DEVELOPMENT OF OPTIMAL OPERATING POLICY FOR PAGLADIA MULTIPURPOSE RESERVOIR

Submitted in Partial Fulfillment of the Requirements for the Degree of
DOCTOR OF PHILOSOPHY

By
JURAN ALI AHMED

DEPARTMENT OF CIVIL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI
JULY 2004

DEVELOPMENT OF OPTIMAL OPERATING POLICY FOR PAGLADIA MULTIPURPOSE RESERVOIR

A Thesis
Submitted in Partial Fulfillment of the Requirements
for the Degree of
DOCTOR OF PHILOSOPHY

By
JURAN ALI AHMED

DEPARTMENT OF CIVIL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI
JULY 2004

DEDICATED TO MY PARENTS AND WIFE

CERTIFICATE

It is certified that the work contained in the thesis entitled “DEVELOPMENT OF OPTIMAL OPERATING POLICY FOR PAGLADIA MULTIPURPOSE RESERVOIR” by Juran Ali Ahmed, Roll Number 01610402, a student in the Department of Civil Engineering, Indian Institute of Technology Guwahati for the award of the degree of Doctor of Philosophy has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Supervisor

Dr. Arup Kumar Sarma
Assistant Professor
Department of Civil Engineering
I.I.T. Guwahati
July 2004

ACKNOWLEDGEMENTS

I would like to place on record my deep sense of gratitude and sincere thanks to my thesis Supervisor **Dr. Arup Kumar Sarma**, **Assistant Professor, Department of Civil Engineering** for his invaluable guidance and full hand cooperation throughout the all aspects of this research work. I also admire his patient explanation of the concepts and basic principles.

I am grateful **to Dr. S. Talukdar, Head of the department of Civil Engineering** for necessary facilities and support. I thank my **Doctoral committee members** for their useful suggestions.

I am indebted to **Dr. V. Chandramouli**, **Dr. Chandan Mahanta**, and **Dr. G. Barua**, **Assistant Professors, Department of Civil Engineering,** for their valuable help and suggestions.

My special thanks to **Dr. B. Talukdar**, **Assistant Professor, Assam Engineering College** for his generous helping attitudes.

I am thankful to **Brahmaputra Board, Government of India**, for providing the necessary data during the research work.

My sincere thanks are also due to all the **Faculty members** of **Hydraulics and Water Resources Engineering** for their keen interest and suggestions.

I wish to thank **Er. K. Ahmed**, Executive Engineer, **Water Resource Department, Government of Assam**, for his kind support.

I express my sincere thanks to the **staff members** of the Civil Engineering Department, IIT Guwahati for their kind cooperation during my work.

I express my special thanks to **Er. B. Choudhury**, **Er. B. Barman, Er. R. Goswami, Er. I. Gogoi, Er. M. Das, Er. S. Chakma** and **Er. D. Devi** for their kind support.

I wish to thank **those** who directly or indirectly helped me for smooth completion of the research work.

I have great pleasure in thanking my family members particularly my wife **Jahanara Ahmed** who took the pain for early completion of my research work at IIT Guwahati.

JURAN ALI AHMED

ABSTRACT

Pagladia multipurpose reservoir, located on the river Pagladia, a major north bank tributary of Brahmaputra, proposes to serve three purposes, namely, flood control, irrigation and power generation. In order to achieve these, a proper operating policy of the reservoir is imperative.

Recent researches have revealed the potential of heuristic methods in deriving reservoir-operating policy. In this study, the potential of Genetic Algorithm (GA) and Artificial Neural Network (ANN) in deriving an optimal operating policy has been explored through their application in the Pagladia multipurpose reservoir. Efficiency of the policies derived by these recent techniques has been assessed through their critical comparison with policies derived by some long-established techniques.

To have the advantage of using a long streamflow series in the development of a reservoir operating policy, an ANN based synthetic streamflow generation model has been developed and compared with the Thomas-Fiering and Autoregressive Moving Average (ARMA) models. Synthetic streamflow series generated by the ANN based model has been used in the development of operating policies, as its statistics have been found to be in better agreement with those of the observed historic series.

For solving the reservoir optimization problem for Pagladia multipurpose reservoir, deterministic Dynamic Programming (DP) has first been solved. Both multiple linear regression and ANN have been used to infer general monthly operating policy from the DP results, and these models are being termed as DPR and DPN models respectively in this study. Stochastic Dynamic Programming (SDP)

model, which uses an explicit stochastic optimization technique, has been developed next for deriving monthly optimal operating policy for the Pagladia multipurpose reservoir. Finally, GA, which is of recent origin and has the capability of solving complex optimization problem, has been used to derive optimal monthly operating policy for the reservoir. Performance of all the operating policies developed by different models has been analyzed on the basis of the reservoir simulation results for 228 months of historic streamflow series (1977-1996). For making a fair comparison among all the models, a total of eight performance criteria covering different aspects of reservoir operation, have been used.

The study has shown that GA, which is a robust optimization technique, is quite capable of developing multipurpose reservoir operating policy and has been found to give the most efficient operating policy for the Pagladia multipurpose reservoir. Policies derived by SDP and GA have been found to be competitive in respect of some of the performance criteria. GA out performs the DPR and the DPN models developed in this study. Although performances of different models vary with different performance criteria, considering overall performance and giving priority to irrigation, the application of operating policy derived by GA model has been suggested to be the most appropriate for the Pagladia multipurpose reservoir.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF NOTATIONS

a, b, c = Coefficients of multiple linear regression

A_{at} = Mean reservoir area during time period t

A_i, B_i = Fourier coefficients

α = Momentum factor

α = Significance level for null hypothesis test

b_j = Bias for the j^{th} neuron.

