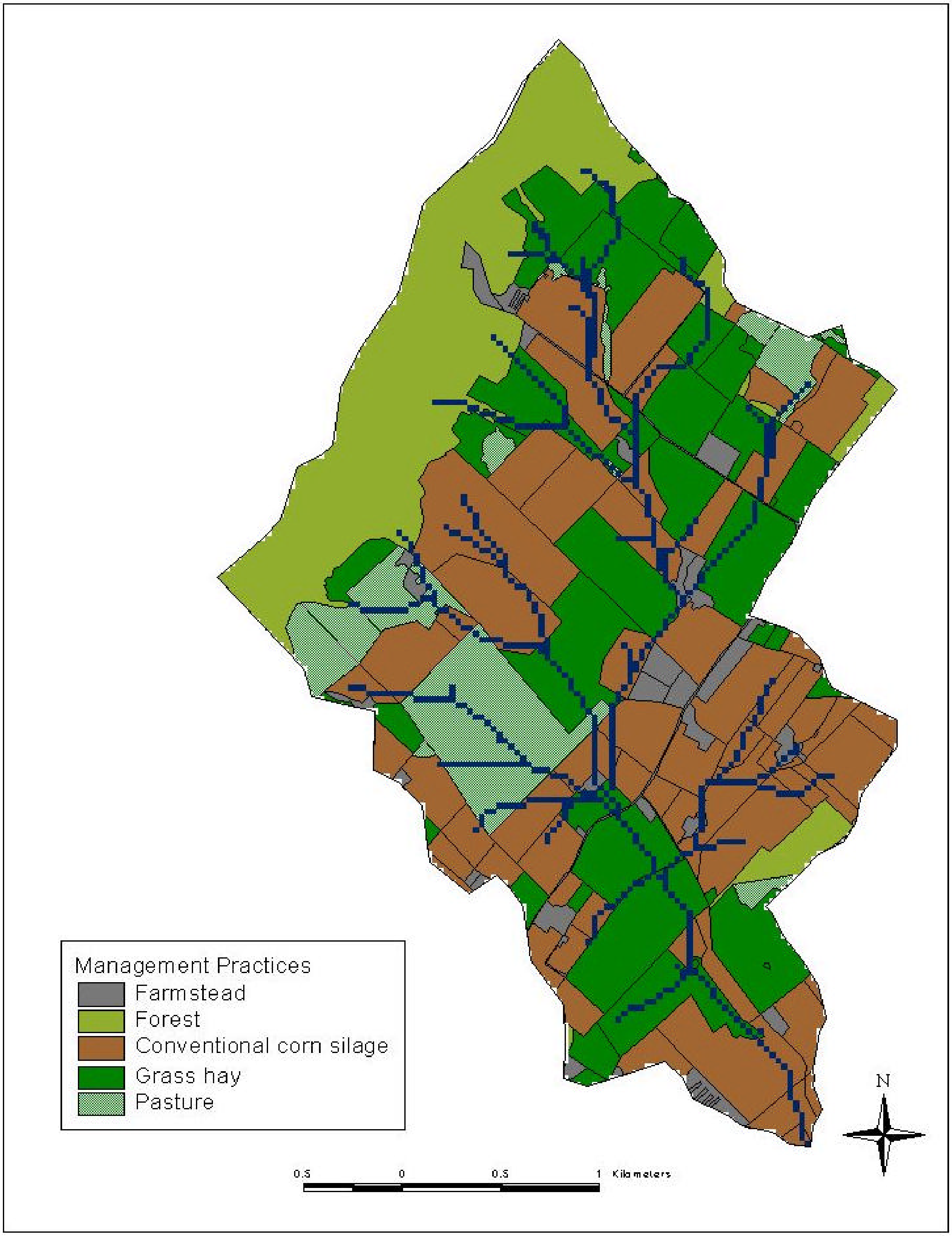

Figure 4.23: Agricultural land use allocation for the baseline scenario in Lola Run

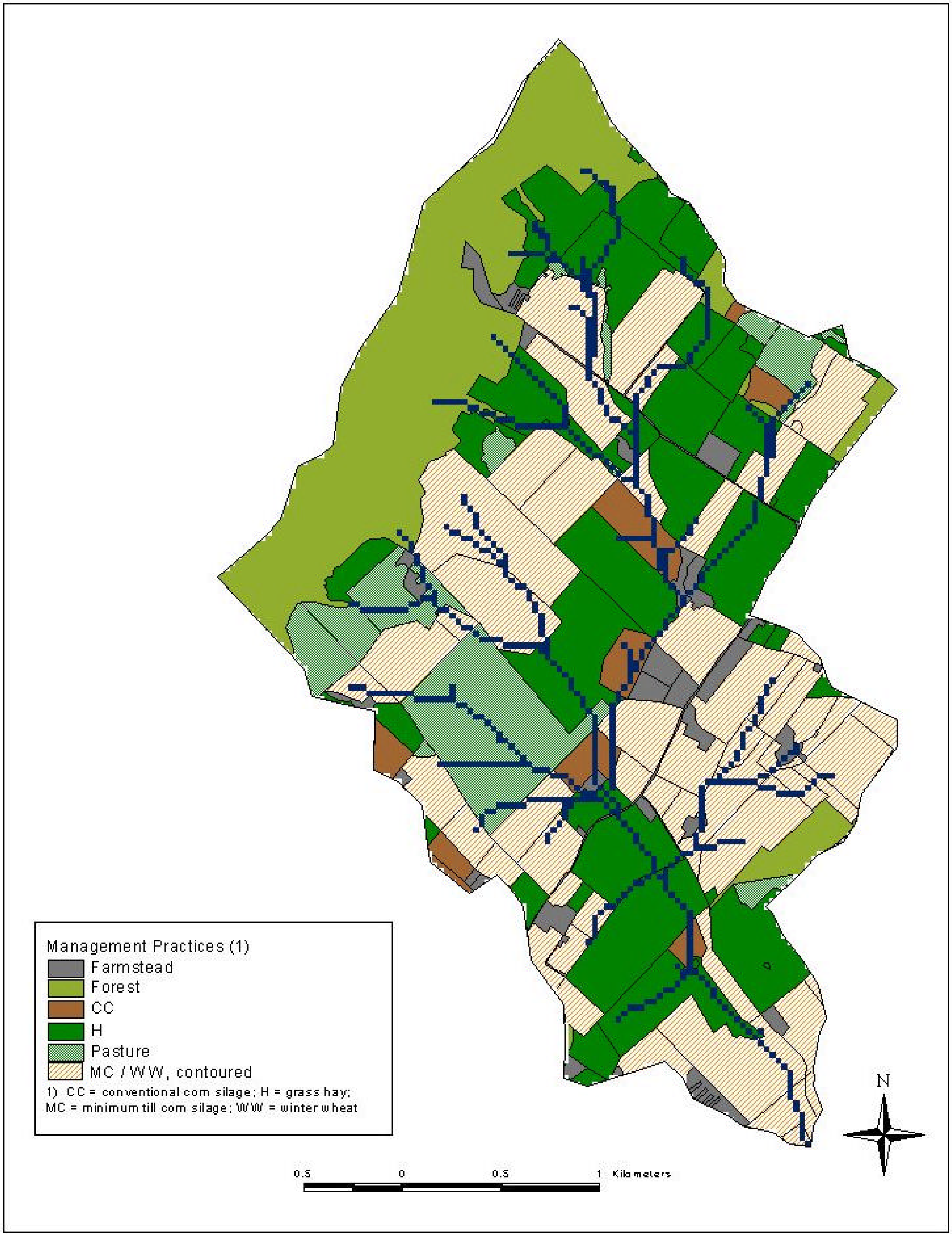

Figure 4.24: Agricultural land use allocation for the targeting strategy in Lola Run

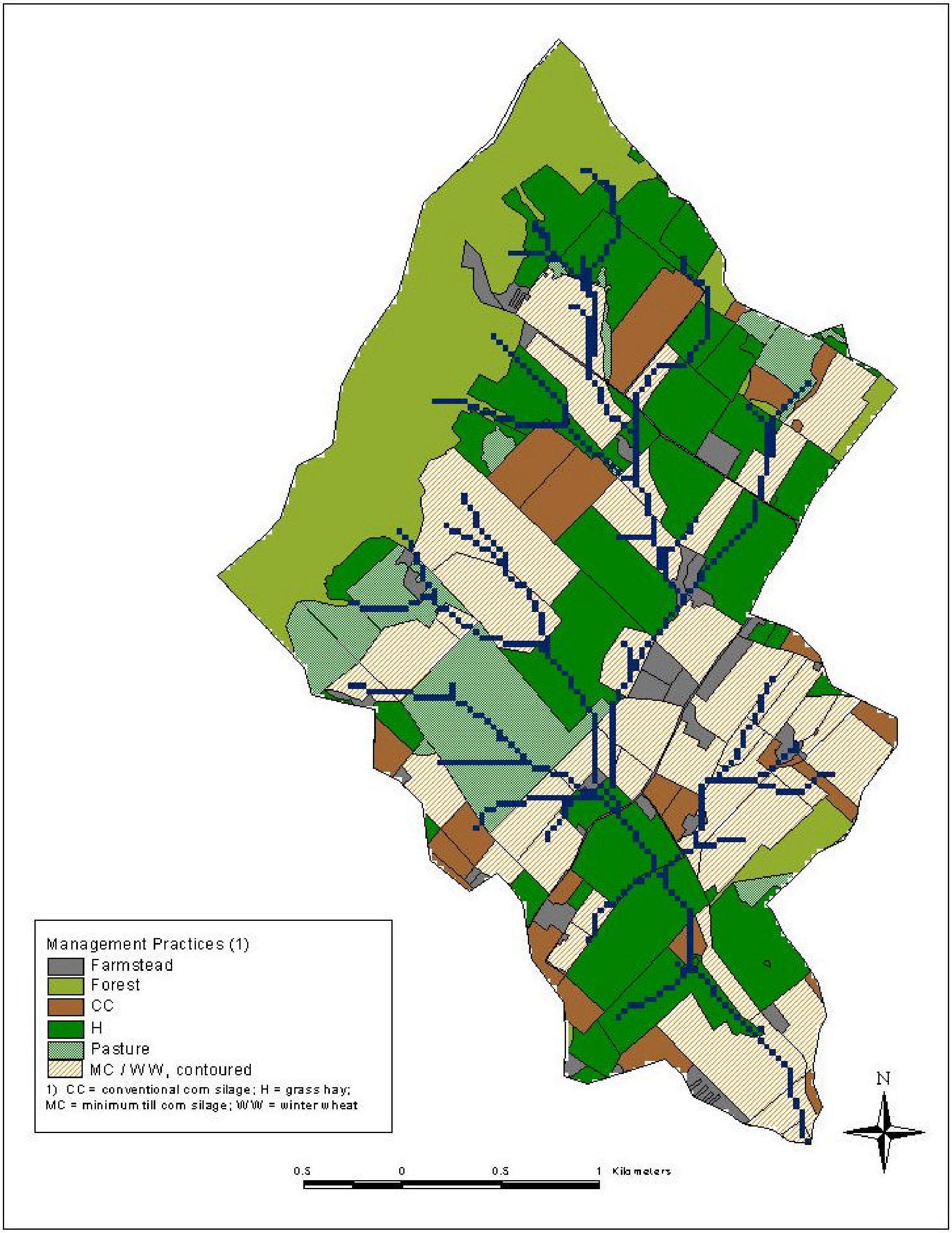

Figure 4.25: Agricultural land use allocation for the Optimization Run 1 solution in Lola Run

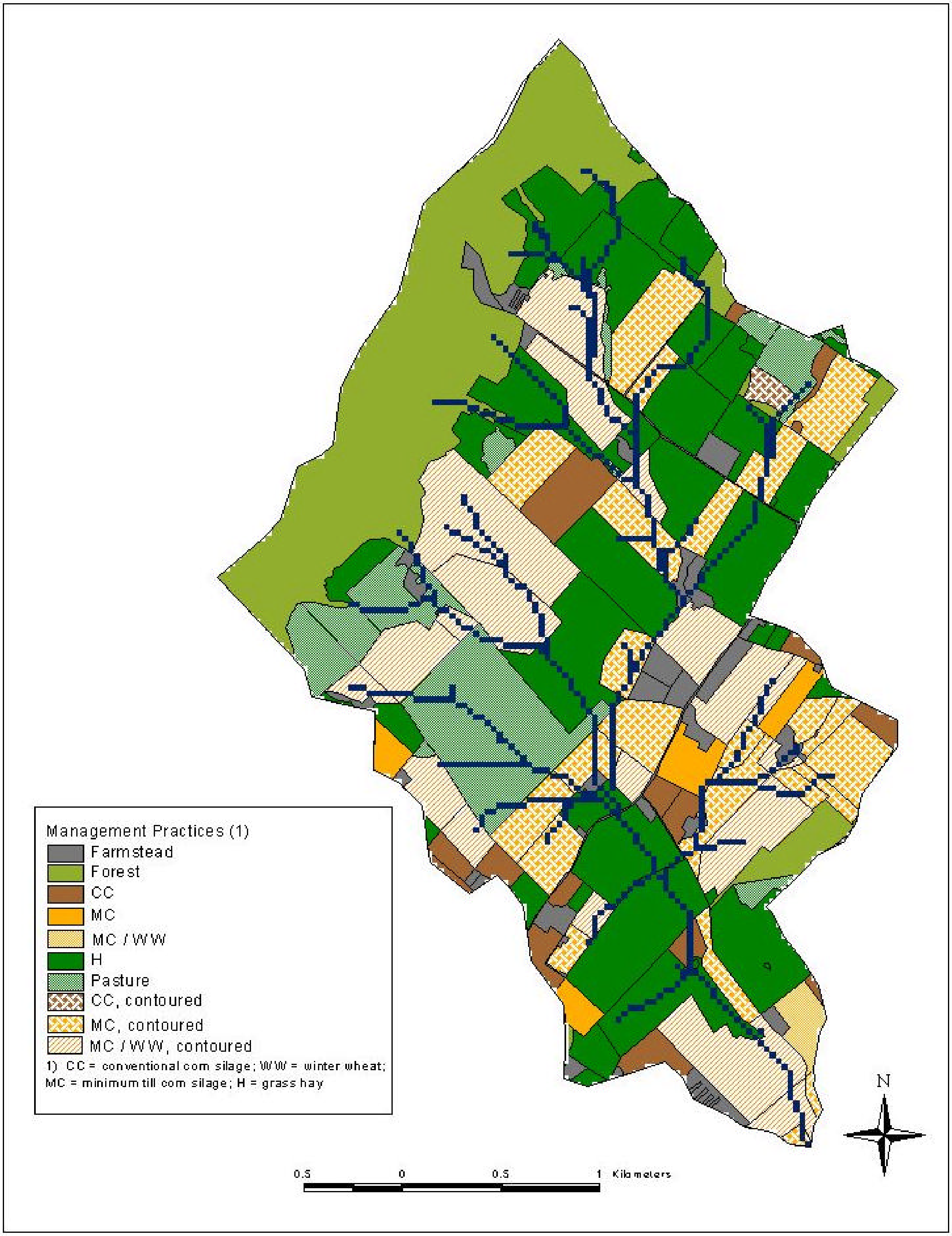

Figure 4.26: Agricultural land use allocation for the Optimization Run 2 solution in Lola Run

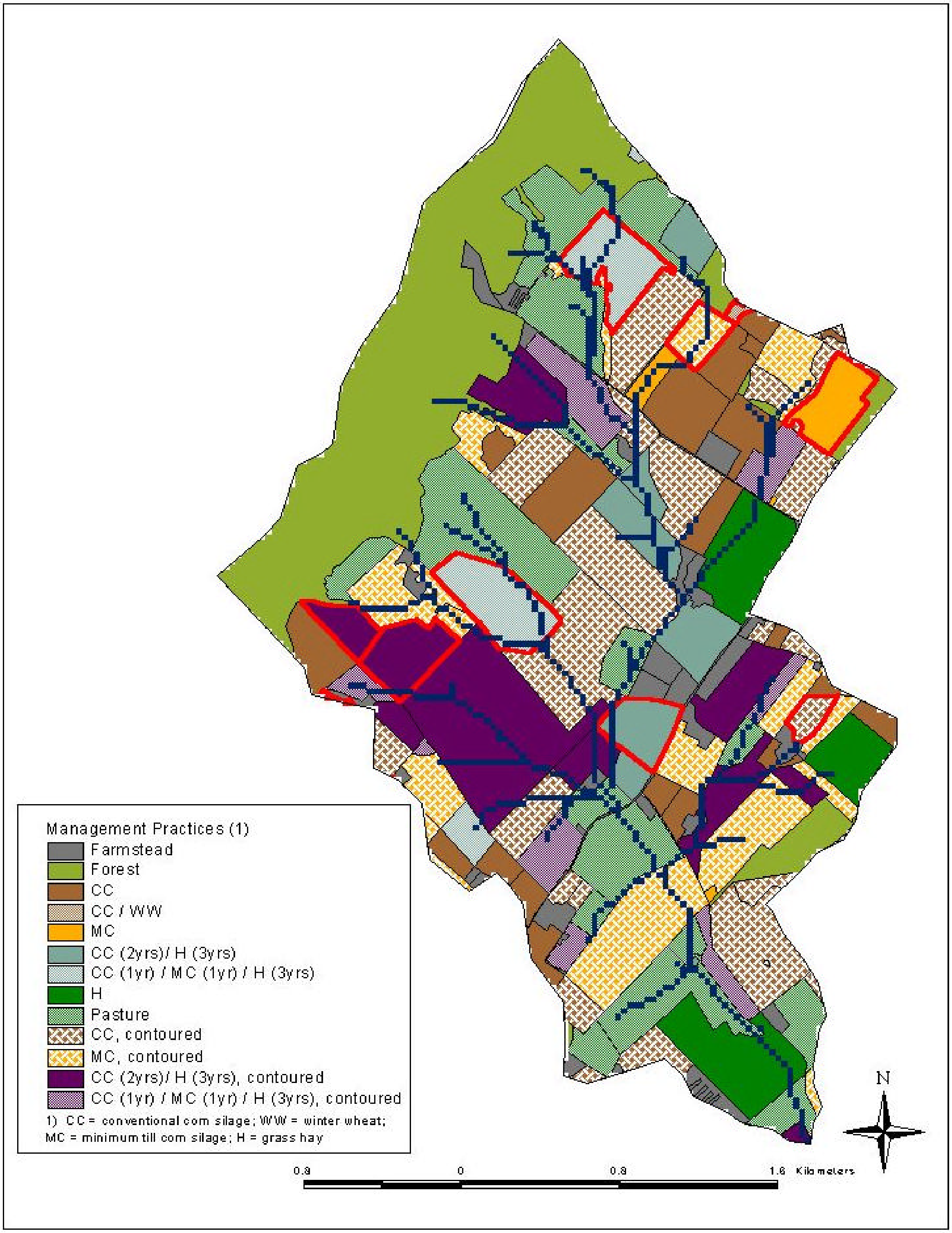

Figure 4.27: Agricultural land use allocation in Lola Run for highest scoring solution of Optimization Run 3 under a maximum acceptable pollutant load equal to the targeting strategy pollutant load; areas in different BMPs for this and an alternate solution of Run 3 (Figure 4.28) outlined in red

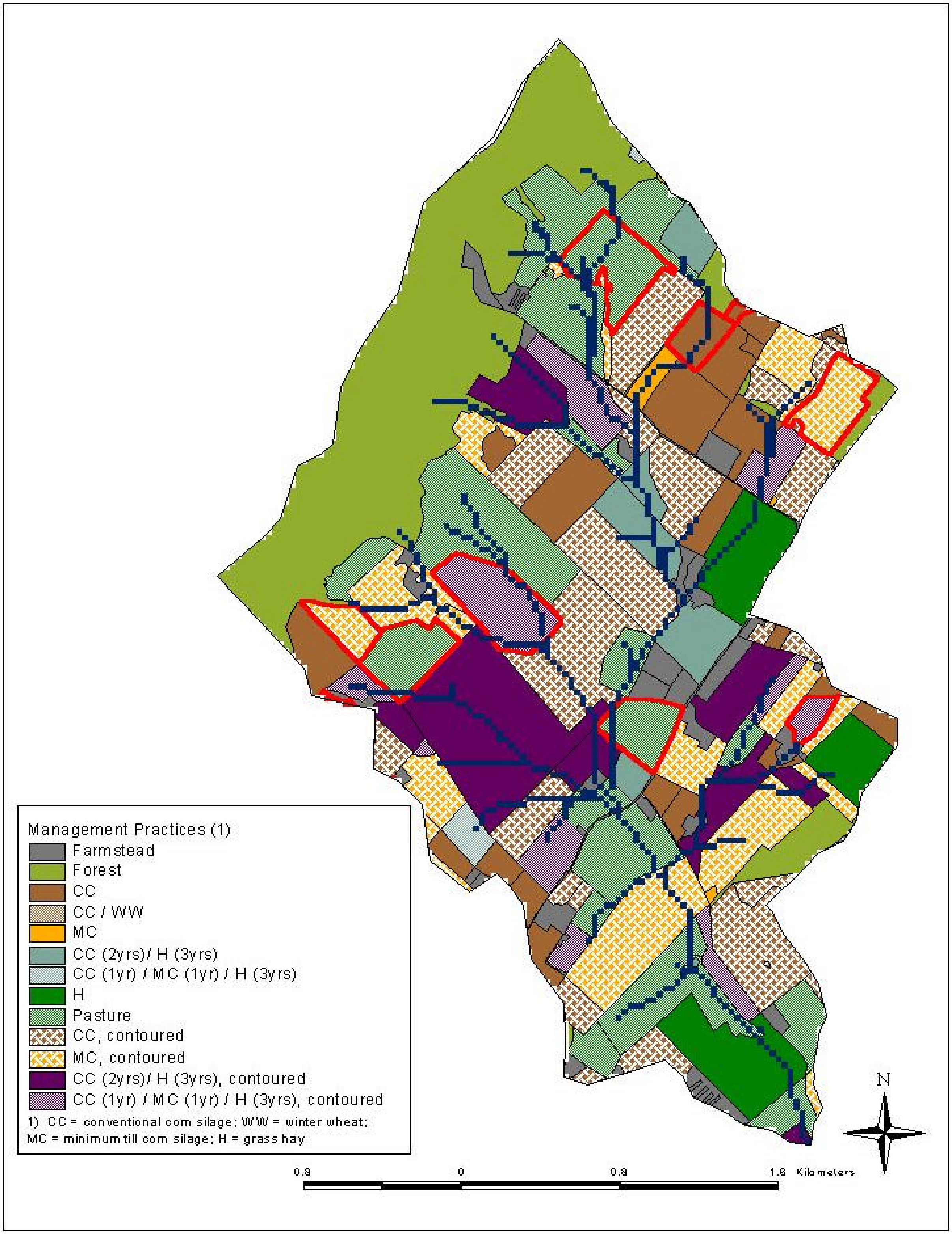

Figure 4.28: Agricultural land use allocation in Lola Run for an alternate solution of Optimization Run 3 under a maximum acceptable pollutant load equal to the targeting strategy pollutant load; areas in different BMPs for this and highest scoring solution of Run 3 (Figure 4.27) outlined in red

can enable addressing specific BMP requests by farmers. For example, a farmer might be willing only to change to a single BMP for all cropland fields whereas the most optimal solution strategies might have assigned different BMPs to those fields. Additionally, adjustments might be desired for field-specific considerations (e.g., a farmer not willing to change management practices) that were not known when data for the optimization procedure were prepared. After a customized scenario has been created, it can be run through the NPS and economic components to confirm that it still meets the target load criterion and to see how it compares with the other optimization and targeting strategy scenarios in terms of cost.

