OPTIMIZATION OF MULTIRESERVOIR SYSTEMS BY GENETIC ALGORITHM

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ONUR HINÇAL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
CIVIL ENGINEERING

JANUARY 2008

Approval of the thesis:

OPTIMIZATION OF MULTIRESERVOIR SYSTEMS BY GENETIC ALGORITHM

submitted by **ONUR HINÇAL** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Civil Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** ____________________

Prof. Dr. Güney Özcebe
Head of Department, **Civil Engineering** ____________________

Assist. Prof. Dr. A.Burcu Altan Sakarya
Supervisor, **Civil Engineering Dept., METU** ____________________

Prof. Dr. Metin Ger
Co-Supervisor, **Civil Engineering Dept., METU** ____________________

Examining Committee Members:

Prof. Dr. Halil Önder ____________________
Civil Engineering Dept., METU

Assist. Prof. Dr. A.Burcu Altan Sakarya ____________________
Civil Engineering Dept., METU

Prof. Dr. Mustafa Göğüş ____________________
Civil Engineering Dept., METU

Prof. Dr. Tülay Özbek ____________________
Civil Engineering Dept., Gazi University

Assoc. Prof. Dr. Osman Börekçi ____________________
Civil Engineering Dept., Boğaziçi University

Date:
11.01.2008

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Onur, HINÇAL

Signature :

ABSTRACT

OPTIMIZATION OF MULTIRESERVOIR SYSTEMS BY GENETIC ALGORITHM

Hınçal, Onur

Ph.D., Department of Civil Engineering

Supervisor : Assist. Prof. Dr. A.Burcu Altan Sakarya

Co-Supervisor: Prof. Dr. A. Metin Ger

January 2008, 85 pages

Application of optimization techniques for determining the optimal operating policy for reservoirs is a major title in water resources planning and management. Genetic algorithms, ruled by evolution techniques, have become popular for solving optimization problems in diversified fields of science. The main aim of this research was to explore the efficiency and effectiveness of the applicability of genetic algorithm in optimization of multi-reservoirs. A computer code has been constructed for this purpose and verified by means of a reference problem with a known global optimum. Three reservoirs in the Colorado River Storage Project were optimized for maximization of energy production. Besides, a real-time approach utilizing a blend of online and a posteriori data was proposed. The results achieved were compared to the real operational data and genetic algorithms were found to be effective, competitive and can be utilized as an alternative technique to other traditional optimization techniques.

Keywords: Genetic Algorithm, Optimization, Reservoirs, Real-time

ÖZ

ÇOK REZERVUARLI SİSTEMLERİN GENETİK ALGORİTMA İLE OPTİMİZASYONU

Hınçal, Onur
Doktora, İnşaat Mühendisliği Bölümü
Tez Yöneticisi : Yrd.Doç.Dr. A.Burcu Altan Sakarya
Ortak Tez Yöneticisi: Prof. Dr. A.Metin Ger

Ocak 2008, 85 sayfa

Rezervuarlar için optimal işletme politikasını belirlemek üzere optimizasyon tekniklerinin uygulanması, su kaynakları planlaması ve yönetiminin önemli bir konusudur. Evrim teknikleriyle yönetilen genetik algoritma, bilimin çeşitli alanlarındaki optimizasyon problemlerini çözmek için gözde hale gelmiştir. Bu araştırmanın ana hedefi, genetik algoritmanın çoklu rezervuarların optimizasyonunda uygulanabilirliğinin verimliliği ve yararlılığını keşfetmekti. Bu amaçla, bir bilgisayar kodu oluşturuldu ve bilinen global optimuma sahip olan bir referans problem aracılığıyla doğrulandı. Colorado Nehri Depolama Projesi'nde yer alan üç rezervuar, enerji maksimizasyonu için optimize edildi. Bunun yanında, güncel ve geçmiş verilerin harmanını kullanan bir gerçek zamanlı yaklaşım önerildi. Elde edilen sonuçlar, gerçek işletme verileriyle karşılaştırıldı ve genetik algoritmaların etkili, rekabet edebilir olduğu ve diğer geleneksel optimizasyon tekniklerine alternatif bir teknik olarak kullanılabileceği tespit edildi.

Anahtar Kelimeler: Genetik Algoritma, Optimizasyon, Rezervuarlar, Gerçek zamanlı

To My Parents

ACKNOWLEDGMENTS

This dissertation marks the culmination of three years work on the development of a new optimization model under the direction of Assist. Prof. Sakarya, B. (advisor and committee member), professors Ger, M. (co-advisor and committee member), Göğüş, M. (committee member) and Börekçi, O. (committee member). I would like to extend my sincere thanks to my advisor for her help, encouragement, and constant enthusiasm. I would also like to thank my co-advisor, professor Ger, M. for his many innovative ideas. My studies have benefited immensely from his knowledge, experience, and teaching and I feel fortunate to have had the opportunity to study and research with him. Finally, I am grateful to Professor Göğüş and Börekçi for their useful comments while serving on my committee. On a personal note, I am thankful for having wonderful parents who have always valued and encouraged my learning.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

This chapter gives the description of the problem, an overview on water resources systems, objectives of this research, road map and a brief outline of this thesis.

1.1 Problem Description

For 5,000 years dams have served mankind, ensuring an adequate supply of water by storing water in times of surplus and releasing it in times of scarcity. Today there are more than 45,000 large dams in the world contributing to the management of scarce water resources and mitigating devastating floods and catastrophic droughts.

Dams regulate the natural runoff with its seasonal variations and climatic irregularities to meet the pattern of demand for irrigated agriculture, power generation, domestic and industrial supply and navigation. They also provide recreation, attract tourism, aquaculture and fisheries, and can enhance environmental conditions. Dams contribute greatly to the world's food production in providing water for irrigation. Many of them generate electricity, clean renewable energy without CO_2 emissions.

In spite of the large investments made in dams and reservoirs worldwide, many are still operated on the basis of experience, rules of thumb or static rules established at the time of construction. Even small improvements in the operating policies can lead to large benefits for many consumers.

Optimization of reservoir operation is an area that has attracted extensive research over the years. Optimization in design, planning and implementation of water resources systems have always been an intensive research area. Optimization of water resources systems is related not only to the physical structures and their functional characteristics but also the criteria by which the system is operated.

There are many decision making problems in the world, which have many constraints. A reservoir operation problem can be considered as a decision making problem. Optimizing reservoir operations incorporate allocation of resources, development of stream flow regulation strategies, operating rules and real-time release decisions in its bodily constitution. A reservoir regulation plan, which is sometimes referred to as operating procedure or release policy is a group of rules quantifying the amount of water to be stored, released or withdrawn from a reservoir or system of reservoirs under various conditions. This study intended to build an operational model to ease the decisions about the optimal volumes to be stored or released from the reservoirs in question, i.e. the operational decisions.

Multi-reservoir operation/management planning is a complex task involving many variables, objectives, and decisions. The complexities of the multiple reservoir system compel that the release decisions are determined by means of optimization or simulation models. Most of the optimization methods are constructed upon the basis of mathematical modeling. So far, optimization methods have been implemented for both planning purposes and for real time operation. Real time reservoir operation deals with the optimal operation of an existing reservoir system and decisions about releases have to be made in reasonably short time periods. In determining optimal policy, storages for the ending time of period optimized are necessarily to meet the required target ending minimum storages at this time point. This system state is desired to be applicable to satisfactory future operations. In other words, it is desired to establish the optimum release policy over the release periods specified, which shall result in a set of target ending minimum storages in the final policy period that makes sure of being adequate for future system operations.

In a typical manner, the optimization model deals with constraints such as: continuity equation, maximum and minimum storages in the reservoirs, maximum and minimum releases from the reservoirs and some case-specific obligations.

The most commonly accepted objectives are the optimality of the water supply for irrigation, industrial and domestic use, hydropower generation, water quality improvement, recreation, fish and wildlife enhancement, flood control and navigation.

1.2 Definition of Water Resources Systems

A dam is a barrier built across a watercourse for impounding water. By erecting dams, humans can obstruct and control the flow of water in a basin. A reservoir is an artificial lake, usually the result of a dam, where water is collected and stored in quantity for various uses. The major function of reservoirs is to smooth out the variability of surface water flow through control and regulation and make water available in case of necessity.

Reservoir is one of the major storage zones of water and forms a crucial part of water resources management. Water resource systems should be designed and operated for the most effective and efficient achievement of overall objectives.

One of the most important uses of reservoirs is to produce electricity. In this case a hydroelectric power plant is provided near the reservoir. The quantity of energy produced by a hydropower plant depends both on the flow through the turbines and the water head. The water head is the difference between upstream water elevation and tailwater elevation, which are the reservoir levels respectively in front of the intake and at the exit of the draft tube.

Several objectives have been considered in the optimization models of water resources systems in the previous researches. Those objectives were set down in the state of the art review of Ralph and Wurbs (1993) as follows:

- Economic benefits and costs
    - maximize water supply and/or hydroelectric power revenues
    - minimize the cost of meeting electric-power commitments

- minimize economic losses due to water shortages
- minimize the cost of pumping water in a distribution system
- minimize the damage associated with a specified flood event
- maximize net benefits of multi-purpose operations

- Water availability and reliability
  - maximize firm yield, yields for specified reliabilities, or reliabilities for specified demands
  - minimize shortage frequencies and/or volumes
  - minimize shortage indices, such as the sum of the squared deviations between target and actual diversions
  - maximize the minimum streamflow
  - maximize reservoir storage at the end of the optimization horizon
  - minimize spills or evaporation losses
  - minimize average monthly storage fluctuations
  - maximize the length of the navigation season
- Hydroelectric power generation
  - maximize firm energy
  - maximize average annual energy
  - minimize energy shortages or energy shortage indices
  - maximize the potential energy of water stored in the system

As can be inferred from the above listed objectives, there is a broad range of benefits to be accomplished from the water resources systems.

One of the most important benefits of the water resources systems is the generation of hydroelectric power. The objective function employed in this study is the maximization of the energy to be produced by the system.

Besides, this study considers a set of reservoirs as a system rather than individually. Dealing with the set of reservoirs jointly, the main purpose was to obtain a greater benefit than that is obtained dealing with this set of reservoirs individually.

Figure 1.1 indicates the model of a single reservoir operation optimization problem. S_n, I_n and R_n are defined as the storage, inflow and release variables at the n^{th} stage, respectively, where the stage parameter n implies the time duration from the first stage to the n^{th} stage.

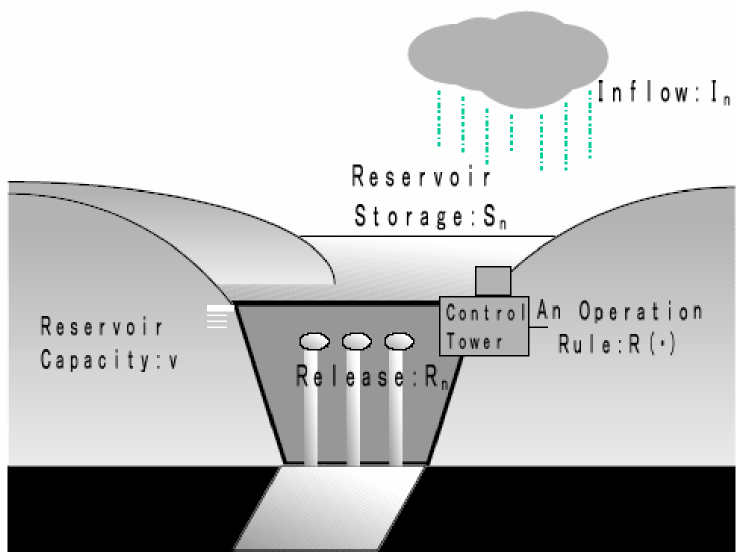

Figure 1.1 Illustration of a single reservoir model

Controlled inflows into a reservoir include all releases from adjacent upstream reservoirs on the same river or its tributaries. Uncontrolled or natural inflows include all other inflows from surface runoffs, streams and undammed rivers. Water may flow out of a reservoir through various outlets such as derivations (to draw water for irrigation or other consumption), spillways (for flood protection) and penstocks (to produce electricity). Also, there may be water losses due to evaporation and seepage into the ground.

Water is a storable commodity, so there is a continuous process of deciding whether to release it now, or to store it and release it at a later time, where the time frame for these decisions can range from minutes to months.

Reservoir operation rule is defined as a function of values of the state vector which quantifies the amount of reservoir release for each time increment considered.

Generally speaking, the optimization problem takes the following form

Objective Function:
Maximization of the Energy to be produced
Constraints:

- Continuity equation be satisfied,
- Storage be within the upper and lower bounds,
- Releases be within the upper and lower bounds,
- Final end storages be satisfied.

1.3 Research Objectives

The main objectives of the research are:

- Comprehensive examination of genetic algorithm, its mechanism, applications,
- Construction of a computer code for the optimization of a multi-reservoir system management by making use of genetic algorithm,
- Verification of the built code by implementing the code to a previously solved well-known model,
- Real case study; implementing the code developed to a real case, a multi-reservoir system under operation,
- Creating a real time approach using Genetic Algorithm for the optimization of operation policy of multi-reservoir systems.

1.4 Research Plan

The initial phase of the research comprised a thorough search and study with regards to the past researches in the fields of water resources systems, optimization of those systems and genetic algorithm and its applications.

Afterwards, the multi-reservoir system optimization problem formulated as a mathematical model incorporating the decision variables, objective function and the constraints. Following the extensive study on genetic algorithm and its applications in optimization problems, genetic algorithm aiming to optimize the mathematical problem under consideration has been constructed and configured including all the necessary operators, the conditional statements in order to meet the constraints and most importantly to find out the optimum solution remaining in strict compliance with the objective function specified. Then, the computer code in which the above mentioned stages are all embedded and employed, has been developed in Fortran Language. Pursuing configuration of the code, a verification process has been administered by making use of the four reservoir problem having a known global optimum solution which has already been adopted as a reference problem in past researches focusing on optimal reservoir system operation. A sensitivity analysis has been applied to the optimization problem in order to evaluate the effects of the variables employed in the genetic algorithm optimization technique proposed.

A real case study followed this verification stage. A multi-reservoir system in the United States has been picked out as a real case. The data pertaining to the multi-reservoir system have been acquired, a real-time optimization has been applied and the real case study has been performed onto this system.

1.5 Outline of the Thesis

The literature review has been carried out with regards to the previous researches concerning the water resources systems management; optimization and genetic algorithm topics have been studied and compiled in Chapter 2, Literature Review.

Problem definition of multi-reservoir operation in water resources systems has thoroughly been investigated and an introductory chapter including the definition of the problem considered in this research has been given in Chapter 3, Problem Definition.

Genetic algorithms which constitute a vital component in this study has been examined in detail and presented circumstantially, the coding mechanism, constraint handling mechanism, fitness function, selection mechanism, selection operators, crossover mechanism and mutation mechanism have all been presented in detail in Chapter 4, Genetic Algorithm.

Chapter 5 includes the construction and verification of computer code developed in the aim of performing the optimization of multi-reservoirs in water resources systems. The algorithm and steps of the code which is intended to be applicable in general rather than being peculiar to a particular problem have also been mentioned. The four reservoir problem of Heidari et al. (1971), having a known global optimum has been examined and the performance of the computer code constructed has been tested with this example. Furthermore, the results obtained by making use of the computer code constructed have been compared to the known global optimum. Moreover, a sensitivity analysis has been performed to see the influence of the genetic algorithm parameters of the problem on the optimum solution.

Chapter 6 gives a real time approach for determination of optimal reservoir release policy by Genetic Algorithm, which is proposed originally in this research. Colorado River Storage Project, a three-reservoir system, all of which is under operation for the purpose of producing hydroelectric energy has been optimized by making use of the data attained. The results achieved after optimization of the multi-reservoir system in the CRSP are examined and presented.

This last chapter, Chapter 7, incorporating summary, conclusion and recommendations is followed by “References” and “Appendices”.

CHAPTER 2

LITERATURE REVIEW

This chapter presents a literature survey focused on genetic algorithms and optimization in water resources problems.

Several approaches have been developed for optimization of reservoir operations, defining reservoir operating rules and many different techniques have been studied with regards to this optimization problem. Numerous optimization models have been proposed and reviewed by many scientists.

Historical background of reservoir operation optimization techniques has been given below.

For a long period, dynamic programming (Bellman, 1957), has a powerful approach in the optimization of reservoir operation. The prime advantage of dynamic programming is its ability to deal with complex objective functions without difficulty. Furthermore, constraints in the optimization problem can easily be embedded into dynamic programming. Young (1967) developed optimal operating rules for a single reservoir using dynamic programming. Larson (1968) proposed a study embracing a four-reservoir problem by making use of incremental dynamic programming. He also studied dynamic programming successive approximation technique in the optimization of reservoir systems, then Trott and Yeh (1973) used the successive approximation technique together with incremental dynamic programming. Hall et al. (1969), using a different form of incremental dynamic programming, studied a two-reservoir system. Heidari et al. (1971) developed a model, setting off from the proposal of incremental dynamic programming, which is called discrete differential dynamic programming. A procedure incorporating a combination of both linear programming and dynamic programming optimization of a multiple reservoir system has been put forth by Becker and Yeh (1974). Procedure suggested by Becker and Yeh (1974) has also been used by Takeuchi and Moreau (1974), Yeh et al. (1979), Yeh and Becker (1982), and Marino and Mohammadi (1983). Howson and Sancho (1975) generated a progressive optimality algorithm for optimization of reservoir operation policies. Loucks and Dorfman (1975) showed that chance constrained models on reservoir planning and operation are overly conservative and generate operational rules that exceed the prescribed reliability levels. Murray and Yakowitz (1979) have developed an effective technique, differential dynamic programming, for optimization of multi-reservoir systems, without any requirement for discretization state and decision variables. Extensive review of dynamic programming applications to reservoir systems is available in the studies of Yakowitz (1982) and Yeh (1985). Braga et al. (1991) applied a stochastic approach to the multi-reservoir system of the Companhia Energetica de Sao Paulo, Brazil, but attempted to account for spatial correlation of inflows. Ko et al. (1992) compared epsilon-constraint method and weighting method for multi-objective evaluation of the Han River Reservoir system in Korea. Karamouz et al. (1992) applied discrete dynamic programming to a multiple site reservoir system in the Gunpowder River Basin near Baltimore. Wurbs (1993) describes several computational models that can be used in the analysis of water resource systems. Crawley and Dandy (1993) applied separable programming to the multi-reservoir Metropolitan Adelaide water supply system in Australia. A stochastic dynamic programming approach is proposed by Archibald et al. (1997) whereby a sequence of three-dimensional stochastic dynamic programming problems are solved, with states representing the current reservoir, aggregate states of upstream reservoirs, and an approximation of the downstream reservoir. Ahmed and Lansey (2001) proposed a method based on the parameter iteration method of Gal (1979) involving quadratic approximation of future benefits and parameterization of operating policies for hydropower systems. Labadie (2004) performed an extensive compilation on the optimal operation of multi-reservoir models. Liu et al. (2006) proposed and used the dynamic programming neural-network simplex (DPNS) model in order to derive refill operating rules in reservoir planning and management.