B_{kilt} = system performance (squared deficit of release from demand) in the period t for initial storage volume corresponding to index k, inflow volume corresponding to index i, and final storage volume corresponding to index l

β_q = A parameter of simulated binary crossover

c_1, c_2 = Children 1 and 2

D_t = Demand in Mm^3 in time period t

D_t = Dependent stochastic component

δ_j = A factor depending on whether neuron j is an output neuron or a hidden neuron

δ_q = A parameter of mutation

η = Learning rate

η = Efficiency of power plant

η_c = Distribution index for crossover

η_m = Distribution index for mutation

e_t = Evaporation rate in meter per unit surface area of the reservoir during time period t

E_t = Evaporation loss in Mm^3 during time period t

F = Fitness value

f_{max} = Objective function value of the worst feasible solution in the population

g_t = Normalized constraint

H_t = Average head in meter of the reservoir during time period t

i, j = Index of characteristic inflow volume in time period t and t+1

ID_t = Irrigation demand in Mm^3 during time period t

k, l = Index of characteristic storages volumes at the beginning of time period t and t+1

k = lags in time series of streamflow

K = Total number of autocorrelation considered for residual series

KWH_t = Power generation in kWh in time period t

m = Number of period in a year

μ = Population mean

μ_v = Mean of the residual series of the transformed streamflow

n = Total number of period remaining including the current period before reservoir operation terminates

n = Number of months for synthetic streamflow generation.

N = Maximum number of piecewise linear release rule function considered to define the operating policy in GA model

N = Total number of observations of time series data

p = total number of training pattern.

p = Number of autoregressive parameters

P_{ij}^{t} = Probability that inflow Q_j occurs in time period t+1, given a known inflow of Q_i in period t

P_t = Periodic component of streamflow time series

$PC_{k,k}$ = Partial autocorrelation function at lag k

PD_t = Water requirement in Mm^3 for power generation during time period t

Φ = Autoregressive parameters

q =Number of output nodes

q = Number of moving average parameters

$\overline{q}_m$ = Observed average of month *m* of historic monthly transformed streamflow series

$q_{m,y}$ = Transformed streamflow in month *m* and year y.

$Q_{m,y}$ = Streamflow in month *m* and year y

Q_t = Inflow in Mm^3 during time period t

r_k = Autocorrelation function at lag k

R_t = Release in Mm^3 during time period t

R_{pt} = Release in Mm^3 meant for power generation in any time period t

R_t^{max} = Maximum release in time period t

R_t^{min} = Minimum release in time period t

s = Epoch/training iteration number

S = Sample standard deviation

s_m^2 = Observed variance of month *m* of historic monthly transformed series

S^{max} = Storage capacity of the reservoir at FRL

S^{min} = Storage capacity of the reservoir at MDDL

S_t = Storage in Mm^3 at the beginning of time period t

S_{t+1} = Storage in Mm^3 at the end of time period t or beginning of time period t+1

S_t = Stochastic component of time series.

σ_0^2 = Variance of the population

σ_v^2 = Variance of independent stochastic series of streamflow

t = Time index in month

T = Total number of time periods.

T = Number of month in a year

T_t = Trend component (a deterministic component)

θ = Moving average parameters

V_t = Independent stochastic component

w_{ji} = Weight of the connection joining the j^{th} neuron in the hidden layer with the i^{th} neuron in the input layer,

W = Total water availability during a month

x_i = Value of the i^{th} neuron in the input layer

y_j = Output from the j^{th} neuron in the hidden layer

$y_j^{(t)}$ = Standardized target value for pattern j

y_1, y_2 = Parent 1 and 2

y_l, y_u = Lower and upper boundary of a parameter

Z_t = Time series variable

$\overline{Z}$ = Mean value of the time series Z_t

ζ_t = Independent standard normal random variables

LIST OF ABBREVIATIONS

AIC	Akaike information criteria
ANN	Artificial Neural Network
ARMA	Autoregressive moving average
BCM	Billion Cubic Meter
BP	Backpropagation
BSDP	Binary State Dynamic Programming
CCA	Cultivable Command Area
DDP	Differential Dynamic Programming
DDDP	Discrete Differential Dynamic Programming
Discrete DP	Discrete Dynamic Programming
DP	Dynamic Programming
DPN	Model, which uses artificial neural network for deriving general operating policies form deterministic dynamic programming optimization result
DPR	Model, which uses multiple linear regression for deriving general operating policies from deterministic dynamic programming optimization result.
DPSA	Dynamic Programming with Successive Approximations
EL	Elevation above mean sea level
FDP	Folded Dynamic Programming
FRL	full reservoir level
GA	Genetic Algorithm
GCA	Gross Command Area
IDP	Incremental Dynamic Programming

IDPSA	Incremental Dynamic Programming with successive approximations
LDR	Linear Decision Rules
LP	Linear Programming
MWL	Maximum water level
MDDL	Minimum draw down level
MIDP	Multilevel Incremental Dynamic Programming
MRE	Mean relative error
MSE	Mean square error
NLP	Nonlinear Programming
SBX	Simulated binary crossover
SDP	Stochastic Dynamic Programming