Some fields in the land use layers used in these evaluations are small or irregular in shape. A farmer might not be willing to place such areas in different management practices than the surrounding fields. However, this issue can be resolved before running the optimization procedure by increasing the amount of field and farm boundary information that is collected and incorporated into the land use layer. As a result, the land use layer will more accurately represent the field boundaries of the watershed and the optimization procedure solutions will be more directly implementable. Alternatively, constraints on the management practices allowed in each field could be imposed based on farmer preferences or field characteristics.

4.7 Overall cost-benefit analysis of optimization procedure

An overall cost-benefit analysis of the optimization procedure (Run 1) with respect to the targeting strategy was performed to address the second research objective and determine the feasibility of using the optimization procedure. The hypothesis for this analysis was:

- The benefit-to-cost ratio of the optimization procedure is greater than that of the targeting strategy.

To address this hypothesis, public costs associated with operation of the targeting and optimization methods were considered as well as costs and benefits resulting from solution implementation. Public operation costs included information costs, processing costs, and evaluation costs. Implementation costs (increase in cost from the baseline) are shown in Table 4.15.

The optimization procedure requires a computer with GIS and analysis software. For this cost analysis, the availability of a high-speed (> 1Ghz) computer with spreadsheet and word processor software (such as Microsoft Office) for analyzing and presenting the results is assumed. The ArcView GIS software, including the Spatial Analyst extension, costs $3700. Installing the GIS software takes a half-hour or less. Installing the optimization procedure program on a computer is an additional half-hour process involving copying the DLL file and loading the ArcView scripts. However, because costs associated with obtaining and installing software are one-time costs, they are not included in the overall cost-benefit analysis.

The results of the overall cost-benefit analysis are summarized in Table 4.17. The benefit-to-cost ratio for the optimization procedure was found to be greater than that of the targeting strategy for all but one optimization run considered in this research. These results were due to meeting the same reduction in pollutant loading as the targeting strategy while reducing total costs. An additional benefit of the optimization procedure was the potential flexibility in solutions to suit stakeholders while still meeting the target load criteria. Both methods are applicable to multiple watersheds.

Table 4.17: Summary of costs and benefits for the targeting strategy and optimization procedure in Lola Run

	Targeting strategy	Optimization procedure[1]
Costs		
Operation		
Information	$25	$1750
Processing	$0	$50
Evaluation	$50	$200
Variable costs for evaluation runs and parameter analysis		$500 - $1000
Total (using maximum of variable costs)	$75	$3000
Implementation		
Increase in cost from baseline	$134 892	$117 426
Total		
	$134 967	***$120 426***
Benefits		
Sediment reduction from baseline	2.81 Mg/ha	2.81 Mg/ha
Applicability to numerous watersheds	yes	yes
Flexibility in solutions	no	yes
Cost-to-benefit ratio		
sediment reduction / total cost	**2.08×10^{-5} Mg/ha/$**	**2.33×10^{-5} Mg/ha/$**

[1]Optimization procedure implementation costs and benefits are shown for Optimization Run 1 (Table 4.15).

4.7.1 Information costs

Information costs included data collection and preparation costs. Implementation of the targeting strategy required only data regarding whether or not a field meets the targeting criterion. For this purpose, a field boundary layer of cropland in the watershed is needed and may need to be determined from aerial photographs. In addition, in the example targeting strategy used in this research, knowledge about the major slope on each field was needed. Aerial photographs and USGS elevation or topographic maps for determining slope can be obtained for many areas through the Internet in an hour or less.

The optimization procedure requires, in digital format, the same field boundary layer and elevation map as for the targeting strategy. Field boundaries may need to be digitized from aerial photographs. Also, if necessary, the USGS topographic map can be digitized and converted into an elevation map. The time involved for digitization depends on watershed size and complexity. Digitization of field boundaries and topography in a watershed such as Lola Run is estimated to take about eight hours.

Information on farm type, size, boundaries, and area requirements as well as a characteristic slope length is needed. Assistance from an NRCS conservationist or an extension agent for the region may be required as well as discussion with farmers. Time to gather this information and digitize the farm boundaries is dependent on watershed size. This process is estimated to take about 16 hours for a watershed such as Lola Run.

Topographic maps, when used, may not contain sufficient detail and may require surveying throughout the watershed. However, the resulting additional costs would apply equally to both the targeting and optimization strategies and were not considered in this analysis.

Because the flow network of the watershed is determined from the elevation map, the optimization procedure also requires a digital data layer providing the watershed outline. This may be determined using the GIS. Using the ArcView hydrology extension, this may take an hour or less. In some cases the watershed may be on the edges of two adjacent map sheets, for one or more of the map types. Lining up and joining the map sheets for all the map types can add another two hours of data preparation work in the GIS.

Additionally, for the optimization procedure, a soils map or knowledge of soil erodibility (the USLE K-factor) across the watershed is required. Digital soil surveys, including both the GIS data layer and associated attribute tables, are available for most regions in the United States. If the digital soil survey is not available, then the data layer can be digitized from the hard copy of the soil survey. Work in this case can be minimized by digitizing areas based on soil erodibility, which is given in a table of the hard copy of the soil survey. Digitizing the map may add two hours (estimated for Lola Run) or more, depending on watershed size and complexity, to the data preparation time.

Next, the baseline scenario and desired BMPs must be determined. These can be based on current practices in the watershed and on practices typical for the region. However, depending on the project and the watershed, it may also be necessary limit BMPs to certain fields or regions. Appropriate C, P, and α factors for all management practices must be determined. These data are estimated to take about 24 hours to obtain, with assistance from an NRCS conservationist or an extension agent for the region.

For the economic portion of the optimization procedure, soil productivity from the soil survey tables is required. Also economic budgets for crop production and maintenance, selling prices, and land rental rates are needed. In Virginia this information can be obtained from Virginia Cooperative Extension (VCE, 1999). Once the economic information is obtained for a climatic region, it can be used for watersheds throughout that region, thus minimizing information costs for subsequent studies. Initial collection and preparation of the economic and soil productivity data into the tables used by the optimization procedure takes about 12 hours.

The final step in data preparation for the optimization procedure, using the data preparation scripts provided by the procedure, takes about an hour or less to complete including basic quality control checking of the prepared results.

Gathering and preparing a complete data set for the targeting strategy was estimated to take an NRCS conservationist or agricultural engineer, familiar with a GIS, about an hour. Data preparation for the optimization procedure was estimated to take the same person about 70 hours. A person with a PhD in agricultural engineering or a related field or with a MS and specialized experience could be hired for this work by the United States Government at the GS-11 pay level, corresponding to an hourly wage of about $25/hour (US-OPM, 2002). Thus, estimating labor at $25/hr, information costs for targeting were calculated at $25 while information costs for the optimization procedure were calculated at $1750.

4.7.2 Processing costs

Processing costs included costs of running the program. The targeting strategy can be applied simply by assessing whether or not each field meets the targeting rules. It does not necessarily require use of a computer. Total processing costs for the targeting strategy are negligible.

Running the optimization procedure involves 15 minutes to set the GA parameters and start the program, after the data preparation scripts have been run. The program then runs unassisted. For lengthy runs in which the ArcView program must be restarted (as discussed in Section 4.3), 15 minutes is required for each restart. Additionally, examining the data before restarting ArcView to determine if an appropriate level of convergence has been achieved is estimated to take an hour or less.

For an NRCS conservationist or agricultural engineer, familiar with a GIS, estimated processing costs of the optimization procedure are $50 or less with labor billed at $25/hr. This assumes two starts of ArcView are needed to complete the run.

For the optimization procedure, determination of the most efficient GA parameters to use may involve costs and computer time to complete and evaluate multiple runs. If population size, replacement level, crossover, and mutation parameters all need to be determined, parameter analysis may result in about ten evaluation runs. However, once the data and scripts are set for a watershed, altering GA parameters and starting the program takes only about five minutes. Additional time is needed to evaluate the results. A variable cost of $500-1000 (20 to 40 hours) for determining GA parameters or performing additional runs for evaluation is included in the costs of the optimization procedure. Additional runs also increase the potential inconvenience of having a computer occupied by the optimization procedure over a long time period.

4.7.3 Evaluation costs

Evaluation costs include labor for analyzing the results and preparing a report. Results for the targeting strategy involve a single watershed scenario with explanation of BMPs used in that scenario. This was estimated to take two hours or less at $25/hr for a total of $50. The optimization procedure involves more analysis, using GIS to compare and present solutions from the final population. Additionally, a report might include suggestions of new scenarios that combine the scenarios of the final population to provide some flexibility of implementation. Evaluation of the optimization procedure results was estimated to take eight hours at $25/hr for a total of $200. When determining GA parameter sets, data evaluation and comparison are needed. These costs depend on the number of runs performed and are included in the variable processing costs discussed in the previous section.

4.7.4 Benefits

To illustrate the benefits of each method, results from Section 4.6 for Lola Run under the targeting strategy and Optimization Run 1 were compared. Both solutions met the target load of 0.64 Mg/ha, resulting in a sediment reduction of 2.81Mg/ha from the baseline (Table 4.15). Compared to the baseline, total implementation costs under the targeting strategy increased by $134 892. The lowest cost solution under the optimization procedure increased total watershed costs by $117 426. Including operation costs, the optimization procedure saved $14 500 as compared to the targeting strategy.

Both strategies are applicable to numerous watersheds and, in both cases, information costs are reduced for additional watersheds in the same region. The targeting strategy, however, is generally rigid in its targeting criterion with a single solution scenario. The optimization procedure allows flexibility in implementation by providing several solution scenarios.

4.7.5 Conclusions

The benefit-to-cost ratio of using the optimization procedure was greater than that of the targeting strategy. Operation costs for the optimization procedure were $2500 to $3000 more than for the targeting strategy. Thus, the optimization procedure is beneficial in any case where a solution is found that reduces watershed costs by at least $3000. With the exception of cost increase due to extensive digitizing, this $3000 reduction margin is true regardless of the watershed or target load criterion. The example of Lola Run easily meets this $3000 reduction margin, reducing implementation costs by about $17 500 and total (operation plus implementation) costs by $14 500. As a result, the overall cost-effectiveness, as measured by the benefit-to-cost ratio of sediment reduction to total cost, was greater for the optimization solution than the targeting strategy (Table 4.17). Similar results for the other Lola Run optimizations and for Mini-Muddy Creek optimization runs 1 and 2 are seen by looking at the sediment reduction and watershed cost increase columns of Table 4.15. Run 3 for Mini-Muddy Creek reduced implementation costs only $1661 as compared to the targeting strategy. However, as this watershed is smaller than Lola Run, operational costs dependent on watershed size may be lower.

4.8 Discussion

Evaluation of the optimization procedure components showed that the program code modeled the physical system as intended. Additionally, the modular nature of the procedure allows improvements or added detail to any of the components without affecting the overall functioning of the optimization procedure.

Evaluation of the optimization procedure as a whole showed that the procedure met the research objectives. The presented optimization procedure was designed to minimize both pollution reduction and cost in such a way that the resulting scenarios are useful to watershed management planners. Hence, pollution reduction was set as a satisfaction constraint after which cost increase was minimized. In contrast, Srivastava et al. (1999) presented two separate optimization functions: optimizing pollution reduction, with a penalty for decreased net returns, or optimizing net returns, with a penalty for increased pollution.

When a termination criterion of small change in watershed cost from the previous generation is used, as discussed in Section 4.3.2, the optimization heuristic potentially results in a set of near optimal solutions in the final generation. Two such solutions for Optimization Run 3 were shown in Section 4.6.2. The diversity of and within solutions will decrease as the optimization converges. However, when the optimization process converges to an identical final population, additional scenarios with similar cost-effectiveness levels can be located by looking at previous generations. In comparison the targeting strategy provides a single solution.

Alternative solution scenarios may be beneficial in giving watershed planners and farmers the flexibility of having several options from which to choose. Based on conditions in the watershed, the scenario most feasible for implementation may not have the highest fitness score. It may be that certain farmers are not willing to change practices or are willing to adopt only a selection of the BMPs considered. In the latter case, a solution scenario that does not have the highest fitness score may best meet the farmers' requests. Alternatively, a combination of cost-effective scenarios may provide a more acceptable (and cost-effective) solution to the majority of the watershed members.

The fitness scores of the optimization procedure were designed, in part, to accommodate a requirement by the GALib program that fitness scores remain positive. As a result, any baseline scenario may be used, which may aid in modeling current watershed practices. For example, the case studies in this research used a baseline scenario based on typical practices for the area.

Even in watersheds that show low pollutant loading levels, it may be desirable to determine if allocating management practices differently across the watershed could reduce costs while maintaining low pollutant levels. If farmers in the watershed are considering management practice changes, the optimization procedure could be used to identify alternative scenarios resulting in similar pollutant loading. This information may help ensure continued low levels of pollutant loading despite management or land use changes.

For example, the baseline used for Mini-Muddy Creek yields only 0.65 Mg/ha of sediment at the watershed outlet. Despite low levels of erosion, running and implementing results of the optimization procedure was predicted to be beneficial in potentially reducing both costs and sediment yield from the baseline. However, when baseline scenarios with low pollutant loading levels are used and further reduction of sediment yield is not required, the pollutant target load, p_t in Equation (3.1), may be set equal to the baseline pollutant load. It is then necessary to set p_b (Equation 3.1) to a pollutant loading value higher than the baseline loading so that the denominator of the equation does not go to zero. The optimization procedure can then locate scenarios that meet the target load criteria and minimize costs.

The optimization procedure provides a more comprehensive economic analysis than Srivastava et al. (1999). In particular, this procedure considers public cost and variations in soil productivity. In addition, this procedure incorporates acreage constraints and promotes more evenly spread costs across farms.

The presented optimization procedure follows the ideas of Braden et al. (1989) in using the USLE and a sediment delivery equation within an optimization model to determine pollution loading at the watershed outlet. However, use of a GIS automated determination of flow paths and enabled a finer resolution of spatial variability than was practical in the method of Braden et al. (1989). By using the USLE with a sediment transport function instead of a more detailed NPS model, the procedure runs within a reasonable timeframe and still finds improved solutions compared to the targeting strategy.

Currently, the optimization procedure includes specifications for contour plowing in addition to different tillage practices and rotations. Additionally, it allows pasture to be placed into alternate forms of forage (i.e., grass hay, alfalfa, or pasture). Additional BMPs or BMP sets can be added as long as their pollutant reduction impacts can be represented by the NPS component that is used.

Chapter 5: Summary and Conclusions

5.1 Summary

This research dealt with determining cost-effective placement of BMPs in an agricultural watershed. A BMP was defined as a management practice that contributed to pollution reduction at the watershed level. A computerized optimization procedure was developed to meet the objectives of this research. This procedure used a multi-objective GA to create and select watershed scenarios. The conflicting aspects of the two objectives were addressed with a lexicographical technique. Each scenario was first evaluated for whether or not it met a predetermined pollutant loading criterion and, secondly, for a decreased level of cost increase as compared to a baseline. Evaluation of meeting the pollutant target criterion was done through a NPS component created specifically for the optimization procedure. Evaluation of the scenario costs was performed by an economic component, also designed for this research.

5.2 Conclusions

The overall research goal, supported by two specific research objectives, was to increase BMP cost-effectiveness within a watershed, as compared to targeting recommendations. This goal was achieved, leading to the conclusion that:

- A functional optimization procedure was developed that locates scenarios for which BMP placement reduces the same amount of pollution and results in a lower cost increase from the baseline as compared to a targeting strategy.

The first specific objective was to optimize BMP placement based on cost and NPS pollution reduction for a watershed. Conclusions resulting from evaluation of the BMP placement procedure were:

- The optimization procedure located scenarios that consistently reduced costs while meeting the specified pollution reduction criterion;
- The solutions found by the optimization procedure for all evaluations resulted in increased cost-effectiveness with regard to scenario implementation costs; and
- The program code of the optimization procedure works as intended.

The second specific research objective was to determine if the benefit-to-cost ratio of the optimization procedure outweighed that of the targeting strategy. Conclusions from this research were as follows:

- Monetary benefits of the optimization procedure outweighed costs in all but one case considered;
- Pollution was reduced to the target load criterion set by the targeting strategy in all cases considered; and
- Because use of the GA provides multiple solutions that meet the objectives, there is some flexibility in selection of the most suitable solution based on the priorities of farmers and other stakeholders.

Additional general conclusions drawn from this research were as follows:

- Among a range of optimization heuristics, the genetic algorithm and simulated annealing heuristics have features most suited to this problem type; and
- Representing cost-effectiveness as a ratio in a single objective function does not define a clear response surface for this problem.

5.3 Future possibilities

Future possibilities for extending this research include customization of the procedure for specific questions or types of watersheds as well as more general extensions of research. A user interface to help format the data tables for the data preparation scripts would decrease preparation time. However, much of the data, once located and entered into a table, is applicable to numerous implementations. For example, crop and management practice information for one watershed will likely extend to most watersheds in that region, requiring limited modifications for additional watersheds.

Enhancement of the economic component might include more qualitative information that impacts the farmers' abilities and desires to adopt BMPs, such as openness to change and willingness to invest in new machinery. Accounting for farm boundaries that extend beyond the watershed boundary would increase realism of the component.

Further exploration and sensitivity analysis of GA parameters, such as population size, replacement type and level, selection scheme, and reproduction probabilities, would help provide parameter suggestions for this type of problem. Additionally, such work might provide a better understanding of the effects and interactions between parameters for this problem. For example, it may be that the relatively high mutation rate used to establish initial conditions favors smaller populations. Parameter evaluation studies might also consider watersheds dissimilar to those examined in this research. Study of additional watersheds would aid in parameter suggestions that might improve the heuristic's convergence efficiency.

Further work on the optimization component might include evaluation of the optimization procedure's performance on a watershed with a known response surface as well as comparison of the heuristic with the performance of another heuristic. This would help to validate the optimization procedure. Also, heuristic customization as a result of such evaluations could improve optimization efficiency and effectiveness for this type of problem.

Adding explicitly adaptive knowledge to the scenario representation and/or algorithm might improve its efficiency by leading to a reduced search space. For example, consideration of the impact of a BMP in connection with its up- and downstream neighbors might allow the heuristic to dynamically modify crossover procedures to explicitly preserve beneficial BMP groupings.

A key feature of the optimization procedure is its modular nature. Because of this aspect the procedure can be used with alternate NPS and economic components. In particular, a component estimating NPS nutrient or bacteria loading could replace or be added to the existing sediment loading component with minimum recoding. Additionally, more detailed components could be added as advances in computer technology result in reduced runtime.

In addition to replacing or adding scenario evaluation components, the objectives considered by the optimization component could be increased. For example, analysis possibilities in watershed planning could be enhanced by expanding the objectives to allow, after pollution criteria are met, for user selection and prioritization of remaining objectives.

While this research met its goal and objectives, it also widened the path towards future exploration into the use of optimization heuristics, particularly GAs, to evaluate and compare watershed responses to spatially distributed influences. The computational capabilities of computer technology continually increase the number and complexity of questions that can be asked and explored with regard to improving the cost-effectiveness of NPS pollution reduction at the watershed-level.

References

Akan, A. Osman. 1993. *Urban Stormwater Hydrology: A Guide to Engineering Calculations*. Lancaster, PA: Technomic Publishing Company, Inc.

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.

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.

Bouraoui, F. and T.A. Dillaha. 1996. ANSWERS-2000: Runoff and sediment transport model. *Journal of Environmental Engineering* 122(6): 493-502.

Braden, John B., Gary V. Johnson, Aziz Bouzaher, and David Miltz. 1989. Optimal spatial management of agricultural pollution. *American Journal of Agricultural Economics* 71(2): 404-413.

Buckles, Bill P. and Fredrick E. Petry. 1992. *Genetic Algorithms*. Los Alamitos, CA: IEEE Computer Society Press.

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.

Byne, Wes. 2000. Predicting sediment detachment and channel scour in the process-based planning model ANSWERS-2000. M.S. thesis. Virginia Tech, Blacksburg, VA.

Carpentier, C. L., D. J. Bosch, and S. S. Batie. 1998. Using spatial information to reduce costs of controlling agricultural nonpoint source pollution. *Agricultural and Resource Economics Review* 27(1): 72-84.

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

Coello, Carlos A. 2000. An updated survey of GA-based multiobjective optimization techniques. *ACM Computing Surveys* 32(2): 109-143.

Cooper, J. R., J. W. Gilliam, R. B. Daniels, and W. P. Robarge. 1987. Riparian areas as filters for agricultural sediment. *Soil Science Society of America Journal* 51(2): 416-420.

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.

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.

Das, Prasanta and Yacov Y. Haimes. 1979. Multiobjective optimization in water quality and land management. *Water Resources Research* 15(6): 1313-1322.

De Jong, K. A. 1975. An analysis of the behavior of a class of genetic adaptive systems. Ph.D. thesis, University of Mishigan, Ann Arbor. Cited in Mitchell (1999).

Degarmo, E. P., W. G. Sullivan, J. A. Bontadelli, and E. M. Wicks. 1997. *Engineering Economy*. New Jersey: Prentice Hall.

Dickinson, W. T., R. P. Rudra, and G. J. Wall. 1990. Targeting remedial measures to control nonpoint source pollution. *Water Resources Bulletin* 26(3): 499-507.

Dillaha, T. A. 1990. Role of Best Management Practices in restoring the health of the Chesapeake Bay. In: Perspectives on the Chesapeake Bay, 1990: Advances in Estuarine Sciences. CBP/TRS41/90. Chesapeake Bay Program USEPA, Washington, DC.

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.

Duan, Q. Y., S. Sorooshian, and V. K. Gupta. 1994. Optimal use of the SCE-UA global optimization method for calibrating watershed models. *Journal of Hydrology* 158(3-4): 265-284.

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

Foster, G. R. 1982. Modeling the erosion process. In *Hydrologic Modeling of Small Watersheds*, ed. C. T. Haan, H. P. Johnson, and D. L. Brakensiek, ch. 8, 297-382. St. Joseph, MI: American Society of Agricultural Engineers.

Garbrecht, Jurgen, and Lawrence W. Martz. 2000. Digitial elevation model issues in water resources modeling. In *Hydrologic and Hydraulic Modeling Support with Geographic Information Systems*, ed. David Maidment and Dean Djokic, ch. 1, 1-28. Redlands, CA: Environmental Systems Research Institute, Inc.

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.

Grefenstette, J. J. 1986. Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics* 16(1): 122-128. Cited in Mitchell (1999).

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.

Guy, B. T., W. T. Dickinson, and R. P. Rudra. 1992. Evaluation of fluvial sediment transport equations for overland flow. *Transactions of the ASAE* 35(2): 545-555.

Haan, C. T., B. J. Barfield, and J. C. Hayes. 1994. *Design Hydrology and Sedimentology for Small Catchments*. San Diego, CA: Academic Press.

Heatwole, C. D. 1999. Muddy Creek Farm Generation Procedure. http://www.isis.vt.edu/watershed/MC/FarmGenerationProcedure.doc Accessed 15 January 2002.