Genetic algorithm was firstly developed by Holland, J. (1975) over the course of the 1960s and 1970s and finally popularized by one of his students, David Goldberg, who was able to solve a difficult problem involving the control of gas pipeline transmission for his dissertation (Goldberg, 1989). Holland was the first to try to develop a theoretical basis for genetic algorithms through his schema theorem. The work of De Jong (1975) showed the usefulness of the genetic algorithm for function optimization and made the first concerted effort to find optimized genetic algorithm parameters.

Genetic algorithm applications in diversified fields of science are mentioned below.

Goldberg and Kuo (1987) developed a study for pipeline optimization by making use of genetic algorithms. Pioneers of genetic algorithm, Goldberg (1989) and Michalewicz (1992) presented satisfying introductions and several papers give general overviews of genetic algorithm. Genetic algorithm has been applied to many real life optimization problems by several researchers. Wang (1991) applied a genetic algorithm to the calibration of a conceptual rainfall-runoff model. Murphy et al. (1993) developed a methodology for optimizing a water supply network using genetic algorithm, having an objective of finding the combination of pipe sizes minimizing the cost of the system. Ritzel et al. (1994) solved a multi-objective ground-water pollution problem using a genetic algorithm, considering reliability and cost of a hydraulic containment system. McKinney and Lin (1994) also solved a ground-water management model by incorporating groundwater simulation models into a genetic algorithm. Simpson et al. (1994) studied pipe network optimization comparing nonlinear programming and genetic algorithm. Cieniawski et al. (1995) studied the multi-objective optimal location of a network of ground-water monitoring wells under conditions of uncertainty by benefiting from genetic algorithm. Davidson and Goulter (1995) used genetic algorithms to optimize the layout of rectilinear branched distribution (natural gas/water) systems. A study similar to that of Wang (1991), for the automatic calibration of conceptual rainfall-runoff models, has been reported by Francini (1996), who made use of a genetic algorithm combined with a local search method; sequential quadratic programming. Genetic algorithm has been developed by Dandy et al. (1996) for cost optimization of pipenetworks. Soh and Yang (1996) used genetic algorithms in combination with fuzzy logic for structural shape optimization problems. Feng et al. (1997) applied genetic algorithm to the problem of cost-time trade-offs in construction projects. Halhal et al. (1997) applied genetic algorithm to a network rehabilitation problem having multiple objectives. A methodology based on genetic algorithms has been developed by Li and Love (1998) for optimizing the layout of construction site level facilities. Wang and Zheng (2002) studied on job shop scheduling with a modified genetic algorithm. Wei et al. (2005) employed genetic algorithm in their research aiming optimization of truss size and shaping with frequency constraints.

Genetic algorithms have many applications in reservoir systems optimization. Researches concerning the application of genetic algorithm in optimization of reservoir operation are summarized below.

Esat and Hall (1994) applied a genetic algorithm to the four-reservoir problem. The objective of this problem was to maximize the benefits from power generation and irrigation water supply, having constraints on both storages and releases from the reservoirs. The study of Esat and Hall indicated that genetic algorithms constitute a significant potential in reservoir operation, and their study clearly put forward the fact that genetic algorithms have superiorities over standard dynamic programming techniques in many aspects. Fahmy et al. (1994) applied genetic algorithm to a reservoir system, and compared performance of the genetic algorithm approach with that of dynamic programming. Raman and Chandramouli (1996) used an artificial neural network for inferring optimal release rules conditioned on initial storage, inflows, and demands. Results of a deterministic DP model for the Aliyar reservoir in Tamil Nadu, India for 20 years of bimonthly data serve as a training set for the artificial neural network. The training of an artificial neural network is an optimization process, usually by a gradient-type back propagation procedure, which determines the values of the weights on all interconnections that best explain the input-output relationship. Oliveira and Loucks (1997) used a genetic algorithm to evaluate operating rules for multireservoir systems and indicated that optimum reservoir operating policies can be determined by means of genetic algorithms. Cai et al. (2001) describe an application of genetic algorithms to solving large-scale nonlinear reservoir operation problems over multiple periods. In this study, the genetic algorithm only optimizes over a limited number of complicating or coupling variables such that when fixed, allow decomposition of the original problem into many small linear programming problems. Chandramouli and Raman (2001) extended the study of Raman and Chandramouli (1996), developing operating rules for multireservoir systems. Sharif and Wardlaw (2000) presented a real case study in Brantas Basin in Indonesia for the optimization of the system using genetic algorithm. Ahmed and Sarma (2005) presented a genetic algorithm model for finding the optimal operating policy of a multi-purpose reservoir, located on the river Pagladia, a major tributary of the river Brahmaputra.

CHAPTER 3

PROBLEM DEFINITION

This chapter covers definition of the problem of this study intending to determine the optimal operating policy for multi-reservoir systems.

The purpose of optimal operating policy is to specify how water is managed throughout the system. Optimal operating policy serves to reach maximum benefit from the system satisfying the flow requirements and system demands. In this study, benefit is considered to be the energy gained throughout the system. Operating policy shows variation from time to time. Operating policy is composed of decision variables which are the releases from each reservoir location at each time interval. Optimization aims to find out optimum combination of releases which will lead to generate maximum energy throughout the system. There are upper and lower boundaries for releases and storages. Besides, the storages at the end of periods considered are to be equal to or above the target ending minimum storages. These limitations form the constraints of the problem. Another constraint of the problem is that continuity equation is to be satisfied throughout the whole system. This is realized by computing storages utilizing continuity equation; hence it is satisfied as a matter of fact.

Generally expressing the optimization function:

The objective, Maximization of Total Energy Produced
Which is subject to:
Continuity equation is satisfied, which is:

$$S_{t+1}^{n} = S_{t}^{n} + I_{t}^{n} - R_{t}^{n} \qquad 3.1$$

where S_t^n, I_t^n and R_t^n are the storage, inflow and releases for the n^{th} reservoir at the t^{th} time step.

Storages will be below maximum and above minimum storages, which is:

$$S_{i,\min} \le S_{i,t+1} \le S_{i,\max} \quad \text{for t = 1,…,T} \qquad 3.2$$

Releases will be equal to or below maximum and equal to or above minimum releases, which is:

$$R_{i,\min} \le R_{i,t} \le R_{i,\max} \quad \text{for t=1,…,T} \qquad 3.3$$

Ending storage will be equal to or above the target ending minimum storages, which is:

$$S_{i,T} \ge d_{i,T} \qquad 3.4$$

where $d^{n}_{i,T}$ is the target ending minimum storage for the i^{th} reservoir at the T^{th} time step and T is the ending time for the problem under consideration.

Continuity equation is readily satisfied since the storages are computed by making use of continuity equation given in Equation 3.1.

Releases are the decision variables in the problem. Decision variables exist in the composition of the individuals of the population in Genetic Algorithm. Constraints of releases are identified during generation of initial population and as a matter of fact they are satisfied. Generation of initial population is mentioned thoroughly in "5.2 Generation of Initial Population".

Other constraints are embedded into the objective function as a penalty function. Thus, constrained optimization problem takes the form of an unconstrained optimization problem. The purpose lying beneath the fact that constraint problem is transformed into an unconstrained problem is to be able to handle the problem by means of Genetic Algorithm.

Objective function incorporating the terms penalizing the constraint violations takes the following form:

$$\sum_{i=1}^{J}\sum_{t=1}^{N}\left(Energy_{i,t}\right) \qquad 3.5$$

If $S_{i,t} > S_{\max}$, then the penalty term

$$\sum_{i=1}^{J}\sum_{t=0}^{N}\left[c_1\left(S_{i,\max} - S_{i,t}\right)^2\right]$$ is introduced into Equation 3.5

If $S_{i,t} < S_{\min}$, then the penalty term

$$\sum_{i=1}^{J}\sum_{t=0}^{N}\left[c_2\left(S_{i,\min} - S_{i,t}\right)^2\right]$$ is introduced into Equation 3.5

If $S_{i,T} < d_T$,then the penalty term

$$\sum_{i=1}^{J}\left[c_3\left(d_{i,T} - S_{i,T}\right)^2\right]$$ is introduced into Equation 3.5

where the deviations from maximum, minimum storages and target ending minimum storages are penalized by square of deviation from constraints. c_1, c_2 and c_3 are constants. Those constants act as a tuner of the weight of the penalty term in order for them to be in the order of the benefit terms.

The optimization problem, the objective function and constraints of which are given above are adapted into the genetic algorithm. Genetic algorithm will thoroughly be mentioned in the following chapter.

CHAPTER 4

GENETIC ALGORITHMS

Chapter 4 gives an overview of genetic algorithms, its working mechanism and components; i.e. coding, constraint handling, fitness, genetic operators; i.e. selection, crossover and mutation.

Genetic algorithm is a search algorithm based on mechanics of natural selection and natural genetics (Goldberg, 1989). Concisely stated, a genetic algorithm is a programming technique that mimics biological evolution as a problem-solving strategy. Genetic algorithms represent a popular approach to optimization, especially as it relates to the global optimization problem of finding the best solution among multiple local optima. As the name implies, genetic algorithm is based on principles of natural evolution and survival of the fittest. In genetic algorithms, a population of candidate solutions to the problem is employed. Genetic algorithms simultaneously consider multiple candidate solutions to the problem and proceed by moving this population of solutions toward a global optimum. In a genetic algorithm, an initial population is generated randomly and this population is exposed to genetic operators. By means of those operators, population evolves and optimum solution is achieved.

Most of the early work in the field came from those in the fields of computer science and artificial intelligence. More recently, interest has extended to essentially all branches of business, engineering, and science where search and optimization are of interest. The widespread interest in genetic algorithms appears to be due to the success in solving many difficult optimization problems.

Genetic algorithm has a main generational process cycle. This cycle is driven mainly by generation number. Within this cycle, an initial population is created; each individual is coded so as to be represented numerically; then each individual of population is assigned a fitness value. Fitness value is a parameter with respect to which each individual is evaluated whether or not to live in subsequent generations. Evaluation and selection of individual which will be awarded to live in subsequent generations are handled by means of genetic operators, selection, crossover and mutation.

Genetic algorithm begins, like other optimization algorithms, by defining decision variables and objective function. It ends like other optimization algorithms too, by testing for convergence. Nevertheless, it is quite different than the others with regards to the steps taking place in the process.

Because of the nature of the algorithm, a special terminology is used in genetic algorithms. Genetic algorithms start by generation of an initial population which is constituted by individuals called chromosomes (or also referred to as string). In other words, genetic algorithm begins by defining a chromosome or an array of variable values to be optimized. These variables are called the decision variables which has an active role in calculation of objective function value.

Population size depends on the nature of the problem, but typically contains several hundreds or thousands of possible solutions. Traditionally, the population is generated randomly, covering the entire range of possible solutions (the search space). Given upper and lower bounds for each chromosome, they are created randomly so as to remain within its upper and lower constraints. The principle is to maintain a population of chromosomes, which represent candidate solutions to the problem that evolves over time through a process of competition and controlled variation. Each chromosome in the population has an assigned fitness to determine which chromosomes are used to form new ones in the competition process which is called selection. The new ones are created using genetic operators such as crossover and mutation.

This algorithm is repeated sequentially until stopping criterion is achieved. The stopping criterion of a genetic algorithm is governed either by number of generations or the rate of change in the objective function value. Fitness values are expected to improve indicating creation of better individuals in new generations.

It is expected that most of the fitness values of the later generations will be improved after a number of iterations from the earlier generations.

The reason for a great part of success of genetic algorithm is its ability to exploit the information accumulated about an initially unknown search space in order to perform subsequent searches into useful subspaces. This constitutes a key feature, especially in large, complex, and poorly understood search spaces, where classical search tools are inappropriate.

A general flowchart of a genetic algorithm indicating the processes within the algorithm is given in Figure 4.1.

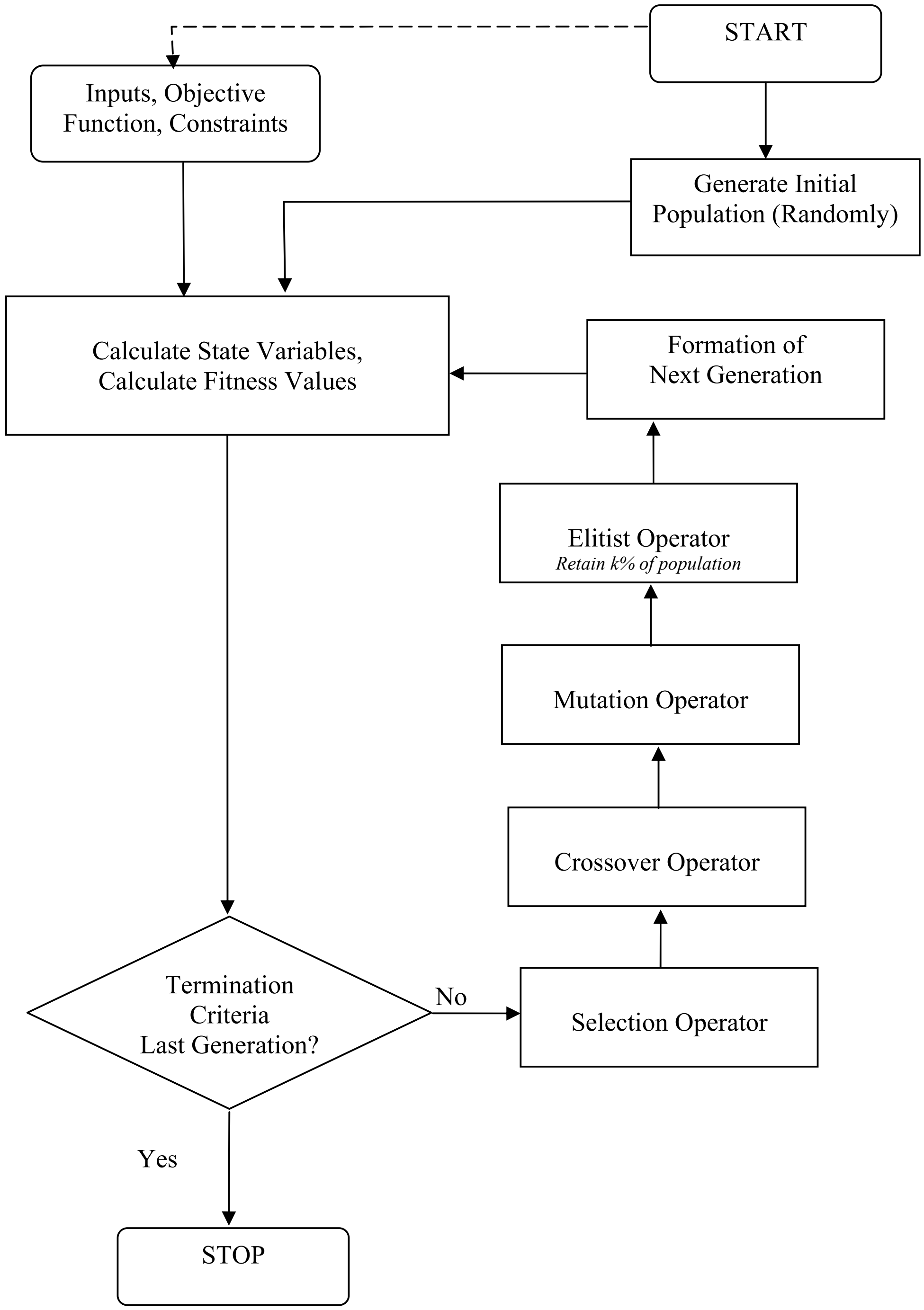

Figure 4.1 Flowchart of a Genetic Algorithm

4.1 Coding

Physical parameters in the problem space constituting the phenotypes are encoded into genotypes, thus into the genetic algorithm space. Genotype of an individual is the chromosome and the potential solution to a problem corresponding to a string is the phenotype. In genetic algorithm space, genetic operators are applied onto the individuals to generate better solutions until the optimum one. Then the individual representing the optimum solution is decoded to phenotypes and transferred into the problem space. The transferred phenotype yields the optimal solution. The decision variables, or phenotypes, in the genetic algorithm are obtained by applying some mapping from the chromosome representation into the decision variable space. Coding in genetic algorithm is the form in which chromosomes and genes are expressed. Coding, mapping from phenotypes to genotypes, is performed in a number of ways such as binary coding, gray coding, e-coding and real coding. However, most common coding mechanisms are binary and real coding. In binary coding the chromosomes are expressed as binary strings.

The most commonly used representation of chromosomes in the genetic algorithm is that of the single-level binary string by making use of 0's and 1's. In this coding, each decision variable in the parameter set is encoded as a binary string and these are concatenated to form a chromosome. Therefore, the search space of the problem is mapped into a space of binary strings through a coder mapping. Then, after implementation of the genetic operators, a decoder mapping is applied to bring them back to their real form in order to compute their fitness function values.

The use of real-valued genes in genetic algorithms is claimed by Wright (1991), to offer a number of advantages in numerical function optimization over binary coding. Efficiency of the genetic algorithm is increased as genotype into phenotype conversion is not required; less memory is required as efficient floating-point internal computer representations can be used directly; there is no loss in precision due to formation of discreteness to binary or other values; and there is greater freedom to use different genetic operators. Nonetheless, the real coding is more applicable and it seems that it fits the continuous optimization problems better than the binary coding.

In real coded genetic algorithms, each individual is coded as a vector of floating point numbers (real numbers) having the same length as that of the solution vector. Real-coded genetic algorithms handle even slight changes since real numbers represent the individuals and they are capable of local tuning the solutions.

Using real coding the representation of the solutions is very close to the natural formulation of many problems, e.g., there are no differences between the genotype (coding) and the phenotype (search space). Therefore, the coding and decoding processes that are needed in the Binary Coded Genetic Algorithms are not required; this increases the speed of process and expressiveness level reached becomes very high.

Real coding allows the domain knowledge to be easily integrated into the Real Coded Genetic Algorithms. Goldberg (1991) and Eshelman and Schaffer (1993) leave to the user the decision for choosing one of these coding mechanisms, suggesting that each one of them has suitable properties for different types of fitness functions. On the other hand, other authors such as Michalewicz (1992) defend the use of real coding, showing their advantages with respect to the efficiency and precision reached as compared to the binary one. After evaluation of advantages and disadvantages of both coding mechanism, real coding is preferred in this research.

4.2 Constraint Handling

In optimization problems, a constraint is a condition which a solution to an optimization problem must satisfy in order to be acceptable. The set of solutions that satisfy all constraints is called the feasible set. They are generally classified as equality and inequality constraints. Constraints are embedded into the objective function in the form of penalty functions. In other words, a cost or a penalty with all constraint violations is associated with the individual and this cost is inserted into the objective function evaluation. Those penalty functions have a force on the objective function. In case individual violates constraints, they put forth a negative impact on the objective function value and weaken fitness of the individual. Hence, individual looses power to survive in the next generation. Otherwise, if constraints are not violated, fitness of the individual is not affected and retains its fitness which determines its chance to live in the next generation.

4.3 Fitness Function

Every chromosome is composed of genes (or also referred to as bits) representing the variables which are used to determine the fitness value of the chromosome. Each and every chromosome has its own fitness value determined by calculating the objective function value for each of them. The fitness value of the chromosome is considered to be a grade for the evaluation of this member of the population whether or not to pass to the next generation. Fitness values are calculated by making use of the objective function; hence fitness value of a chromosome can be taken into consideration as the objective function value of this member. The aim in genetic algorithm is to end up with the best chromosome yielding the optimum objective function value, i.e. the best fitness value.