CHAPTER 1

INTRODUCTION

1.1 PURPOSE OF THE STUDY

The Northeastern region of India accounts for about one third of the total water resources potential of the country. The river Brahmaputra and its tributaries carry about 600 BCM of surface water annually through this region. However, development of this huge water resources potential has remained limited so far and on the other hand, the region is suffering from disastrous flood every year. The high degree of temporal and spatial variations of available water makes the problem of harnessing water resources quite complex and calls for strategic planning and management. To utilize the available water as well as to mitigate the flood hazard, a numbers of dam projects such as Pagladia, Tipaimukh, Dibang, Lohit have been proposed by the Government of India in this region.

Pagladia Dam Project, which is under construction, is a multipurpose reservoir project located on the river Pagladia, a major north bank tributary of the river Brahmaputra. This multipurpose reservoir is proposed to serve three purposes, namely, flood control, irrigation and power generation. In order to achieve these various purposes proper operation of the reservoir is imperative. Deriving an optimal operating policy for a multipurpose reservoir is always considered to be an important aspect of the project, as operational effectiveness and efficiency of the reservoir system can be improved significantly with an efficient operating policy.

A number of optimization techniques have been used during the last four decades to determine the optimal operating policy of reservoirs for various situations. In spite of extensive research in reservoir optimization, researchers are still in search of better optimization techniques that can derive more efficient reservoir operating policy for better management of reservoir. The choice of an optimization technique depends on the availability of data, objective function of the problem and the constraints specified. Generally, dynamic programming (DP), linear programming (LP), nonlinear programming (NLP) and simulation techniques are applied for deriving optimal operating policy. Recently, some researchers have used artificial neural network (ANN) and fuzzy rule-based model for inferring reservoir-operating policy. Researchers have also attempted using genetic algorithm (GA) for deriving reservoir-operating policy. Recent researches have shown that these new techniques do have the potential of developing improved reservoir operating policy. However, potential of these new techniques are yet to be fully explored for their application in reservoir operation in different situations. Although extensive researches have been done towards finding better methods of deriving optimal operating policy, superiority of a particular method in general could not be claimed.

Therefore, in this study, an attempt has been made to explore the potential of ANN and GA in deriving optimal operating policy for a reservoir in general, and for the Pagladia multipurpose reservoir in particular. The efficiency of the policies derived by these recent techniques has been assessed through their critical comparison with policies derived by different long-established techniques so as to suggest the most efficient operating policy for the Pagladia multipurpose reservoir.

1.2 METHOD OF INVESTIGATION

With an objective to make a systematic approach towards the problem, the proposed work has been divided into different phases and dealt separately in different chapters as outlined below.

Development of an optimal operating policy for a reservoir has been an active area of research for the scientific community for the last four decades. Uncertainty associated with the hydrological processes has made the problem of deriving a general operating policy for future implementation a challenging one. Non-availability of a long series of historical streamflow data for most of the reservoir projects in the developing countries has added another dimension to this complex problem. Therefore at the very outset, in chapter 2, a detail discussion on the previous work conducted on various topics, relevant to the proposed study, has been made in a systematic manner.

An overview of the Pagladia River system, its climatic condition, geology of the dam site, and salient features of the proposed Pagladia multipurpose reservoir has been presented in the chapter 3.

Total understanding of the reservoir operation problem is necessary before applying any optimization method for deriving an operating policy. Problem must be formulated with utmost care so that the objective sought can be achieved through its solution. Therefore in chapter 4, an attempt has been made to formulate the reservoir operation problem considering all the important aspects of the proposed reservoir. The objective function considered for the problem along with the constraints is explained in this chapter.

A long time series of streamflow data is generally required for the development of good operating policy of a reservoir. Therefore in chapter 5, an attempt has been made to generate synthetic monthly streamflow for the Pagladia River at the dam site. Three different models, namely, Thomas-Fiering model, autoregressive moving average (ARMA) model and artificial neural network (ANN) based model, used for developing synthetic monthly streamflow are presented in this chapter. Finally, a statistical comparison carried out to decide the best synthetic monthly streamflow series, is presented.

Chapter 6 deals with the development of optimal monthly operating policy for the Pagladia multipurpose reservoir using deterministic DP. The deterministic DP has been solved for different numbers of characteristic storages of the reservoir. Two approaches, namely multiple linear regression and ANN have been used to infer general operating policy from the deterministic DP results. These models are called DPR and DPN models respectively in this study. Finally, the effect of considering different numbers of characteristic storage of the reservoir on the performance of DPR and DPN models has been analyzed through reservoir simulation.

An explicit stochastic optimization technique, namely, stochastic dynamic programming (SDP) used to develop monthly operating policy of the Pagladia multipurpose reservoir has been presented in chapter 7. Three different SDP models are developed considering different number of characteristic storages of the reservoir. Finally, the effect of considering different numbers of characteristic storage of the reservoir on the performance of SDP models has been analyzed through reservoir simulation.

Genetic algorithm (GA) is a robust optimization technique, which has the potential of solving very complex optimization problem. In chapter 8, an attempt has been made to derive optimal operating policy for the Pagladia multipurpose reservoir using GA. Performances of four different GA models developed in this chapter have been analyzed through reservoir simulation.