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.

Heatwole, C. D., A. B. Bottcher, and L. B. Baldwin. 1987a. Modeling cost-effectiveness of agricultural nonpoint pollution abatement programs on two Florida basins. *Water Resources Bulletin* 23(1): 127-131.

Heatwole, C. D., K. L. Campbell, and A. B. Bottcher. 1987b. Modified CREAMS hydrology model for Coastal Plain flatwoods. *Transactions of the ASAE* 30(4): 1014-1022.

Heatwole, C. D., P. L. Diebel, and J. M. Halstead. 1990. Management and policy effects on potential groundwater contamination from dairy waste. *Water Resources Bulletin* 26(1): 25-34.

Hersch, Valerie. 1993. The price of clean water. *Florida Trend* 35(9): 30-35.

Hrissanthou, V. 1990. Application of a sediment routing model to a middle European watershed. *Water Resources Bulletin*. 26(5): 801-810.

Huang, Wen-yuan, David Shank, and Tracy Hewitt. 1997. On-farm costs of timing nitrogen applications to reduce nitrogen losses. *Agricultural and Resource Economics Review* 26(2): 262.

Huber, Bill. 2000. [ArcView-L] SUM: error msg during for each loop. ESRI Online Support Center. http://support.esri.com/search/ViewListPost.asp?dbid=4134. Accessed 15 January 2002.

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.

Johnson, C. W., G. A. Schumaker, and J. P. Smith. 1980. Effects of grazing and sagebrush control on potential erosion. *Journal of Range Management* 33(6): 451-454.

Julien, P. Y. and D. B. Simons. 1985. Sediment transport capacity of overland flow. *Transactions of the ASAE* 28 (3): 755-762.

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.

Kothyari, Umesh C. and Sanjay K. Jain. 1997. Sediment yield estimation using GIS. *Hydrological Sciences-Journal-des Sciences Hydrologiques* 42(6): 833-843.

Koza, J. R.1992. *Genetic programming: on the programming of computers by means of natural selection*. Cambridge, MA: The MIT Press.

Lawler, E. L. 1976. *Combinatorial Optimization: Networks and Matroids*. New York, NY: Holt, Rinehart, and Winston.

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.

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

Maas, R. P., S. L. Birchford, M. P. Smolen, and J. Spooner. 1988. Agricultural nonpoint source control: experiences from the Rural Clean Water Program. *Journal of Lake and Reservoir Management* 4(1): 51-56. Cited in Dillaha (1990).

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

Mulligan, Ann E. and Linfield C. Brown. 1998. Algorithms for calibrating water quality models. *Journal of Environmental Engineering* 124(3): 202-211.

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

Nicks, A. D., L. J. Lane and G. A. Gander. 1995. Technical Documentation USDA Water Erosion Prediction Project (WEPP), Ch. 2 Weather Generator. West Lafayette, Indiana.

Novotny, Vladimir and Harvey Olem. 1994. Water Quality: Prevention, Identification, and Management of Diffuse Pollution. New York: Van Nostrand Reinhold.

Prato, Tony and Shunxiang Wu. 1991. Erosion, sediment, and economic effects of conservation compliance in an agricultural watershed. *Journal of Soil and Water Conservation* 46(3): 211-214.

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

Rentmeesters, Mark J., Wei K. Tsai, and Kwei-Jay Lin. 1996. A theory of lexicographic multi-criteria optimization. In *Proc. Second IEEE International Conference on Engineering of Complex Computer Systems*, 76-79. Piscataway, NJ: IEEE.

Roumasset, James A. 1976. *Rice and Risk.* Amsterdam: North-Holland Publishing Company, Inc.

Royer, Jack P. 1987. Determinants of reforestation behavior among southern landowners. *Forest Science* 33(3): 654-667.

Schaffer, J. D., Caruana, R. A., Eshelman, L. J., and R. Das. 1989. A study of control parameters affecting online performance of genetic algorithms for funcion optimization. In J. D. Schaffer, ed., *Proc. Third International Conference on Genetic Algorithms*. Morgan Kaufmann. Cited in Mitchell (1999).

Schwab, Glenn O., Delmar D. Fangmeier, William J. Elliot, and Richard K. Frevert. 1993. *Soil and Water Conservation Engineering*. New York, NY: John Wiley & Sons, Inc.

Sheridan, J. M., C. V. Booram, Jr., and L. E. Asmussen. 1982. Sediment-delivery ratios for a small coastal plain agricultural watershed. *Transactions of the ASAE* 25(3): 610-615, 622.

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.

Spurlock, Stan R. and Ivery D. Clifton. 1982. Efficiency and equity aspects of nonpoint source pollution controls. *Southern Journal of Agricultural Economics* 14(2): 123-129.

Srivastava, P. 1999. A GIS, genetic algorithm, and AnnAGNPS integrated system for non-point pollution assessment. Ph.D. dissertation, The Pennsylvania State University, PA.

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.

Stone, N.D. 1995. Whole-farm planning for crop/livestock farms: integrating nutrient management. In *FACTS 95 Conference Proceedings*, 255-261. Cornell University, Ithaca, NY.

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.

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.

Theurer, Fred D., Ron Bingner, and Roger Cronshey. 2001. AnnAGNPS Pollutant Loading Model. Ver 2.0. Oxford, MS: USDA-NRCS and USDA-ARS National Sedimentation Lab. http://www.sedlab.olemiss.edu/agnps.html. Accessed 12 July 2001.

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.

US-OPM (United States Office of Personnel Management). 2002. USDA Salary Table 2002-RUS. http://www.opm.gov/oca/02tables/RUS_h.htm. Accessed April 19.

USDA-ARS (Agricultural Research Service). 2002. WEPP Validation Data Sets. http://topsoil.nserl.purdue.edu/nserlweb/weppmain/ahtml/readme1.html. Accessed March 04.

USDA-NRCS (Natural Resources Conservation Service). 1988. Crop Management Factors - Mountains and Valleys - Area II. NRCS Field Office Technical Guide 120-411. Washington, DC: USDA.

USDA-NRCS (Natural Resources Conservation Service). 2001. SSURGO Documentation. http://gis.itc.nrcs.usda.gov/docs/SSURGO_Documentation.html. Accessed January 15.

USDA-NRCS (Natural Resources Conservation Service). 2002. Conservation Technical Assistance. http://www.nrcs.usda.gov/programs/cta/. Accessed April 19.

VA-DCR (Virginia Department of Conservation and Recreation). 2002. Soil & Water Conservation: Best Management Practices for Virginia Agriculture. http://www.dcr.state.va.us/sw/costshar.htm. Accessed April 19.

VASS (Virginia Agricultural Statistics Service). 2000. Virginia Agricultural Statistics Bulletin & Resource Directory, Number 75. Richmond, VA.

VCE (Virginia Cooperative Extension). 1999. Virginia farm management crop and livestock enterprise budgets. Publication 446-047. Virginia Tech. Blacksburg, VA.

VCE. 2001. Land Rental Guide for the Shenandoah Valley. http://www.ext.vt.edu/news/periodicals/fmu/2001-02/landrental.html. Accessed October 31.

VA-DCR (Virginia Department of Conservation and Recreation). 1993. Nutrient Management Handbook. Richmond, VA.

Walling, D. E. 1983. The sediment delivery problem. *Journal of Hydrology* 65(1-3): 209-237.

White, David C. 1988. The economics of sediment management using an aggregate model of a large agricultural watershed. Ph.D. dissertation, University of Illinois, IL.

White, G. B. and E. J. Partenheimer. 1980. Economic impacts of erosion and sedimentation control plans: Case studies of Pennsylvania dairy farms. *Journal of Soil and Water Conservation* 35(2): 76-78.

Willis, L. M., S. B. Forrest, J. A. Nissen, J. G. Hiscock, and P. V. Kirby. 1994. Analysis of on-farm best management practices in the Everglades agricultural area. In *Environmentally Sound Agriculture: Proc. Second Conference*, 93-99. Orlando, FL, 20-22 April.

Yitayew, M., S. J. Pokrzywka, and K. G. Renard. 1999. Using GIS for facilitating erosion estimation. *Applied Engineering in Agriculture* 15(4): 295-301.

Appendix A: Alternative net return calculations

This appendix gives example calculations for two alternative methods for calculating annual net return. These calculations are for a multi-year rotation on a 1-ha field of single soil type (mapping unit symbol 1B: Allegheny fine sandy loam).

Table A.1: Annual net return calculated by determining the net return for each year, discounting all rotation years to a present value, and annualizing

Net return calculation

year	crop	selling price [$/tons]	yield [tons/ha] for a 1-ha area	gross return	production cost [$/ha]	net return	bring to end-of-year 1	
				price * yield		gross return-production cost	discount factor[1]	discount * net return
1	conventional-tillage corn silage	26.1	44.46	1 160.41	744.33	416.076	-- 1	416.08
2	conventional-tillage corn silage	26.1	44.46	1 160.41	744.33	416.076	(P/F,9%,1) = 0.9174	381.71
3	grass hay	46	9.26	425.96	527.86	-101.9	(P/F,9%,2) = 0.8417	-85.77
4	grass hay	46	9.26	425.96	359.14	66.82	(P/F,9%,3) = 0.7722	51.60
5	grass hay	46	9.26	425.96	359.14	66.82	(P/F,9%,4) = 0.7084	47.34
Present value total for all crops in rotation [$]:								**810.95**

Average annual net return

=Present value total * 5-year annualization factor[2] (A/F,9%,5) = 810.95 * 0.1671 **= $135.51/ha**

[1](P/F,9%,n) is the single payment, present worth factor for discounting a value *n* years in the future to a present value, using a 9% interest rate. (Degarmo et al., 1997; Table C.12)
[2](A/F,9%,n) is the uniform series, sinking fund factor for distributing a future value evenly over *n* years, using a 9% interest rate. (Degarmo et al., 1997; Table C.12)

Table A.2: Annual net return calculated as the difference between annualized gross return and annualized production cost

Production cost calculation

year	crop	production cost [$/ha]	bring to end-of-year 1 discount factor[1]		discount * production cost	5-year annualized production cost [$/tons] discounted cost * annualization factor[2] ((A/F,9%,5) =0.1671)
1	conventional-tillage corn silage	744.33	--	1	744.33	124.38
2	conventional-tillage corn silage	744.33	(P/F,9%,1) =	0.9174	682.85	114.10
3	grass hay establishment and harvest	527.86	(P/F,9%,2) =	0.8417	444.30	74.24
4	grass hay maintenance	359.14	(P/F,9%,3) =	0.7722	277.33	46.34
5	grass hay maintenance	359.14	(P/F,9%,4) =	0.7084	254.41	42.51

Gross return calculation

year	crop	selling price [$/tons]	yield [tons/ha] for a 1-ha area	gross return price * yield	bring to end-of-year 1 discount factor[1]		discount * gross return	5-year annualized gross return [$/tons] discounted price * annualization factor[2] ((A/F,9%,5) =0.1671)
1	conventional-tillage corn silage	26.1	44.46	1 160.41	--	1	1 160.41	193.90
2	conventional-tillage corn silage	26.1	44.46	1 160.41	(P/F,9%,1) =	0.9174	1 064.56	177.89
3	grass hay	46	9.26	425.96	(P/F,9%,2) =	0.8417	358.53	59.91
4	grass hay	46	9.26	425.96	(P/F,9%,3) =	0.7722	328.93	54.96
5	grass hay	46	9.26	425.96	(P/F,9%,4) =	0.7084	301.75	50.42

Annual net return calculation

year	annualized gross return [$/tons]	annualized production cost [$/tons]	annualized gross return - annualized production cost
1	193.90	124.38	69.53
2	177.89	114.10	63.78
3	59.91	74.24	-14.33
4	54.96	46.34	8.62
5	50.42	42.51	7.91
Average annual net return			**$135.51/ha**

[1](P/F,9%,n) is the single payment, present worth factor for discounting a value *n* years in the future to a present value, using a 9% interest rate. (Degarmo et al., 1997; Table C.12)
[2](A/F,9%,n) is the uniform series, sinking fund factor for distributing a future value evenly over *n* years, using a 9% interest rate. (Degarmo et al., 1997; Table C.12)

Appendix B: Program code

Appendix B contains the scripts and DLL used in the optimization procedure. Figure B.1 shows interactions among the various pieces of code. Three scripts (*Prepdat.ave*, *Prepdat1.ave*, and *Prepdat2.ave*) were used to prepare the data for the procedure. The *Baseline.ave* script runs the baseline scenario through the components of the procedure and stores the results. Once the data are prepared, *CallDLL.ave* calls the DLL portion of the procedure. For each scenario, the DLL calls the *Main.ave* script, which calls the *Econ.ave* and *NPS-sed.ave* scripts to calculate cost and pollutant loading and uses this information to determine the economic and pollutant fitness scores. Control is returned to the DLL, which determines the total fitness score and sends the next scenario to *Main.ave* for evaluation. Additional scripts (*Calcarea.ave*, *Intersect.ave*, and *MakeGtheme.ave*) are called by the other scripts as needed; these scripts perform general functions and were separated from the function specific scripts to increase the object-oriented nature of the code.

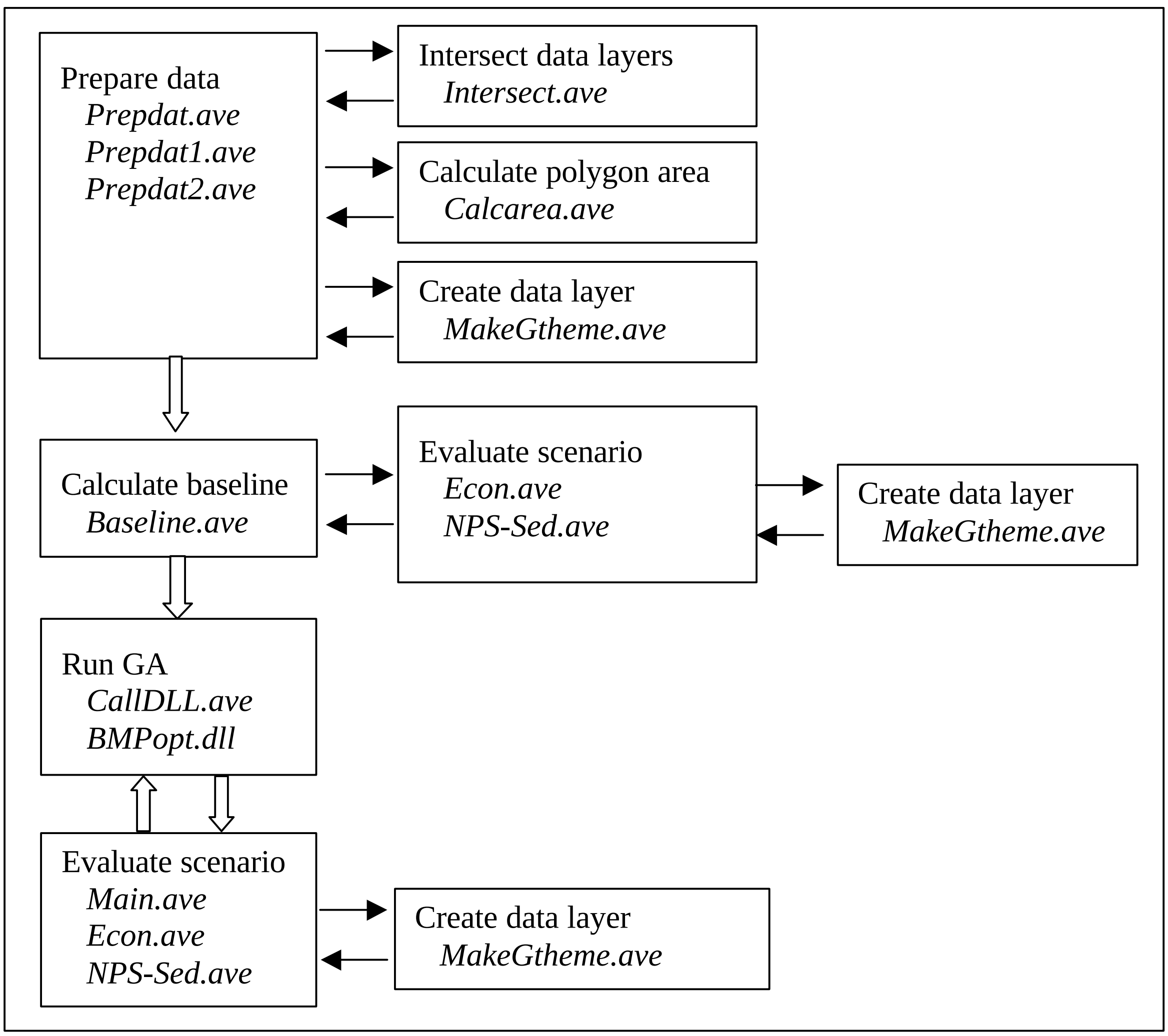

Figure B.1: Interactions among ArcView scripts and DLL used to code the optimization procedure

B.1 Prepdat.ave

```
'Prepares data for the NPS component.
'calculates length-across-cell and slope grids
'calculates S and L factors and combines with R and K factors
'R-factor of 2800; 'characteristic slope length of 45m

theView = av.GetActiveDoc
'**input data
 'DEM
  theDEMGrid = theView.FindTheme("Fill1").GetGrid
 'watershed grid
  wshedGrid = theView.FindTheme("Wshed_mini").GetGrid
 'R factor in SI units
  constRfactor = 2800
 'K factor in  converted to SI units (conversion = 0.1317)
  KgridSI = (theView.FindTheme("Kfactus_mini").GetGrid)*(0.1317)
  'save data set
   KgridSI.SaveDataSet("Kfact-si".asFileName)
  'create theme and add to view
   theView.AddTheme(av.run("MakeGtheme",{KgridSI,"Kfact-si"}))

'**calc slope grids
 'Decimal slope grid
  slopeDecimalGrid = (theDEMGrid.Slope (1, TRUE))/100
   'slopeDegreeGrid = theView.FindTheme("Slope in m/m").GetGrid
  'save data set
   slopeDecimalGrid.SaveDataSet("slopedec".asFileName)
  'create theme and add to view
   theView.AddTheme(av.run("MakeGtheme",{slopeDecimalGrid,"Slope in m/m"}))

 'degree slope grid
  slopeDegreeGrid = theDEMGrid.Slope (1, FALSE)
```

```
 'convert to radians
   slopeRadGrid = slopeDegreeGrid*(Number.getPi.asGrid)/(180.asGrid)

'**create flow-length-across-cell grid
 'calc flow direction
   flowDirGrid = theDEMGrid.FlowDirection (FALSE)
   'flowDirGrid = theView.FindTheme("Flow Direction").GetGrid
    'save data set
      flowDirGrid.SaveDataSet("Flow Direction".asFileName)
     'create theme and add to view
      theView.AddTheme(av.run("MakeGtheme",{flowDirGrid,"Flow Direction"}))
  'make flow accumulation for later
   flwaccGrid = flowDirGrid.FlowAccumulation(NIL)
   'save data set
     flwaccGrid.SaveDataSet("Flow Accumulation".asFileName)
   'create theme and add to view
     theView.AddTheme(av.run("MakeGtheme",{flwaccGrid,"Flow Accumulation"}))

 'locate cells with diagonal flow directions
   tempGrid1=( flowDirGrid = 2.AsGrid) or (flowDirGrid = 8.AsGrid) or (flowDirGrid =
32.AsGrid) or (flowDirGrid = 128.AsGrid)
 'assign diagonal flow cells a length of cell width times SQRT(2)
   tempGrid2=( tempGrid1  * 30.AsGrid * 2.Sqrt.AsGrid)
 'locate cells with non-diagonal flow directions
   tempGrid1=( flowDirGrid= 1.AsGrid) or (flowDirGrid = 4.AsGrid) or (flowDirGrid =
16.AsGrid) or (flowDirGrid = 64.AsGrid)
 'assign non-diagonal flow cells a length of cell width
   lengthGrid=tempGrid2+(tempGrid1  * 30.AsGrid)

 'save data set
   lengthGrid.SaveDataSet("lngtcell".asFileName)
 'create theme and add to view
   theView.AddTheme(av.run("MakeGtheme",{lengthGrid,"Length Across Cell"}))
```

```
'**calc L factor for USLE
  'create beta grid
    betaNumGrid = (slopeRadGrid.Sin)/((0.0896).AsGrid)
    betaDenomGrid = ((3.AsGrid)*((slopeRadGrid.Sin).Pow(0.8)))+((0.56).AsGrid)
    betaGrid = betaNumGrid/betaDenomGrid

  'create m grid
    mGrid =   betaGrid/(1.AsGrid+betaGrid)

  'calculate L factor (slope length of 45m)
    lGrid = ((45.AsGrid)/(22.AsGrid)).Pow(mGrid)

  'save data set
    lGrid.SaveDataSet("Lfact".asFileName)
  'create theme and add to view
    theView.AddTheme(av.run("MakeGtheme",{lGrid,"Lfact"}))

'**calc S factor for USLE
  'locate cells with slope < 9%
    flagShortGrid=(slopeDegreeGrid < ((5.1428).AsGrid))
  'assign S-factor for short slopes
    tempGrid1=( (((10.8).AsGrid)*(slopeRadGrid.Sin)) + ((0.03).AsGrid))*flagShortGrid
  'locate cells with slope >= 9%
    flagLongGrid=(slopeDegreeGrid >= ((5.1428).AsGrid))
  'assign S-factor for long slopes
    SGrid=tempGrid1+(((((16.8).AsGrid)*(slopeRadGrid.Sin)) -
((0.50).AsGrid))*flagLongGrid)

  'save data set
    SGrid.SaveDataSet("Sfact".asFileName)
  'create theme and add to view
    theView.AddTheme(av.run("MakeGtheme",{SGrid,"Sfact"}))
```

```
'**calc RKSL grid in SI units
  'create grid
    theRKSLgrid = wshedGrid*(constRfactor.AsGrid)*KgridSI*lGrid*SGrid

  'save data set
    theRKSLgrid.SaveDataSet("RKSL-si".asFileName)
  'create theme and add to view
    theView.AddTheme(av.run("MakeGtheme",{theRKSLgrid,"RKSL-si"}))
```