Fitness function determination is an important step in the optimization process, especially when an “optimum” solution is based on more than one variable. The fitness, or objective function, is the “figure of merit” for each individual chromosome, and thus determines its probability of taking part in selection process.

4.4 Selection

Selection is the survival of the fittest within the genetic algorithm. The key notion in selection is to give higher priority of preference to better individuals. During each generation, a proportion of the existing population is selected to breed a new generation. This operator is an artificial version of natural selection, a Darwinian survival of the fittest among string creatures. In natural populations fitness is determined by a creature's ability to survive predators, pestilence and other obstacles to adulthood and subsequent reproduction. In the artificial setting of a genetic algorithm, the objective function is the final arbiter of the string-creature's life or death.

In the stage of application of selection operator, the chromosomes that will be awarded to live in the subsequent generation are determined. Those chromosomes selected to live in the subsequent generation form the mating pool from which the parents of the new generation undergoes the process of crossover. Two chromosomes are selected from the mating pool of chromosomes to produce two new offspring.

Chromosomes having sufficient fitness values to be the candidates for becoming the parents of the new population, the children of who will live in the next generation are transferred into the next generation. The remaining ones are considered to be dead and excluded from the population. This operation is repeated in the subsequent iterations so that the good ones shall survive to reach the best solutions at the end of generations. In the application of the selection operator, the idea of natural selection is imposed. Selection probability is derived by making use of the ratio of the fitness of the individual to that of the total population.

A selection operator combines the relative fitness of the chromosomes of the population with some randomness in order to determine parents of the offspring generation. There are different techniques which a genetic algorithm can use to select the individuals to be copied over into the next generation.

One of the techniques used as a selection operator is the "Roulette Wheel Selection" operator. In this technique, for each and every chromosome, the ratio of the fitness value of the chromosome to the total of the fitness values of the chromosome of the whole population is calculated and this parameter computed for each chromosome is considered for this member of the population as the probability of survival into the next generation. As explained by Ansari and Hou (1999), this approach enables chromosomes with higher fitness values to have a greater probability of survival. In addition, the number of chromosomes in a population is kept constant for each generation and hence the selection operator will generate a new population of the same size. This implies that chromosomes with higher fitness values will eventually dominate the population (Ansari and Hou, 1999).

The basic implementation of a roulette wheel selection operator assigns each chromosome a "slice" of the wheel, with the size of the slice proportional to the fitness value of the chromosome. In other words, the fitter a member is the bigger the slice of the wheel it gets. To select a chromosome for selection, the roulette wheel is "spun," and the chromosome corresponding to the slice at the point where the wheel stops on is grabbed as the one to survive in the offspring generation.

The algorithm of roulette selection may be generalized in steps as follows:

1. Fitness of each individual, f_i, in a population size of N and sum of them are calculated.
2. A real random number, *ran( )*, within the range [0,1] is generated and s is set to be equal to the multiplication of this random number by the sum of the fitness values, $s = ran\,()\ x\ f_{sum}$
3. Minimal k is determined such that $\sum_{i=1}^{k} f_i \geq s$, and the k^{th} individual is selected in the next generation, $t+1$
4. Steps 2 and 3 are repeated until the number of selected individuals becomes equal to the population size, N.

Considering the recommendations and comparisons of the past researches and articles investigating the selection methods, roulette wheel selection method is preferred in this study.

Selection techniques other than Roulette Wheel Selection technique is given in "Appendix A: Overview of Selection Techniques in Genetic Algorithms".

4.5 Crossover

Next genetic operator to be applied to the generation is the crossover operator. Crossover operator is a method for sharing information between chromosomes.

If genetic algorithms were to do nothing but selection, the trajectory of populations could contain nothing but changing proportions of the chromosomes in the original population. To do something more sensible, the algorithm needs to explore different structures. A primary exploration operator used in many genetic algorithms is crossover. Without crossover, each individual solution is on its own, exploring the search space in its immediate vicinity without reference to what other individuals may have discovered. However, with crossover in place, there is a transfer of information between successful candidates - individuals can benefit from what others have learned, and schemata can be mixed and combined, with the potential to produce an offspring that has the strengths of both its parents and the weaknesses of neither.

Selected parents reproduce the offspring by performing a crossover operation on the chromosomes. It has always been regarded as the main search operator in genetic algorithms because it exploits the available information in previous samples to influence future searches. In nature, crossover implies two parents exchange parts of their corresponding chromosomes. Since more fit individuals have a higher probability of producing offspring than less fit ones, the new population will possess on the average an improved fitness. This is why the most real coded research has been focused on developing effective real-parameter crossover operators, and as a result, many different possibilities have been proposed.

The performance of real coded genetic algorithms on a particular problem will be strongly determined by the degrees of exploration and exploitation associated to the crossover operator being applied.

When two genes $c_i^1, c_i^2 \in [a_i, b_i]$ which are to be combined with $\alpha_i = \min\{c_i^1, c_i^2\}$ and $\beta_i = \max\{c_i^1, c_i^2\}$ are considered, the action interval $[a_i, b_i]$ of these genes may be divided into three intervals as shown in Figure 4.2. These intervals bind three regions to which the resultant genes of some combination of the former may belong. In addition, considering a region $[\alpha_i^{'}, \beta_i^{'}]$ so that $\alpha_i^{'} \leq \alpha_i$ and $\beta_i^{'} \geq \beta_i$ would seem reasonable.

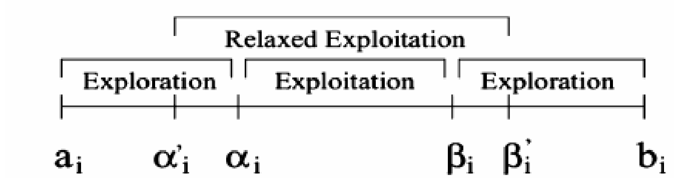

Figure 4.2 Action interval for $c_i^1, c_i^2 \in [a_i, b_i]$

Exploration and/or exploitation degrees may be assigned to any crossover operator for real coded genetic algorithms depending on the way in which these intervals are considered to generate genes.

4.5.1 Crossover Operators for Real Coding

Assuming that $C_1 = (c_1^1, \ldots, c_n^1)$ and $C_2 = (c_1^2, \ldots, c_n^2)$ are two chromosomes that have been selected to apply the crossover operator to them, below is described the operation of the crossover operators for Real Coded Genetic Algorithms considered and their effects are shown in graphical form.

4.5.1.1 Random crossover

Two offspring are created,

$$H_k = \left(h_1^k, \ldots\ldots, h_i^k \ldots\ldots, h_n^k\right),\ \text{k=1,2} \quad 4.1$$

The value of each gene in the offspring is determined by the random uniform choice of the values of this gene in the parents:

$$h_i^k = \begin{cases} c_i^1 \ldots if \ldots u = 0 \\ c_i^2 \ldots if \ldots u = 1 \end{cases} \quad 4.2$$

u being a random number which can have a value of zero or one (Syswerda, 1989).

4.5.1.2 Arithmetic crossover

Two offspring are produced,

$$H_k = \left(h_1^k, \ldots\ldots, h_i^k \ldots\ldots, h_n^k\right)\ \text{k=1,2} \quad 4.3$$

$$h_i^1 = \lambda \cdot c_i^1 + (1-\lambda) \cdot c_i^2 \quad 4.4$$

$$h_i^2 = \lambda \cdot c_i^2 + (1-\lambda) \cdot c_i^1 \quad 4.5$$

Where $\lambda \in [0,1]$

Below is shown the region for arithmetical crossover in Figure 4.3 (Michalewicz, 1996).

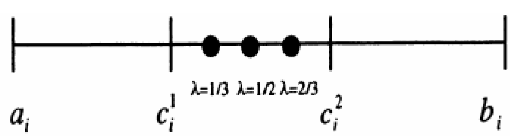

Fig 4.3 Arithmetical crossover with different values for $\lambda \in [0,1]$

4.5.1.3 BLX-α Crossover

Two offspring are generated.

$$H_k = \left(h_1^k, \ldots\ldots, h_i^k \ldots\ldots, h_n^k\right)\ \text{k=1,2} \quad 4.6$$

where h_i^k is a randomly (uniformly) chosen number from the interval

$$\left[C_{\min} - I\alpha, C_{\max} + I\alpha\right], \quad 4.7$$

where $C_{\max} = \max\{c_i^1, c_i^2\}$, $C_{\min} = \min\{c_i^1, c_i^2\}$ and $I = C_{\max} - C_{\min}$

Generally, BLX-α crossover allows the best final results to be obtained. It may be observed that the higher the α is, the better the results are. As α grows, the exploration level is higher, since the relaxed exploitation zones spread over exploration zones, increasing the diversity levels in the population. This allows good zones to be reached. Considering the final results for $\alpha = 0.5$, it seems natural that under this case an efficient exploration and exploitation relationship was induced (Eshelman, 1993).

Other crossover techniques for real coding is given in “APPENDIX B: Overview of Crossover Techniques in Genetic Algorithms”.

4.6 Mutation

One further operator in genetic algorithm is the mutation operator which does play a role of local random search within the framework of the generational process cycle.

Mutation is an insurance policy against lost genes. Mutation in genetic algorithms serves as an operator to reintroduce “lost genes” into the population. It works on the level of chromosome genes by randomly altering a gene value. The operation is designed to prevent genetic algorithm from premature termination.

Mutation is a random process where once the genes are replaced by another to produce a new genetic structure. In genetic algorithms, mutation is randomly applied with low probability and modifies elements in the chromosomes. Usually considered as a background operator, the role of mutation is often seen as providing a guarantee that the probability of searching any given chromosome will never be zero and acting as a safety net to recover good genetic material that may be lost through the action of selection and crossover.

CHAPTER 5

CONSTRUCTION AND VERIFICATION OF CODE

This chapter discusses the steps pursued in the construction of the optimization code intending to optimize the multi-reservoir systems by Genetic Algorithm and its verification.

5.1 Random Number Generation

Random numbers are essential in genetic algorithm as in simulation of majority of numerical computations. There are two important statistical properties for a sequence of random numbers, which are uniformity and independence. In other words, each random number generated is an independent sample drawn from a continuous uniform distribution between 0 and 1. Since random number generation shall be imported in the algorithm, necessary criteria are to be maintained.

There are numerous techniques for generating random numbers. The most widely used technique is linear congruential method, first introduced by Lehmer (1951), containing a recursive formula (based on linear recurrences) of the following form in its bodily constitution:

$$i_n = MOD_m(ai_{n\_1} + c) \qquad 5.1$$

where $MOD_m(k)$ is the module operation which returns the remainder after k is divided by m. It can generate up to m random numbers with the right choice of constants a and c. The larger m, the better it is but unfortunately there is a limit on the maximum one-word integer; 32 bit computers typically allow integers up to $w = 2^{31}$ (one bit for the sign in Fortran) or $w = 2^{32}$ (in C and Pascal). Numerical research by Park and Miller (1988) has identified a theoretical "best" set of parameters. For the linear congruential algorithm to be effective, a and m can take only a very few values, with m most certainly being prime. Park and Miller (1988) identified the parameter values a = 16807, m = 2147483647, c = 0 as producing random values for 32-bit integers.

In order to get a different sequence each time, the seed of the random number function is initialized with the sum of the current hour, minute, and second.

5.2 Generation of Initial Population

Operating rules prescribe how water is to be released or stored during the subsequent month based on current state of the system.

A chromosome (individual) representing all reservoirs in all time steps has been constructed having the following form:

$$[N_{var}] = [R_1(1),\ R_2(1),\ \ldots,R_J(1);\ldots;R_1(n),\ R_2(n),\ \ldots,\ R_J(n);\ldots\ldots;R_1(N),\ R_2(N),\ \ldots,\ R_J(N)] \qquad 5.2$$

where J is the number of reservoirs in the system considered, n is an index specifying a time period, N is the total number of time periods into which the time horizon is divided. $[N_{var}]$ is the set of genes forming a chromosome of the population. Each chromosome contains J x N genes. Each gene within chromosome represents release made from a reservoir at a specific time period and can take up any value between the upper and lower bounds of releases. N_{var} is the total number of genes in a chromosome. Number of genes in a chromosome is defined by the product of number of reservoirs and the total number of time periods considered in the system.

Since a reservoir has a finite capacity for water storage, reservoir releases do have an upper boundary and is to be a positive value. Releases are required to stay within upper and lower bounds on release. Since the objective function is based on reservoir releases in each time step, releases are the decision variables upon which the genetic algorithm is based. Maximum and minimum releases are known for each reservoir.

Hence, initial population comprising of individuals each containing release sets at each time period and reservoir location is created so that release values will remain within their known boundaries.

With ***J*** number of reservoirs and ***N*** time steps, there are ***JxN*** different variables necessary to create an individual of the population in genetic algorithm. Each of those variables is considered to be a gene. Real coding is considered while constructing the chromosomes. Hence, ***JxN*** random real numbers within the upper and lower boundaries of the releases for each reservoir shall constitute a chromosome (individual) of the population.

How genes are arranged in a chromosome is of high importance. There are two basic approaches.

1) Grouping releases by time step; such that the chromosomes contained in ***N*** groups of ***J*** genes representing the release from each reservoir in a particular time step;

2) Grouping releases by reservoir; ***J*** groups of ***N*** genes with each group containing the time series of releases from an individual reservoir.

Objective is to find a gene sequence that yields the best chromosome generating the maximum energy.

In order for the genetic algorithm to be initialized, N_{ip} chromosomes are identified. N_{ip} is the population size of the problem. Therefore, a matrix of N_{ip} rows and N_{var} columns considered. Each row of the initial population in Equation 5.3 represents a chromosome (individual) of the population.

Initially, with an identified number of individuals, i.e. population size of N_{ip}, random numbers are generated to form a matrix of $N_{ip}\ x\ N_{var}$.

$$\begin{bmatrix} Initial \\ Population \end{bmatrix}_{N_{ip} x N_{\text{var}}} = (R_{J,max} - R_{J,min})\ x \begin{bmatrix} Random \\ numbers \end{bmatrix}_{N_{ip} x N_{\text{var}}} + R_{J,min} \qquad 5.3$$

where,

$R_{J,max}$ and $R_{J,min}$ are the maximum and minimum values that the variable may assume for reservoir *J*, respectively. N_{ip} is the total number of chromosomes in a population which is an input.

Since real coding is preferred and real numbers are used, random number is not just necessarily to be either 0 or 1, but it can take up any real number between 0 and 1.

5.3 Calculation of State Variables

After generation of initial population which is composed of individuals containing releases (decision variables), calculation of storages (state variables) comes next. Storage for each and every gene of the individuals is computed making use of continuity equation (3.1) which is the equality constraint of the problem. Usage of Equation 3.1 in calculation of storages ensures that continuity equation is satisfied for every gene created. However, this does not enable the state variables (storages) determined by using the continuity equation be within their boundaries. The inequality constraints providing storages remain within their limits are satisfied by incorporating the related penalty terms into the objective function (see Equation 3.5)

5.4 Calculation of Fitness Values

Next step in the algorithm is the computation of fitness values. Fitness assigned to each gene has direct influence on eligibility for each chromosome to live in the next generation. Fitness value is the bodily constitution of objective function and the penalty terms originating from violation of the constraints, if exists. Constraints are embedded into the objective function as penalty terms in order to penalize the violation of the constraints related to storages. (See Equation 3.5).

In order to overcome negative fitness values which may cause instability in the code, they are assigned zero value. Negative fitness may occur when the negative influence of penalty functions embedded into the fitness function to account for the violation of the constraints exceeds the amount of benefit function which is defining the amount of energy generated.

Next step is the calculation of sum of the fitness values assigned to the chromosomes. This sum is a parameter used during implementation of selection operator.

5.5 Genetic Algorithm Operators

In this phase, genetic algorithm operators; selection, crossover, and mutation operators are implemented onto the population.

5.5.1 Selection Operator

At this stage, mates, whose child to live in the subsequent generation are selected. Among the selection operators mentioned in Section 4.4, roulette wheel selection operator, recommended for its superiorities over the remaining ones has been used as the selection operator. After the fitness values and the sum of the fitness values in the generation are computed, roulette wheel selection, mentioned in Section 4.4 has been placed within the code. The higher the fitness value of an individual in the current population, the higher its probability of being selected as one of the mates whose children will live in the next generation is. Selection probability is the ratio of the fitness of the individuals in the population to the sum of fitness of each individual in the population.

5.5.2 Crossover Operator

After implementation of the selection operator, selected mates are subjected to crossover operator which provides sharing of the information between the mates selected and exploits the available information in the previous samples to influence future searches. There are different crossover techniques. The mechanisms of those crossover techniques are given in Section 4.5. The computer code has been developed so as to enhance the comparison of the crossover techniques, arithmetic crossover, average crossover, random crossover and BLX-α, with different values of α. Parent chromosomes undergo crossover process to give birth to the child individuals which may have higher fitness values sharing the strong genes of the parent chromosomes, providing approximation to the optimum solution. Crossover probability has been configured as an input variable in the code. It is a governing value for the code to decide whether or not to put the parent chromosomes under the process of crossover. A random number is generated and compared with the crossover probability for the computer code to specify whether or not to apply the crossover operators. Decision to apply crossover to the selected chromosomes depend on whether a random number generated is greater than the probability of crossover or not. If it is greater, crossover operator is applied; otherwise it is not.

5.5.3 Mutation Operator

One further operator in genetic algorithm is the mutation operator which plays a role of local random search within the framework of the generational process cycle. Mutation is a random process where a gene of an individual is replaced by a new one to produce a new genetic structure. In the genetic algorithm code constructed, mutation is randomly applied with low probability, typically in the range 0.001 and 0.02 to modify the genes of some individuals. Usually considered as a background operator, the role of mutation is often seen as a safety net to recover good genetic material that may be lost through implementation of selection and crossover operators.

Mutation operator has been constructed so as to alter the gene randomly with consideration to probability of mutation. Mutation probability is configured as an input the code. In the event that the random number generated is greater than the probability of mutation, the gene is reproduced at random; otherwise it remains the same.

5.6 Verification of the Code

In order to verify the code, the four-reservoir problem which was formulated and first solved by Larson (1968), and elaborated further by Heidari et al. (1971), has been used. The fact that this problem has a known global optimum made it eligible for verification.

5.6.1 The Four Reservoir Problem, Heidari et al. (1971)

The four-reservoir problem permits to test the performance of genetic algorithms against a known global optimum and to perform sensitivity analysis. There are four reservoirs in the system, the layout of which is shown in Figure 5.1

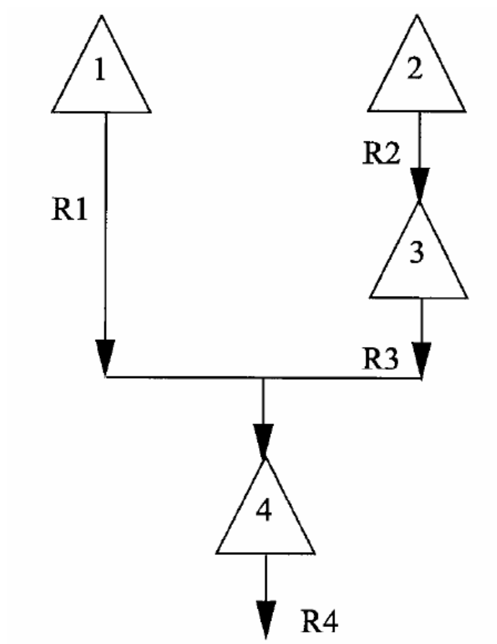

Figure 5.1 Layout of the reservoirs for the system considered

Details given by Heidari et al. (1971) with regards to the four reservoir system may be summarized as follows.