In chapter 9, a critical comparison of all the policies derived in chapters 6,7, and 8 has been carried out on the basis of their performance in reservoir simulation. Based on this comprehensive comparison the most efficient operating policy for the Pagladia multipurpose reservoir has been chosen and recommended for implementation.

In the concluding chapter (chapter 10), a comprehensive conclusion and general discussions on various works, performed in this study has been presented. Some suggestions have also been put forward for further extension of the present work.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

Extensive researches have been done in the past by different investigators to develop efficient reservoirs operating policy for various situations using different optimization techniques. Efforts are still on for the development of a more efficient operating policy to improve the operational effectiveness of the reservoir. A brief review of the previous works reported on reservoir operation and its related topics has been presented in this chapter.

2.2 PREVIOUS WORKS ON RESERVOIR OPERATION

2.2.1 INTRODUCTION

Reservoir operation plays a vital rule in maximizing beneficial use of the reservoir. Development of an optimal operating policy for a reservoir has been an active area of research for the scientific community for the last four decades. Development and adoption of optimization techniques for planning, design, and management of complex water resources systems can be marked as a significant advancement made in the field of water resources engineering.

A number of methods are in use for optimization of reservoir operation. The choice of methods depends on the characteristic of the reservoir system being considered, on the availability of data, and on the objective and constraint specified (Yeh 1985). The available methods can be classified as linear programming (LP), dynamic programming (DP), nonlinear programming (NLP) and simulation. Combinations of these methods have also been reported in literature. Fuzzy rule based modeling, artificial neural network (ANN) approach, and genetic algorithms (GAs) are some of the recent techniques that have been used in the reservoir operation. Yeh (1985) presented the state-of-the-art review and discussed in detail the usefulness of various models for reservoir operation. Wurbs et al. (1985) presented an annotated bibliography for various optimization and simulation models. Simonovic (1992) presented a short review of mathematical models used in reservoir management and operation. This review was intended to present conclusions reached by previous state- of –the –art reviews and to provide ideas for closing the gap between theory and practice. Wurbs (1993) reviewed the reservoir system simulation and optimization models. This review was done with the objective to contribute to ongoing efforts throughout the water management community in sorting through the numerous reservoir system analysis models. Yakowitz (1982) exclusively reviewed the dynamic programming models used in various water resources planning problems. Most recently Labadie (2004) provided the state -of-the- art-review of multireservoir systems operation. This review was done with the purpose to assess the state-of-the-art in optimization of reservoir system management and operations and consider future directions for additional research and application. Optimization methods designed to prevail over the high dimensional, dynamic, nonlinear, and stochastic characteristics of the reservoir systems are scrutinized, as well as extensions into multiobjective optimization. Application of heuristic programming methods using evolutionary and genetic algorithms were described, along with application of neural networks and fuzzy rule-based systems for inferring reservoir system operating rules.

The reservoir operation models developed were under deterministic or stochastic environment. The deterministic models use a specific sequence of stream flows - either historical or synthetically generated, while the stochastic models use a statistical description of the streamflow process instead of a specific streamflow sequence (Karamouz and Houck, 1987).

2.2.2 LINEAR PROGRAMMING (LP) MODELS

2.2.2.1 DETERMINISTIC LP MODELS

Linear programming (LP) has been a valuable tool in the optimization of reservoir operations. LP can be applied to a reservoir operation problem provided the objective function and the constraints are linear in nature. Although objectives functions as well as some of the constraints of reservoir are often nonlinear, linearization techniques such as piecewise linearization can be successfully used. Computational simplicity and availability of computer code are the basic reasons behind the popularity of LP. Also in LP formulations, where solution is feasible, these algorithms always converge to global optimum (Mujumdar and Narulkar, 1993).

Dorfman (1962) initiated the application of LP technique in reservoir system planning problems. Meier and Beightler (1967) introduced a branch-compression technique for decomposing parallel reservoir systems, but their approach did not consider temporal allocation over seasons. Parikh (1966) put forth the idea of spatial decomposition by applying dynamic programming to subsystems under an initial set of prices. Then, releases over time from all subsystems were allocated over space by using LP, and from which dual values were used to adjust the initial prices assigned to the subsystems. Roefs and Bodin (1970) suggested an extension of this methodology to decomposition over time, but substantial computational difficulties were encountered when it was applied to a system of three reservoirs.

Windsor (1973) developed a methodology employing a recursive LP as the optimization tool for analyzing a multi reservoir flood control system.

Martin (1983) used a successive linear approximation technique to determine long term operation of a system consisting of 27 reservoirs.

Reznicek and Simonovic (1990) developed a successive LP algorithm to optimize the operation of a hydroelectric utility.

Recently, Mujumdar and Ramesh (1998) presented an integrated model for short-term yearly reservoir operation for irrigation of multiple crops using deterministic LP, which was a modification of earlier model by Mujumdar and Ramesh (1997).

2.2.2.2 STOCHASTIC LP MODELS

2.2.2.2.1 STOCHASTIC LP FOR MARKOV PROCESS

The earliest application of stochastic linear programming model for reservoir operation was by Manne (1962). He demonstrated the application of LP for optimization with a hypothetical single reservoir example taking into account that inflows are subject to random as well as seasonal influences and follows Markov process.