B.2 Prepdat1.ave

```
'Prepares first part of econ data for use in opt program.

'*** hand join SoilYld table to Soils dataset first***
theView = av.GetProject.FindDoc("View1")
'***Identify data sets ***********************
  theFLDtheme = theView.FindTheme("Fields_mini.shp")
  theFLDftab = theFLDtheme.GetFTab
  theSoilstheme = theView.FindTheme("Soil_srgkt_mini.shp")
  theSoilsftab = theSoilstheme.GetFTab
  theMPvtab = av.GetProject.FindDoc("MP.txt").GetVTab

'  theSoilYldvtab = av.GetProject.FindDoc("SoilYld.txt").GetVTab
  theCropsvtab = av.GetProject.FindDoc("Crops.txt").GetVTab
'  theConstvtab = av.GetProject.FindDoc("Constraints.dbf").GetVTab

'***Locate fields **********************
  theMPMPIDfield = theMPvtab.FindField("Mpid")
  thePCfield = theMPvtab.FindField("Prodcost")
  cropsFld = theCropsvtab.FindField("Crop")
  unitPriceFld= theCropsvtab.FindField("Unitprice")
  theCropMPfield = theCropsvtab.FindField("Mpid")

'*** Add new fields to FLDftab
  '**Make theFLDftab editable.
    'If you can't edit the theme inform the user.
      if (theFLDftab.CanEdit.Not) then
        MsgBox.Info("Cannot edit table for theme:"++theFLDftab.AsString,"")
      end
    theFLDftab.SetEditable(TRUE)
  '**make and add fields
    'Add two fields for sorting areas into categories
```

```
    'Add one field for storing working MPID
    'Add one field for storing flag for & calculating public cost calculations
    'Add one field for calculating private cost in Econ component
    theFLDftab.AddFields({Field.Make("areaC",#FIELD_DOUBLE,16,3),
                          Field.Make("areaH",#FIELD_DOUBLE,16,3),
                          Field.Make("MPID",#FIELD_LONG,8,0),
                          Field.Make("PubFlag",#FIELD_LONG,16,2),
                          Field.Make("PrivC",#FIELD_DOUBLE,16,2)})

    '**Add profit max and opp cost fields
     'make new field for profit max MPID
       profMaxMPField = Field.Make("ProfMaxMP",#FIELD_LONG,8,0)
     'make new field for opp cost of each FLD [$]
       oppCostField = Field.Make("oppCost",#FIELD_DOUBLE,16,2)
     'add fields to table
       theFLDAreaVTab.AddFields({profMaxMPField,oppCostField})

  '**Make theFLDftab not editable.
    theFLDftab.SetEditable(FALSE)

'***join MP table (C,P,alpha,type,prod cost) to FLD dataset****
  theJoinTofield = theFLDftab.FindField("Mpid")
  theFLDftab.Join (theJoinTofield,theMPvtab,theMPMPIDfield)
'
'***join SoilYld table (prod of each crop by soil [qty/ha]) to Soils dataset****
'**something wierd about this join. use with caution. join by hand seems fine. **
'  theJoinFromfield = theSoilYldvtab.FindField("Musym")
'  theJoinTofield = theSoilsftab.FindField("Musym")
'  theSoilsftab.Join (theJoinTofield,theMPvtab,theJoinFromfield)

'**Refresh joins
  theFLDftab.Refresh
  theSoilsftab.Refresh
```

```
'***Intersect Soils dataset with FLD dataset****
  '**caution - intersect may fail if joins are performed multiple times.
  theSoilbyFLDfname =
av.run("intersec",{theView,theFLDtheme,theSoilstheme,"SoilbyFld.shp".AsFileName})

'***Calculate area of soils within FLDs ****
   av.run("CalcArea",{theView,theSoilbyFLDfname})

'***Calc crop selling prices by area of soils in FLDs ****
  theSoilbyFLDtheme = theView.FindTheme("SoilbyFld.shp")
  theSbyFftab = theSoilbyFLDtheme.GetFtab
  '**locate new fields
    farmtypeFld = theSbyFftab.FindField("F_typeS")
    areaFld = theSbyFftab.FindField("Area")

  '**Make theSbyFftab editable.
    'If you can't edit the theme inform the user.
      if (theSbyFftab.CanEdit.Not) then
        MsgBox.Info("Cannot edit table for theme:"++theSbyFftab.AsString,"")
      end
    theSbyFftab.SetEditable(TRUE)

  '**Initialize lists for storing SellingPrice and Enum fields - for summarizing
    theMPSellPriceList = List.Make
    theEnumList = List.Make

  '**Calculate selling price by MP
   'look through MP table
    for each i in theMPvtab
     theMPMPIDvalue = theMPvtab.ReturnValue(theMPMPIDfield,i)
     'make new field for selling price of the MP by soil in field [$]
       newSellPriceField =
Field.Make("SP"+theMPMPIDvalue.AsString,#FIELD_DOUBLE,16,2)
```

```
    'add field to table
      theSbyFftab.AddFields({newSellPriceField})
    'look through crops table for crops in this MP
      for each j in theCropsvtab
       'find crop record in Crops table to match MP
        if (theMPMPIDvalue=theCropsvtab.ReturnValue(theCropMPfield, j)) then
         'get UnitPrice [$/qty] from Crops table
          unitPrice = theCropsvtab.ReturnValue(unitPriceFld, j)
          'find associated crop in theSbyFftab (from soilyld join)
          cropName = theSbyFftab.FindField(theCropsvtab.ReturnValue(cropsFld, j))
          for each k in theSbyFftab
           'get Area [ha] from SoilbyFLD table
             area = theSbyFftab.ReturnValue(areaFLD, k)
           'get cropYld [qty/ha] for this soil in SoilbyFLD table
             soilYld = theSbyFftab.ReturnValue(cropName, k)
           'get current selling price
             currentSP = theSbyFftab.ReturnValue(newSellPriceField,k)
           'add for this crop
             '[qty/ha]*[$/qty]*ha = selling price by soil in field [$]
             theSbyFftab.SetValue(newSellPriceField,k,currentSP+(soilYld*unitPrice*area))
          end 'for k in theSbyFftab
        end 'if crop match
      end 'for each j in theCropsvtab
    'add field to list for summarizing
      theMPSellPriceList.Add(newSellPriceField)
      theEnumList.Add(#VTAB_SUMMARY_SUM)
   end 'for each i in theMPvtab
'MsgBox.List(theCropSellPriceList,"","")
'MsgBox.ListAsString(theEnumList,"","")
  '**Make theSbyFftab not editable.
   theSbyFftab.SetEditable(FALSE)
```

```
'***Summarize selling prices and area of soils by FLDs ****
  'add areaFLD onto lists to Sum
    theMPSellPriceList.Add(areaFld)
    theEnumList.Add(#VTAB_SUMMARY_SUM)
  'calc Sum
   theFLDAreaVTab = theSbyFftab.Summarize(("FLDArea").AsFileName, dBASE,
theSbyFftab.FindField("Field_ID")
                        ,theMPSellPriceList,theEnumList)

'***Calc MP net return ****
'  theFLDAreaVTab = av.GetProject.FindDoc("fldarea.dbf").GetVTab

  '**locate new fields
    areaFLDfield = theFLDAreaVTab.FindField("Sum_area")

  '**Make theFLDAreaVTab editable.
    'If you can't edit the theme inform the user.
      if (theFLDAreaVTab.CanEdit.Not) then
        MsgBox.Info("Cannot edit table for theme:"++theFLDAreaVTab.AsString,"")
      end
    theFLDAreaVTab.SetEditable(TRUE)

  '**calculate production cost
    'look through MP table
     for each j in theMPvtab
      'make new field for each MPID
        MPretfield = Field.Make("MPret"+j.AsString,#FIELD_DOUBLE,16,2)
      'add field to table
        theFLDAreaVTab.AddFields({MPretfield})
      'get Production cost [$/ha] from MP table
        PCvalue = theMPvtab.ReturnValueNumber(thePCfield, j)
        for each k in theFLDAreaVTab
         'calc MP return as -prodcost[$/ha]*area[ha]
```

```
        MPretvalue = -PCvalue*(theFLDAreaVTab.ReturnValue(areaFLDfield, k))
        'get selling price calculated and summed above
        SPvalue =
theFLDAreaVTab.ReturnValue(theFLDAreaVTab.FindField("Sum_sp"+j.AsString), k)
        theFLDAreaVTab.SetValue(MPretfield, k, MPretvalue+SPvalue)
        end 'each k in theFLDAreaVTab
    end 'for j record in theMPvtab

  '**Make theFLDAreaVTab not editable.
   theFLDAreaVTab.SetEditable(FALSE)

 '***join Fldarea table to FLD dataset****
   theJoinFromfield = theFLDAreaVTab.FindField("Field_id")
   theJoinTofield = theFLDftab.FindField("Field_id")
   theFLDftab.Join (theJoinTofield,theFLDAreaVTab,theJoinFromfield)

 '***Summarize areas by farm, for constraints ****
   theFarmAreaVTab = theSbyFftab.Summarize(("FarmArea").AsFileName, dBASE,
theSbyFftab.FindField("Farm_ID")
                   ,{farmtypeFld, areaFld}
                   ,{#VTAB_SUMMARY_FIRST, #VTAB_SUMMARY_SUM})
```

B.3 Prepdat2.ave

```
'Second part of prepare script
'Split script because join is not working for constraint table

'*** hand join Constraint table to FarmAreatables first***
theView = av.GetProject.FindDoc("View1")
theFarmAreaVTab = av.GetProject.FindDoc("FarmArea.dbf").GetVTab

' '***join Constraint table (ha and % constraints) to FarmArea dataset****
'  theConstvtab = av.GetProject.FindDoc("Constraints.dbf").GetVTab
'  theJoinFromfield = theConstvtab.FindField("F_types")
'  theJoinTofield = theFarmAreaVTab.FindField("First_f_ty")
'  test  = theFarmAreaVTab.Join (theJoinTofield,theConstvtab,theJoinFromfield)

 '**Make theFarmAreaVTab editable.
  'If you can't edit the theme inform the user.
    if (theFarmAreaVTab.CanEdit.Not) then
      MsgBox.Info("Cannot edit table for theme:"++theFarmAreaVTab.AsString,"")
    end
  theFarmAreaVTab.SetEditable(TRUE)

 '***make new fields and add to theFarmAreaVTab table
   '** one field for total farm cost
   '**three fields for constraints
     theFarmAreaVTab.AddFields({
            Field.Make("TotCost".AsString,#FIELD_DOUBLE,16,2),
            Field.Make("cst_c".AsString,#FIELD_DOUBLE,16,2),
            Field.Make("cst_h".AsString,#FIELD_DOUBLE,16,2),
            Field.Make("cst_b".AsString,#FIELD_DOUBLE,16,2)    })
   '**four fields for weighting area categories
     theFarmAreaVTab.AddFields({Field.Make("ac",#FIELD_DOUBLE,16,3),
                              Field.Make("ah",#FIELD_DOUBLE,16,3),
```

```
                              Field.Make("ab",#FIELD_DOUBLE,16,3),
                              Field.Make("a",#FIELD_DOUBLE,16,3)})
  '**field for holding baseline area weight
    theFarmAreaVTab.AddFields({Field.Make("aBase",#FIELD_DOUBLE,16,3)})

 ' '**two fields for Baseline pub and private costs
 '  theFarmAreaVTab.AddFields({Field.Make("BasePrivC",#FIELD_DOUBLE,16,2),
 '                            Field.Make("BasePubC",#FIELD_DOUBLE,16,2)})
   theFarmAreaVTab.AddFields({Field.Make("BaseOppC",#FIELD_DOUBLE,16,2)})

'***calc min allowable crop and hay for each farm based on constraints
   areaFarm = theFarmAreaVTab.FindField("Sum_area")
   cstCfield = theFarmAreaVTab.FindField("cst_c")
   cstHfield = theFarmAreaVTab.FindField("cst_h")
   cstBfield = theFarmAreaVTab.FindField("cst_b")'
 'determine min allowable area in const type as max of ha or %*area
   for each record in theFarmAreaVTab
     theFarmAreaVTab.SetValue(cstCfield, record,
       theFarmAreaVTab.ReturnValue(theFarmAreaVTab.FindField("ha_c"), record) MAX
       (theFarmAreaVTab.ReturnValue(theFarmAreaVTab.FindField("%_c"), record)*
         (theFarmAreaVTab.ReturnValue(areaFarm, record))/100)
     )
     theFarmAreaVTab.SetValue(cstHfield, record,
       theFarmAreaVTab.ReturnValue(theFarmAreaVTab.FindField("ha_h"), record) MAX
       (theFarmAreaVTab.ReturnValue(theFarmAreaVTab.FindField("%_h"), record)*
         (theFarmAreaVTab.ReturnValue(areaFarm, record))/100)
     )
     theFarmAreaVTab.SetValue(cstBfield, record,
       theFarmAreaVTab.ReturnValue(theFarmAreaVTab.FindField("ha_b"), record) MAX
       (theFarmAreaVTab.ReturnValue(theFarmAreaVTab.FindField("%_b"), record)*
         (theFarmAreaVTab.ReturnValue(areaFarm, record))/100)
     )
   end 'for
```

```
 '**Make theFarmAreaVTab not editable.
   theFarmAreaVTab.SetEditable(FALSE)

'**run economic script to calc baseline costs
'Although this is done in the Baseline script, it is done first here
'to make the TempSum1 table first
  exitSucess = av.run("Econ",{theView})
```

B.4 Baseline.ave

```
'Calculates public and private costs for the baseline scenario.
'Should be run after preparing data and before running optimization component.
'Can be run separately from Prepare Data scripts in the case of Baseline changes only.
'Uses values in BaseMP column for calculations (overwriting values in MPID column).

'**first add FarmArea.dbf and TempSum1.dbf to Project, if not already****

 theView = av.GetProject.FindDoc("View1")
 theFLDftab = theView.FindTheme("Fields_mini.shp").GetFTab
 theMPvtab= av.GetProject.FindDoc("MP.txt").GetVTab
 theFarmAreaVTab = av.GetProject.FindDoc("FarmArea.dbf").GetVTab
 theTempSum1vtab = av.GetProject.FindDoc("TempSum1.dbf").GetVTab
 theFarmAreaVTab.UnjoinAll

'***set the MPID values = the Baseline MP values
  '**Make theFLDftab editable.
    theFLDftab.SetEditable(TRUE)
  '**set MPID = to BaseMP
     for each record in theFLDftab
     theFLDftab.SetValue (theFLDftab.FindField("Mpid"),record,
                      theFLDftab.ReturnValue(theFLDftab.FindField("BaseMP"),record))
    end 'for
  '**Make theFLDftab not editable.
    theFLDftab.SetEditable(FALSE)
  '**Refresh joins
    theFLDftab.Refresh

'**run each NPS script to calc baseline pollution and add to pollut.dbf table
  thePollList = {}
  'run NPS component (SedYld script) and add result to list
    theSedYld = av.run("NPS-Sed",{theView})
```

```
    thePollList.Add(theSedYld)
  '**call scripts for other pollutants here
  '**add result to list in order listed in pollut.dbf file

  thePollVtab = av.GetProject.FindDoc("pollut.dbf").GetVTab
  PBfield = thePollVtab.FindField("Base")
  thePollVtab.SetEditable(TRUE)
  for each record in thePollVtab
    PW = thePollList.Get(record)
    thePollVtab.SetValue(PBfield, record, PW)
    if (PW <= thePollVtab.ReturnValue(thePollVtab.FindField("target"),record) ) then
      'warn user
      MsgBox.Info("Base load <= Target load for pollutant"+
          thePollVtab.ReturnValue(thePollVtab.FindField("Pid"),record).AsString, "Warning")
    end 'if
  end 'for each record
  thePollVtab.SetEditable(FALSE)
  thePollList.Empty

'**run economic script to calc baseline costs
   exitSuccess = av.run("Econ",{theView})

'**Calculate Oppcost for each record
  MPtypeFromFldField = theFLDftab.FindField("MPtype")
  MPtypeFromMPfield = theMPvtab.FindField("Mptype")
  MPIDFromFldField = theFLDftab.FindField("Mpid")
  MPIDFromMPfield = theMPvtab.FindField("Mpid")
  profMaxMPField = theFLDftab.FindField("ProfMaxMP")
  oppCostField = theFLDftab.FindField("oppCost")
  theFLDftab.Calculate ("0", profMaxMPField)
  theFLDftab.Calculate ("0", oppCostField)
  theMPtypeList = List.Make
  '**Make theFLDftab editable.
```

```
  theFLDftab.SetEditable(TRUE)
  for each recFLD in theFLDftab
   'get MP type
    MPtypeFromFldValue =(theFLDftab.ReturnValueString(MPtypeFromFldField,
recFLD)).trim
    'add all MPIDs to list that are in same MPtype
     if (MPtypeFromFldValue = "f") then
      theMPtypeList.Add(theFLDftab.ReturnValueNumber(MPIDFromFldField, recFLD))
     else
      for each recMP in theMPvtab
       MPtypeFromMPValue = theMPvtab.ReturnValue(MPtypeFromMPfield, recMP)
       if ( ( MPtypeFromFldValue = MPtypeFromMPValue                    )
         or ( (MPtypeFromFldValue = "c") and (MPtypeFromMPValue = "h") )) then
           theMPtypeList.Add(theMPvtab.ReturnValueNumber(MPIDFromMPfield, recMP))
       end 'if
      end 'for each recMP
     end 'if-else
'MsgBox.ListAsString (theMPtypeList, " ", "list")

   'determine Opp Cost and Profit Max MP for this record based on MPtypeList
    for each recList in theMPtypeList
     'calc Opp Cost and identify Profit Max MPID
       oppCostvalue = theFLDftab.ReturnValue(oppCostField, recFLD)
       MPretfieldValue = theFLDftab.ReturnValueNumber
        (theFLDftab.FindField("MPret"+recList.AsString), recFLD)
       if (oppCostvalue <= MPretfieldValue) then  'MPret is pos for return so want max
       theFLDftab.SetValue(profMaxMPField,recFLD,recList)
       theFLDftab.SetValue(oppCostField,recFLD,MPretfieldValue)
      end 'if
    end 'for each recList
    theMPtypeList.Empty
  end 'for each recFLD
```

```
  '**Make theFLDftab not editable.
    theFLDftab.SetEditable(FALSE)

'**Calculate farm-level values
  '**join tempsum1 table with Farmarea table by FarmID
    theJoinFromfield = theTempSum1vtab.FindField("Farm_id")
    theJoinTofield = theFarmAreaVTab.FindField("Farm_id")
    theFarmAreaVTab.Join (theJoinTofield,theTempSum1vtab,theJoinFromfield)

  '**Make theFarmAreaVTab editable.
    theFarmAreaVTab.SetEditable(TRUE)

  '**update baseline values
    for each record in theFarmAreaVTab
      theFarmAreaVTab.SetValue(theFarmAreaVTab.FindField("aBase"),record,
         theFarmAreaVTab.ReturnValue(theFarmAreaVTab.FindField("a"),record) )
      theFarmAreaVTab.SetValue(theFarmAreaVTab.FindField("BaseOppC"),record,
         theFarmAreaVTab.ReturnValue(theFarmAreaVTab.FindField("Sum_oppcos"),record)
)
'     theFarmAreaVTab.SetValue(theFarmAreaVTab.FindField("BasePrivC"),record,
'        theFarmAreaVTab.ReturnValue(theFarmAreaVTab.FindField("Sum_Privc"),record) )
'     theFarmAreaVTab.SetValue(theFarmAreaVTab.FindField("BasePubC"),record,
'        theFarmAreaVTab.ReturnValue(theFarmAreaVTab.FindField("Sum_Pubfla"),record)
)
    end 'for
  '**Make theFarmAreaVTab not editable.
    theFarmAreaVTab.SetEditable(FALSE)

'**Report Finished
MsgBox.Info("Baseline script is finished", "Info")
```

B.5 CallDLL.ave

```
'** Set wshed size (by grid size) in NPS-sed script!!

'** Check theme names.

'** Run "PrepDat" script to prepare the USLE layers for the NPS-Sed component.

'** Run "PrepDat1" and "PrepDat2" scripts to prepare the (mainly economic) data. Follow the instructions at the top of each file.

'** Run the "Baseline" script to calculate the baseline values and determine extent of NPS problem


'** Now run the GA!
  'aDLL = DLL.Make("C:\My_Research\GAs\galib244-win\projects\BMPdll\Debug\BMPdll.dll".asFileName)
  aDLL = DLL.Make("C:\My_GIS_work\GA-Fin-MiniMC\BMPdll.dll".asFileName)


  'Create an instance of DLLProc
  avFunction = DLLProc.Make(aDLL,"SomeFunction", #DLLPROC_TYPE_STR,{})


  'Call the function in the DLL, supplying the proper parameters
  xStr = avFunction.Call({})


  'Report the results of the function call
  MsgBox.Info("The result is " + xStr+".", "The GA is finished!!")
  Return NIL
```