"Supplies from the system are used for hydropower generation and for irrigation. The objective is to maximize the energy produced from the system over 12 two-hour operating periods. The objective function is explicated as

$$\text{Max I} = \sum_{i=1}^{4}\sum_{t=0}^{11}\left(b_i(t)\cdot R_i(t)\right)+\sum_{t=0}^{11}\left(b_5(t)\cdot R_4(t)\right) \qquad 5.4$$

where $b_i(t)$ is the unit return due to activity i, i=1,2, …5 during a period starting at stage n and lasting at stage $n+1$. There are a total of five activities in the above criterion; four generation activities ($b_1(t)$, $b_2(t)$, $b_3(t)$, $b_4(t)$) and one irrigation activity ($b_5(t)$). The numerical values of unit returns have been given in "APPENDIX C: Benefit function constants proposed by Heidari et al. (1971)"

There are inflows to the first and second reservoirs only, and these are 2 and 3 units, respectively, in all time periods. The initial storage in all reservoirs is 5 units.
Constraints on reservoir storages for all times are:

$$0 \le S_1, S_s, S_3 \le 10 \qquad 5.5$$

$$0 \le S_4 \le 15 \qquad 5.6$$

Constraints on releases for all times are as follows:

$$0 \le R_1 \le 3 \qquad 5.7$$

$$0 \le R_2, R_3 \le 4 \qquad 5.8$$

$$0 \le R_4 \le 7 \qquad 5.9$$

Continuity equation for each reservoir over each time period, t is as follows:

$$S_{t+1}^n = S_t^n + I_t^n - R_t^n \qquad 5.10$$

In accordance with the layout of the four reservoir problem, continuity equation throughout the system may be expressed as follows:

$$\begin{Bmatrix} S_1(t+1) \\ S_2(t+1) \\ S_3(t+1) \\ S_4(t+1) \end{Bmatrix} = \begin{Bmatrix} S_1(t) \\ S_2(t) \\ S_3(t) \\ S_4(t) \end{Bmatrix} + \begin{Bmatrix} I_1(t) \\ I_2(t) \\ I_3(t) \\ I_4(t) \end{Bmatrix} + \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 1 & -1 \end{bmatrix} \begin{Bmatrix} R_1(t) \\ R_2(t) \\ R_3(t) \\ R_4(t) \end{Bmatrix} \qquad 5.11$$

Additionally, the target ending minimum storages, d_i's are as follows:

$$d_1 = d_2 = d_3 = 5 \tag{5.12}$$

$$d_4 = 7 \tag{5.13}$$

and

$$g_i\left(S_i(12), d_i\right) = -40\left[S_i(12) - d_i\right]^2 \text{ for } S_i(12) \le d_i \tag{5.14}$$

$$g_i\left(S_i(12), d_i\right) = 0 \text{ for } S_i(12) > d_i \tag{5.15}$$

where $g_i\left(S_i(12), d_i\right)$ is a function that reflects a penalty to the system when the final state of the i^{th} component of the system at stage N is $S_i(12)$ instead of the desired minimum state d_i. Such a penalty function is necessary to meet the requirements related to the target ending minimum storages. The desired state vectors of the initial and final stages for i=1,2,3,4 are assumed. If the constraints of storages are violated, the following penalty terms are embedded into the objective function.

If $S_{i,t} > S_{\max}$, then the penalty term

$\sum_{i=1}^{R}\sum_{t=0}^{T}\left[c_1\left(S_{i,\max} - S_{i,t}\right)^2\right]$ is introduced into Equation 5.4

If $S_{i,t} < S_{\min}$, then the penalty term

$\sum_{i=1}^{R}\sum_{t=0}^{T}\left[c_2\left(S_{i,\min} - S_{i,t}\right)^2\right]$ is introduced into Equation 5.4

Computer code constructed for optimization of multi-reservoir systems by genetic algorithm has been structured as mentioned in the preceding parts of this chapter by using Fortran programming language.

Since four-reservoir problem has been studied formerly and has a known global optimum, it is treated in the field of reservoir optimization problems as a reference model for verification. Hence, for the purpose of verification, code constructed has been applied to the model proposed by Heidari et al. (1971). Code created has been compiled and executed under several combinations of different input parameters of the problem.

The inputs of the computer code created for the optimization of reservoir management by Genetic Algorithm may be listed as follows:

- population size
- number of generations
- crossover technique
- probability of crossover
- probability of mutation

As can be inferred from the objective function in equation 5.4, benefit function constants (*b*'s) are used in the objective function in order to reveal the relationship between the energy generation and the decision variables, i.e. releases at each time step and each reservoir location. Those benefit constants have been studied and proposed by Heidari et al. (1971). For the verification of the code constructed, other parameters of the problem, such as initial and boundary conditions, objective function, system layout have all been adapted to the code as they are used by Heidari et al. (1971).

5.6.2 Comparison of Results

Adopting the four reservoir problem as an appropriate reference model for verification, objective function and constraints indicated in Section 5.6.1 has been studied and examined for testing performance of the computer code constructed for the optimization of multi-reservoir systems by genetic algorithm.
For different ranges of input parameters listed above, the variation of outcomes has been explored.

The computer code has been run to observe the effect of considered different crossover techniques, namely arithmetic crossover, average crossover, random crossover and BLX-α (with different values of α) techniques. The known global optimum for the energy produced in the four-reservoir problem was given by Wardlaw and Sharif (1999) as 401.3 units of energy. Energy was given as product of benefit constants and release. Based on the above mentioned input parameters, the computer code has been run and known global optimum has been achieved. The output storages and releases obtained after execution of the code fit perfectly to those stated for the four reservoir problem by Wardlaw and Sharif (1999). The fact that the target ending minimum storages are satisfied, another constraint of the four reservoir model examined, has also been checked and confirmed for each reservoir location. Besides, the results obtained after optimization by the utilization of the computer code revealed that the inequality constraints defined in the four reservoir system have been met without any violation.

Furthermore, as expected it was confirmed that CPU time increases with increasing generation number and also with increasing population size.

Sensitivity analysis has been performed to achieve the influence of the change in the input parameters on fitness. Sensitivity analysis is mentioned in detail in Section 5.7. In the light of the recommended values for input parameters; i.e. population size, generation number, probability of crossover, probability of mutation and the results of sensitivity analysis, the following set for input parameters were employed:

Population size: 5,000
Generation number: 5,000
Probability of crossover=0.70
Probability of mutation=0.02

The variation of the fitness values obtained after test runs for different crossover techniques, namely, arithmetic crossover, random crossover, average crossover and BLX-α Crossover technique for different values of α = 0.10, 0.25 and 0.50 is shown in Figure 5.2. After exploration of the influence of different crossover techniques examined and given in Figure 5.2, it is seen that BLX-α Crossover technique exhibits a faster converging behavior with respect to that of the other crossover techniques. Fitness values determined by execution of code for different crossover techniques are given for generation numbers, 2000, 3000, 4000, 5000, in Table 5.1. This fact lead us to prefer BLX-α Crossover technique with α = 0.10.

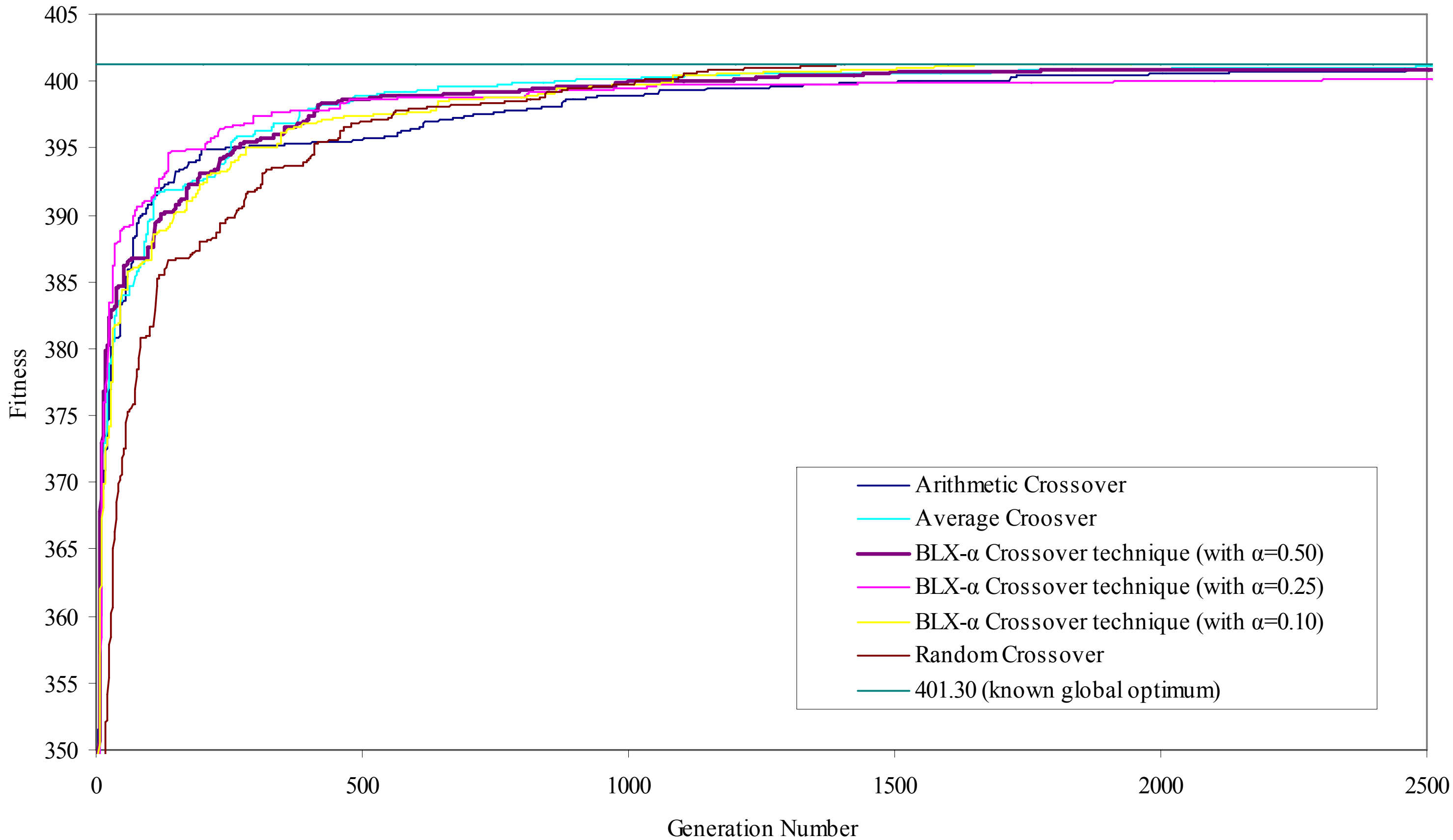

Figure 5.2 Influence of Crossover Technique on Fitness

Table 5.1 Fitness values for different crossover techniques

Crossover technique \ Generation number	**2000**	**3000**	**4000**	**5000**
Arithmetic crossover	400,589	400,887	400,942	401,006
Average crossover	400,912	401,064	401,190	401,199
BLX-α Crossover (α = 0.50)	400,812	401,102	401,278	401,289
BLX-α Crossover (α = 0.25)	401,014	401,188	401,282	401,299
BLX-α Crossover (α = 0.10)	**401,177**	**401,278**	**401,294**	**401,301**
Random crossover	399,470	400,610	400,988	401,234

5.7 Sensitivity Analysis

Sensitivity analysis is the investigation of how the variation in the output of a model can be apportioned, qualitatively or quantitatively, to different sources of variation. It is the formal technique of determining those parameters in a system that controls its performance. It identifies those parameters that are important as well as those that are unimportant. The sensitivity analysis approach of genetic algorithms parameters such as crossover probability, mutation probability, population size, generation number is discussed. The most common sensitivity analysis is sampling-based. A sampling-based sensitivity is the one in which the model is executed repeatedly for combinations of values sampled from a set of different input parameters and establishing a relationship between inputs and outputs using the model results at the sample points.

5.7.1 Sensitivity to Crossover Probability

Firstly, sensitivity analysis has been performed with respect to the probability of crossover. Used input parameters were mutation probability of 0.02 and BLX-α Crossover technique (with α=0.10). In order to see the effect of change in crossover probability on proportion of maximum fitness for different sets of population size and generation numbers. As depicted in Figure 5.3, crossover probability seems to have no significant effect on fitness for the range covered. Besides, as demonstrated by Figures 5.4, 5.5 and 5.6, with increasing generation number, the amplitude of fluctuations becomes smaller. Moreover, as generation number increases, the bandwidth which is formed by the change in population size, becomes narrower. Those variations indicate that the change in proportion of maximum fitness becomes insignificant with increasing generation number.

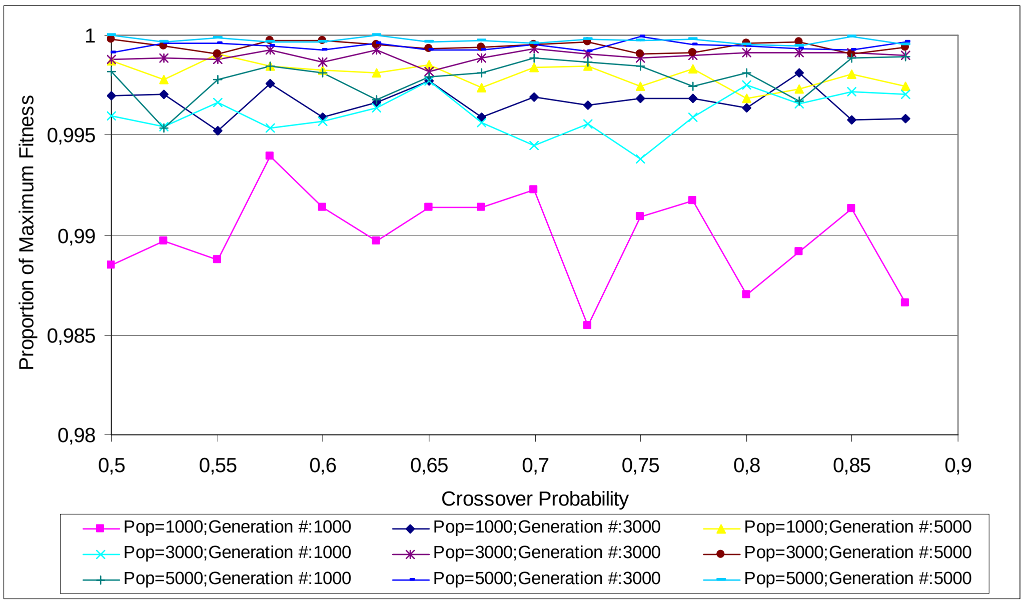

Figure 5.3. Effect of Crossover Probability on Fitness

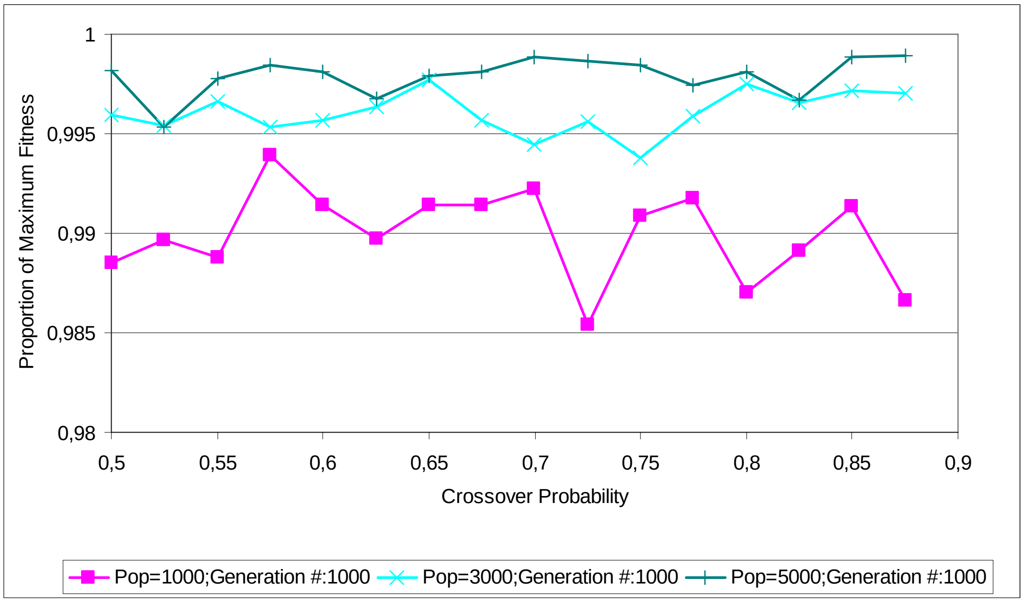

Figure 5.4 Effect of Crossover Probability on Fitness (Generation Number=1000; Population Size=1000, 3000, 5000)

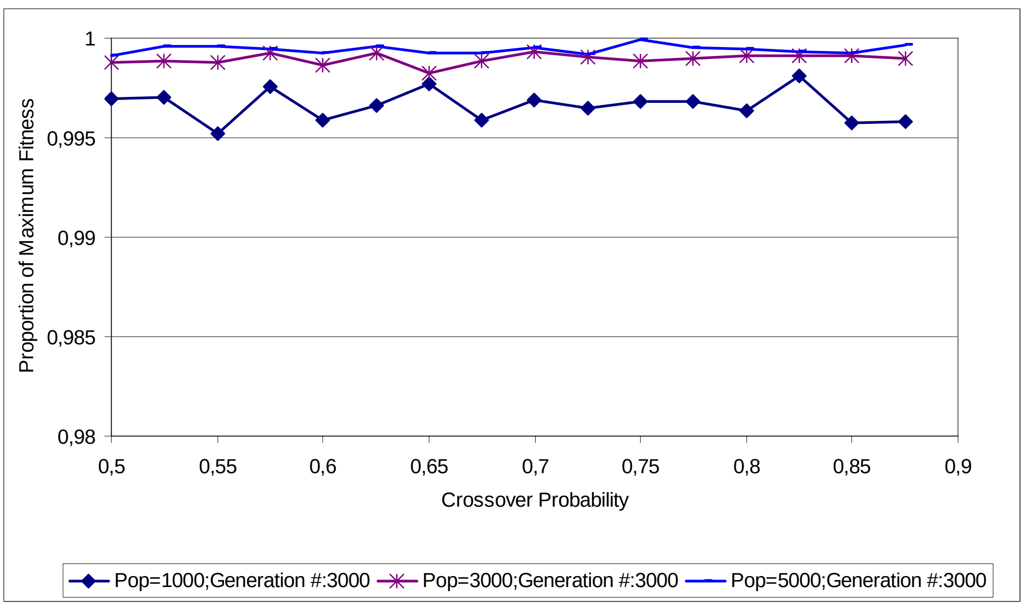

Figure 5.5 Effect of Crossover Probability on Fitness (Generation Number=3000; Population Size=1000, 3000, 5000)

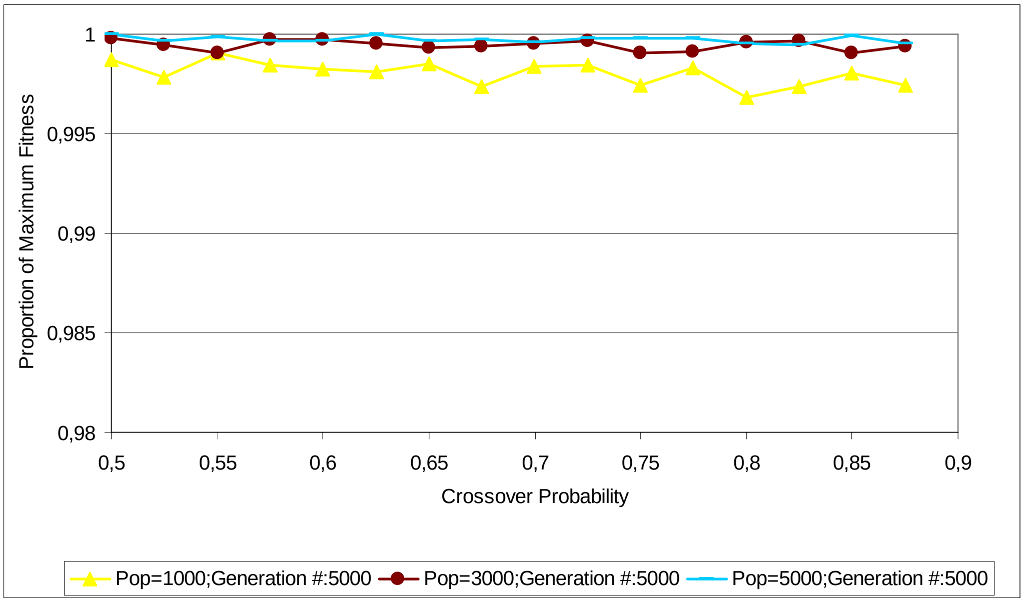

Figure 5.6 Effect of Crossover Probability on Fitness (Generation Number=5000; Population Size=1000, 3000, 5000)

5.7.2 Sensitivity to Population Size and Generation Number

Sensitivity to population size and generation number has also been investigated. Input parameters used were crossover probability of 0.70, mutation probability of 0.02 and BLX-α Crossover technique (with α=0.10). Fitness is again expressed as a proportion of the known optimum for the four-reservoir problem. Variation of proportion of maximum fitness was examined against generation number for a series of different population sizes. As demonstrated in Figure 5.7, the proportion of maximum fitness increases while the effect of population size on the proportion of maximum fitness becomes less appreciable.