Thomas and Watermeyer (1962) extended Manne's work by defining the initial state of the system as inflow and storage rather than storage alone.

Loucks (1968) developed a stochastic LP model for a single reservoir subject to random, serially correlated, net inflows. The net inflows for each time period were described by a first order Markov chain and transition probabilities of inflows were estimated from historical inflows. The stochastic LP model was applied to a simplified representation of the Finger Lakes within the Oswego river basin. He pointed the dimensionality problem associated with this type of model in real situations, which can easily exceed several thousand constraints.

Houck and Cohon (1978) assumed a discrete Markov structure for streamflows to find design and management policy for a multipurpose multiple-reservoir system by solving two LP problems sequentially in order to approximate a non-linear formulation.

2.2.2.2.2 CHANCE CONSTRAINED LP MODELS

The application of chance constrained LP to reservoir system optimization was initiated by ReVelle et al. (1969). For application of chance constrained LP to reservoir operation they first proposed the use of linear decision rules (LDR) that relate releases to storage, inflow and decision parameters. Jores et al. (1971) applied chance constrained LP, synthetic streamflow generation and simulation in modelling multiple source water supply system for Baltimore. Loucks and Dorfman (1975) proposed an LDR with a weighting factor for current inflow.

Houck and Datta (1981) performed a comparative study to evaluate multiple LDR models. They found multi-LDR models to be less conservative than single LDR models.

Stedinger (1984) evaluated the performance of single and multiple LDR models for reservoir screening and operating policy and found LDR to be less efficient when compared to standard operating policy.

2.2.3 NONLINEAR PROGRAMMING (NLP) MODEL

NLP has not enjoyed the popularity that LP and DP have in water resource system analysis as reported in the review presented by Yeh (1985). This is particularly due to the fact that the optimization process is usually slow and takes up large amounts of computer storage and time when compared with other methods. The mathematics involved in the nonlinear models is much more complicated than in the linear case, and NLP unlike DP cannot easily accommodate the stochastic nature of inflow.

However with the rapid development of computing facilities researchers have attempted using NLP for reservoir operation where non-linear formulation cannot be avoided and for the situation where linearization is expected to introduce serious error. NLP can effectively handle a nonseparable objective function and nonlinear constraints which many programming techniques cannot. NLP includes quadratic programming, geometric programming, and separable programming as special cases, which can be used iteratively as master programs or as a subprogram in large-scale system problems.

Simonovic and Marino (1980) applied gradient projection method with a two-dimensional Fibonacci search to solve a reliability programming problem for single reservoir management. They considered both random inflow and demand in their continuity equation, and both benefit and risk in their objective function for discrete determination of reliability concerning flood and drought.

Marino and Loaiciga (1985) developed a quadratic model for monthly release policies for Northern California Central Valley Project (NCVP). They compared the quadratic model with a simplified linear model and found that optimal release schedules are robust to the choice of the model, both yielding an increase of nearly 27% in the total annual energy production with respect to conventional operation procedures, although quadratic model is more flexible and of general applicability.

Tejada-Guibert et al. (1990) applied successive quadratic programming (SQP) to a five-resrvoir portion of the central valley project (CVP) of California

Peng and Buras (2000) applied generalized reduced gradient (GRG) method to five major upstream Lakes in the west Branch Penobscot River, Maine.

Barros et al. (2003) applied the successive linear programming (SLP) technique to the Brazilian hydropower system, one of the largest in the world.

Labadie (2004) in his recent review on multireservoir operation has enlisted some of the recent use of NLP in water resources system optimization.

2.2.4 DYNAMIC PROGRAMMING (DP) MODELS

2.2.4.1 DYNAMIC PROGRAMMING (DP)

Dynamic programming (DP), a method first introduced by Bellman (1957), is an optimization procedure for solving a multistage decision process. DP has a wide variety of applications in engineering and economic decision problems (Yakowitz, 1982; Yeh, 1985). The popularity and success of this technique can be attributed to the fact that the non-linear and stochastic features, which characterize a large number of water resources systems, can be translated into a DP formulation. In addition, it has the advantage of effectively decomposing highly complex problems with large number of variables into a series of subproblems, which are solved recursively.

A major limitation in the use of dynamic programming is the well-known ‘curse of dimensionality’ (Bellman and Dreyfus, 1962). The computational requirement for DP increases exponentially with increase in state dimension. Chow et al. (1975) discussed the computational requirement of discrete dynamic programming applied to multireservoir problems.

2.2.4.2 DETERMINISTIC DYNAMIC PROGRAMMING (DP) MODELS

Young (1967) was the first to apply the deterministic dynamic programming algorithm in reservoir operation. He studied a finite horizon single reservoir operation problem. After Young (1967) a number of modified DP algorithm have been specially developed to alleviate the computational burden in DP when applied to multireservoir planning and operation problems.

A brief review of past work reported on application of different types of deterministic DP in reservoir operation is presented below.

Discrete Dynamic Programming (Discrete DP):

Bellman and Dreyfus (1962) first developed discrete dynamic programming (Discrete DP). The Discrete DP requires that the control and state space be discretized by a finite set of vector.

Young (1967) first applied this Discrete DP approach to find the optimal operating policy of a single reservoir. He used release for current period as decision variable and initial storage for current period as state variable. Young (1967) first proposed a means to obtain general operating rules from the results of the deterministic optimization model: he suggested performing least squares regression against the preceding characteristics of the optimal operation such as previous seasons' releases, inflows, and storages. In this way some of the stochastic nature of the optimal deterministic operation is captured in a general operating rule.