B.6 Bmpopt.dll

The DLL, coded in C++, runs the optimization component within the optimization procedure. To create the DLL, this code must be compiled in conjunction with the GALib program.

```
/* --------------------------------------------------------------------------
  BMPopt13DLL.C   by Tamie L Veith 31Oct01
-------------------------------------------------------------------------- */
// Header files
    //for dll
    #include <windows.h>
    #include <process.h>
    #include <C:\My_Research\GAs\galib244-win\projects\BMPopt\avexec32.h>
    //for all
    #include <fstream.h>   //read files
    #include <math.h>           //math functions
    #include <stdlib.h>         //standard library routines
    #include <ga/ga.h>          //contains GA function definations

//These functions are not exported.
    void RNG(int number, char *word[]);
    GAAlleleSetArray<int> CreateGenomeTemplate();
    void BaselineInitializer(GAGenome &g);
//  void ReadExistingPop(GAGenome &g);
    void ReadExistingPop(GAGenome &g, GAPopulation & p);
    void PopInitializer(GAPopulation &p);
    void PopEvaluator(GAPopulation &p);
    float Objective(GAGenome &);

//These functions are exported and available to calling applications.
    extern "C" __declspec(dllexport) char* SomeFunction(void);

//Define global variables and constants.
    //define max size of fields allowed
```

```
        const int iBaseRowARRAY_MAX = 300;
        int iBaseArray[iBaseRowARRAY_MAX];
        int iExistPopArray[iBaseRowARRAY_MAX];
        const int iMP_MAX = 20;
        char chMPTypeArray[iMP_MAX];
    //define max size of tables allowed
        const int iColn_MAX = 8; //# of cols in each table
         //(PollScore,CostScore,unlimitedSedScore, PollYld,WshedC,FarmC,MedianC,CE)
        float fAVReturnArray[iColn_MAX];
        float fBestArray[iColn_MAX];
    //define best genome score thus far
        float bestGenome;
    //define output files
        ofstream ofCommentary;
        ofstream ofGenomeScore;
        ofstream ofEvalOutput;

//Main function.
char* SomeFunction() //int main(int argc, char *argv[])
{
//Read in DLL settings
    char readTemp[50];
    float readCross;
    float readMut;
    int readPopSize;
    int readNumGen;
    float readFractRep;
    char readRepeat;
    ifstream ifDLLsettings;
    ifDLLsettings.open("ifDLLsettings.dat");
    ifDLLsettings >> readTemp >> readCross
                  >> readTemp>> readMut>> readTemp >> readPopSize
                  >> readTemp >> readNumGen >> readTemp >> readFractRep
```

```
                >> readTemp >> readRepeat;
    ifDLLsettings.close();

//Open output files.
    if (readRepeat == 'Y') {
        ofCommentary.open("ofCommentary.dat", (ios::out | ios::app));
        ofGenomeScore.open("ofGenomeScore.dat", (ios::out | ios::app));
        ofEvalOutput.open("ofEvalOutput.dat", (ios::out | ios::app));
    } else {
        ofCommentary.open("ofCommentary.dat", (ios::out | ios::trunc));
        ofGenomeScore.open("ofGenomeScore.dat", (ios::out | ios::trunc));
        ofEvalOutput.open("ofEvalOutput.dat", (ios::out | ios::trunc));
        //Print program description to console window.
            ofCommentary << "BMPopt program ver 13 by tlv (01-06-02) \n\n"
                        << "This program optimizes BMP placement based on \n"
                        << "routed sediment and public & private cost by farm \n"
                        << "Each scenario is compared to the baseline. \n \n";
            ofCommentary << "The GA is SteadyState, Tournament selection \n"
                        << "with no scaling and, not limited by ArcView software, \n"
                        << "terminates by 99% convergence over the user-specified \n"
                        << "number of generations. \n"<<endl;
        //Print DLL settings to output file
                ofCommentary << "crossover fraction: " << readCross <<endl;
                ofCommentary << "mutation fraction: " << readMut <<endl;
                ofCommentary << "populationSize: " << readPopSize <<endl;
                ofCommentary << "numberGenerations: " << readNumGen <<endl;
                ofCommentary << "fractionReplacement: " << readFractRep <<endl;
                ofCommentary << "Continue run: " << readRepeat <<endl<<endl;
        //Print file headers
            ofEvalOutput <<"gencnt"<< "\t"<<"P"<<"\t"<<"E"<<"\ t"
            <<"SedScr"<<"\t"<<"SedYld"<<"\t"<<"WshdCst"<<"\t"
            <<"FarmCst"<<"\t"<<"MedianC"<<"\t"<< endl;
            ofEvalOutput <<"[-]"<< "\t"<<
```

```
            "[-]"<<"\t"<<"[-]"<<"\t"<<"[-]"<<"\t"<<"[Mg/ha]"<<"\t"<<"[$]"<<"\t"<<
            "[$]"<<"\t"<<"[$]"<<"\t"<< endl;
    }

//Identify and start RNG.
    int seed = 0;
    // Identify the RNG.
    ofCommentary << "Using the " << GAGetRNG()
        << " random number generator (RNG).\n";
    // Initialize RNG by calling the routine with the specified seed.
    GARandomSeed(seed);
    ofCommentary << "Library thinks the random seed is: "
        << GAGetRandomSeed() << "\n \n";
  //RNG(argc,argv);

//Create Genetic Algorithm.
    //Create genome template from the baseline.
     GA1DArrayAlleleGenome<int> genome(CreateGenomeTemplate(), Objective);
    ofCommentary << "Template has been created. " << endl;

    //Create the population.
     GAPopulation gaPop(genome,readPopSize);
    //Initialize GA using allele set.
     //If ofFinalPop exists then initialize with it
     if (readRepeat == 'Y') {
        //Assign the initialization operator.
         ReadExistingPop(genome, gaPop);
     } else {
     // (else initialize with random individuals).
     gaPop.initializer(PopInitializer);
     gaPop.initialize();
     }
```

```
    //Set the popuation evaluator.
      gaPop.evaluator(PopEvaluator);
    //Assign the mutation operator.
      genome.mutator(GA1DArrayAlleleGenome<int>::FlipMutator);
    //Assign the evaluation operator.
      genome.evaluator(Objective);

    //Evaluate and keep inital genome, gBase, for future use.
      GA1DArrayAlleleGenome<int> gBase(genome);
      gBase.initializer(BaselineInitializer);
      gBase.initialize();
      gBase.evaluator(Objective);
      ofCommentary <<"\n evaluating baseline \n";
      gBase.evaluate();

/*
//Test Genetic Algorithm operators
    // test the initializer
        ofCommentary << "Genome after initialization:\n" << genome << endl;
    // test the mutator
        genome.mutate(.5);
        ofCommentary << "genome after mutation:\n" << genome << endl;
    // check the base genome
        ofCommentary << "Baseline Genome:\n" << gBase << endl;
    // test the crossover function
      GA1DArrayAlleleGenome<int>* a = new GA1DArrayAlleleGenome<int>(genome);
      GA1DArrayAlleleGenome<int>* b = new GA1DArrayAlleleGenome<int>(genome);
      GA1DArrayAlleleGenome<int>* c = new GA1DArrayAlleleGenome<int>(genome);
      GA1DArrayAlleleGenome<int>* d = new GA1DArrayAlleleGenome<int>(genome);
      a->initialize();
      b->initialize();
      b->mutate(1.00);
      ofCommentary << "parents:\n" << *a << "\n" << *b << "\n";
```

```
    SinglePointCrossover(*a, *b, c, d);   // test two child crossover
    ofCommentary << "children of crossover:\n" << *c << "\n" << *d << "\n";
    SinglePointCrossover(*a, *b, c, 0);   // test single child crossover
    ofCommentary << "child of crossover:\n" << *c << "\n";endl;
  // delete the extra genomes
    delete a;
    delete b;
    delete c;
    delete d;
*/

//Set GA characteristics
  // Set the default values of the parameters.
      GAParameterList params;
      GASteadyStateGA::registerDefaultParameters(params);
      params.set(gaNpopulationSize, readPopSize);     // population size
       //popsize is set when gaPop created but is not const so
       //the default param value must be overridden here as well.
      params.set(gaNpCrossover, readCross);    // probability of crossover
      params.set(gaNpMutation, readMut); // probability of mutation
      //params.set(gaNnGenerations, readNumGen);     // number of generations
      params.set(gaNpReplacement, readFractRep); // fraction of pop to replace each gen
       //OR (gaNnReplacement, 5);  // # of individuals to replace each gen
      params.set(gaNscoreFrequency, 1); // how often to record scores
      params.set(gaNpConvergence, 0.99); // ratio of BOG's that signals convergence
      params.set(gaNnConvergence, readNumGen); // how far back to get BOG for conv
      params.set(gaNselectScores, GAStatistics::AllScores);

  //set scaling function
    GANoScaling noScale;
 //set selection scheme
  GATournamentSelector tourn;
  //set GA type
```

```
    GASteadyStateGA  ga(gaPop);
   //set GA output
   GAStatistics stats;

   //assign features
    ga.parameters(params);
   //assign the crossover operator
    ga.crossover(GA1DArrayGenome<int>::OnePointCrossover);
   //assign scaling
    ga.scaling(noScale);
   //Assign selection method
    ga.selector(tourn);
   //define termination
    ga.terminator(GAGeneticAlgorithm::TerminateUponConvergence);

//Run genetic algorithm and print to file while evolving.
//   ofCommentary << "\n\n evolving..." << endl;
   int gencnt = 0;

while(gencnt < 360)
   {
      ofCommentary <<"\n"<<"evaluating generation " << gencnt<< endl;
      ofEvalOutput <<gencnt<<"\t";
      //calculate values for this generation
         ga.step();
      //write results to file
         genome = ga.statistics().bestIndividual();
         stats = ga.statistics();
         ofGenomeScore <<genome.score()<< "\t"<<genome<<endl;
//       ofEvalOutput <<ga.statistics().current(GAStatistics::Deviation)<<endl;
      gencnt++;
   } //end while
   ofCommentary << "\n\n";
```

```
//Report results
    //Print initial individual to file.
        ofCommentary << "The baseline genome is (score: "<< gBase.score() <<") \n"
         << gBase  <<"\n";
    //Print best individual to file.
        genome = ga.statistics().bestIndividual();
        ofCommentary << "The ga generated following string (objective score is ";
        ofCommentary << genome.score() << "):\n" << genome << "\n";
    //Print final population to file.
        ofCommentary << "Printing population to file 'ofFinalPop.dat'..." <<endl;
//      ofCommentary << "Stats: "<<"\n"<<stats;
        ofCommentary.flush();
        ofstream ofFinalPop;
        ofFinalPop.open("ofFinalPop.dat", (ios::out | ios::trunc));
        for(int kk=0; kk<ga.population().size(); kk++)
        {
            genome = ga.population().individual(kk);
            ofFinalPop <<  genome.score() << "\t"<<genome<<"\n";
        } //end for

//Close files
    ofFinalPop.close();
    ofCommentary.close();
    ofGenomeScore.close();
    ofEvalOutput.close();

//Exit program
    return "EXIT_SUCCESS";
}
/* end main routine------------------------------------------------------- */
```

```
/* ---------------------------------------------------------------------------
RNG function
  This function checks for a seed entered on the command-line. If none then
  seed is randomly selected. The function then starts the random number
  generator set in the program, prints out the seed and generator being used
  and returns control to the main routine.
--------------------------------------------------------------------------- */
void
RNG(int iMyNum, char *sMyWord[])
{
  // See if random seed was specified for testing purposes.
      int seed = 0;
      for(int i=1; i<iMyNum; i++)
      {    //loop until k reaches number entered on command line
      if(strcmp("seed", sMyWord[i]) == 0)
      {    //compare "seed" with word entered on command line
           if(++i >= iMyNum)
           {          //if next k is as big as your number, report error and exit
                cerr << "You must enter a number when specifying a random seed.\n";
                exit(1);
           } //end if, do else
           else
           {
                seed = atoi(sMyWord[i]);
           } //end else-then
      } //end if-then
      }    //end for

      // Tell us which RNG we're using...
      ofCommentary << sMyWord[0] << ": Random Number Test\n";
      ofCommentary << "Using the " << GAGetRNG()
           << " random number generator (RNG).\n";
```

```
    // initialize the RNG by calling the seed routine with our seed
    if(seed) ofCommentary << "Using specified random seed " << seed << "\n";
    GARandomSeed(seed);
    ofCommentary << "Library thinks the random seed is " <<
        GAGetRandomSeed() << "\n";
}
/* end RNG function------------------------------------------------------- */

/* -------------------------------------------------------------------------
CreateGenomeTemplate function
  This function creates baseline array & uses it to choose an allele set
  for each gene. The baseline array is created using the baseline values
  (MP.txt) as its elements. This is done, instead of reading the inputs
  directly into a genome, because of the constructor limitations for the
  defined genome classes; i.e., an array genome can only be created from
  another such genome, or from the set of alleles and the objective
  function, or from the objective function - but not from scratch or array.
------------------------------------------------------------------------- */
GAAlleleSetArray<int>
CreateGenomeTemplate()
{
    //Read in baseline scenario
    ifstream ifMPID;
    ifMPID.open("ifMPID.dat");
    //ifMPID.dat is a space or tab delimited text file of one column.
    for (int iGenomeLen = 0; !ifMPID.eof(); iGenomeLen++)
    {
        //read line from input file into genome array
        ifMPID >> iBaseArray[iGenomeLen];
        //if - break line corrects for looping one more eof since
        //line is not read until after for statement
        if (ifMPID.eof()) break;
    } //end for
```

```
ifMPID.close();

/* -------------------------------------------------------------------------
CreateAlleleSets sub function
  This function reads the list of MPs and MPtypes from Mp.txt and uses it
  to make allele sets.
------------------------------------------------------------------------- */
//Define allele sets
     GAAlleleSet<int> alleleSetH;
     GAAlleleSet<int> alleleSetC;

//Read in MP.txt
     int iReadMPID;
     char readTemp[50];
     char cReadMPtype;
     ifstream ifMPlist;
     ifMPlist.open("MP.txt");
     //discard first line
     ifMPlist >> readTemp >> readTemp >> readTemp >>
               readTemp >> readTemp >> readTemp >> readTemp;
     //1st column is MPID, 2nd is MPdesc, 3rd is MPtype, next 4 colns extra.
          //read line from input file into temp vars
     while (ifMPlist >> iReadMPID >> readTemp >> cReadMPtype >>
               readTemp >> readTemp >> readTemp >> readTemp)
     {
          //store MPtypes in array for creating genome template
          chMPTypeArray[iReadMPID] = cReadMPtype;
          //make crop and hay allele sets

          if (cReadMPtype == 'h')
               //if hay type land use then value may be in hay types
               //also add hay types to crop set (see else-if)
               //add MP to hay list and crop list
```

```
            {
                alleleSetH.add(iReadMPID);
                alleleSetC.add(iReadMPID);
            } //end if 'h'
            else
            {
                if (cReadMPtype == 'c')
                //if crop type land use then value may be in hay or crop types
                {
                    alleleSetC.add(iReadMPID);
                  //this line added to allow all hay and crop types for both
                    //to evaluate optimization on both beef and dairy farms
                    alleleSetH.add(iReadMPID);
                } //end if 'c'
            } //end else 'h'
        } //end while
        ifMPlist.close();
    /* end CreateAlleleSets subfunction---------------------------------------- */

    // This genome is created using an array of allele sets. This means that
    // each element of the genome will assume a value in its corresponding
    // allele set. The alleles are defined based on the baseline scenario.
    // This allows specialization of each allele set based on baseline genome.

    GAAlleleSetArray<int> alleleSetArray;
    for (int i=0; i<iGenomeLen; i ++)
    {
        if (chMPTypeArray[iBaseArray[i]] == 'f')
            //if a non-changing land use then fix value
        {
            GAAlleleSet<int> alleleSetFix1;
            alleleSetFix1.add(iBaseArray[i]);
            alleleSetArray.add(alleleSetFix1);
```

```
        }
        else
        {
            if (chMPTypeArray[iBaseArray[i]] == 'h')
            {
                alleleSetArray.add(alleleSetH);
            }
            else
            {
                if (chMPTypeArray[iBaseArray[i]] == 'c')
                {
                    alleleSetArray.add(alleleSetC);
                }//end if 'c'
            } //end else 'h'
        } //end else 'f'
    } //end for
    return alleleSetArray;
}
/* end CreateGenomeTemplate function-------------------------------------- */

/* ------------------------------------------------------------------------
BaselineInitializer function
  This initializer fills the values from the baseline into the genome template.
------------------------------------------------------------------------ */
void
BaselineInitializer(GAGenome &g)
{
    GA1DArrayAlleleGenome<int> &genome=(GA1DArrayAlleleGenome<int> &)g;
    for(int i=0; i<genome.size(); i++)
    {
        genome.gene(i, iBaseArray[i]);
    }
}
```

```
/* end BaselineInitializer function----------------------------------------- */


/* ---------------------------------------------------------------------------
ReadExistingPop function
  This initializer reads the values from the finalpop.dat file into an
  array, one genome at a time, for the initial population.
--------------------------------------------------------------------------- */
void
ReadExistingPop(GAGenome &g, GAPopulation & p)
{
    float flScore;
    GA1DArrayAlleleGenome<int> &genome=(GA1DArrayAlleleGenome<int> &)g;
    //Read in finalpop.dat file
ofCommentary << "Initial population (previous final population) \n";
    ifstream ofFinalPop;
        ofFinalPop.open("ofFinalPop.dat");
    for (int i = 0; i<p.size(); i++)
    {
        for(int iGenomeLen=0; iGenomeLen<=genome.size(); iGenomeLen++)
        {
            if (iGenomeLen == 0) {
                //skip genome score
                ofFinalPop >>flScore;
            } else {
            //read line from input file into genome array
            ofFinalPop >> iExistPopArray[iGenomeLen-1];
            genome.gene((iGenomeLen-1),iExistPopArray[iGenomeLen-1]);
ofCommentary <<iExistPopArray[iGenomeLen-1]<<" ";
            }
        }
        p.individual(i) = genome;
ofCommentary <<endl;
    }
```

```
}
/* end ReadExistingPop function------------------------------------------ */

/* ----------------------------------------------------------------------------
PopInitializer function
  This function initializes the population based on the first genome. This
  is done so that the GA can run on the population and not the indivual.
---------------------------------------------------------------------------- */
void
PopInitializer(GAPopulation & p)
{
    for (int i = 0; i<p.size() ; i++)
    {        p.individual(i).initialize();
    }
    ofCommentary <<"Population has been initialized."<<endl;
}
/* end PopInitializer function-------------------------------------------------- */

/* --------------------------------------------------------------------------------
PopEvaluator function
  This function evaluates the population by calling the objective function to
  evaluate each individual and then ranking all individuals in the population.
-------------------------------------------------------------------------------- */
void
PopEvaluator(GAPopulation & p)
{
    for (int i = 0; i<p.size(); i++)
    {
        p.individual(i).evaluate(); //evaluate() calls Objective function
    }
    //print data for best genome
    for (int j = 0; j<iColn_MAX; j++) {
        ofEvalOutput <<fBestArray[j]<<"\t";
```

```
    }
    ofEvalOutput <<endl;
}
/* end PopEvaluator function-------------------------------------------------- */

/* -------------------------------------------------------------------------
Objective function
  // This objective tries to maximize the cost-effectiveness of the genome
  //as a whole.
-------------------------------------------------------------------------- */
float
Objective(GAGenome& g)
{
    GA1DArrayAlleleGenome<int>& genome = (GA1DArrayAlleleGenome<int>&)g;
    char myBigStr[2000];
    char buffer[5];
    strcpy(myBigStr, "");

    for(int i=0; i<genome.length(); i++)
    {
        //add values for this gene to the string
          //char *_itoa(int value, char *string, int radix);(radix 10=base 10)
        strcat(myBigStr,_itoa(genome.gene(i),buffer,10));
        strcat(myBigStr," ");
    } //end for

/* ** BEGIN CALL TO ARCVIEW ** to change gene ID into 7 float outputs */
    //define vars and create "call-to-script" command
        //char *pMyAvenueStr = "av.run(\"Main\",\"NIL)";
    char MyAvenueStr[2000]="av.run(\"Main\",{\"";
    strcat(MyAvenueStr,myBigStr);
    strcat(MyAvenueStr,"\"})");
    char *pResultStr;
```

```
    //call ArcView script
    pResultStr = AVExec(MyAvenueStr);
ofCommentary <<" ArcView Result: "<< pResultStr << "\t";
//  theResultStr = P.AsString++E.AsString++PACTUAL.AsString
//       ++theSedYld.AsString++WshedCost.AsString++(sumFarmCost.Sqrt).AsString
//       ++MedianCost.AsString++theCostEff.AsString

    fAVReturnArray[0] = atof(strtok(pResultStr, " "));
    fAVReturnArray[1] = atof(strtok(0, " "));
    fAVReturnArray[2] = atof(strtok(0, " "));
    fAVReturnArray[3] = atof(strtok(0, " "));
    fAVReturnArray[4] = atof(strtok(0, " "));
    fAVReturnArray[5] = atof(strtok(0, " "));
    fAVReturnArray[6] = atof(strtok(0, " "));
    fAVReturnArray[7] = atof(strtok(0, " "));
 /* ** END CALL TO ARCVIEW ** */

    if (fAVReturnArray[0] <1) {
        genome.score(fAVReturnArray[0]); }
    else {
        genome.score(fAVReturnArray[1]); }

ofCommentary <<" genome score: "<< genome.score() <<endl;

    if (genome.score() > bestGenome) {
        for (i = 0; i<iColn_MAX; i++) {
            fBestArray[i] = fAVReturnArray[i];
        }
        bestGenome = genome.score();
    }
  return genome.score();
}
/* end Objective function--------------------------------------------------- */
```

B.7 Main.ave

```
'This script accepts the MPID's for the working scenario from each GA call and
'runs the NPS and Econ components.
'It calculates the fitness scores and returns the results to the GA.

 theView = av.GetProject.FindDoc("View1")
'***Extract MPIDs From Genome************************************
  'read MPIDs in from script parameter call
    myString = self.Get(0)
  'locate data
    theMPIDftab = theView.FindTheme("Fields_mini.shp").GetFTab
    theMPIDfield = theMPIDftab.FindField("Mpid")
  'update MPID field
    theMPIDftab.SetEditable(TRUE)
     for each record in theMPIDftab
        theMPIDvalue = myString.Extract(record).AsNumber
       theMPIDftab.SetValue(theMPIDfield, record, theMPIDvalue)
      end 'for (record)
    theMPIDftab.SetEditable(FALSE)
  'recalculate joins
   theMPIDftab.Refresh

'***Score for the NPS component**********************************
  thePollList = {}
  'run NPS component (SedYld script) and add result to list
    theSedYld = av.run("NPS-Sed",{theView})
    thePollList.Add(theSedYld)
  '**call scripts for other pollutants here
  '**add result to list in order listed in pollut.dbf file

  thePollVtab = av.GetProject.FindDoc("pollut.dbf").GetVTab
  PBfield = thePollVtab.FindField("Base")
```

```
  PTfield = thePollVtab.FindField("Target")
  Weightfield = thePollVtab.FindField("weight")

  PIweightedSum = 0
  WeightSum = 0
  P = 0
  thePollVtab.SetEditable(TRUE)
  for each record in thePollVtab
    PB = thePollVtab.ReturnValue(PBfield, record)
    PT = thePollVtab.ReturnValue(PTfield, record)
    Weight = thePollVtab.ReturnValue(Weightfield, record)
    PW = thePollList.Get(record)
    ' calc sediment score specifically, for method/data analysis
    PACTUAL = ((PB-PW)/(PB-PT))
    if (PB <= PW) then
      PI = 0
    elseif (PW <= PT) then
      PI = 1
    else
      PI = PACTUAL
    end
    PIweightedSum = PIweightedSum+(Weight*PI)
    WeightSum = WeightSum+Weight
  end 'for
  P = PIweightedSum/WeightSum
  thePollVtab.SetEditable(FALSE)
  thePollList.Empty

'***Econ component**********************************
  exitSucess = av.run("Econ",{theView})
  theFarmAreaVTab = av.GetProject.FindDoc("FarmArea.dbf").GetVTab
  theFarmAreaVTab.refresh
```

```
'  BasePrivField = theFarmAreaVTab.FindField("BasePrivC")
'  BasePubField = theFarmAreaVTab.FindField("BasePubC")
  OppCostField = theFarmAreaVTab.FindField("Sum_OppCos")

'  WorkPrivField = theFarmAreaVTab.FindField("Sum_privc")
'  WorkPubField = theFarmAreaVTab.FindField("Sum_pubfla")
  WorkTotCostField = theFarmAreaVTab.FindField("TotCost")

  aField = theFarmAreaVTab.FindField("a")
  aBaseField = theFarmAreaVTab.FindField("aBase")

  OppCostSum = 0
  ai = 0
  x2i = 0
  WshedCost = 0
  sumFarmCost = 0
  Wsum = 0
  E = 0

  for each record in theFarmAreaVTab
    OppCostSum = OppCostSum+ theFarmAreaVTab.ReturnValue(OppCostField, record)
    ai = ( (theFarmAreaVTab.ReturnValue(aField, record))
            /(theFarmAreaVTab.ReturnValue(aBaseField, record)) ) Min 1
    x2i = (theFarmAreaVTab.ReturnValue(WorkTotCostField, record))^2
    WshedCost = WshedCost + (theFarmAreaVTab.ReturnValue(WorkTotCostField, record))
    sumFarmCost = sumFarmCost+x2i
    Wsum = Wsum + (x2i/ai)
  end 'for
   E = 1+ ( OppCostSum/((Wsum+1).Sqrt) )
```

```
'**errorcheck**
  If ((theSedYld.isNull ) or (exitSucess <> 1)) then
    MsgBox.Info("NPS or Econ component did not run correctly", "Error")
  End 'if

'** calculate median farm cost****
    'determine middle row or one above midpoint from top (count includes '0')
      middleRow = ((theFarmAreaVTab.GetNumRecords)/2 ).Truncate
    'sort farmarea.dbf table descending
      myTable = av.GetProject.FindDoc( "Farmarea.dbf" )
      aField = myTable.GetVTab.FindField( "TotCost" )
      if ((myTable.GetWin.IsOpen).Not) then   ' Is myTable open
        myTable.GetWin.Open
      end
      myTable.Sort( aField, TRUE )
    'find the record associated with the midpoint row
      MedianRecord = myTable.ConvertRowToRecord (middleRow)
    'close table
      myTable.GetWin.Close
    'find cost
      MedianCost = theFarmAreaVTab.ReturnValue(WorkTotCostField, MedianRecord)
'** END calculate median farm cost****

'***Return results and clean up********************
  'set output variables
    theResultStr = P.AsString++E.AsString++PACTUAL.AsString++
              theSedYld.AsString++WshedCost.AsString++(sumFarmCost.Sqrt).AsString++
              MedianCost.AsString++0.AsString
'MsgBox.Info (theResultStr, "from AV: theResultStr")

  av.PurgeObjects
  Return theResultStr
```