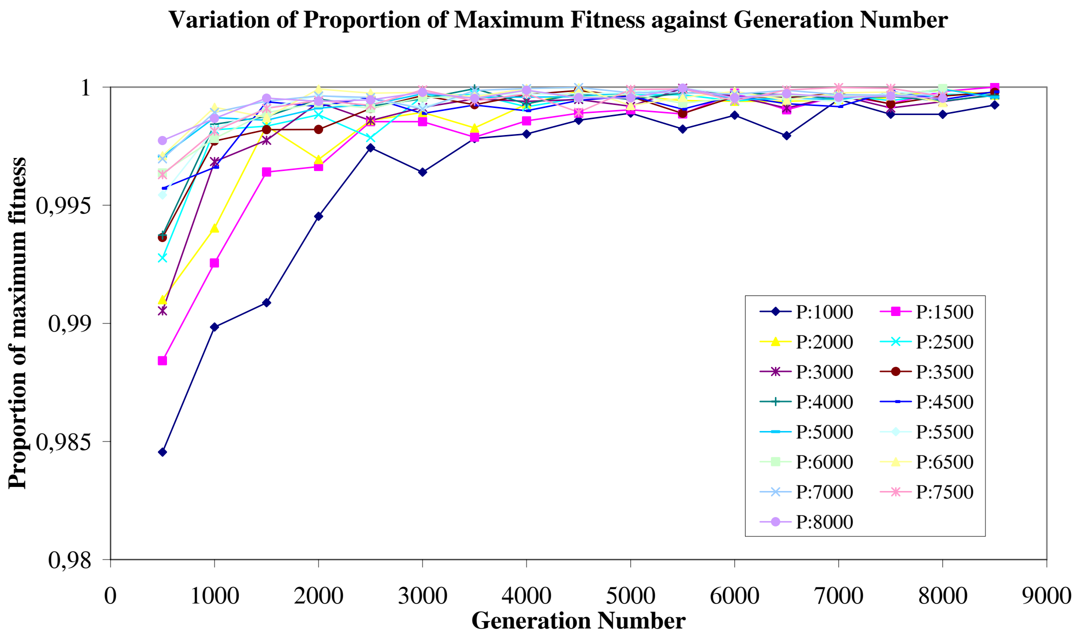

Figure 5.7 Effect of Population Size and Generation Number on Fitness

5.7.3 Sensitivity to Mutation Probability

The effect of change in mutation probability has been investigated for different sets of population size and generation numbers. Input parameters used were crossover probability of 0.70 and BLX-α Crossover technique (with α=0.10). Variation of proportion of maximum fitness was examined against mutation probability for a series of different population sizes and generation numbers. Irrespective of the population size and generation number, the proportion of maximum fitness decreases significantly for mutation probability larger than 0.06 as depicted in Figure 5.8. For mutation probability between 0.02 and 0.06, effect of mutation probability on the proportion of maximum fitness is insignificant

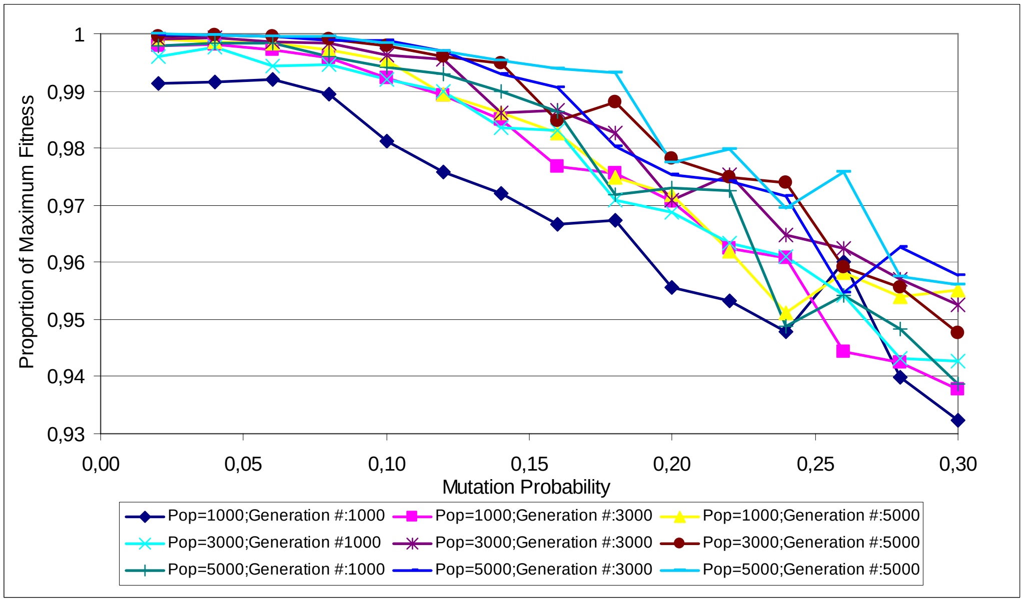

Figure 5.8 Effect of Mutation Probability on Fitness

CHAPTER 6

DEVELOPMENT OF REAL TIME APPROACH, APPLICATION AND DISCUSSION OF RESULTS

This chapter explores the multi-reservoir system in the U.S., so called Colorado River Storage Project, including the Blue Mesa, the Morrow Point and the Crystal Reservoirs, establishes the mathematical model for optimization of this system by the optimization code developed and optimizes this model for different considerations. Furthermore, it proposes a real-time optimization approach by making use of the code generated.

6.1 Definition of Problem in the Colorado River Storage Project (CRSP)

The four reservoir problem utilized in many of the past researches has been studied thoroughly for the purpose of verification as a reference model. The objective function utilized in the four reservoir problem incorporates the constraints specifically determined for that problem solely and therefore this objective function can not flexibly be applied to any other reservoir. Setting off from this idea, a more general objective function which can be applied to other real case problems has been attempted to be formed.

The objective function for determination of reservoir release policy for the maximization of the power generated has been configured, being subject to the constraints such that:

- Continuity equation is satisfied,
- Storages will be below maximum and above minimum storages,
- Releases will be equal to or below maximum and equal to or above minimum releases,
- Ending storage will be equal to or above the target ending minimum storage.

The energy generated by the hydroelectric power plant is a function of both release from the reservoir (discharge) and head. Derived from the elevation versus volume (elevation-storage curve) relationship, the power generated may be determined by fitting an appropriate curve onto the elevation-storage curve. There is a relationship between the head and storage which is determined by elevation-storage curve. Furthermore, storage is related to release by using continuity equation. Since releases are the decision variables, energy can be determined.

Colorado River Storage Project (CRSP) was examined and realized operational data with regards to the Blue Mesa, the Morrow Point and the Crystal Reservoirs were compared to those achieved by means of the optimization code developed. General description of CRSP and information related to the multi-reservoir system comprising of aforementioned reservoirs are given in detail in “APPENDIX D: Colorado River Storage Project Facts”. Realized operational data pertaining to the time period between 2002 and 2006 together with information concerning the characteristics of the reservoirs considered were obtained from the US Bureau of Reclamation, Water Resources Group, Salt Lake City Office.

The data included all of the constraints, operational data; inflows, releases, power generated, water levels in the reservoirs, current status of the dams and reservoirs.

Objective function formulation used in the four reservoir problem considered in verification process has been re-structured so as to be applicable to real world water resources problems. Objective function formulation in the four reservoir problem which was formulated and first solved by Larson (1968), and developed further by Heidari et al. (1971) included benefit constants which are only applicable to that system. Benefit constants incorporated in the energy formulation is valid solely for that four reservoir problem. Therefore, objective function to maximize the total energy production in the multi-reservoir system considered is as follows:

The objective function formulation of Barros et al. (2003) has been adopted in this study.

$$\max f = \sum_t \sum_i \left(\xi_{i,t} R_{i,t} \right) \qquad 6.1$$

where,

$\xi_{i,t}$ is the energy production function in MW.s^2/m^3; such that:

$$\xi_{i,t} = \varepsilon_i \Delta H_{i,t} = \varepsilon_i \left(HF_{i,t} - HT_{i,t} \right) \qquad 6.2$$

with ε_i is the specific productibility in MW.s^2/m^4. $HF_{i,t}$ is the reservoir upstream water level and $HT_{i,t}$ is the tailwater level in m.

Energy versus $\Delta H_{i,t} R_{i,t}$ values pertaining to the past data acquired from CRSP has been plotted for each reservoir examined. Slope of the line fitted to those plotted data reveals the specific productibility as depicted in Figures 6.1, 6.2 and 6.3, for the Blue Mesa, the Morrow Point and the Crystal Reservoirs, respectively.

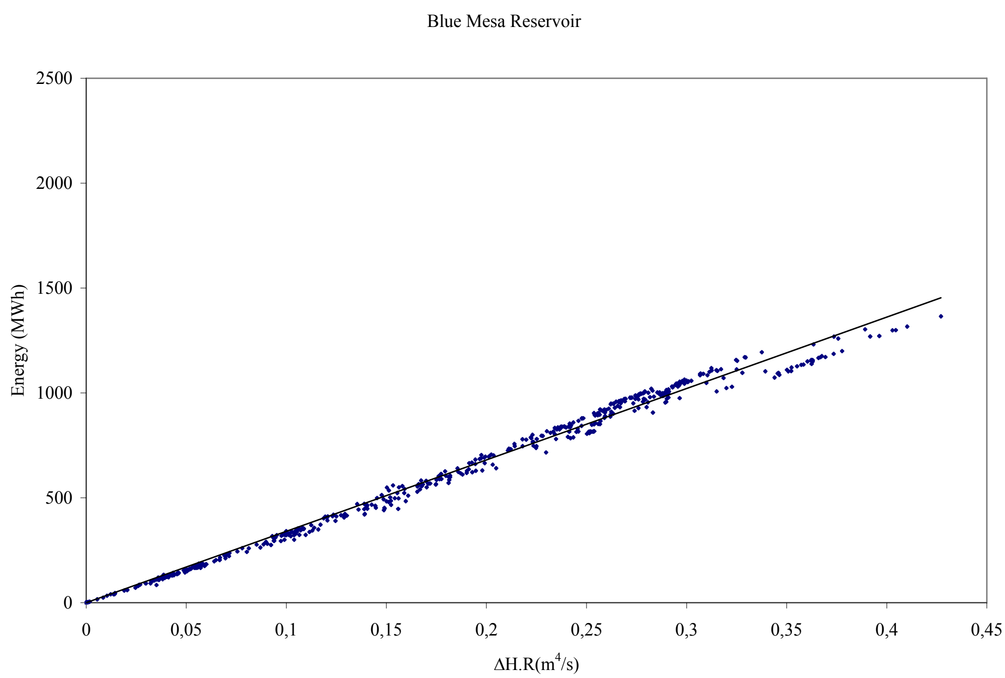

Figure 6.1 Specific productibility in Blue Mesa Reservoir

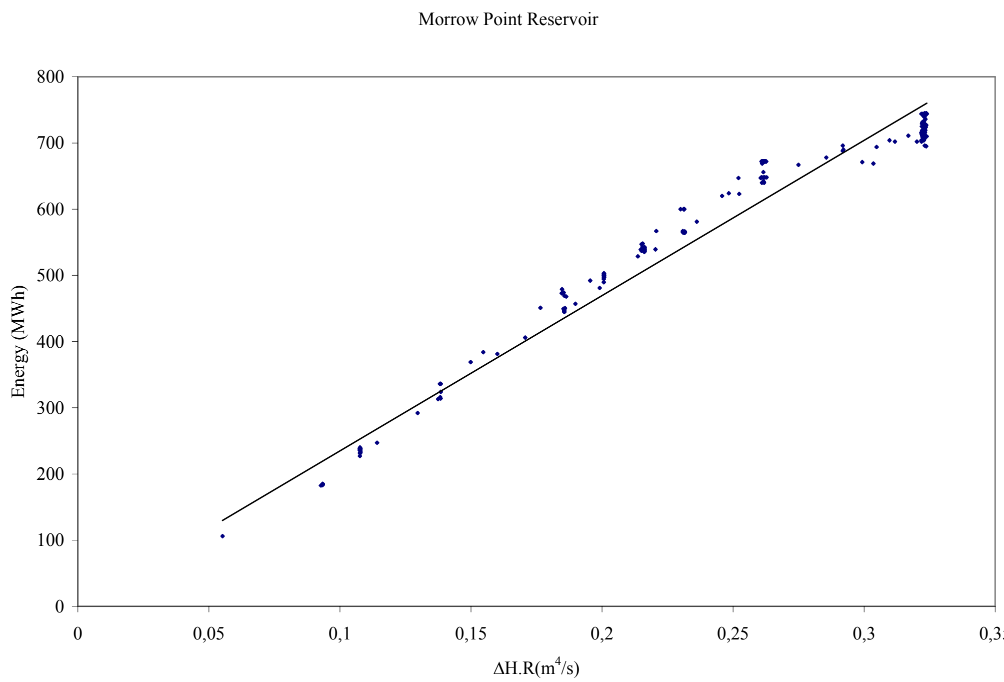

Figure 6.2 Specific productibility in the Morrow Point Reservoir

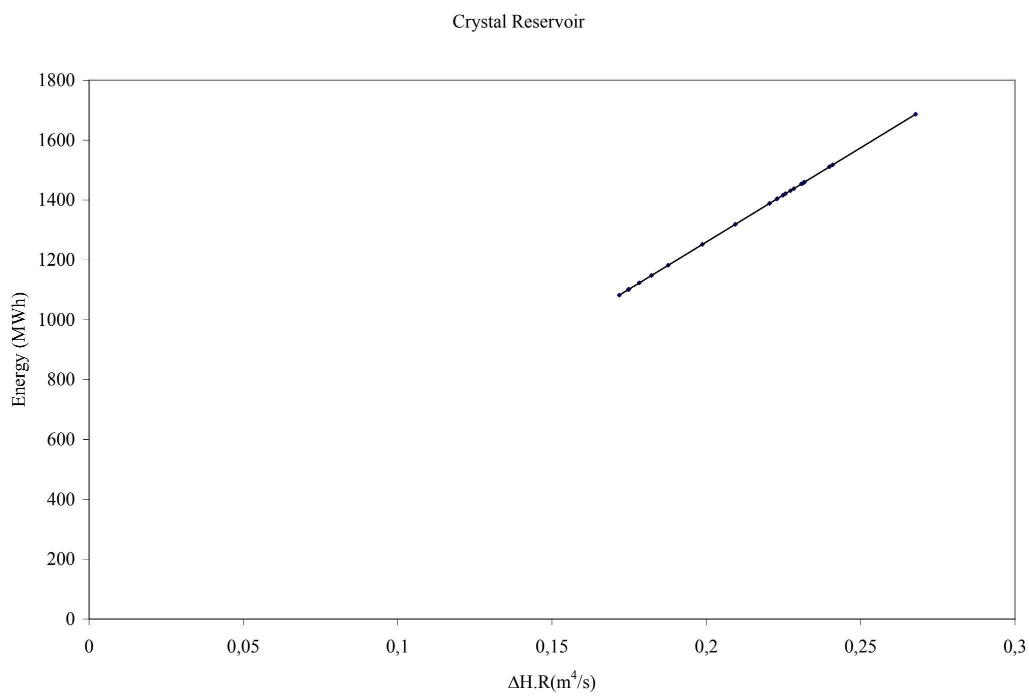

Figure 6.3 Specific productibility in the Crystal Reservoir

Specific productibility for the Blue Mesa, the Morrow Point and the Crystal Reservoirs were computed as 12,249,000, 8,449,200 and 22,671,720 in MWs^2/m^4, respectively

Upstream water level is a function of the storage value and by means of the stage-area-capacity curves obtained the relationship between the upstream water level and the storage values are determined as given below in detail.

The variation of the storage with the upstream water level and the storage in the Blue Mesa Reservoir is shown in Figure 6.4.