Hall et al. (1968) used the Discrete DP technique for optimizing operation of a multipurpose reservoir. The objective of their analysis was to determine for a given initial state of the system, price schedule, and sequence of inflows the set of decision regarding release of water from the reservoir that will maximize the total return from the operation subject to physical and other constraints.

Hall et al. (1969) modified their earlier method (Hall et al., 1968) by incorporating additional factors like firm water and on-peak energy constraints, energy pricing, and flood control desiderata, etc.

Roefs and Bodin (1970) made a critical analysis of reservoir operating rules using deterministic, implicit stochastic and explicit stochastic algorithm and extended the Discrete DP to a multireservoir case.

Liu and Tedrow (1973) used the Discrete DP algorithm along with a multivariable pattern search technique and applied the model to find out the operating rule curves for Oswego river system with five major lakes in the system.

Karamouz and Houck (1982) developed general reservoir system operating rules by deterministic optimization. They constructed a dynamic programming regression based model. They used a hypothetical loss function, which involved only reservoir release.

Karamouz et al. (1992) applied Discrete DP to multiple reservoir systems in the Gunpowder River basin near Baltimore. Total of 1500 months of multi-site, synthetic streamflow data were input to the Discrete DP model for implicit stochastic optimization.

Mujumdar and Ramesh (1997) developed a real time reservoir operation model for irrigation of multiple crops using deterministic dynamic programming. The reservoir storage, soil moistures of individual crops, and a crop production measure constituted the state space. The model was applied to the Malprabha reservoir in Karnataka (India).

Incremental DP (IDP) and Discrete Differential DP (DDDP):

IDP or DDDP provides a means to alleviate the curse of dimensionality associated with discrete DP. The IDP for reservoir operation studies was reported by Hall et al. (1969) and Trott and Yeh(1971) and systematized and referred to by Heidari et al. (1971) as discrete differential dynamic programming (DDDP). IDP uses incremental concept for the state variable, a concept first introduced by Larson (1968). The major difference between Larson's state incremental DP and IDP is the time interval used in the computation, which is variable in the former and fixed in the later.

Heidari et al. (1971) considered a prototypical four-reservoir problem, which was probably beyond the discrete dynamic programming because of curse of dimensionality. They proposed a computational technique called DDDP and gave a detail account of their numerical solution. The same hypothetical problem was also solved in the work of Larson (1968) to illustrate his computational procedure IDP.

Fults and Hancock (1972) applied the DDDP technique to a five-reservoir problem of Central Valley Project (CVP) system in California, USA for an operation on daily basis. Later on Fults et al. (1976) applied the same technique in the same system for operation of nine reservoirs on monthly basis. In both the cases only the major storage reservoirs were analyzed for maximization of power generation.

Nopmongcol and Askew (1976) analyzed the difference between IDP and DDDP and concluded that DDDP is the generalization of IDP.

Turgeon (1982) demonstrated that IDP might converge to non-optimal solution if the same state increment is used in every stage.

Incremental Dynamic Programming with Successive Approximations (IDPSA or DPSA):

IDPSA or DPSA is another way of alleviating curse of dimensionality associated with Discrete DP. The incremental DP with successive approximations (IDPSA) decomposes an original multiple state variable DP into a series of sub problems of one state variable in such a manner that the sequence of optimizations over the sub problems converges to the solution of the original problem. Larson (1968), Trott and Yeh (1973), and Giles and Wunderlich (1981) have applied this technique to problems involving multiple reservoirs.

Trott and Yeh (1973) applied this algorithm to find operating policy of a six-reservoir system in northern California.

Giles and Wunderlich (1981) used DPSA for the weekly operation of Tennessee Valley Authority (TVA) reservoir system consisting of nineteen storage reservoirs.

Yakowitz (1983) demonstrated that if the discretization of the state space is sufficiently fine and if the limiting trajectory is an interior point of the admissible policies, then the state increment DP technique has linear convergence.

Multilevel Incremental Dynamic Programming (MIDP):

Nopmongcol and Askew (1976) were the first to develop this algorithm and demonstrated it through the Larson's (1968) four reservoir hypothetical problem.

This algorithm is somewhat a combination of DPSA and DDDP. No further application of this algorithm has so far been reported in literature.

Binary State Dynamic Programming (BSDP):

BSDP algorithm was first proposed by Ozden (1984) and was demonstrated through the standard four-reservoir problem of Larson (1968) and a case study of a planning problem of four-reservoir system in Northeast Turkey. Ozden (1984) showed that the algorithm is computationally less expensive than the DDDP technique.

Differential Dynamic Programming (DDP):

Jocobson and Mayne (1970) first introduced the differential dynamic programming (DDP) to alleviate dimensionality difficulties in DP through use of analytical solutions rather than resorting to discretization of state space.

Murray and Yakowitz (1979) proposed a constrained differential DP (CDDP) applicable to multiple reservoir systems by modifying the differential DP. Murray and Yakowitz (1979) illustrated the efficiency of the CDDP using a hypothetical four-reservoir problem. They also enlarged the problem to a ten-reservoir case and computed the optimal policy.