B.8 Econ.ave

```
'farm level private costs and public costs with area requirements

'***Set Constant Values**********************************************
'  '***Economic Data**********
   cnstPubPrice = 194.42+153.40 'public cost per farm [$]
     '$/farm annualized over 5 years (Carpentier et al., 1998)
'***END Set Constant Values*******************************************

'***Locate Input Data************************************************
  theView = self.Get(0)
  theMPIDftab = theView.FindTheme("Fields_mini.shp").GetFTab
  theMPIDfield = theMPIDftab.FindField("Mpid")
  theMPIDftab.refresh

  theFarmAreaVTab = av.GetProject.FindDoc("FarmArea.dbf").GetVTab
  theFarmAreaVTab.refresh

  theBaseMPfield = theMPIDftab.FindField("BaseMP")

  theOppCostField = theMPIDftab.FindField("OppCost")
  thePrivCostfield = theMPIDftab.FindField("PrivC")

  theFarmfield = theMPIDftab.FindField("Farm_id")
  theMPtypefield = theMPIDftab.FindField("Mptype")
  theFLDareafield = theMPIDftab.FindField("Sum_area")
  theareaCfield = theMPIDftab.FindField("areaC")
  theareaHfield = theMPIDftab.FindField("areaH")

  theCfactfield = theMPIDftab.FindField("Cfact")
  thePfactfield = theMPIDftab.FindField("Pfact")
  theALPHAfield = theMPIDftab.FindField("ALPHA")
```

```
  thePubFlagfield = theMPIDftab.FindField("PubFlag")
'***END Locate Input Data*********************************************

theMPIDftab.SetEditable(TRUE)
'***reset vars (to clear out previous scenario values)*****************
  'reset private cost values to zero (to clear out previous scenario values)
   theMPIDftab.Calculate ("0", thePrivCostfield)
  'reset area constraint values to zero (to clear out previous scenario values)
   theMPIDftab.Calculate ("0", theareaCfield)
   theMPIDftab.Calculate ("0", theareaHfield)
  'reset PubFlag values to zero (to clear out previous scenario values)
   theMPIDftab.Calculate ("0", thePubFlagfield)

for each record in theMPIDftab
  '***calc public cost by field*******************************************
    'set flag if MP has changed
    if( theMPIDftab.ReturnValue(theMPIDfield,record)<>
        theMPIDftab.ReturnValue(theBaseMPfield,record) ) then
      theMPIDftab.SetValue(thePubFlagfield,record, "1")
    end 'if
  '***END calc public cost by field***********************

  '***calc private cost***********************
    'get opportunity cost
     theOppCostValue = theMPIDftab.ReturnValue(theOppCostField,record)
    'get this scenario cost (pos is a return, neg is a cost)
     theMPID = theMPIDftab.ReturnValue(theMPIDfield, record)
     theCurRetName =  ("MPret"+ theMPID.AsString.Trim).AsString
     theCurRetField = theMPIDftab.FindField(theCurRetName)
     theCurRetValue = theMPIDftab.ReturnValue(theCurRetField,record)
    'update private cost var (now pos is a cost, neg is a return)
     theMPIDftab.SetValue(thePrivCostfield, record, theOppCostValue-theCurRetValue)
  '***END calc private cost***********************
```

```
  '***calc area constraint weights by field************************************
    'sort area of each field into crop or hay for constraint purposes
     if   ((theMPIDftab.ReturnValue(theMPtypefield, record)).trim="c") then
       theMPIDftab.SetValue(theareaCfield,record,
(theMPIDftab.ReturnValue(theFLDareafield, record)))
     elseif ((theMPIDftab.ReturnValue(theMPtypefield, record)).trim="h") then
       theMPIDftab.SetValue(theareaHfield,record,
(theMPIDftab.ReturnValue(theFLDareafield, record)))
     end 'if
  '***END area constraint weights by field************************
end 'for (record)
theMPIDftab.SetEditable(FALSE)

'***summarize by farm**********************
   sumFarmTemp1ftab = theMPIDftab.Summarize( "tempsum1".AsFileName, Dbase,
theFarmfield,
           {thePrivCostfield, thePubFlagfield, theareaCfield, theareaHfield,theOppCostField},

{#VTAB_SUMMARY_SUM,#VTAB_SUMMARY_SUM,#VTAB_SUMMARY_SUM,#V
TAB_SUMMARY_SUM,#VTAB_SUMMARY_SUM})

'***Locate Data*********************************************
  'public cost
    sumPubFlagField = sumFarmTemp1ftab.FindField("Sum_PubFla")
  'private cost
    sumPrivCostField = sumFarmTemp1ftab.FindField("Sum_privc")
  'total farm cost
    totcostfield = theFarmAreaVTab.FindField("TotCost")
  'area categories
    sum_areaC = sumFarmTemp1ftab.FindField("Sum_areaC")
    sum_areaH = sumFarmTemp1ftab.FindField("Sum_areaH")
  'area weight variables
    acfield = theFarmAreaVTab.FindField("ac")
    ahfield = theFarmAreaVTab.FindField("ah")
```

```
   abfield = theFarmAreaVTab.FindField("ab")
   afield  = theFarmAreaVTab.FindField("a")
'***END Locate Data*********************************************

'******calc Public cost by farm*****************************
sumFarmTemp1ftab.SetEditable(TRUE)
 for each record in sumFarmTemp1ftab
   thePubFlagValue = sumFarmTemp1ftab.ReturnValue(sumPubFlagField,record)
   if (thePubFlagValue > 0) then
    'update public cost var
     sumFarmTemp1ftab.SetValue(sumPubFlagField, record, cnstPubPrice)
   end 'if
  end 'for (record)
sumFarmTemp1ftab.SetEditable(FALSE)
'******END calc Public cost by farm*****************************

'**Refresh joins
  theFarmAreaVTab.Refresh

'set table editable
 theFarmAreaVTab.SetEditable(TRUE)

  '***reset vars (to clear out previous scenario values)*****************
   'area weight variables
     theFarmAreaVTab.Calculate ("0", totcostfield)
     theFarmAreaVTab.Calculate ("0", acfield)
     theFarmAreaVTab.Calculate ("0", ahfield)
     theFarmAreaVTab.Calculate ("0", abfield)
     theFarmAreaVTab.Calculate ("0", afield)
  '***sum up area categories by farm***********************
   'update values in Farmarea table using values from temp farm sum table
     for each record in theFarmAreaVTab
```

```
    'locate area constraint values
     cstCvalue = theFarmAreaVTab.ReturnValue(theFarmAreaVTab.FindField("Cst_c"),
record)
     cstHvalue = theFarmAreaVTab.ReturnValue(theFarmAreaVTab.FindField("Cst_h"),
record)
     cstBvalue = theFarmAreaVTab.ReturnValue(theFarmAreaVTab.FindField("Cst_b"),
record)
    'run through temp table and match Farm_IDs, just in case order is different
    '(first time through this script, the two tables aren't joined.)
     for each record2 in sumFarmTemp1ftab
      if ( theFarmAreaVTab.ReturnValue(theFarmAreaVTab.FindField("Farm_id"),
record) =
         sumFarmTemp1ftab.ReturnValue(sumFarmTemp1ftab.FindField("Farm_id"),
record2) ) then
       'add together costs (for output calculations)
        sumPrivCostvalue = sumFarmTemp1ftab.ReturnValue(sumPrivCostField, record2)
        sumPubFlagvalue = sumFarmTemp1ftab.ReturnValue(sumPubFlagField, record2)
        theFarmAreaVTab.SetValue(totcostfield,record,
sumPrivCostvalue+sumPubFlagvalue )


       'locate current area in each category
        areaCvalue = sumFarmTemp1ftab.ReturnValue(sum_areaC, record2)
        areaHvalue = sumFarmTemp1ftab.ReturnValue(sum_areaH, record2)
        areaBvalue = areaCvalue + areaHvalue
       'update area constraint penalty
        'count of constraints used, for determining final weighting factor
         acount = 0
         avalue = 0
        if (cstCvalue <> 0) then
         theFarmAreaVTab.SetValue(acfield,record, (areaCvalue/cstCvalue) Min (1) )
         acount = acount + 1
        end
        if (cstHvalue <> 0) then
        theFarmAreaVTab.SetValue(ahfield,record, (areaHvalue/cstHvalue) Min (1) )
```

```
          acount = acount + 1
        end
        if (cstBvalue <> 0) then
        theFarmAreaVTab.SetValue(abfield,record, (areaBvalue/cstBvalue) Min (1) )
          acount = acount + 1
        end
        if (acount = 0) then
          avalue = 1
          theFarmAreaVTab.SetValue(afield,record, avalue )
        else
          avalue = (theFarmAreaVTab.ReturnValue(acfield,record) +
                   theFarmAreaVTab.ReturnValue(ahfield,record) +
                   theFarmAreaVTab.ReturnValue(abfield,record)  )/acount
          if (avalue <> 0) then
            theFarmAreaVTab.SetValue(afield,record, avalue )
          else
            theFarmAreaVTab.SetValue(afield,record,0.001)
          end 'if (avalue <> 0)
        end 'if (acount = 0)
      'break out of loop and go to next farm in theFarmAreaVTab
        break
    end 'if match
   end 'for each record2 in sumFarmTemp1ftab
 end 'for each record in theFarmAreaVTab

'set table not editable
 theFarmAreaVTab.SetEditable(FALSE)
'***END calc area constraint weights by farm************************************
  Return 1
```

B.9 NPS-sed.ave

```
'Calculates the sediment loading for the NPS component.

'***Set Constant Values********************************************
   cnstCellArea = 0.09  'cell size [ha]
   cnstWshedArea = 686.07 'watershed size [ha]

  '**set cell size and extent for shapefile-to-grid conversion
   theCellSize = 30 'length of one side of square cell [m]
   theExtent = Rect.MakeXY(669830.142059,4253895,683150.142059,4271685)
    'MakeXY(x1,y1,x2,y2)

'***Locate Input Data**********************************************
  theView = self.Get(0)
  theMPIDftab = theView.FindTheme("Fields_mini.shp").GetFTab
  theMPIDfield = theMPIDftab.FindField("Mpid")
  theFlowDirGrid = theView.FindTheme("Flow Direction").GetGrid
  theFlowAccumGrid = theView.FindTheme("Flow Accumulation").GetGrid
  theRKSLgrid = theView.FindTheme("RKSL-si").GetGrid
  theLengthGrid  = theView.FindTheme("Length Across Cell").GetGrid
  theSlopeDecGrid  = theView.FindTheme("Slope in m/m").GetGrid
  thePTheme = theView.FindTheme("Outlet_mini.shp")
   thePointField = thePTheme.GetFTab.FindField("shape")
   thePointValue = thePTheme.GetFTab.ReturnValue(thePointField,0)

'***NPS Prediction Component******************************************
'  '***Check MP Placement *************************************
'   '**locate within land use shapefile and grid MP description
'    theMPIDgrid = Grid.MakeFromFTab(theMPIDftab, Prj.MakeNull, theMPIDfield,
{theCellSize,theExtent})
'   '** create theme and add to view
'     theView.AddTheme(av.run("MakeGtheme",{theMPIDgrid,"MPID"}))
```

```
 '***Calculate Gross Erosion ***********************************
  '**grid C factor of USLE equation
   theCfield = theMPIDftab.FindField("Cfact")
   theCgrid = Grid.MakeFromFTab(theMPIDftab, Prj.MakeNull, theCfield,
{theCellSize,theExtent})
   '** create theme and add to view
'      theView.AddTheme(av.run("MakeGtheme",{theCgrid,"C factor"}))


  '**grid P factor of USLE equation
   thePfield = theMPIDftab.FindField("Pfact")
   thePgrid = Grid.MakeFromFTab(theMPIDftab, Prj.MakeNull, thePfield,
{theCellSize,theExtent})
   '** create theme and add to view
'     theView.AddTheme(av.run("MakeGtheme",{thePgrid,"input P factor"}))
   'calc P for contour
    truePgrid= (((thePgrid = 2.AsGrid) and (theSlopeDecGrid <=(0.02).AsGrid))*0.6)
     +(((thePgrid = 2.AsGrid) and (theSlopeDecGrid >(0.02).AsGrid) and(theSlopeDecGrid
<=(0.07).AsGrid))*0.5)
     +(((thePgrid = 2.AsGrid) and (theSlopeDecGrid >(0.07).AsGrid) and(theSlopeDecGrid
<=(0.12).AsGrid))*0.6)
     +(((thePgrid = 2.AsGrid) and (theSlopeDecGrid >(0.12).AsGrid) and(theSlopeDecGrid
<=(0.18).AsGrid))*0.8)
     +(((thePgrid = 2.AsGrid) and (theSlopeDecGrid >(0.18).AsGrid) and(theSlopeDecGrid
<=(0.24).AsGrid))*0.9)
     +(((thePgrid = 2.AsGrid) and (theSlopeDecGrid >(0.24).AsGrid)                    )*1)
     +((thePgrid = 1.AsGrid) *1)
   '** create theme and add to view
'    theView.AddTheme(av.run("MakeGtheme",{truePgrid,"calc P factor"}))


  '**calculate USLE cell-level gross erosion
   theErosGrid = theCgrid*truePgrid*theRKSLgrid*cnstCellArea.asGrid ' [Mg]
   '** create theme and add to view
'     theView.AddTheme(av.run("MakeGtheme",{theErosGrid,"Gross Erosion by Cell in
Mg"}))
```

```
 '***Route Sediment Delivery*************************************
  '**locate within land use shapefile and grid ALPHA factor of sediment delivery equation
   theALPHAfield = theMPIDftab.FindField("Alpha")
   theALPHAgrid = Grid.MakeFromFTab(theMPIDftab, Prj.MakeNull, theALPHAfield,
{theCellSize,theExtent})
   '** create theme and add to view
'     theView.AddTheme(av.run("MakeGtheme",{theALPHAgrid,"Alpha grid"}))

  '**calculate delivery from cell with equation: alpha*SQRT(slope/length)
   theDgrid = theALPHAgrid*((theSlopeDecGrid/theLengthGrid).SQRT)
   '** create theme and add to view
'     theView.AddTheme(av.run("MakeGtheme",{theDgrid,"Unbounded Delivery Function
by Cell"}))

'   '**limit (delivery in streams) or (delivery > 1) to 1
      theMaxDgrid =  ( (  theDgrid=0                              ) * (0.001) )
                +( ( ((theDgrid>0) and (theDgrid<1))
                     and (theFlowAccumGrid<60) ) * (theDgrid) )
                +( ( (theDgrid>=1) and (theFlowAccumGrid<60) ) * (1)        )
                +( ( (theFlowAccumGrid>=60)  and (theFlowAccumGrid<200) ) * (0.98) )
                +( ( (theFlowAccumGrid>=200)                         ) * (0.9997) )

   '** create theme and add to view
'     theView.AddTheme(av.run("MakeGtheme",{theMaxDgrid,"Delivery Function by
Cell"}))

  theLnDgrid = (theMaxDgrid.Log) *(-1)
      'mult by -1 to make ln numbers pos for flow accum
   '** create theme and add to view
'     theView.AddTheme(av.run("MakeGtheme",{theLnDgrid,"-ln(Del)"}))

  theDelAccumGrid = theFlowDirGrid.FlowLength((theLnDgrid/theLengthGrid),FALSE)
   '** create theme and add to view
```

```
'     theView.AddTheme(av.run("MakeGtheme",{theDelAccumGrid,"flowlength (-
ln(Del)/length)"}))


   theExpDwnstrmAccum = (theDelAccumGrid*(-1)).Exp
      'mult by -1 to return ln numbers to proper sign
    '** create theme and add to view
'     theView.AddTheme(av.run("MakeGtheme",{theExpDwnstrmAccum,"Downstream
Delivery Surface"}))
'
theView.AddTheme(av.run("MakeGtheme",{((theExpDwnstrmAccum+(0.005))*100).Int,"%
erosion delivered"}))


  '***Calculate NPS Results *************************************
'    '***Gross Erosion * Delivery Function at cell *******************
'     theCellSedYldGrid = theErosGrid*theMaxDgrid/cnstCellArea
'     '** create theme and add to view
'      theView.AddTheme(av.run("MakeGtheme",{theCellSedYldGrid,"Sediment from cell in
Mg/ha"}))
'
'   '***Gross Erosion * Delivery Product ************************
'     theCellSedLossGrid = theErosGrid*theExpDwnstrmAccum
'     '** create theme and add to view
'      theView.AddTheme(av.run("MakeGtheme",{theCellSedLossGrid,"Sediment delivered
to outlet in Mg"}))
'
'    '***Total Gross Erosion *************************************
'     theTotGEGrid = theFlowDirGrid.FlowAccumulation(theErosGrid)
'     '** create theme and add to view
'     '  theView.AddTheme(av.run("MakeGtheme",{theTotGEGrid,"Total gross erosion in
Mg"}))
'        ' this is an accumulation grid, used to get and check value at outlet.
'     '**Get value of outlet point in Mg/ha
'      theTotGE = (theTotGEGrid.CellValue(thePointValue,Prj.MakeNull))/cnstWshedArea
```

```
"*for placement test*
'    '***Field Total Gross Erosion ***********************************
'    theFldTotGEGrid =
theFlowDirGrid.FlowAccumulation(theErosGrid*(theALPHAgrid>9.6))
'     '** create theme and add to view
"      theView.AddTheme(av.run("MakeGtheme",{theFldTotGEGrid,"Field Tot gross erosion
in Mg"}))
'          ' this is an accumulation grid, used to get and check value at outlet.
'     '**Get value of outlet point in Mg/ha
'      theFldTotGE = (theFldTotGEGrid.CellValue(thePointValue,Prj.MakeNull))/(40*0.09)
"*for placement test*


   '***Total Sediment Yield ************************************
    theTotSedLossGrid =
theFlowDirGrid.FlowAccumulation(theErosGrid*theExpDwnstrmAccum)
     '** create theme and add to view
     ' theView.AddTheme(av.run("MakeGtheme",{theTotSedLossGrid,"Total Sed Loss in
Mg"}))
         ' this is an accumulation grid, used to get and check value at outlet.
     '**Get value of outlet point in Mg/ha
      theSedYld =
(theTotSedLossGrid.CellValue(thePointValue,Prj.MakeNull))/cnstWshedArea


'***END of NPS Prediction Component****************************************
'    '**for field pacement test
'      MsgBox.Info(theTotGE.asString++theFldTotGE.asString++theSedYld.asString,"")



Return theSedYld
```

B.10 Calcarea.ave

```
'Calculates area, converting from meters (map unit) to hectares.
'Script is a customization of ArcView 3.2 Sample Script, "CalcAcre"

'**Edited from ArcView Sample Script: CalcAcre *******************
' Calculates area in hectares for polygon themes.
' *Assumes View map units are set to meters.*
' The script will add an "Area" field to the theme if it does not exist.
' If the field exists, the values are recalculated.

'Get proper view and theme from script call
  theView = self.Get(0)
'  thetheme = theView.FindTheme((self.Get(1)).AsString)
  thetheme = theView.FindTheme("SoilbyFld.shp")
  theFTab = thetheme.GetFTab

'Get the view's projection, if any.
  thePrj = theView.GetProjection
  if (thePrj.IsNull) then
    hasPrj = false
  else
    hasPrj = true
  end

'If you can't edit the theme inform the user.
  if (theFTab.CanEdit.Not) then
   MsgBox.Info("Cannot edit table for theme:"++thetheme.AsString,"")
  end

'Make the FTAB editable.
  theFTab.SetEditable(TRUE)
```

```
'Check for existence of "Area" field; If it does not exist, create it.
  if (theFTab.FindField("Area") = nil) then
   theAreaField = Field.Make("Area",#FIELD_DOUBLE,16,3)
   theFTab.AddFields({theAreaField})
  else
   ok = MsgBox.YesNo("Update Area for"++thetheme.getName+"?", "Calculate", true)
   theAreaField = theFTab.FindField("Area")
  end

'Loop through the FTAB and find the projected area and perimeter of each
'shape and set the field values appropriately.
  theShape = theFTab.ReturnValue(theFTab.FindField("shape"),0)
  For Each rec in theFTab
    theFTab.QueryShape(rec,thePrj,theShape)
    theArea = (theShape.ReturnArea)/10000   '(convert from meters to hectares)
    theFTab.SetValue(theAreaField,rec,theArea)
  end

'Make the FTAB not editable.
  theFTab.SetEditable(FALSE)
Return Nil
```