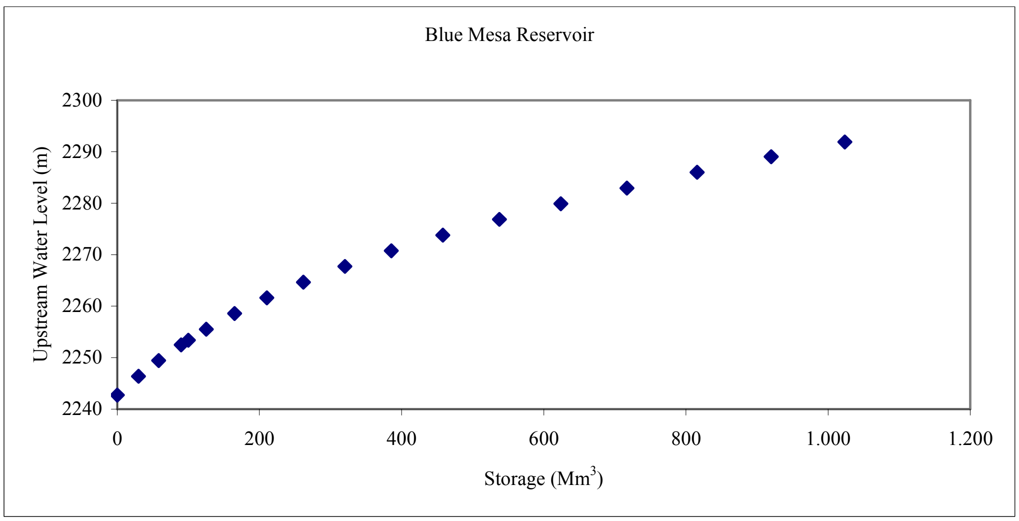

Figure 6.4 Relationship between upstream water level and storage in Blue Mesa Reservoir

The equation of the best fit curve obtained to represent the functional relationship between the water level, *HF* and the storage, *S* is:

$$HF_{1,t} = 2242.9 + 0.1196 S_{1,t} - 0.0002 S_{1,t}^2 + 2x10^{-7} S_{1,t}^3 - 6x10^{-11} S_{1,t}^4 \quad 6.3$$

The variation of the storage with the upstream water level and the storage in the Morrow Point Reservoir is shown in Figure 6.5.

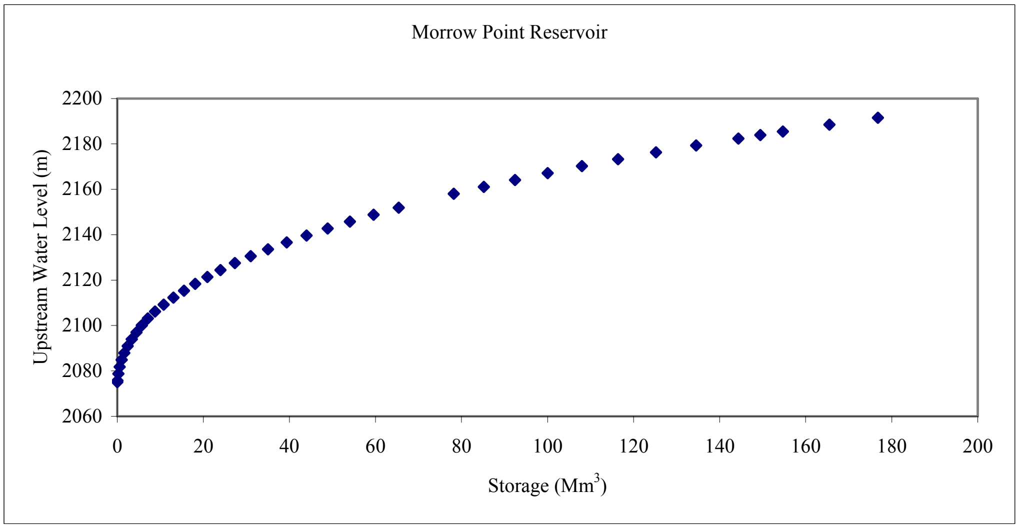

Figure 6.5 Relationship between upstream water level and storage in Morrow Point Reservoir

The equation of the best fit curve obtained to represent the functional relationship between the water level, *HF* and the storage, *S* is:

$$HF_{2,t} = 2083 + 2.4131 S_{2,t} - 0.0331 S_{2,t}^{2} + 0.0002 S_{2,t}^{3} - 6x10^{-7} S_{2,t}^{4} \qquad 6.4$$

The variation of the storage with the upstream water level and the storage in the Crystal Reservoir is shown in Figure 6.6.

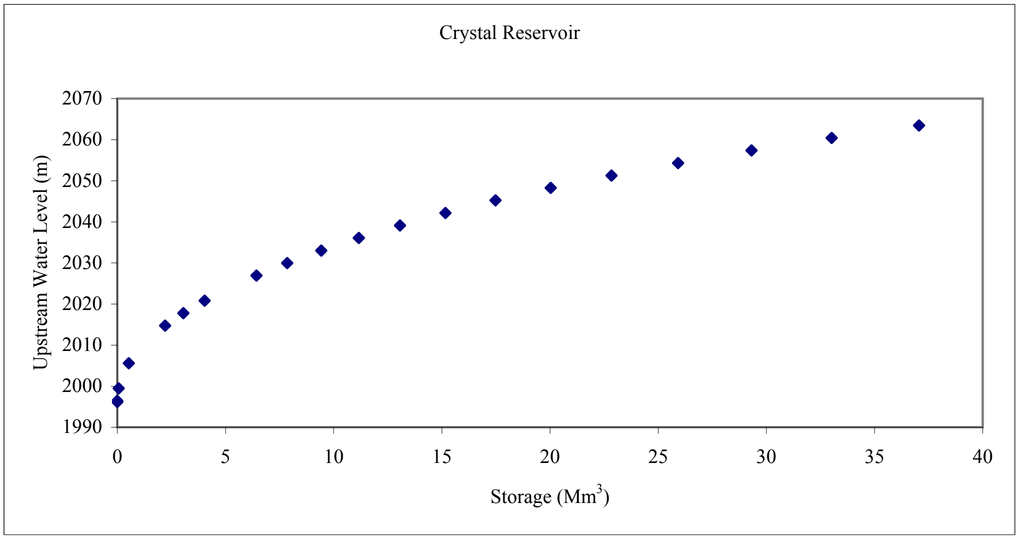

Figure 6.6 Relationship between upstream water level and storage in Crystal Reservoir

The equation of the best fit curve obtained to represent the functional relationship between the water level, *HF* and the storage, *S* is:

$$HF_{3,t} = 1999.1 + 6.6048 S_{3,t} - 0.4146 S_{3,t}^2 + 0.0133 S_{3,t}^3 - 0.0002 S_{3,t}^4 \qquad 6.5$$

Tailwater depths for the Blue Mesa, the Morrow Point and the Crystal Reservoirs are 2180, 2057 and 1990 m, respectively.

Then, energy formulation for each reservoir location in the CRSP within a specified time takes the following form:

For the Blue Mesa Reservoir;

$$E_{1,t} = 3402.5 \cdot (R_{1,t}) \cdot (62.9 + 0.1196 S_{1,t} - 0.0002 S_{1,t}^2 + 2x10^{-7} S_{1,t}^3 - 6x10^{-11} S_{1,t}^4) \qquad 6.6$$

For the Morrow Point Reservoir;

$$E_{2,t} = 2347 \cdot (R_{2,t}) \cdot (26 + 2.4131 S_{2,t} - 0.0331 S_{2,t}^2 + 0.0002 S_{2,t}^3 - 6x10^{-7} S_{2,t}^4) \qquad 6.7$$

For the Crystal Reservoir;

$$E_{3,t} = 6297.7 \cdot (R_{3,t}) \cdot (9.1 + 6.6048 S_{3,t} - 0.4146 S_{3,t}^2 + 0.0133 S_{3,t}^3 - 0.0002 S_{3,t}^4) \quad 6.8$$

Constraints on reservoir storages, S_i (Mm^3) for all times are:

$$328.991 \le S_1 \le 997.097 \quad 6.9$$

$$119.253 \le S_2 \le 142.098 \quad 6.10$$

$$16.429 \le S_3 \le 21.413 \quad 6.11$$

Constraints on releases, R_i (m^3/s) for all times are as follows:

$$0 \le R_1, R_2, R_3 \le 60 \quad 6.12$$

Continuity equation for each reservoir over each time period, t is as follows:

$$S_{t+1}^n = S_t^n + I_t^n - R_t^n \quad 6.13$$

Where I_t^n is the inflow in time period t, to reservoir n, in m^3/s.

In accordance with the layout, continuity equation throughout the system may be expressed as follows:

$$\begin{Bmatrix} S_1(t+1) \\ S_2(t+1) \\ S_3(t+1) \end{Bmatrix} = \begin{Bmatrix} S_1(t) \\ S_2(t) \\ S_3(t) \end{Bmatrix} + \begin{Bmatrix} I_1(t) \\ I_2(t) \\ I_3(t) \end{Bmatrix} + \begin{bmatrix} -1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix} \begin{Bmatrix} R_1(t) \\ R_2(t) \\ R_3(t) \end{Bmatrix} \quad 6.14$$

Additionally, the target ending minimum storages at the end of first year examined, d_i's in Mm^3 are as follows:

$$d_1 = 716.985 \quad 6.15$$

$$d_2 = 136.362 \quad 6.16$$

$$d_3 = 19.870 \quad 6.17$$

Besides, the target ending minimum storages at the end of second year examined, d_i's are as follows:

$$d_1 = 796.591 \quad 6.18$$

$$d_2 = 130.871 \quad 6.19$$

$$d_3 = 16.784 \tag{6.20}$$

Setting the boundary and initial conditions, the objective function, penalty terms and henceforth the fitness function; identifying the remaining inputs of the problem in the light of the outcomes of the sensitivity analyses; code has been executed for different comparison approaches.

Within the aforementioned considerations, the program has been run with the following set of values of genetic algorithm parameters which were already verified in the sensitivity analysis to yield convergence:
Initial Population: 5,000
Generation Number: 5,000
Probability of Crossover: 0.70
Probability of Mutation: 0.02
Roulette Wheel Selection Operator and BLX-α (α=0.10) Crossover Technique

6.2 Comparison Approaches

Energy production of the CRSP has been compared to those determined by application of genetic algorithm with conventional approach and real-time approach.

6.2.1 Conventional Approach

As an initial consideration, developed optimization code has been executed taking into account a period of 1-year (12 months of 2005). Secondly, year 2006 was optimized separately by means of the developed code. Available operational data included twelve months in 2005 and eleven months in 2006.

One further consideration was optimization of both of the separately explored 1-year periods one at a time; in other words, considering a 2-years period. The results obtained after optimization have been compared to those achieved in realized operational results.

6.2.2 Real-time Approach

A real-time approach was attempted in the final stage for the multi-reservoir system considered in the CRSP. The main goal of this real-time approach was intended to ensure real-time optimization with respect to energy maximization of the multi-reservoir system by making use of the developed code.

Firstly, a period of 1 year (2005) is optimized by utilizing the code considering the past realized operational data. Optimized solution with respect to energy maximization criterion formed a template baseline, housing the historical background of the conditions concerning the system being examined.

This template baseline is used for future real-time optimizations. Second year (2006) is optimized by using this approach. In this approach, optimization is refreshed every month. At the end of each month, inflow value becomes known and the realized inflow value is set equal to the inflow in first month of the second year (month 13). Release in this month is assumed to be the same as in the first month of the baseline. Then, continuity equation is applied to determine the storage at the end of month 13. Storage at the end of month 13 is checked so as not to violate its constraints. In case of constraint violation, release in month 13 is adjusted so that the storage at the end of month 13 remains within its upper and lower boundaries. Storage at the end of month 13 is set as the target ending minimum storage of the up-to-date template baseline. Besides, the initial storage of the up-to-date template baseline is replaced by the storage which is the successor of the initial storage, in the template baseline. The template baseline is then shifted and the code is run with the inputs of the shifted template baseline. This template baseline is shifted every month following the same flow mechanism until the end of the period considered.

As demonstrated in Figure 6.7, a brief flow scheme including the steps of the approach may be summarized as follows:

- Set inflow in time period 13 equal to the value known at the end of this month.
- Assume that the release in time period 13 is the same as in the 1st month of the baseline, (See 1 in Figure 6.7)

- Compute the storage at timepoint 13 (end of January of the second year) using Equation 6.14. Check whether the storage computed is within the upper and lower boundaries of storages. If it is not, adjust the release in time period 13 which was assumed to be the release in the 1st month of the baseline is adjusted so that the storage constraints will not be violated (See 2 in Figure 6.7 and equations 6.21-6.25)

  Assume $R_{13} = R_1$

  $$S_{13} = S_{12} + I_{13} - R_{13} \qquad 6.21$$

  If $S_{13} > S_{max}$ then,

  $$\Delta S = S_{13} - S_{max} \qquad 6.22$$

  $$R_{13} = R_1 + \Delta S \qquad 6.23$$

  If $S_{13} < S_{min}$ then,

  $$\Delta S = S_{min} - S_{13} \qquad 6.24$$

  $$R_{13} = R_1 - \Delta S \qquad 6.25$$

- Set storage at timepoint 13 as the target ending minimum storage of the up-to-date template baseline (See 3 in Figure 6.7)
- Replace storage in the 0th timepoint of the up-to-date template baseline by the storage in the 1st timepoint of the template baseline (See 4 in Figure 6.7)
- Shift up-to-date template and run the code considering the inputs of the shifted up-to-date template baseline.
- Follow the same procedure and shift the template for the remaining months.

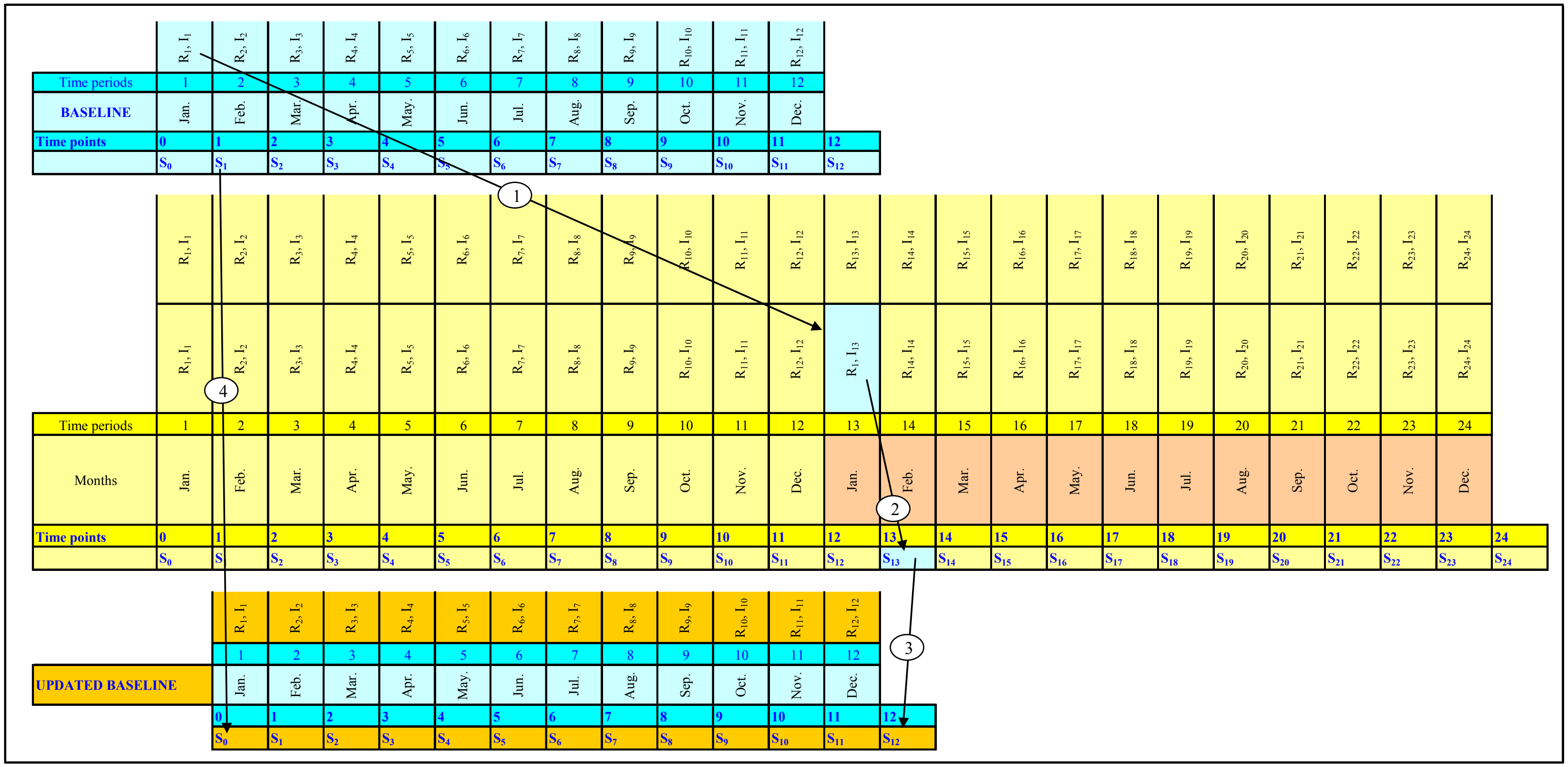

Figure 6.7 Illustration of Real-time Approach

6.3 Comparison and Results

Maximized energy amounts determined by using conventional approach and real time approach, and real operational energy amounts, as well, are shown in Table 6.1 and Figure 6.8.

Table 6.1. Comparison of maximized energy amounts

	Total Energy Generated (kWh)			
	Year 2005	***Year 2006***	**Years 2005 & 2006**	***Percentage (%)***
Conventional Approaches				
Year 2005	627,880		***627,880***	
Year 2005 and Year 2006, separately	627,880	694,302	***1,322,182***	*101.2*
Year 2005 and Year 2006, combined	644,685	711,341	***1,356,026***	*103.8*
Real-time Approach	627,880	639,223	***1,267,103***	*97.0*
Real Operational Data	620,971	685,627	***1,306,598***	*100.0*

Realized energy amounts was considered as a reference line in order to figure out the improvement and/or approximation to the realized/generated energy amount in the multi-reservoir system in the CRSP.

From the investigations of the multi-reservoir system in the CRSP for different considerations, it is evident that:

- Optimizing a 1-year period, year 2005; energy of 627,880 kWh was achieved; indicating an improvement of 1.1% when compared to that gained through realized/produced energy of 620,971 kWh,

- Next 1-year period, year 2006 was optimized and an energy amount of 694,302 kWh was determined which meant an improvement of 1.27% compared to the reference realized energy production,
- Two separately examined periods, years 2005 and 2006 have been optimized considering a single period of 2 years. Energy amount achieved after optimization of this multi-reservoir system considering 2-years period was 1,356,026 kWh which is by 3.8% improved from the energy actually realized/generated.
- When aforementioned real-time approach is performed; energy obtained in 2-years period is 1,267,103 kWh exhibiting an approximation of 3% to the realized/generated energy amount. It is to be noted that conventional approach is a posteriori, while real time approach proposed is online and is heavily dependent on the template baseline. In the event that the template baseline is formed embracing a long period, it is very likely that it will give better results. The realized values include tacit operational knowledge which have not been reflected on one year long data which have been used to establish the template.
- As the period considered for optimization increases, improvement in the amount of optimized energy rises. Energy amount received from optimization of 2-years period is by 2.6% higher than the sum of the optimized energy amounts obtained through optimization of two separate 1-year period.