Jones et al. (1986) applied the DDP approach to implicit stochastic optimization of the Mad River system in northern California. A total of 101 sets of 64 years of stochastically generated monthly inflows were input to the DDP algorithm for minimizing downstream water deficits.

Folded Dynamic Programming (FDP):

The folded dynamic programming (FDP) is the most recent technique of DP developed by Nagesh Kumar and Baliarsingh (2003) for application to multireservoir system operation. They applied their FDP algorithm to the hypothetical four-reservoir problem of Larson (1968). Unlike IDP, DDDP, IDPSA and DPSA, in which initial trial trajectory is necessary to start iteration, the FDP does not require any initial trial trajectory to start iteration. Nagesh Kumar and Baliaringh (2004) showed that FDP requires less number of iteration than other algorithms to reach optimal solution.

Philbrick and Kitanidis (1999) discussed the limitations of the deterministic optimization applied to reservoir operations and showed that deterministic optimization can produce suboptimal reservoir control policies by failing to incorporate adequately the impact of the low-probability events. They also showed that the resulting operating policies might not efficiently balance the costs of rationing, minor flooding or other short-term impacts with the severe impacts of extreme flood or drought.

2.2.4.3 STOCHASTIC DYNAMIC PROGRAMMING (SDP) MODELS

Stochastic dynamic programming (SDP) is an explicit stochastic optimization technique, which operate directly on the probabilistic descriptions of random streamflow process (as well as other random variables) rather than deterministic hydrologic sequence (Labadie 2004). This means that optimization is performed without the presumption of perfect foreknowledge of future events.

Stochastic dynamic programming (SDP) optimization considers the inflow into the reservoir as Markov process (Howard 1960).

The SDP models are very useful for long term operating policy. They can handle non-linear objective function and constraints and are well suited for sequential decision process

The earliest reservoir operation study in English language reported in literature was by Little (1955), who considered operation of a single reservoir. Little's (1955) work was based on stochastic discrete dynamic programming (Yakowitz 1982). The inflows were considered as a stochastic sequence and assumed that it satisfies the Markov assumption. In his SDP model, Little (1955) considered current period storage and previous periods inflow to the reservoir as state variables. He applied his model to data from the Grand Cooley generation plant on the Columbia River. The transition probability matrix was derived from 39 years of historical flows. The model was found to give improve optimal strategy by 1% when compared to conventional rule curves.

Yakowitz (1982), and Yeh (1985) presented the state-of-the-art review of application of stochastic dynamic programming in water resources systems. Stedinger et al. (1984) mentioned almost all the models developed using SDP.

Loucks and Falkson (1970) reviewed and compared three algorithms, namely, dynamic programming, linear programming and policy iteration under stochastic environment. Each of the models was used to derive optimal operating policy. The operating policy defines the release as a function of the current storage volume and inflow. The inflow was assumed to be serially correlated stochastic quantity and approximated by discrete quantities. They found that all the three models yielded the identical policies but their computational efficiency differs.

Dynamic programming algorithm took minimum computational time; policy iteration method took somewhat longer while linear programming algorithm took highest computational time for the same problem.

Butcher (1971) applied the discrete SDP to find out the optimal stationary strategy for operating the Wataheamu Dam on California-Nevada border. He derived the operating policy on a monthly basis. The optimal operating policy for the multipurpose reservoir was stated in terms of the state of the reservoir indicated by the storage volume and the river flow in the preceding month.

Torabi and Mobasheri (1973) presented a SDP model useful in determining the optimal operating policy of a single multipurpose reservoir. The model was applied to the Folsom reservoir and power plant on the American river in Northern California. In their model the stochastic nature of the streamflow was taken into account by considering the correlation between the streamflows of each pair of consecutive time interval. The operational purposes of the reservoir included firm water supply, on- peak power production, and flood and water quality control downstream of the reservoir. Flood control and downstream water quality control was considered as constraints. The firm water supply was treated as a parameter. For a given level of firm water supply the model determines the optimum monthly release policy to maximize the expected annual level of on peak energy production.

Su and Deininger (1974) studied extensively the application of Markovian process in reservoir operation problem. The inflows were treated as stochastic random variable and applied to the model as serially correlated and seasonally independent inflows.

Roefs and Guitron (1975) compared three models, namely linear programming formulations, dynamic programming formulations and policy iteration models on the basis of computational efforts. Some quantitative guidelines, which discriminate dynamic programming model and policy iteration models, have also been developed. The models were used to solve for optimal operating rules for a single reservoir under stochastic environment and the conclusion was that the SDP model was the preferred algorithm.

Bogle and O'Sullivan (1979) developed an explicit stochastic dynamic programming algorithm and applied to a water supply system. The inflows to the reservoir was described by a piecewise linear probability density function. The method permits the use of a smaller number of storage states than the conventional DP algorithms.

Oven-Thomson et al. (1982) developed a tradeoff relationship between agriculture and hydropower for High Aswan dam using stochastic dynamic programming. Varying the monthly reservoir release for agricultural purposes, the consequent impacts on hydropower production were studied. The discretised monthly storage and monthly inflows were considered as state variables of the model and the optimal operating policy was derived by minimizing a specially derived cost function. Setting the monthly irrigation requirements at a certain level, the monthly release policy was derived using the SDP model. Then the derived operation policy was simulated to get the value of hydropower production. Using different values of monthly irrigation requirements, different hydropower values were obtained and the trade-off relationship have been drawn. The results showed an 11-20% increase in firm monthly hydropower production when summer irrigation demands were reduced by 25%.