B.11 Intersec.ave

```
'Intersects Soil and Field shapefiles;
'edited from ArcView3.2 System Script, "GeoProc.Intersect.Finish"

'***Name modifications to customize script for Optimization program***
    theView = self.Get(0)   'av.GetActiveDoc
    Theme1  = self.Get(1)
    Theme2  = self.Get(2)
    outFName = self.Get(3)

'**Below is same as system script except a few unneeded parts cut out (no name changes)***
countq=0
geowait=av.finddialog("GeoWait")
geowait.FindByName("changeme").SetLabel("Intersecting 2 themes")
geowait.open

Tab1 = Theme1.GetFTab
Tab2 = Theme2.GetFTab
Tab1shpField = Tab1.FindField("shape")
Tab2shpField = Tab2.FindField("shape")

shapeType = Tab1.FindField("Shape").GetType
if (shapeType = #FIELD_SHAPEPOLY) then
  outClass = POLYGON
else
  MsgBox.Error("Invalid shape field type.", "Theme Intersection Error")
  return nil
end

OutputFTab = FTab.MakeNew( outFName, outClass )
Theme1Fields = {}
Theme2Fields = {}
```

```
theme1Fields2={}
'---------------------
' Get the list of corrected dbase field names for the overlay theme
Theme2Fielddict=dictionary.make(Tab2.getFields.count)
for each f in Tab2.getFields
  if (f.getname = "Shape") then
    continue
  end
  thecopy=f.clone
  therealname=f.getname

  fldtest=Theme2Fielddict.get(thecopy.getname)
  if (fldtest <> NIL) then
    while (fldtest <> NIL)
      thecopy.setname(thecopy.getname.left(9)+"_")
      fldtest=Theme2Fielddict.get(thecopy.getname)
      countq=countq+1
      if (countq=10) then
        break
      end
    end
  end
  thecopy.setalias(thecopy.getname)
  Theme2Fields.add(thecopy)
  Theme2fieldDict.add(thecopy.getname,{therealname,"Theme2",thecopy})
end 'f

' Get the list of corrected dbase field names for the input theme
Theme1FieldDict=dictionary.make(Tab1.getFields.count)
for each f in Tab1.getFields
  if (f.getname = "Shape") then
    continue
  end
```

```
  thecopy=f.clone
  therealname=f.getname
  'getname
  thecopy.setalias(thecopy.getname)
  Theme1Fields.add(thecopy)
  Theme1FieldDict.add(thecopy.getname,{therealname,"Theme1",thecopy})
end 'f

Theme1FieldDictCopy=Theme1FieldDict.clone
for each akey in Theme1FieldDict.returnKeys
  if (Theme2FieldDict.get(akey)<> NIL) then
    test=Theme2FieldDict.get(akey)
    oldvalue=Theme1FieldDict.get(akey)
    countq=0
    while (test<>NIL)
      newname=akey.left(9)+"_"
      test=Theme2FieldDict.get(newname)
      countq=countq+1
      if (countq=10) then
        break
      end

    end
    for each afld in Theme1Fields
      if (afld.getname = aKey) then
        found=afld
        break
      end
    end
    anIndex=Theme1Fields.find(found)
    avalue=Theme1Fields.get(anIndex)
    avalue.setname(newname)
    Theme1Fields.remove(anIndex)
```

```
    Theme1Fields.add(avalue)
    Theme1FieldDict.remove(akey)
    Theme1FieldDict.add(newname,{oldvalue.get(0),oldvalue.get(1),avalue})
  end
end ' akey

''''''''''''''''''''''''''''''''''
if (Theme1Fields.Count > 0) then
  OutputFTab.AddFields(Theme1Fields)
end
if (Theme2Fields.Count > 0) then
  OutputFTab.AddFields(Theme2Fields)
end
outshpfld = OutputFtab.findfield("Shape")

Tab1_oldselection = Tab1.getselection.clone
Tab2_oldselection = Tab2.GetSelection.Clone

  if (Tab1.getnumselrecords<>0) then
    Theme1.clearselection
  end
  Theme1.getftab.getselection.setall
  Theme1.getftab.updateselection
  Tab1Records = Tab1
  nrecords=Tab1.GetNumRecords

  if (Tab2.getnumselrecords<>0) then
    Theme2.clearselection
  end
  Theme2.getftab.getselection.setall
  Theme2.getftab.updateselection
  Tab2Records = Theme2.getftab
  nrecords2 = Tab2.GetNumRecords
```

```
  SelType2= #VTAB_SELTYPE_NEW
  numSelected2 = 0

OutputFtab.seteditable(False)
OutputFtab.seteditable(True)

''''''''''''''
'' Create the temp file
'-----------------------------------------------
' First pass will get all parts of tab1
count=0
av.showmsg("Processing, on First Pass")
for each aRecord in tab1records
  count=count+1
  test=av.SetStatus(count/nrecords  * 100)
  av.showstopbutton
  if (test=FALSE) then
    geowait=av.finddialog("GeoWait")
    geowait.close
    return(FALSE)
  end

   ' get the mainshape
   themainshape=tab1.returnvalue(tab1shpfield, arecord)

    for each ashape in themainshape.explode
      theleftover=ashape.clone
      ashape=ashape.clean
      if (SelType2 = #VTAB_SELTYPE_AND) then
        tab2.setselection(Tab2_oldselection)
        tab2.updateselection
      end
      if (theView.getprojection.isNULL) then
```

```
      theme2.selectbyshapes({ashape},Seltype2)
    else
      theme2.selectbyshapes({ashape}.returnProjected(theView.getprojection),Seltype2)
    end
    tab2.updateselection

  '  msgbox.report(ashape.asstring,"")

  '  if
((msgbox.yesno(tab2.getnumselrecords.asstring+nl+SelType2.asstring,theme2.getname,False
))) then
  '     return(nil)
  '   end

    if (tab2.getnumselrecords >= numSelected2) then  ' tests if any more records were
selected
      ' this piece intersects some
      for each interRecord in tab2.getselection
        intersectedPiece=tab2.returnvalue(tab2shpfield,interRecord)
        theIntersection=ashape.returnintersection(intersectedPiece)
        if (theIntersection.isNull.NOT) then
          if (theIntersection = ashape) then
            ' the 2 pieces are exactly the same
            ' Write it out
            outrecord=outputftab.addrecord

            outputftab.setvalue(outshpfld,outrecord,ashape)

            for each akey in Theme1FieldDict.returnkeys
              therealname=Theme1FieldDict.get(akey).get(0)
              theThemeType=Theme1FieldDict.get(akey).get(1)
              theSRCfield=Theme1FieldDict.get(akey).get(2)
              if (theThemeType="Theme1") then
```

```
        thevalue=tab1.returnvalue(tab1.findfield(therealname),arecord)
        outputftab.setvalue(theSRCField,outrecord,thevalue)
      end 'if theme1
    end

    for each akey in Theme2FieldDict.returnkeys
      therealname=Theme2FieldDict.get(akey).get(0)
      theThemeType=Theme2FieldDict.get(akey).get(1)
      theSRCfield=Theme2FieldDict.get(akey).get(2)

      if (theThemeType="Theme2") then
        thevalue=tab2.returnvalue(tab2.findfield(therealname),interRecord)
        outputftab.setvalue(theSRCField,outrecord,thevalue)
      end 'if theme1

    end 'for each akey

    continue
  end ' the shape are equal

  ' WRITE out the INTERSECTION to  Output
    ' Write it out to TMP and output
    outrecord=outputftab.addrecord
    outputftab.setvalue(outshpfld,outrecord,theIntersection)
    for each akey in Theme1FieldDict.returnkeys
      therealname=Theme1FieldDict.get(akey).get(0)
      theThemeType=Theme1FieldDict.get(akey).get(1)
      theSRCfield=Theme1FieldDict.get(akey).get(2)
      if (theThemeType="Theme1") then
        thevalue=tab1.returnvalue(tab1.findfield(therealname),arecord)
        outputftab.setvalue(theSRCfield,outrecord,thevalue)
      end 'if theme1
    end
```

```
       for each akey in Theme2FieldDict.returnkeys
        therealname=Theme2FieldDict.get(akey).get(0)
        theThemeType=Theme2FieldDict.get(akey).get(1)
        theSRCfield=Theme2FieldDict.get(akey).get(2)

        if (theThemeType="Theme2") then
         thevalue=tab2.returnvalue(tab2.findfield(therealname),interRecord)
         outputftab.setvalue(theSRCfield,outrecord,thevalue)
        end 'if theme1

       end 'for each akey
     end ' is NOT NULL
    end ' intersectedPiece
   end ' test for selection
  end ' ashape
end 'aRecord in tab1

outputftab.seteditable(false)
tab2.setselection(Tab2_oldselection)
tab2.updateselection
tab1.setselection(Tab1_oldselection)
tab1.updateselection
theNewTheme = FTheme.Make( OutputFTab )
theView.AddTheme( theNewTheme )
for each afld in outputftab.getfields
 afld.setalias(afld.getname)
end

geowait.close
av.clearstatus
av.purgeobjects
return outFName
```

B.12 MakeGtheme.ave

'makes Gtheme from grid

```
theGrid = self.Get(0)
theGTheme = GTheme.Make(theGrid)
' check if output is ok
if (theGrid.HasError) then
  return NIL
else
  ' set name of theme
  theGTheme.SetName(self.Get(1))
  ' add theme to the view
  return theGTheme
end
```

Appendix C: Answers-2000 input files

Appendix C contains the main input files used in the Answers-2000 simulation runs. The weather file is not included here. Although only one main input file for each watershed is given, this file contains the complete set of information used for all evaluation runs. The input files are edited as indicated, due to repetition within and length of the cell-level part of these files.

C.1 Mini-Muddy Creek main input file

```
  Mini-Muddy sed yld test - ctcrn-wht (5)
METRIC  UNITS ARE USED ON INPUT/OUTPUT                PRINT
STORM BY STORM OUTPUT = 1
EXTRA OUTPUT ON DAYS  =
PRINT HYDROGRAPHS = 00
RAINFALL DATA FOR 1 RAINGAGES
BEGINNING JULIAN DAY OF SIMULATION 001 1980
DURATION OF SIMULATION DAYS 1827
GAUGE NUMBER    1
 SIMULATION CONSTANTS FOLLOW
NUMBER OF LINES OF HYDROGRAPH OUTPUT =0101
TIME INCREMENT =030.0 SECONDS
INFILTRATION CAPACITY CALCULATED EVERY00030 SECONDS
EXPECTED RUNOFF PEAK =0050.00 MM/HR
 SOIL INFILTRATION, DRAINAGE AND GROUNDWATER CONSTANTS FOLLOW
NUMBER OF SOILS =0027
S01, TP =.51, FP =.55, FC =00.27, A =1.000, DF =228.6, ASM =.55
CONDUCTIVITY OPTION = 0
15.0 54.0 21.0 2.50 05.0 09.0
S02, TP =.50, FP =.24, FC =00.12, A =1.000, DF =152.4, ASM =.24
CONDUCTIVITY OPTION = 0
11.5 31.1 02.4 1.25 22.5 01.1
S03, TP =.48, FP =.83, FC =00.42, A =1.000, DF =228.6, ASM =.83
CONDUCTIVITY OPTION = 0
20.0 23.8 43.7 1.50 02.5 18.8
S04, TP =.48, FP =.83, FC =00.42, A =1.000, DF =228.6, ASM =.83
CONDUCTIVITY OPTION = 0
20.0 23.8 43.7 1.50 02.5 18.8
S05, TP =.48, FP =.83, FC =00.42, A =1.000, DF =228.6, ASM =.83
CONDUCTIVITY OPTION = 0
20.0 23.8 43.7 1.50 02.5 18.8
S06, TP =.48, FP =.83, FC =00.42, A =1.000, DF =228.6, ASM =.83
CONDUCTIVITY OPTION = 0
20.0 23.8 43.7 1.50 02.5 18.8
S07, TP =.45, FP =.71, FC =00.36, A =1.000, DF =228.6, ASM =.71
CONDUCTIVITY OPTION = 0
33.5 26.8 27.2 1.25 02.5 11.8
S08, TP =.48, FP =.67, FC =00.33, A =1.000, DF =228.6, ASM =.67
CONDUCTIVITY OPTION = 0
20.0 21.5 21.0 1.50 07.5 09.0
S09, TP =.48, FP =.67, FC =00.33, A =1.000, DF =228.6, ASM =.67
CONDUCTIVITY OPTION = 0
20.0 21.5 21.0 1.50 07.5 09.0
S10, TP =.46, FP =.43, FC =00.22, A =1.000, DF =228.6, ASM =.43
```

```
CONDUCTIVITY OPTION = 0
20.0 20.8 01.7 1.50 07.5 00.8
S11, TP =.48, FP =.83, FC =00.42, A =1.000, DF =228.6, ASM =.83
CONDUCTIVITY OPTION = 0
20.0 25.1 34.9 1.50 10.0 15.1
S12, TP =.48, FP =.83, FC =00.42, A =1.000, DF =228.6, ASM =.83
CONDUCTIVITY OPTION = 0
20.0 25.1 34.9 1.50 10.0 15.1
S13, TP =.48, FP =.83, FC =00.42, A =1.000, DF =228.6, ASM =.83
CONDUCTIVITY OPTION = 0
20.0 25.1 34.9 1.50 10.0 15.1
S14, TP =.47, FP =.91, FC =00.46, A =1.000, DF =254.0, ASM =.91
CONDUCTIVITY OPTION = 0
20.0 28.1 41.9 2.00 05.0 18.1
S15, TP =.51, FP =.41, FC =00.21, A =1.000, DF =152.4, ASM =.41
CONDUCTIVITY OPTION = 0
13.5 40.0 11.5 2.50 12.5 05.0
S16, TP =.51, FP =.41, FC =00.21, A =1.000, DF =152.4, ASM =.41
CONDUCTIVITY OPTION = 0
13.5 40.0 11.5 2.50 12.5 05.0
S17, TP =.51, FP =.47, FC =00.24, A =1.000, DF =152.4, ASM =.47
CONDUCTIVITY OPTION = 0
18.5 25.2 23.8 2.50 10.0 10.2
S18, TP =.51, FP =.47, FC =00.24, A =1.000, DF =152.4, ASM =.47
CONDUCTIVITY OPTION = 0
18.5 25.2 23.8 2.50 10.0 10.2
S19, TP =.51, FP =.47, FC =00.24, A =1.000, DF =152.4, ASM =.47
CONDUCTIVITY OPTION = 0
18.5 25.2 23.8 2.50 10.0 10.2
S20, TP =.51, FP =.39, FC =00.20, A =1.000, DF =152.4, ASM =.39
CONDUCTIVITY OPTION = 0
17.0 23.2 24.8 3.00 12.5 10.7
S21, TP =.47, FP =.89, FC =00.45, A =1.000, DF =177.8, ASM =.89
CONDUCTIVITY OPTION = 0
26.5 20.6 47.9 3.00 02.5 20.6
S22, TP =.48, FP =.83, FC =00.42, A =1.000, DF =228.6, ASM =.83
CONDUCTIVITY OPTION = 0
20.0 23.8 43.7 1.50 02.5 18.8
S23, TP =.48, FP =.83, FC =00.42, A =1.000, DF =228.6, ASM =.83
CONDUCTIVITY OPTION = 0
20.0 23.8 43.7 1.50 02.5 18.8
S24, TP =.47, FP =.79, FC =00.39, A =1.000, DF =228.6, ASM =.79
CONDUCTIVITY OPTION = 0
21.0 30.0 46.5 1.25 00.1 20.0
S25, TP =.47, FP =.66, FC =00.33, A =1.000, DF =203.2, ASM =.66
CONDUCTIVITY OPTION = 0
15.5 40.9 31.1 2.00 05.0 13.4
S26, TP =.51, FP =.43, FC =00.22, A =1.000, DF =177.8, ASM =.43
CONDUCTIVITY OPTION = 0
21.0 20.0 11.5 2.50 00.1 05.0
S27, TP =.51, FP =.43, FC =00.22, A =1.000, DF =177.8, ASM =.43
CONDUCTIVITY OPTION = 0
21.0 20.0 11.5 2.00 00.1 05.0
 PARTICLE SIZE AND TRANSPORT DATA FOLLOWS
NUMBER OF PARTICLE SIZE CLASSES  = 05
NUMBER OF WASH LOAD CLASSES      = 01
```

```
  SIZE      SPECIFIC GRAVITY  FALL VELOCITY
000000.0020000000000000002.6500000000.0000030
000000.0100000000000000002.6500000000.0000800
000000.2000000000000000002.6400000000.0240000
000000.0300000000000000001.8000000000.0003500
000000.5000000000000000001.6000000000.0400000
00.15000.21000.54000.09000.050 S01
00.11500.02400.31100.01100.225 S02
00.20000.43700.23800.18800.025 S03
00.20000.43700.23800.18800.025 S04
00.20000.43700.23800.18800.025 S05
00.20000.43700.23800.18800.025 S06
00.33500.27200.26800.11800.025 S07
00.20000.21000.21500.09000.075 S08
00.20000.21000.21500.09000.075 S09
00.20000.01700.20800.00800.075 S10
00.20000.34900.25100.15100.100 S11
00.20000.34900.25100.15100.100 S12
00.20000.34900.25100.15100.100 S13
00.20000.41900.28100.18100.050 S14
00.13500.11500.40000.05000.125 S15
00.13500.11500.40000.05000.125 S16
00.18500.23800.25200.10200.100 S17
00.18500.23800.25200.10200.100 S18
00.18500.23800.25200.10200.100 S19
00.17000.24800.23200.10700.125 S20
00.26500.47900.20600.20600.025 S21
00.20000.43700.23800.18800.025 S22
00.20000.43700.23800.18800.025 S23
00.21000.46500.30000.20000.001 S24
00.15500.31100.40900.13400.050 S25
00.21000.11500.20000.05000.001 S26
00.21000.11500.20000.05000.001 S27
003.8670020.0000004.0000000.0500
002.4116020.0000004.0000000.0500
005.7599020.0000004.0000000.0500
005.7599020.0000004.0000000.0500
005.7599020.0000004.0000000.0500
005.7599020.0000004.0000000.0500
007.8014020.0000004.0000000.0500
004.8508020.0000004.0000000.0500
004.8508020.0000004.0000000.0500
004.0784020.0000004.0000000.0500
005.4086020.0000004.0000000.0500
005.4086020.0000004.0000000.0500
005.4086020.0000004.0000000.0500
005.6901020.0000004.0000000.0500
003.1800020.0000004.0000000.0500
003.1800020.0000004.0000000.0500
004.6646020.0000004.0000000.0500
004.6646020.0000004.0000000.0500
004.6646020.0000004.0000000.0500
004.4036020.0000004.0000000.0500
007.2263020.0000004.0000000.0500
005.7599020.0000004.0000000.0500
005.7599020.0000004.0000000.0500
```

```
006.0750020.0000004.0000000.0500
004.3645020.0000004.0000000.0500
004.6700020.0000004.0000000.0500
004.6700020.0000004.0000000.0500
DRAINAGE EXPONENT =03
DRAINAGE COEFFICIENT FOR TILE DRAINS =09.55 MM/24HR
GROUNDWATER RELEASE FRACTION =000000.005
FERTILIZER APPLIED =00
 IMPOUNDMENT SPECIFICATIONS FOLLOW
NUMBER OF IMPOUNDMENTS = 00
 SURFACE ROUGHNESS AND CROP CONSTANTS FOLLOWS
NUMBER OF CROPS AND SURFACES =012
C01,    Pasture ,    00.40    0.96    0.55    050.00    3.000
095.0 005.0 000.8 008.0 002.0 085.0 0.07 0.07 0.04
0.00 0.70 1.80 3.00 3.00 3.00 2.90 2.70 1.96 0.90 0.50
001 365 0.00 00.000 00.00 00000.0 100 3.00
010.0 0.085 0.070 00.50 01.00 0.040 0.050 01 00
C02,    Hay    ,    00.80    0.96    0.45    030.00    3.000
096.0 004.0 001.0 010.0 002.0 099.9 0.07 0.07 0.04
0.00 0.15 0.40 1.90 2.60 3.00 2.96 2.92 2.30 1.15 0.50
001 365 2.30 -0.208 02.25 03020.0 120 3.00
005.0 0.085 0.450 00.50 01.00 0.050 0.200 01 00
C03,    Corn-Sil,    01.10    0.90    0.60    076.20    5.000
070.0 030.0 060.0 020.0 000.0 005.0 0.20 0.20 0.10
0.00 0.09 0.20 0.32 0.55 1.30 3.00 3.00 2.90 2.00 0.00
121 250 0.40 -0.548 01.35 44800.0 1200 3.00
043.0 0.336 2.400 00.50 01.00 0.150 0.200 00 00
C04,    Forest ,    02.00    0.95    0.50    090.00    3.500
095.0 005.0 000.8 010.0 002.0 095.0 0.25 0.20 0.13
2.50 2.50 4.50 4.50 4.50 4.50 4.50 4.50 4.50 2.50 2.50
001 365 1.30 -0.264 02.50 09400.0 900 4.50
005.0 0.000 3.000 00.50 01.00 0.100 0.200 01 00
C05,    Imp    ,    00.01    0.01    0.10    000.05    5.000
001.0 099.0 001.0 100.0 002.0 099.9 0.00 0.00 0.00
1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
001 365 0.00 00.000 00.00 00000.0 000 1.00
001.0 0.010 0.010 00.01 01.00 0.012 0.013 00 01
C06,    fallow ,    00.01    0.10    0.45    038.60    7.000
020.0 080.0 010.0 040.0 040.0 040.0 0.05 0.05 0.02
0.00 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01
092 120 0.00 00.000 00.00 00000.0 000 0.01
043.0 0.050 0.000 00.01 01.00 0.080 0.100 00 00
C07,    W-wht ,    00.65    0.10    0.55    063.50    7.000
080.0 020.0 060.0 018.0 002.0 060.0 0.22 0.16 0.08
0.00 0.47 0.90 0.90 0.90 0.90 1.62 3.00 3.00 3.00 0.00
251 091 1.00 -0.301 02.50 03360.0 400 3.00
043.0 0.500 0.750 00.50 01.00 0.080 0.100 00 00
C08,    CTwinfal,    00.01    0.10    0.52    050.80    1.000
005.0 095.0 003.0 080.0 005.0 005.0 0.05 0.05 0.02
0.00 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01
251 120 0.00 00.000 00.00 00000.0 000 0.01
043.0 0.050 0.000 00.01 01.00 0.120 0.180 00 00
C09,    MTcrnsil,    01.10    0.90    0.50    050.80    3.000
070.0 030.0 045.0 010.0 010.0 050.0 0.20 0.20 0.10
0.00 0.09 0.20 0.32 0.55 1.30 3.00 3.00 2.90 2.00 0.00
121 250 0.40 -0.548 01.35 44800.0 1200 3.00
```

```
030.0 0.336 2.400 00.50 01.00 0.070 0.120 01 00
C10,     ctcrn-w ,    01.10     0.90     0.60    076.20    1.000
070.0 030.0 060.0 020.0 000.0 005.0 0.20 0.20 0.10
0.00 0.09 0.20 0.32 0.55 1.30 3.00 3.00 2.90 2.00 0.00
121 250 0.40 -0.548 01.35 44800.0 1200 3.00
043.0 0.336 2.400 00.50 01.00 0.150 0.200 00 00
C11,     mtcrn-w ,    01.10     0.90     0.50    050.80    3.000
070.0 030.0 045.0 010.0 010.0 050.0 0.20 0.20 0.10
0.00 0.09 0.20 0.32 0.55 1.30 3.00 3.00 2.90 2.00 0.00
121 250 0.40 -0.548 01.35 44800.0 1200 3.00
030.0 0.336 2.400 00.50 01.00 0.070 0.120 01 00
C12,     MTwinfal,    00.01     0.10     0.35    038.60    3.000
005.0 095.0 000.0 040.0 050.0 050.0 0.05 0.05 0.02
0.00 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01
251 120 0.00 00.000 00.00 00000.0 000 0.01
030.0 0.050 0.000 00.01 01.00 0.050 0.080 00 00
NUMBER OF ALL ROTATIONS      =008
01 04 1980365 04 1981365 04 1982365 04 1983365 04 1984365 04 1985365 04 1986365
   04 1987365 04 1988365 04 1989365 04 1990365
```

{42 blank lines excluded}

```
02 05 1980365 05 1981365 05 1982365 05 1983365 05 1984365 05 1985365 05 1986365
   05 1987365 05 1988365 05 1989365 05 1990365
```

{42 blank lines excluded}

```
03 08 1980120 03 1980250 08 1981120 03 1981250 08 1982120 03 1982250 08 1983120
   03 1983250 08 1984120 03 1984250 08 1985120 03 1985250 08 1986120 03 1986250
   08 1987120 03 1987250 08 1988120 03 1988250 08 1989120 03 1989250 08 1990120
   03 1990250
```

{38 blank lines excluded}

```
04 12 1980120 09 1980250 12 1981120 09 1981250 12 1982120 09 1982250 12 1983120
   09 1983250 12 1984120 09 1984250 12 1985120 09 1985250 12 1986120 09 1986250
   12 1987120 09 1987250 12 1988120 09 1988250 12 1989120 09 1989250 12 1990120
   09 1990250
```

{38 blank lines excluded}

```
05 07 1980091 06 1980120 10 1980250 07 1981091 06 1981120 10 1981250 07 1982091
   06 1982120 10 1982250 07 1983091 06 1983120 10 1983250 07 1984091 06 1984120
   10 1984250 07 1985091 06 1985120 10 1985250 07 1986091 06 1986120 10 1986250
   07 1987091 06 1987120 10 1987250 07 1988091 06 1988120 10 1988250 07 1989091
   06 1989120 10 1989250 07 1990091 06 1990120 10 1990250
```