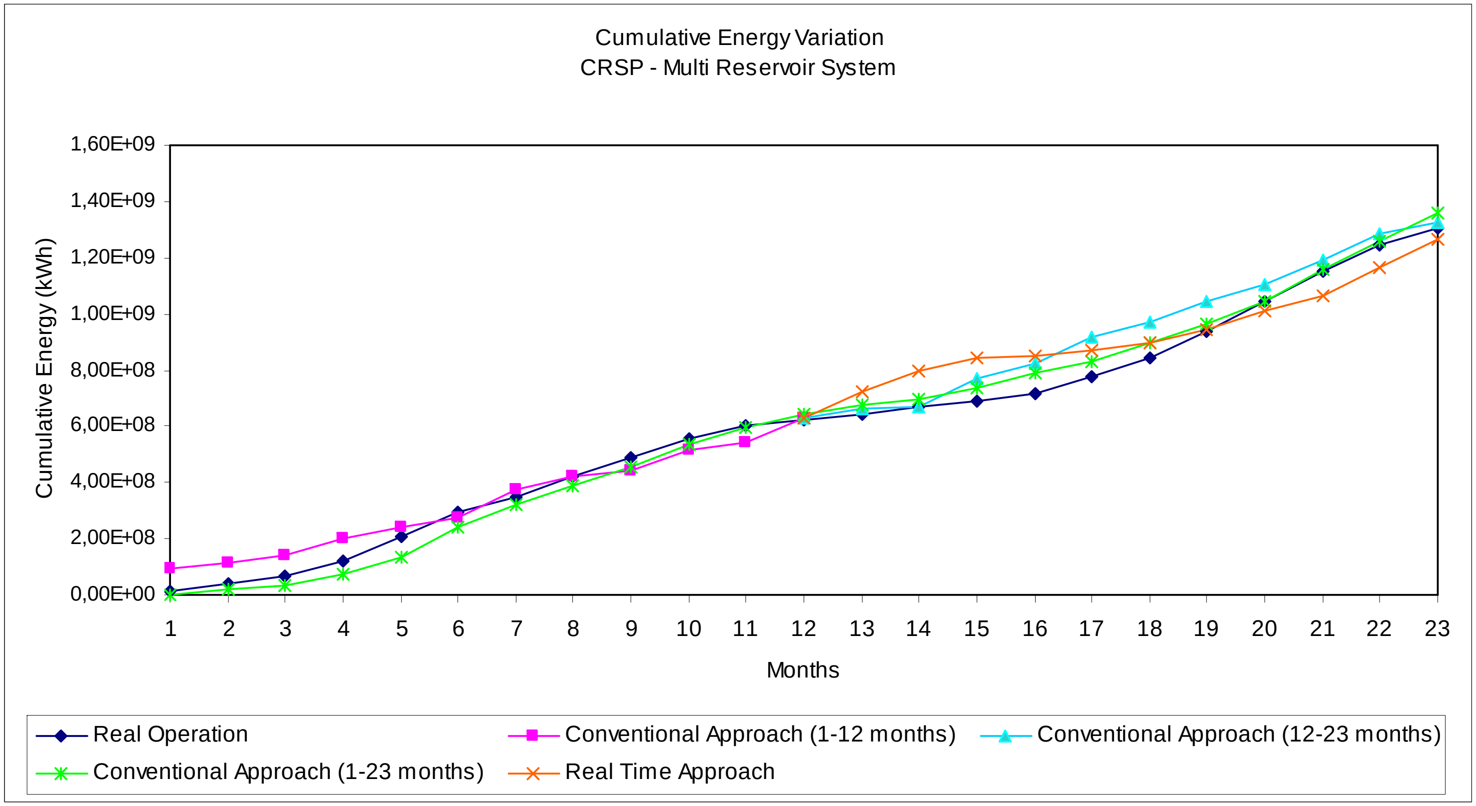

Figure 6.8 Comparison of Cumulative Energy for 2005-2006 in CRSP with respect to different considerations in optimization

CHAPTER 7

SUMMARY, CONCLUSION AND RECOMMENDATIONS

In the beginning, a literature survey was performed to achieve a sound comprehension and see the course of development in the fields of genetic algorithm and optimization in water resources problems, specifically optimization of reservoirs.

Subsequently, problem definition and accordingly construction of the mathematical model took place. Then, the mathematical framework defining the optimization problem with its objective, initial and boundary conditions was formed. Next step was adaptation of this mathematical framework to genetic algorithm which would be employed in the optimization process. After configuration of this adaptation, a computer code in Fortran programming language was constructed to solve this optimization problem by means of processors. This code would include the steps and principles which were necessary for genetic algorithm.

Following the construction of the code, its verification was necessary. A previously studied and proven reference multi-reservoir model with a known global optimum was used in verification of the code. The mathematical model of this reference system was embedded with its objective function, initial and boundary conditions, into the code constructed. Results achieved through employment of the code well-fit to the known global optimum of the system. Hence, the code has been verified. Beside verification, a sensitivity analysis was performed to see how the variation in the output of the model was with respect to the controlling parameters in the system.

Following verification process, the code was attempted to be employed in a real case multi-reservoir system under operation. The Blue Mesa, the Morrow Point and the Crystal Reservoirs within the Colorado River Storage Project in the U.S. Data which would be required in the optimization process have been obtained. Moreover, operational data belonging to the same period has also been determined. Since the objective function used in the verification was solely valid for the considered reference model, objective function has been modified to be applicable in the CRSP. Computer code was executed and the energy produced in the system was optimized by using two different approaches, conventional and real-time approach. In conventional approach, past data were utilized in optimization for one year and two years periods. Two years period has been considered in two different cases; 2-years time period as a sole time horizon, and in the second case 2- years time period was optimized in two separate 1-year time period. In real-time approach past data contributed in formation of a template baseline which is continuously updated in accordance with real-time data. As expected, the comparison of the results revealed that the energy amounts optimized by using conventional approach were higher than the energy produced in real operation. On the other hand, by using real-time approach, a close approximation to the real operational data has been achieved.

While conventional approaches make use of a priori data which belongs to occurred time periods, in real-time approach a combination of a priori and posterior data are used. A priori data constitutes a template baseline which will be updated by means of so called posterior real-time data. Template baseline is constructed benefiting from the past data. This baseline reflects the behavior of the flow regime in the considered system. In future researches, it can be further improved by being constructed upon past data belonging to a longer period of time. It is recommended for future researches that a learning capability is brought in this approach so as to cover a long period. After this study, it has been shown that genetic algorithms can successfully be applied in optimization of reservoir operations.

REFERENCES

- Ahmed, I., and Lansey, K. (2001). "Optimal operation of multi-reservoir systems under uncertainty." Proc, World Water and Environmental Resources Congress, D. Phelps, ed., Environmental and Water Resources Institute of ASCE, Reston, Va.

- Ahmed, J. A., and Sarma A.K. (2005). “Genetic Algorithm for Optimal Operating Policy of a Multipurpose Reservoir”, Water Resources Management, Vol. 19, No.2, 145-161

- Ansari, N., and Hou, E. (1999). Computational Intelligence for Optimization, Kluwer Academic, Boston, MA.

- Archibald, T., McKinnon, K., and Thomas, L. (1997). "An aggregate stochastic dynamic programming model of multireservoir systems." Water Resour. Res., 33(2), 333-340.

- Barros, M., Tsai, F., Yang, S.-L., Lopes, J., and Yeh, W. (2003). "Optimization of large-scale hydropower system operations." J. Water Resour. Plan. Manage., 129(3), 178-188.

- Becker, L., and Yeh, W. (1974). "Optimization of real-time operation of a multiple reservoir system." Water Resour. Res., 10(6), 1107—1112.

- Bellman, R. (1957). Dynamic programming, Princeton University Press, Princeton, N.J.

- Braga, B., Yeh, W., and Barros, M. (1991). "Stochastic optimization of multiple-reservoir-system operation." J. Water Resour. Plan. Manage., 117(4), 471-481.

- Cai, X., McKinney, D., and Lasdon, L. (2001). "Solving nonlinear water management models using a combined genetic algorithm and linear programming approach." Adv. Water Resour, 24(6), 667—676.

- Chandramouli, V, and Raman, H. (2001). "Multireservoir modeling with dynamic programming and neural networks."/ Water Resour. Plan. Manage., 127(2), 89-98.

- Cieniawski, S. E., Eheart, J. W., and Ranjithan, S. (1995). ‘‘Using genetic algorithms to solve a multiobjective groundwater monitoring problem.’’ Water Resour. Res., 31(2), 399–409.

- Crawley, P., and Dandy, G. (1993). "Optimal operation of multiple-reservoir system." J. Water Resour. Plan. Manage., 119(1), 1-17.

- Dandy, G. C., Simpson, A. R., and Murphy, L. J. (1996). ‘‘An improved genetic algorithm for pipe network optimization.’’ Water Resour. Res., 32(2), 449–458.

- Davidson, J. W., and Goulter, I. C. (1995). ‘‘Evolution program for the design of rectilinear branched distribution systems.’’ J. Comp. in Civ. Engrg., ASCE, 9(2), 112–121.

- De Jong K.A. (1975), An Analysis of the Behavior of a Class of Genetic Adaptive Systems, Doctoral dissertation, University of Michigan, Ann Arbor, MI

- Esat, V. and Hall, M.J., (1994), “Water resources system optimization using genetic algorithms”, Proc. 1st Int. Conf. on Hydroinformatics, Balkema Rotterdam, The Netherlands, 225-231

- Eshelman, L.J., Caruana, A., and Schaffer, J.D. (1989). Biases in the Crossover Landscape. Proc. of the Third Int. Conf. on Genetic Algorithms, J. David Schaffer (Ed.) (Morgan Kaufmann Publishers, San Mateo), 86–91.

- Eshelman LJ, and Schaffer JD (1993) Real-coded genetic algorithms and interval-schemata. In: Foundations of genetic algorithms 2, Whitley LD (ed) Morgan Kaufmann Publishers, San Mateo, CA, pp 187–202

- Fahmy, H. S., King, J. P., Wentzel, M. W., and Seton, J. A. (1994). ‘‘Economic optimization of river management using genetic algorithms.’’ Paper No. 943034, ASAE 1994 Int. Summer Meeting, Am. Soc. Of Agricultural Engrs., St. Joseph, Mich.

- Feng, C.W., Liu, L., and Burns, S.A. (1997) “Using genetic algorithms to solve construction time-cost trade-off problem”J.Comp. in Civ. Engrg., ASCE, 11(3), 184-189

- Francini, M. (1996) “Use of a genetic algorithm combined with a locak search method for the automatic calibration of conceptual rainfall-runoff models”, Journal of Hydro. Sci., 41(1), 21-40

- Gal, S. (1979) ‘‘Optimal management of a multi-reservoir water supply system.’’ Water Resour. Res., 15(4), 737–749.

- Goldberg, D. E., and Kuo, C. H. (1987). ‘‘Genetic algorithms in pipeline optimization.’’ J. Comp. in Civ. Engrg., ASCE 1(2), 128–141.

- Goldberg, D. (1989). Genetic algorithms in search optimization andma-chine learning, Addison-Wesley, Reading, Mass.

- Goldberg D.E. (1991). Real Coded Genetic Algorithms, Virtual Alphabets, and Blocking. Complex Systems 5, 139–167.

- Halhal, D., Walters, G. A., Ouazar, D., and Savic, D. A. (1997). ‘‘Water network rehabilitation with structured genetic algorithm.’’ J. Water Resour. Plng. and Mgmt., ASCE, 123(3).

- Hall, W. A., Harboe, W., Yeh, W. W.-G., and Askew, A. J. (1969). ‘‘Optimum firm power output from a two reservoir system by incremental dynamic programming.’’Water Resour. Ctr. Contrib., University of Calif., 130, 273–282.

- Heidari, M., Chow, V., Kotovic, P., and Meredith, D. (1971). "Discrete differential dynamic programming approach to water resources system optimization." Water Resour. Res, 7(2), 273-282.

- Holland, J. H. (1975). Adaptation in natural and artificial systems, MIT Press, Cambridge, Mass.

- Howson, H. R., and Sancho, N. G. F. (1975). ‘‘A new algorithm for the solution of multistate dynamic programming problems.’’ Math. Programming, 8(1), 104–116.

- Karamouz, M., Houck, M., and Delleur, J. (1992). "Optimization and simulation of multiple reservoir systems." J. Water Resour. Plan. Manage., 118(1), 71-81.

- Ko, S.-K., Fontane, D., and Labadie, J. (1992). "Multiobjective optimization of reservoir systems operations." Water Resour. Bull, 28(1), 111-127.

- Labadie, J.W., (2004) “Optimal operation of Multireservoir Systems: State-of-the-Art-Review”, Journal of Water Resources Planning and Management, March/April, 93-111

- Larson, R. E. (1968) State increment dynamic programming, Elsevier Science, New York

- Lehmer, D.H. (1951) “Mathematical methods in large-scale computing units. In Proc. 2nd Sympos. on Large-Scale Digital Calculating Machinery”, Cambridge, MA. Harvard University Press, 141-146

- Li, H. and Love, P.E.D. (1998). “Site level facilities layout using genetic algorithms” J.Comp.in Civ.Engrg., ASCE, 12(4), 227-231

- Liu, P., Guo, S., Xiong, L., Li, W., and Zhang, H., (2006). “Deriving Reservoir Refill Operating Rules by Using the Proposed DPNS Model”, Journal of Water Resources Management, Vol. 20, No.3, 337-357.

- Loucks, D., and Dorfman, P. (1975). "An evaluation of some linear decision rules in chance-constrained models for reservoir planning and operation." Water Resour. Res., 11(6), 777—782.

- Marino, M. A., and Mohammadi, B. (1983). ‘‘Reservoir operation by linear and dynamic programming.’’ J. Water Resour. Plng. and Mgmt., ASCE, 109(4), 303–319.

- McKinney, D.C., and Lin, M.D. (1994). “Genetic algorithm solution of groundwater management models” Water Resour.Res. 30(6), 1897-1906

- Michalewicz, Z. (1992). Genetic algorithms 1 data structures = evolution programs, Springer, New York.

- Michalewicz, Z. (1996). Genetic algorithms+data structures= evolution programs, Springer, Berlin.

- Murray, D. M., and Yakowitz, S. (1979). ‘‘Constrained differential dynamic programming and its application to multireservoir control.’’ Water Resour. Res., 15(5), 1017–1027.

- Murphy, L. J., Simpson, A. R., and Dandy, G. C. (1993). ‘‘Design of a network using genetic algorithms.’’ Water, 20, 40–42.

- Oliveira, R., and Loucks, D. (1997). “Operating rules for multireservoir systems.” Water Resour. Res., 33(4), 839-852.

- Park, S.K., and Miller K.W. (1988). “Random number generators: good ones are hard to find.” Comm. ACM, 31:1192-1201.

- Ralph, A., and Wurbs, A. (1993) “Reservoir System Simulation and Optimization Models” J.of Water Resources Planning and Management, Vol.119, No.4, July/August 1993

- Raman, H., and Chandramouli, V. (1996). ‘‘Deriving a general operating policy for reservoirs using neural network.’’ J. Water Resour. Plan. Manage., 122(5), 342–347.

- Ritzel, B. J., Eheart, J. W., and Ranjithan, S. (1994). ‘‘Using genetic algorithms to solve a multiple objective groundwater pollution problem.’’ Water Resour. Res., 30(5), 1589–1603.

- Sharif, M., and Wardlaw, R. (2000). "Multireservoir systems optimization using genetic algorithms: Case study." J. Comput. Civ. Eng., 14(4), 255-263.

- Simpson, A. R., Dandy, G. C., and Murphy, L. J. (1994). ‘‘Genetic algorithms compared with other techniques for pipe optimization.’’ J. Water Resour. Plng. and Mgmt., ASCE, 120(4), 423–443.

- Soh, C.K and Yang, J. (1996) “Fuzzy controlled genetic algorithm search for shape optimization” J.Comp. in Civ. Engrg., ASCE, 10(2), 143-150

- Syswerda G. (1989). Uniform Crossover in Genetic Algorithms, In Proceedings of the rd International Conference on Genetic Algorithms, J.D. Scha#er (eds), Morgan Kaufmann Publishers, CA, 2-9.

- Takeuchi, K., and Moreau, D. H. (1974). ‘‘Optimal control of multiunit interbasin water resource system.’’Water Resour. Res., 10(3), 407–414.

- Trott, W. J., and Yeh, W. W.-G. (1973). ‘‘Optimization of multiple reservoir system.’’ J. Hydr. Div., ASCE, 99(10), 1865–1884.

- Wardlaw, R., and Sharif, M., (1999). “Evaluation of Genetic Algorithms for Optimal Reservoir System Operations”, *Journal Of Water Resources Planning and Management,* February, 25-33

- Wang, Q. J. (1991). ‘‘The genetic algorithm and its application to calibrating conceptual rainfall-runoff models.’’ Water Resour. Res., 27(9), 2467–2471.

- Wang, L., and Zheng, D. Z. (2002) “A Modified Genetic Algorithm for Job Shop Scheduling “, The International Journal of Advanced Manufacturing Technology, Springer London, 72-76

- Wei, L., Mei, Z., Guangming, W. and Guang, M., (2005) “Truss optimization on shape and sizing with frequency constraints based on genetic algorithm”, Computational Mechanics, Vol 35, No.5, 361-368.

- Wright, A. H (1991)., “Genetic Algorithms for Real Parameter Optimization”, In Foundations of Genetic Algorithms, . E. Rawlins (Ed.), Morgan Kaufmann, pp. 205-218.

- Wurbs, R. A. (1993). ‘‘Reservoir-system simulation and optimization models.’’ J. Water Resour. Plng. and Mgmt., ASCE, 119(4), 455–472.

- Yakowitz, S. (1982). ‘‘Dynamic programming applications in water resources.’’ Water Resour. Res., 18(4), 673–696.

- Yeh, W. (1985). "Reservoir management and operations models: A state-of-the-art review." Water Resour. Res, 21(12), 1797-1818.

- Yeh, W., and Becker, L. (1982). "Multi-objective analysis of multireservoir operations." Water Resour. Res., 18(5), 1326-1336.

- Yeh, W. W.-G., Becker, L., and Chu, W. S. (1979). ‘‘Real time hourly reservoir operation.’’ J. Water Resour. Plng. and Mgmt., ASCE, 105(2), 187–203.

- Young, G. K. (1967). ‘‘Finding reservoir operating rules.’’ J. Hydr. Div., ASCE, 93(6), 297–321.

APPENDIX A

OVERVIEW OF SELECTION TECHNIQUES IN GENETIC ALGORITHMS

A.1 Tournament Selection

Another selection technique is “tournament selection,” randomly selected pairs of chromosomes “fight” to become parents in the mating pool through their fitness function value (Goldberg, 1989). Tournament selection runs a "tournament" among a few individuals chosen at random from the population and selects the winner in accordance with their fitness values, such that the one with the best fitness is selected for crossover. Selection pressure can be easily adjusted by changing the tournament size. If the tournament size is larger, weak individuals have a smaller chance to be selected. In general tournament selection n individuals are selected at random and the fittest is selected. The most common type of tournament selection is binary tournament selection, where just two individuals are selected.

Roulette Wheel Selection and Tournament Selection techniques are considered to be more popular than the other techniques. However, there are many other selection techniques. Among the other selection techniques, are the elitist selection, scaling selection, rank selection, generational selection, Steady-state selection and hierarchical selection technique. Brief introduction of those selection techniques are given below.