{37 blank lines excluded}

```
06 07 1980091 06 1980120 11 1980250 07 1981091 06 1981120 11 1981250 07 1982091
   06 1982120 11 1982250 07 1983091 06 1983120 11 1983250 07 1984091 06 1984120
   11 1984250 07 1985091 06 1985120 11 1985250 07 1986091 06 1986120 11 1986250
   07 1987091 06 1987120 11 1987250 07 1988091 06 1988120 11 1988250 07 1989091
   06 1989120 11 1989250 07 1990091 06 1990120 11 1990250
```

{37 blank lines excluded}

```
 07 02 1980365 02 1981365 02 1982365 02 1983365 02 1984365 02 1985365 02 1986365
    02 1987365 02 1988365 02 1989365 02 1990365
```

{42 blank lines excluded}

```
 08 01 1980365 01 1981365 01 1982365 01 1983365 01 1984365 01 1985365 01 1986365
    01 1987365 01 1988365 01 1989365 01 1990365
```

{42 blank lines excluded}

```
  CHANNEL SPECIFICATIONS FOLLOW
 NUMBER OF CHANNEL NETWORKS  =001
 NUMBER OF TYPES OF CHANNELS =003
 CHAN01 WID =01.0(m),  SOIL N =00.045 CHAN N =00.055 0.01 1.00
 CHAN02 WID =01.5(m),  SOIL N =00.030 CHAN N =00.040 0.01 1.00
 CHAN03 WID =02.0(m),  SOIL N =00.020 CHAN N =00.030 0.01 1.00
  ELEMENT SPECIFICATIONS FOR BASELINE SENSITIVITY ANALYSIS
 EACH ELEMENT IS0030.00m. SQUARE
 NETWORK 1 OUTFLOW FROM ROW0174 COLUMN 0154     07623
 44120 0103 312       2   1    1  0     0  0   0   0    4610   46  184    4
  262    6  808    31
 45119 0114 307       2   1    1  0     0  0   0   0    4610   46  184    4
  262    6  808    31
```

{remainder of cell-level input excluded}

C.2 Lola Run main input file

```
  Lola Sed yld test -CTcrnwht(5)
METRIC  UNITS ARE USED ON INPUT/OUTPUT              PRINT
STORM BY STORM OUTPUT = 1
EXTRA OUTPUT ON DAYS  =
PRINT HYDROGRAPHS = 00
RAINFALL DATA FOR 1 RAINGAGES
BEGINNING JULIAN DAY OF SIMULATION 001 1980
DURATION OF SIMULATION DAYS 1827
GAUGE NUMBER    1
 SIMULATION CONSTANTS FOLLOW
NUMBER OF LINES OF HYDROGRAPH OUTPUT =0101
TIME INCREMENT =030.0 SECONDS
INFILTRATION CAPACITY CALCULATED EVERY00030 SECONDS
EXPECTED RUNOFF PEAK =0050.00 MM/HR
 SOIL INFILTRATION, DRAINAGE AND GROUNDWATER CONSTANTS FOLLOW
NUMBER OF SOILS =0034
S01, TP =.51, FP =.55, FC =00.27, A =1.000, DF =228.6, ASM =.55
CONDUCTIVITY OPTION = 0
15.0 54.0 21.0 2.50 05.0 09.0
S02, TP =.50, FP =.24, FC =00.12, A =1.000, DF =152.4, ASM =.24
CONDUCTIVITY OPTION = 0
11.5 31.1 02.4 1.25 22.5 01.1
S03, TP =.51, FP =.76, FC =00.38, A =1.000, DF =177.8, ASM =.76
CONDUCTIVITY OPTION = 0
17.5 35.6 41.9 1.25 00.1 18.1
S04, TP =.48, FP =.83, FC =00.42, A =1.000, DF =228.6, ASM =.83
CONDUCTIVITY OPTION = 0
20.0 23.8 43.7 1.50 02.5 18.8
S05, TP =.48, FP =.83, FC =00.42, A =1.000, DF =228.6, ASM =.83
CONDUCTIVITY OPTION = 0
20.0 23.8 43.7 1.50 02.5 18.8
S06, TP =.48, FP =.83, FC =00.42, A =1.000, DF =228.6, ASM =.83
CONDUCTIVITY OPTION = 0
20.0 23.8 43.7 1.50 02.5 18.8
S07, TP =.48, FP =.83, FC =00.42, A =1.000, DF =228.6, ASM =.83
CONDUCTIVITY OPTION = 0
20.0 23.8 43.7 1.50 02.5 18.8
S08, TP =.51, FP =.43, FC =00.22, A =1.000, DF =228.6, ASM =.43
CONDUCTIVITY OPTION = 0
21.0 41.5 15.0 2.50 10.0 06.5
S09, TP =.48, FP =.67, FC =00.33, A =1.000, DF =228.6, ASM =.67
CONDUCTIVITY OPTION = 0
20.0 21.5 21.0 1.50 07.5 09.0
S10, TP =.48, FP =.83, FC =00.42, A =1.000, DF =228.6, ASM =.83
CONDUCTIVITY OPTION = 0
20.0 25.1 34.9 1.50 10.0 15.1
S11, TP =.48, FP =.83, FC =00.42, A =1.000, DF =228.6, ASM =.83
CONDUCTIVITY OPTION = 0
20.0 25.1 34.9 1.50 10.0 15.1
S12, TP =.48, FP =.83, FC =00.42, A =1.000, DF =228.6, ASM =.83
CONDUCTIVITY OPTION = 0
20.0 25.1 34.9 1.50 10.0 15.1
S13, TP =.48, FP =.83, FC =00.42, A =1.000, DF =228.6, ASM =.83
```

CONDUCTIVITY OPTION = 0
20.0 25.1 34.9 1.50 10.0 15.1
S14, TP =.48, FP =.83, FC =00.42, A =1.000, DF =228.6, ASM =.83
CONDUCTIVITY OPTION = 0
20.0 25.1 34.9 1.50 10.0 15.1
S15, TP =.47, FP =.91, FC =00.46, A =1.000, DF =254.0, ASM =.91
CONDUCTIVITY OPTION = 0
20.0 28.1 41.9 2.00 05.0 18.1
S16, TP =.51, FP =.41, FC =00.21, A =1.000, DF =152.4, ASM =.41
CONDUCTIVITY OPTION = 0
13.5 40.0 11.5 2.50 12.5 05.0
S17, TP =.51, FP =.41, FC =00.21, A =1.000, DF =152.4, ASM =.41
CONDUCTIVITY OPTION = 0
13.5 40.0 11.5 2.50 12.5 05.0
S18, TP =.51, FP =.47, FC =00.24, A =1.000, DF =152.4, ASM =.47
CONDUCTIVITY OPTION = 0
18.5 25.2 23.8 2.50 10.0 10.2
S19, TP =.51, FP =.39, FC =00.20, A =1.000, DF =152.4, ASM =.39
CONDUCTIVITY OPTION = 0
17.0 23.2 24.8 3.00 12.5 10.7
S20, TP =.51, FP =.39, FC =00.20, A =1.000, DF =152.4, ASM =.39
CONDUCTIVITY OPTION = 0
17.0 23.2 24.8 3.00 12.5 10.7
S21, TP =.51, FP =.82, FC =00.41, A =1.000, DF =457.2, ASM =.82
CONDUCTIVITY OPTION = 0
18.5 39.8 34.2 3.00 02.5 14.8
S22, TP =.48, FP =.83, FC =00.42, A =1.000, DF =228.6, ASM =.83
CONDUCTIVITY OPTION = 0
20.0 23.8 43.7 1.50 02.5 18.8
S23, TP =.47, FP =.89, FC =00.45, A =1.000, DF =177.8, ASM =.89
CONDUCTIVITY OPTION = 0
26.5 20.6 47.9 3.00 02.5 20.6
S24, TP =.48, FP =.83, FC =00.42, A =1.000, DF =228.6, ASM =.83
CONDUCTIVITY OPTION = 0
20.0 23.8 43.7 1.50 02.5 18.8
S25, TP =.48, FP =.83, FC =00.42, A =1.000, DF =228.6, ASM =.83
CONDUCTIVITY OPTION = 0
20.0 23.8 43.7 1.50 02.5 18.8
S26, TP =.48, FP =.83, FC =00.42, A =1.000, DF =228.6, ASM =.83
CONDUCTIVITY OPTION = 0
20.0 23.8 43.7 1.50 02.5 18.8
S27, TP =.47, FP =.79, FC =00.39, A =1.000, DF =228.6, ASM =.79
CONDUCTIVITY OPTION = 0
21.0 30.0 46.5 1.25 00.1 20.0
S28, TP =.49, FP =.41, FC =00.20, A =1.000, DF =152.4, ASM =.41
CONDUCTIVITY OPTION = 0
14.0 23.6 19.9 3.00 07.5 08.6
S29, TP =.49, FP =.41, FC =00.20, A =1.000, DF =152.4, ASM =.41
CONDUCTIVITY OPTION = 0
14.0 23.6 19.9 3.00 07.5 08.6
S30, TP =.47, FP =.62, FC =00.31, A =1.000, DF =177.8, ASM =.62
CONDUCTIVITY OPTION = 0
14.0 40.6 30.4 1.25 07.5 13.1
S31, TP =.47, FP =.66, FC =00.33, A =1.000, DF =203.2, ASM =.66
CONDUCTIVITY OPTION = 0
15.5 40.9 31.1 2.00 05.0 13.4

S32, TP =.51, FP =.43, FC =00.22, A =1.000, DF =177.8, ASM =.43
CONDUCTIVITY OPTION = 0
21.0 20.0 11.5 2.00 00.1 05.0
S33, TP =.51, FP =.43, FC =00.22, A =1.000, DF =177.8, ASM =.43
CONDUCTIVITY OPTION = 0
21.0 20.0 11.5 2.00 00.1 05.0
S34, TP =.51, FP =.59, FC =00.29, A =1.000, DF =177.8, ASM =.59
CONDUCTIVITY OPTION = 0
18.5 21.2 20.3 0.25 15.0 08.7
 PARTICLE SIZE AND TRANSPORT DATA FOLLOWS
NUMBER OF PARTICLE SIZE CLASSES = 05
NUMBER OF WASH LOAD CLASSES   = 01
  SIZE   SPECIFIC GRAVITY FALL VELOCITY
000000.0020000000000000002.6500000000.0000030
000000.0100000000000000002.6500000000.0000800
000000.2000000000000000002.6400000000.0240000
000000.0300000000000000001.8000000000.0003500
000000.5000000000000000001.6000000000.0400000
00.15000.21000.54000.09000.050 S01
00.11500.02400.31100.01100.225 S02
00.17500.41900.35600.18100.001 S03
00.20000.43700.23800.18800.025 S04
00.20000.43700.23800.18800.025 S05
00.20000.43700.23800.18800.025 S06
00.20000.43700.23800.18800.025 S07
00.21000.15000.41500.06500.100 S08
00.20000.21000.21500.09000.075 S09
00.20000.34900.25100.15100.100 S10
00.20000.34900.25100.15100.100 S11
00.20000.34900.25100.15100.100 S12
00.20000.34900.25100.15100.100 S13
00.20000.34900.25100.15100.100 S14
00.20000.41900.28100.18100.050 S15
00.13500.11500.40000.05000.125 S16
00.13500.11500.40000.05000.125 S17
00.18500.23800.25200.10200.100 S18
00.17000.24800.23200.10700.125 S19
00.17000.24800.23200.10700.125 S20
00.18500.34200.39800.14800.025 S21
00.20000.43700.23800.18800.025 S22
00.26500.47900.20600.20600.025 S23
00.20000.43700.23800.18800.025 S24
00.20000.43700.23800.18800.025 S25
00.20000.43700.23800.18800.025 S26
00.21000.46500.30000.20000.001 S27
00.14000.19900.23600.08600.075 S28
00.14000.19900.23600.08600.075 S29
00.14000.30400.40600.13100.075 S30
00.15500.31100.40900.13400.050 S31
00.21000.11500.20000.05000.001 S32
00.21000.11500.20000.05000.001 S33
00.18500.20300.21200.08700.150 S34
003.8670020.0000004.0000000.0500
002.4116020.0000004.0000000.0500
005.1938020.0000004.0000000.0500
005.7599020.0000004.0000000.0500

```
005.7599020.0000004.0000000.0500
005.7599020.0000004.0000000.0500
005.7599020.0000004.0000000.0500
004.8208020.0000004.0000000.0500
004.8508020.0000004.0000000.0500
005.4086020.0000004.0000000.0500
005.4086020.0000004.0000000.0500
005.4086020.0000004.0000000.0500
005.4086020.0000004.0000000.0500
005.4086020.0000004.0000000.0500
005.6901020.0000004.0000000.0500
003.1800020.0000004.0000000.0500
003.1800020.0000004.0000000.0500
004.6646020.0000004.0000000.0500
004.4036020.0000004.0000000.0500
004.4036020.0000004.0000000.0500
005.0879020.0000004.0000000.0500
005.7599020.0000004.0000000.0500
007.2263020.0000004.0000000.0500
005.7599020.0000004.0000000.0500
005.7599020.0000004.0000000.0500
005.7599020.0000004.0000000.0500
006.0750020.0000004.0000000.0500
003.6078020.0000004.0000000.0500
003.6078020.0000004.0000000.0500
004.0363020.0000004.0000000.0500
004.3645020.0000004.0000000.0500
004.6700020.0000004.0000000.0500
004.6700020.0000004.0000000.0500
004.5226020.0000004.0000000.0500
DRAINAGE EXPONENT =03
DRAINAGE COEFFICIENT FOR TILE DRAINS =09.55 MM/24HR
GROUNDWATER RELEASE FRACTION =000000.005
FERTILIZER APPLIED =00
 IMPOUNDMENT SPECIFICATIONS FOLLOW
NUMBER OF IMPOUNDMENTS = 00
 SURFACE ROUGHNESS AND CROP CONSTANTS FOLLOWS
NUMBER OF CROPS AND SURFACES =012
C01,    Pasture ,    00.40    0.96    0.55    050.00    3.000
095.0 005.0 000.8 008.0 002.0 085.0 0.07 0.07 0.04
0.00 0.70 1.80 3.00 3.00 3.00 2.90 2.70 1.96 0.90 0.50
001 365 0.00 00.000 00.00 00000.0 100 3.00
010.0 0.085 0.070 00.50 01.00 0.040 0.050 01 00
C02,    Hay    ,    00.80    0.96    0.45    030.00    3.000
096.0 004.0 001.0 010.0 002.0 099.9 0.07 0.07 0.04
0.00 0.15 0.40 1.90 2.60 3.00 2.96 2.92 2.30 1.15 0.50
001 365 2.30 -0.208 02.25 03020.0 120 3.00
005.0 0.085 0.450 00.50 01.00 0.050 0.200 01 00
C03,    Corn-Sil,    01.10    0.90    0.60    076.20    5.000
070.0 030.0 060.0 020.0 000.0 005.0 0.20 0.20 0.10
0.00 0.09 0.20 0.32 0.55 1.30 3.00 3.00 2.90 2.00 0.00
121 250 0.40 -0.548 01.35 44800.0 1200 3.00
043.0 0.336 2.400 00.50 01.00 0.150 0.200 00 00
C04,    Forest ,    02.00    0.95    0.50    090.00    3.500
095.0 005.0 000.8 010.0 002.0 095.0 0.25 0.20 0.13
2.50 2.50 4.50 4.50 4.50 4.50 4.50 4.50 4.50 2.50 2.50
```

```
001 365 1.30 -0.264 02.50 09400.0 900 4.50
005.0 0.000 3.000 00.50 01.00 0.100 0.200 01 00
C05,     Imp     ,     00.01     0.01     0.10    000.05    5.000
001.0 099.0 001.0 100.0 002.0 099.9 0.00 0.00 0.00
1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
001 365 0.00 00.000 00.00 00000.0 000 1.00
001.0 0.010 0.010 00.01 01.00 0.012 0.013 00 01
C06,     fallow  ,     00.01     0.10     0.45    038.60    7.000
020.0 080.0 010.0 040.0 040.0 040.0 0.05 0.05 0.02
0.00 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01
092 120 0.00 00.000 00.00 00000.0 000 0.01
043.0 0.050 0.000 00.01 01.00 0.080 0.100 00 00
C07,     W-wht   ,     00.65     0.10     0.55    063.50    7.000
080.0 020.0 060.0 018.0 002.0 060.0 0.22 0.16 0.08
0.00 0.47 0.90 0.90 0.90 0.90 1.62 3.00 3.00 3.00 0.00
251 091 1.00 -0.301 02.50 03360.0 400 3.00
043.0 0.500 0.750 00.50 01.00 0.080 0.100 00 00
C08,     CTwinfal,     00.01     0.10     0.52    050.80    1.000
005.0 095.0 003.0 080.0 005.0 005.0 0.05 0.05 0.02
0.00 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01
251 120 0.00 00.000 00.00 00000.0 000 0.01
043.0 0.050 0.000 00.01 01.00 0.120 0.180 00 00
C09,     MTcrnsil,     01.10     0.90     0.50    050.80    3.000
070.0 030.0 045.0 010.0 010.0 050.0 0.20 0.20 0.10
0.00 0.09 0.20 0.32 0.55 1.30 3.00 3.00 2.90 2.00 0.00
121 250 0.40 -0.548 01.35 44800.0 1200 3.00
030.0 0.336 2.400 00.50 01.00 0.070 0.120 01 00
C10,     ctcrn-w ,     01.10     0.90     0.60    076.20    1.000
070.0 030.0 060.0 020.0 000.0 005.0 0.20 0.20 0.10
0.00 0.09 0.20 0.32 0.55 1.30 3.00 3.00 2.90 2.00 0.00
121 250 0.40 -0.548 01.35 44800.0 1200 3.00
043.0 0.336 2.400 00.50 01.00 0.150 0.200 00 00
C11,     mtcrn-w ,     01.10     0.90     0.50    050.80    3.000
070.0 030.0 045.0 010.0 010.0 050.0 0.20 0.20 0.10
0.00 0.09 0.20 0.32 0.55 1.30 3.00 3.00 2.90 2.00 0.00
121 250 0.40 -0.548 01.35 44800.0 1200 3.00
030.0 0.336 2.400 00.50 01.00 0.070 0.120 01 00
C12,     MTwinfal,     00.01     0.10     0.35    038.60    3.000
005.0 095.0 000.0 040.0 050.0 050.0 0.05 0.05 0.02
0.00 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01
251 120 0.00 00.000 00.00 00000.0 000 0.01
030.0 0.050 0.000 00.01 01.00 0.050 0.080 00 00
NUMBER OF ALL ROTATIONS      =008
01 04 1980365 04 1981365 04 1982365 04 1983365 04 1984365 04 1985365 04 1986365
   04 1987365 04 1988365 04 1989365 04 1990365
```

{42 blank lines excluded}

```
02 05 1980365 05 1981365 05 1982365 05 1983365 05 1984365 05 1985365 05 1986365
   05 1987365 05 1988365 05 1989365 05 1990365
```

{42 blank lines excluded}

```
03 08 1980120 03 1980250 08 1981120 03 1981250 08 1982120 03 1982250 08 1983120
   03 1983250 08 1984120 03 1984250 08 1985120 03 1985250 08 1986120 03 1986250
   08 1987120 03 1987250 08 1988120 03 1988250 08 1989120 03 1989250 08 1990120
```

```
   03 1990250
```

{38 blank lines excluded}

```
 04 08 1980120 09 1980250 08 1981120 09 1981250 08 1982120 09 1982250 08 1983120
   09 1983250 08 1984120 09 1984250 08 1985120 09 1985250 08 1986120 09 1986250
   08 1987120 09 1987250 08 1988120 09 1988250 08 1989120 09 1989250 08 1990120
   09 1990250
```

{38 blank lines excluded}

```
 05 07 1980091 06 1980120 10 1980250 07 1981091 06 1981120 10 1981250 07 1982091
   06 1982120 10 1982250 07 1983091 06 1983120 10 1983250 07 1984091 06 1984120
   10 1984250 07 1985091 06 1985120 10 1985250 07 1986091 06 1986120 10 1986250
   07 1987091 06 1987120 10 1987250 07 1988091 06 1988120 10 1988250 07 1989091
   06 1989120 10 1989250 07 1990091 06 1990120 10 1990250
```

{37 blank lines excluded}

```
 06 07 1980091 06 1980120 09 1980250 07 1981091 06 1981120 09 1981250 07 1982091
   06 1982120 09 1982250 07 1983091 06 1983120 09 1983250 07 1984091 06 1984120
   09 1984250 07 1985091 06 1985120 09 1985250 07 1986091 06 1986120 09 1986250
   07 1987091 06 1987120 09 1987250 07 1988091 06 1988120 09 1988250 07 1989091
   06 1989120 09 1989250 07 1990091 06 1990120 09 1990250
```

{37 blank lines excluded}

```
 07 02 1980365 02 1981365 02 1982365 02 1983365 02 1984365 02 1985365 02 1986365
   02 1987365 02 1988365 02 1989365 02 1990365
```

{42 blank lines excluded}

```
 08 01 1980365 01 1981365 01 1982365 01 1983365 01 1984365 01 1985365 01 1986365
   01 1987365 01 1988365 01 1989365 01 1990365
```

{42 blank lines excluded}

```
  CHANNEL SPECIFICATIONS FOLLOW
 NUMBER OF CHANNEL NETWORKS  =001
 NUMBER OF TYPES OF CHANNELS =004
 CHAN01 WID =01.0(m),  SOIL N =00.045 CHAN N =00.055 0.01 1.00
 CHAN02 WID =01.5(m),  SOIL N =00.030 CHAN N =00.040 0.01 1.00
 CHAN03 WID =02.0(m),  SOIL N =00.020 CHAN N =00.030 0.01 1.00
 CHAN04 WID =02.5(m),  SOIL N =00.015 CHAN N =00.025 0.01 1.00
  ELEMENT SPECIFICATIONS FOR BASELINE SENSITIVITY ANALYSIS
 EACH ELEMENT IS0030.00m. SQUARE
 NETWORK 1 OUTFLOW FROM ROW0298 COLUMN 0120     11220
113 78 0 93 278      26   1    1  0     0  0   0   0    4610   46  184    4
  262    6  808    31
113 79 0158 295       2   1    1  0     0  0   0   0    4610   46  184    4
  262    6  808    31
```

{remainder of cell-level input excluded}

Vita

Tamie Lynne Veith was born and raised west of the Mississippi and east of the Rockies. In 1992 she received her B.A. in Mathematics from Reed College in Portland Oregon, with a thesis titled *Watered Silk: A Mathematical Look At Moiré Patterns*. Tamie stayed on the west coast and worked for three years as a research assistant at the University of Washington before driving across country to start graduate school on the east coast. She studied operations research in the Industrial and Systems Engineering Department at Virginia Tech and received her M.S. in 1997. Her focus was on simulation and WWW applications. Her M.S. thesis was titled *Netsim: A Java™ –Based WWW Simulation Package*.

In the interest of applying her knowledge and skills in a manner beneficial to the world, Tamie pursued a Ph.D. in Biological Systems Engineering at Virginia Tech. Her research in this field has included the use of modeling to understand and reduce NPS pollution at the watershed level. She received a three-year USDA National Needs Fellowship in Water Science to support her Ph.D. work. After receiving her Ph. D., Tamie plans to continue learning from and contributing to the area of land and water engineering through research and academia. The current question is where in the country (or world) will she end up next?