A.2 Elitist selection

The most fit members of each generation are guaranteed to be selected. (Most GAs do not use pure elitism, but instead use a modified form where the single best, or a few of the best, individuals from each generation are copied into the next generation just in case nothing better turns up.)

A.3. Scaling selection

As the average fitness of the population increases, the strength of the selective pressure also increases and the fitness function becomes more discriminating. This method can be helpful in making the best selection later on when all individuals have relatively high fitness and only small differences in fitness distinguish one from another.

A.4. Rank selection

Each individual in the population is assigned a numerical rank based on fitness, and selection is based on this ranking rather than absolute difference in fitness. The advantage of this method is that it can prevent very fit individuals from gaining dominance early at the expense of less fit ones, which would reduce the population's genetic diversity and might hinder attempts to find an acceptable solution.

A.5. Generational selection

The offspring of the individuals selected from each generation become the entire next generation. No individuals are retained between generations.

A.6 Steady-state selection

The offspring of the individuals selected from each generation go back into the pre-existing gene pool, replacing some of the less fit members of the previous generation. Some individuals are retained between generations.

A.7. Hierarchical selection

Individuals go through multiple rounds of selection each generation. Lower-level evaluations are faster and less discriminating, while those that survive to higher levels are evaluated more rigorously. The advantage of this method is that it reduces overall computation time by using faster, less selective evaluation to weed out the majority of individuals that show little or no promise, and only subjecting those who survive this initial test to more rigorous and more computationally expensive fitness evaluation.

APPENDIX B

OVERVIEW OF CROSSOVER TECHNIQUES IN GENETIC ALGORITHMS

$C_1 = \left(c_1^1,.....,c_n^1\right)$ and $C_2 = \left(c_1^2,.....,c_n^2\right)$ are two chromosomes that have been selected to apply the crossover operator to them.

B.1 Two-point crossover

Two points of crossover are randomly selected $i, j \in \left(1,2,....,n-1\right)$ provided that i<j and the segments of the parent, defined by them, are exchanged for generating two offspring (Eshelman et al., 1989):

$$H_1 = \left(c_1^1, c_2^1,....,c_i^2, c_{i+1}^2,.....,c_j^2, c_{j+1}^1,.....,c_n^1\right) \quad \text{B.1}$$

$$H_2 = \left(c_1^2, c_2^2,....,c_i^1, c_{i+1}^1,.....,c_j^1, c_{j+1}^2,.....,c_n^2\right) \quad \text{B.2}$$

B.2. Geometrical crossover

Two offspring are built,

$$H_k = \left(h_1^k,.....,h_i^k.....,h_n^k\right) \text{ k=1,2} \quad \text{B.3}$$

where

$$h_i^1 = c_i^{1^{\omega}} \cdot c_i^{2^{(1-\omega)}} \quad \text{B.4}$$

$$h_i^2 = c_i^{2^{\omega}} \cdot c_i^{1^{(1-\omega)}} \quad \text{B.5}$$

For $\omega \in [0,1]$.

Below is shown the region for geometric crossover in Figure B.1 (Michalewicz, 1996).

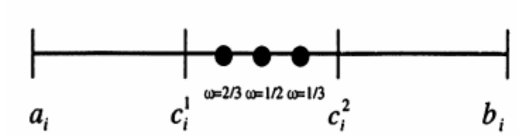

Figure B.1 Geometrical crossover with different values for $\omega \in [0,1]$

APPENDIX C

BENEFIT FUNCTION CONSTANTS OF THE FOUR RESERVOIR PROBLEM

Table C.1 Benefit function used to calculate the optimal policies of the system considered proposed by Heidari et al. (1971)

t	$b_1(t)$	$b_2(t)$	$b_3(t)$	$b_4(t)$	$b_5(t)$
0	1.1	1.4	1.0	1.0	1.6
1	1.0	1.1	1.0	1.2	1.7
2	1.0	1.0	1.2	1.8	1.5
3	1.2	1.0	1.8	2.5	1.9
4	1.8	1.2	2.5	2.2	2.0
5	2.5	1.8	2.2	2.0	2.0
6	2.2	2.5	2.0	1.8	2.0
7	2.0	2.2	1.8	2.2	1.9
8	1.8	2.0	2.2	1.8	1.8
9	2.2	1.8	1.8	1.4	1.7
10	1.8	2.2	1.4	1.1	1.6
11	1.4	1.8	1.1	1.0	1.6

APPENDIX D

COLORADO RIVER STORAGE PROJECT FACTS

General Description

The Gunnison River is a tributary of the Colorado River, approximately 180 mi (290 km) long, in the U.S. state of Colorado.

It rises in west central Colorado, in eastern Gunnison County, formed by the confluence of Taylor and East rivers. Just past the town of Gunnison, the river begins to swell into the expanse of Blue Mesa Reservoir, a 40 mile (64 km) long reservoir formed by Blue Mesa Dam, where it receives the Lake Fork of the Gunnison. Just downstream it is dammed again to form Morrow Point Reservoir, then just downstream of that dammed for the final time to form Crystal Reservoir. The reservoirs produce hydroelectric power and supply water for the surrounding ares for both municipal and irrigation use. The reservoirs are the upper part of the Black Canyon of the Gunnison, one of the longest, narrowest, and deepest gorges in the world. Below Crystal Dam it begins to roar through massive cataracts and flows through the deepest part of the gorge. At the outlet of the canyon it receives the North Fork River, then downstream near Delta is joined by the Uncompahgre River. It then winds through desert canyonlands until it empties into the Colorado near Grand Junction, carrying almost as much water as the former.

The Colorado River Storage Project (CRSP) provides for the comprehensive development of the Upper Colorado River Basin. The project furnishes the long-time regulatory storage needed to permit States in the upper basin to meet their flow obligation at Lees Ferry, Arizona, as defined in the Colorado River Compact, and still utilize their apportioned water.

Water stored by the project provides a portion for direct use in the upper basin. Sediment and flooding are better controlled and recreation development and fish and wildlife conservation have benefited. Because of project development, a significant amount of electrical energy is produced to meet the needs of the upper basin and adjacent areas.

The project includes four storage units: Glen Canyon on the Colorado River in Arizona near the Utah border; Flaming Gorge on the Green River in Utah near the Wyoming border; Navajo on the San Juan River in New Mexico near the Colorado border; and the Wayne N. Aspinall Storage Unit on the Gunnison River in west-central Colorado. Figure D.1 and D.2 shows the map of the region under consideration.

Authorized with, but not part of, are a number of participating projects which will share in the power revenues of the larger project to help pay for irrigation construction costs. These participating projects are listed in the authorization paragraphs.

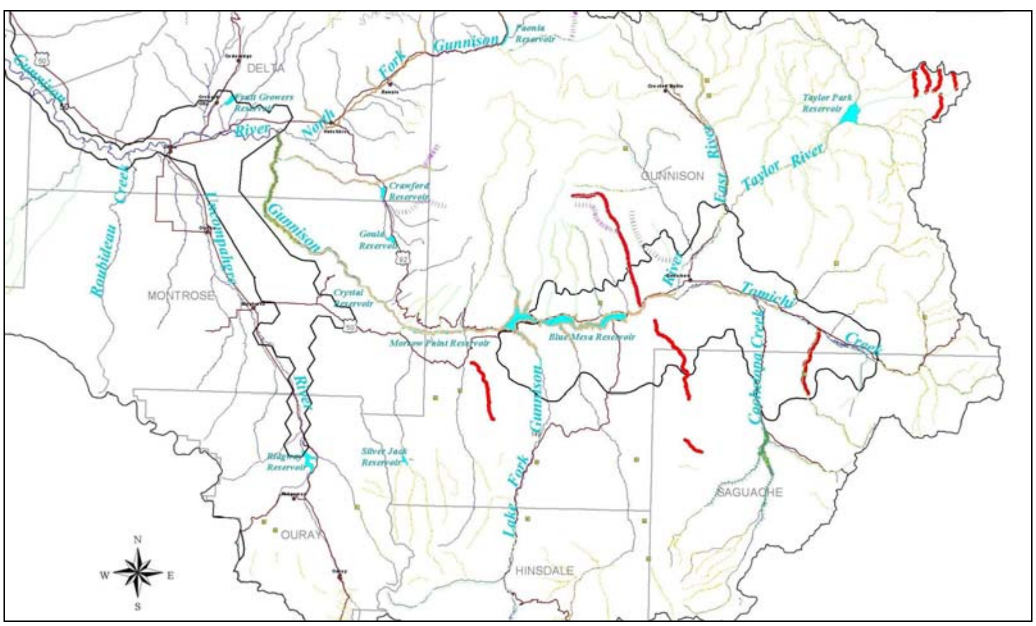

Figure D.1 Map under consideration ***(www.coloradowatertrust.org)***

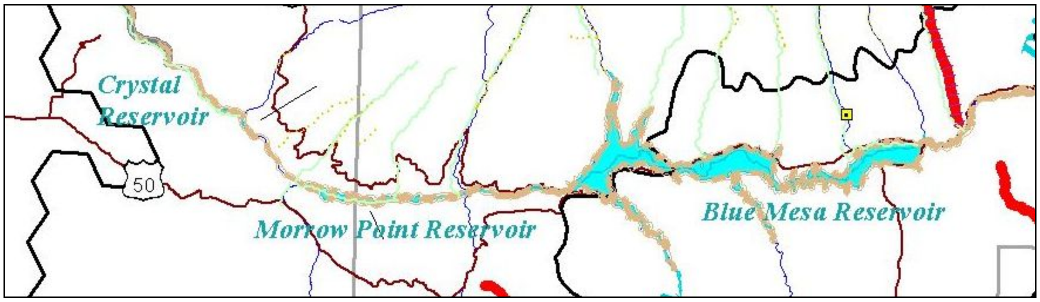

Figure D.2 Map of Basin Considered (www.coloradowatertrust.org)

Aspinall Unit

The Aspinall Unit developed the water storage and hydroelectric power generating potential along a 40-mile (64 km) section of the Gunnison River in Colorado by the construction of three dams and powerplants: Blue Mesa, Morrow Point, and Crystal.

Blue Mesa Dam, Reservoir, and Powerplant

Blue Mesa Dam is on the Gunnison river about 30 miles below Gunnison, and 1.5 miles (2.4 km) below Sapinero, Colorado. The zoned earthfill embankment has a structural height of 390 feet (119 m), a crest length of 785 feet (239 m), and a volume of 3,080,000 cubic yards (2,354,829 cubic meters) of materials.

The spillway consists of a concrete intake structure with two 25 (7.62 m)- by 33.5 (10.21 m)-foot radial gates, concrete-lined tunnel, concrete flip bucket structure, and stilling basin. Maximum discharge of the spillway is 34,000 cubic feet (963 cubic meters) per second.

The outlet works consists of an intake structure, tunnel, and manifold anchor block. The outlet works is controlled by one 16 (4.9 m) - by 18 (5.5 m)-foot fixed-wheel gate in the intake structure and by two 84-inch ring-follower gates and two 84-inch (2.13 m) hollow-jet valves in a gate house at the terminus of the outlet conduits. Maximum discharge from the outlet works is 5,000 cubic feet (141.6 cubic meters) per second at maximum water surface elevation, with two 84-inch (2.13 m) hollow-jet valves 62 percent open.

Blue Mesa Reservoir has a total capacity of 940,700 acre-feet (1,161 million cubic meters) and an active capacity of 748,430 acre-feet (923 million cubic meters). At maximum water surface elevation, the reservoir occupies 9,180 acres (3715 hectares).

The Blue Mesa Powerplant consists of two 30,000-kilowatt generators, driven by two 41.55-horsepower turbines. Each Turbine is designed to operate at a maximum head of about 360 feet (109 m).

One 16-foot-diameter (4.9 m) penstock conveys water to the two turbines and also carries water for the outlet works. After branching from the main penstock, each of the penstock laterals is controlled by 156-inch (3.9 m) butterfly valves. The main penstock is reduced by a wye branch to the outlet works control valves.

Plant Facts: The Blue Mesa Powerplant consists of two 43,200-kilowatt generators, driven by two 41,500-horsepower turbines. Each turbine is designed to operate at a maximum head of about 360 feet (109 m).
River: Gunnison River
Location 1.5 mi (2.4 km) below Sapinero
Turbine Type: Francis
Installed Capacity: 86,400 kW
Rated Head: 332 feet (101.2 m)
Year of Initial Operation: 1967
Hydraulic Height: 33.4 ft (10.18 m)
Crest Elevation of Dam: 7528.0 ft (2294.5 m)
Structural Height of Dam: .   502.0 ft (153 m)

Crest Length: 785.0 ft (239.3 m)

Top of Joint Use: 7519.4 ft (2291.9 m)

Top of Active Conservation: 7519.4 ft (2291.9 m)

Top of Inactive Conservation: 7393.0 ft (2253.4 m)

Spillway Crest: 7487.9 ft (2282.3 m)

Top of Dead Storage: 7358.0 ft (2242.7 m)

Streambed at Dam Axis : 7186.0 ft (2190.3 m)

Morrow Point Dam, Reservoir, and Powerplant

Morrow Point Dam, 12 miles (19.3 km) downstream from Blue Mesa Dam, is Reclamation's first thin-arch, double-curvature dam. It is 468 feet (142.6 m) high, 52 feet (15.8 m) thick at the base, and 12 feet (3.65 m) thick at the crest. The dam has a crest length of 720 feet (219.5 m) and a volume of 360,000 cubic yards (275,240 cubic meters) of concrete.

The spillway consists of four orifice-type openings in the top central part of the dam, providing a free-fall discharge higher than 350 feet (106.68 m) to the concrete stilling basin at the toe of the dam. Each of the four spillway openings is controlled by a 15 (4.57 m)- by 16.83-foot (5.13 m) fixed-wheel gate. Maximum capacity of the spillway is 41,000 cubic feet (1161 cubic meters) per second.

The outlet works consists of one stainless-steel lined 4-foot-square (0.37 square meters) conduit through the dam. Control is by two 3.5-square-foot (0.32 square meters) slide gates. Discharge capacity of the outlet works is 1,500 cubic feet per second.

Reservoir capacity behind Morrow Point Dam is 117,190 acre-feet (144.5 million cubic meters) at maximum water surface. The active capacity is 42,120 acre-feet (51.9 million cubic meters). Surface area for Morrow Point Reservoir is 817 acres (330 hectares) at an elevation of 7,160.0 ft (2182.3 m).

The powerplant chamber is tunneled into the canyon wall in the left abutment about 400 feet below the ground surface. The powerplant chamber is 231 feet (70.4 m) long and 57 feet (17.4 m) wide with a height ranging from 65 (19.8 m) to 134 feet (40.8 m). There are two 60,000-kilowatt generators driven by two 83,000-horsepower turbines. The power penstocks consist of 13.5-foot-diameter (4.1 m) steel liners in 18-foot-diameter (5.5 m) tunnels.

River: Gunnison River

Turbine Type: Francis

Installed Capacity: 173,334 kW

Year of Initial Operation: 1970

Rated Head: 396 feet (120.7 m)

Crest Elevation 7165.0 ft (2183.9 m)

Structural Height 468 ft (142.6 m)

Hydraulic Height 400 ft (121.9 m)

Crest Length . 724 ft (220.7 m)

Crest Width . 12 ft (3.65 m)

Base Width . 52 ft (15.85 m)

Volume of Concrete . 365,180 cu yd (279,200 cubic meters)

Location . 22 mi (35.4 km) from Montrose, CO

Crystal Dam, Reservoir, and Powerplant

Crystal Dam is located 6 miles (9.7 km) downstream from Morrow Point Dam and approximately 20 miles (32.2 km) east of Montrose, Colorado. The dam is a double-curvature thin-arch type, 323 feet (98.45 m) high, with a crest length of 620 feet (188.98 m), and a volume of 154,400 cubic yards (118,000 cubic meters) of materials.

The spillway consists of an ungated ogee crest on the right side of the dam and a plunge pool at the toe of the dam. The crest is at an elevation of 6,756.0 feet (2059.2 m), 1 foot (30.5 cm) above normal water surface. The plunge pool is unlined except for a downstream retaining wall to contain the river fill material.

Water is conveyed from the reservoir to the hydraulic turbine by an 11.5-foot (3.5 m) - diameter concrete penstock, the lower portion of which is steel lined. The intake structure consists of a metal trashrack, a 10.58 (3.22 m)- by 17.27-foot (5.26 m) bulkhead gate, an 8.33 (2.53 m)- by 13.58-foot (4.13 m) fixed-wheel gate, and a transition. The fixed-wheel gate is provided for emergency closure and for inspection and maintenance of the penstock. Water from the turbine exits through the draft tube to the tailrace.

The river outlets consist of an intake structure on the upstream face of the dam and two 54-inch (137 cm) pipes through the dam and powerplant. The 54-inch (137 cm) ring-follower emergency gates and 48-inch (122 cm) jet-flow regulating gates in the powerplant control outlet flows. The intake structure includes a metal trashrack, a concrete arch conduit to convey water to the 54-inch pipes, and provisions for installing a bulkhead gate. The Morrow Point Dam river outlet bulkhead gate can be used to close off the outlet pipes for inspection or maintenance.

The reservoir has a total capacity of 25,236 acre-feet (31.11 million cubic meters) and an active capacity of 12,891 acre-feet (15.89 million cubic meters) at an elevation of 6,700 ft (2042.1 m), with a surface area of 301 acres (121.8 hectares).

The powerplant, completed in 1978, has a generating capacity of 28,000 kilowatts from one unit driven by a 39,000-horsepower hydraulic turbine. It is connected to the main CRSP transmission system at the Curecanti substation by a 115-kilovolt line.
Turbine Type: Francis
Installed Capacity: 31,500 kW
Year of Initial Operation: 1978
Rated Head: 207 feet (63.1 m)
Location . 20 mi (32.2 km) E of Montrose, CO
Crest Elevation 6772.0 ft (2064.1 m)
Structural Height 323 ft (98.45 m)
Hydraulic Height 227 ft (69.19 m)
Crest Length . 635 ft (193.5 m)

Crest Width . 10 ft (3 m)

Base Width 29 ft (8.83 m)

Volume of Concrete . 147,000 cu yd (112,400 cubic meters)

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Hınçal, Onur
Nationality: Turkish (TC)
Date and Place of Birth: 23 May 1976 , Edirne
Marital Status: Single
Phone: +90 542 595 62 87
email: ohincal@gmail.com

EDUCATION

Degree	**Institution**	**Year of Graduation**
MS	METU Civil Engineering	2000
BS	METU Civil Engineering	1998
High School	Edirne Anadolu High School, Edirne	1994

WORK EXPERIENCE

Year	**Place**	**Enrollment**
1999- Present	SOYAK İnşaat ve Ticaret A.Ş.	Project Coordinator
1998-1999	METU Department of Civil Engineering	Research Assistant

FOREIGN LANGUAGES

Advanced English

PUBLICATION

Aydin I., Ger A. M. and Hincal O. "Measurement of Small Discharges in Open Channels by Slit Weir", Journal of Hydraulic Engineering, Volume 128, Issue 2, pp. 234-237 (February 2002)

HOBBIES

Tennis, Computer Technologies, Movies