Esmaeil –Beik and Yu (1984) used a SDP model to develop optimal weekly policies for operating the multipurpose pool of the Elk City Lake in Kansas. The operation of the pool was treated as a periodic decision process with finite state and discrete time. The model determined a long term operating policy to minimize the expected average annual loss and showed a marked improvement in meeting the target operation. They also found that that the use of previous period's inflow as state variable in the SDP model resulted simulated weekly losses almost twice as large as when current period's inflow was used to develop the model. Sensitivity analysis of the optimal policy and computational effort to the number of discrete states representing the reservoir system was also investigated.

Buras (1985) presented a SDP model for seasonal operation of Sardar Sarovar Project in Narmada river system of Central India. The seasonal inflows were considered as serially correlated and Markovian decision problem was formulated for discrete state space. Further, he derived the steady state probabilities of initial storages, inflows, and final storages. Finally reservoir releases were estimated corresponding to 75%, 80% and 90% reliability levels.

Goulter and Tai (1985) discussed practical implications in respect of storage states while using a SDP model for reservoir operation problem. The computer time required to reach steady state condition with changes in the number of storage states were investigated. The increase in computer time required to develop the storage probability distributions with increase in the number of storage states has also been investigated. It showed that for a small number of storage states, the storage probability distribution becomes unrealistically skewed. The conclusion of this paper supports the Doran's (1975) statement that with appropriate discretization technique the use of minimum 5 to 10 storage states is advisable.

Karamouz and Houck (1987) tested and compared two algorithms to generate reservoir-operating rules by deterministic and stochastic optimization for single reservoir sites. The deterministic model comprised a deterministic dynamic program, regression analysis and simulation. The stochastic model was a SDP. The SDP model described streamflow with a discrete lag-one Markov process. To compare the models, 12 single reservoirs, monthly operating test cases were selected. Authors showed that SDP model performed better than deterministic model for small reservoir (capacity 20 percent of mean annual flow), but for large reservoir (capacity exceeding 50 percent of mean annual flow) the deterministic model performs better. They found that in SDP model for small and medium reservoir (capacity of 20-50 percent of mean annual flow), 20-30 characteristic storages are sufficient; but for large reservoirs (capacity of 100 percent of mean annual flow), 50 or more characteristic storages should be used. For very large reservoirs (capacity greater than mean annual flow), up to 150 characteristic storages may be required to generate good operating rules by SDP model.

Druce (1990) developed a monthly SDP model to provide decision support for short term energy export and applied it to the British Columbia Hydro Systems. The flood control objective has been incorporated in the SDP model on a daily basis. The daily and forecasted monthly flows were input to the SDP model. The state variables used in this model were the current period reservoir storage and historical weather pattern. The return function was calculated through a simulation model embedded in the SDP model, which accounts for the return from hydropower production as well as from downstream flood damages.

Vedula and Mujumdar (1992) developed a SDP model for optimal operating policy of an irrigation reservoir under multiple crop scenarios. The stochasticity in the model was incorporated by assuming the reservoir inflow as a first order Markov chain and the process was assumed to be stationary. Three state variables, namely, current period storage, current period inflow, and average initial soil moisture were considered in the model. The optimal operating policy specifies the reservoir release and crop water allocation for various crops in any given intraseason period for known initial storage, inflow and initial soil moisture in the cropped areas. The model was applied to the Malaprabha reservoir in Karnataka state, India.

Many researchers have examined ways of improving SDP models by improving the models of reservoir inflows. Alarcon and Marks (1979) and Bras et al. (1983) employed non-stationary dynamic programming models for high Aswan dam. Bras et al. (1983) employed a non-stationary "real time adaptive closed loop control" SDP algorithm to optimize reservoir operation, which handle situations where streamflow transition probabilities are continuously updated over a finite transition period.

Datta and Houck (1984) developed a real –time operation model primarily useful for daily operation of a reservoir. The model emphasized the incorporation of uncertainties inherent in short-term hydrologic forecasts into the decision making process.

Stedinger et al. (1984) developed a SDP model, which employed the best forecast of the current period's inflow as hydrologic state variable to define a reservoir release policy and to calculate the expected benefits from future operations. The model which used one-month ahead flow forecast equations, outperformed the correspondingly non-stationary reservoir operating algorithm which employed the preceding period's inflow as hydrologic state variable.

Trezos and Yeh (1987) developed a differential SDP algorithm that can be applied to large-scale reservoir system without discretizing the state and control variables under the limitations that the recursion equation is a concave function of the state variables.

Foufoula-Georgiou and Kitanidis (1988) developed a gradient dynamic programming that reduces the computational efforts for the stochastic optimal control of multidimensional water resources systems, using an alternative interpolation algorithm.

Paundyal et al. (1990) presented a model of long-term operational aspects of multiunit hydropower system for maximizing firm energy and maximizing expected annual energy. An incremental DP algorithm was first used with the objective of maximizing firm energy from the system. In the second step, SDP, which incorporates the uncertainties inherent in the streamflow series, has been used to derive long-term joint operation policy of the system of reservoir in the configuration selected from the first step.

Karamouz (1990) suggested the revision of state transition probabilities in classical stochastic dynamic programming (SDP) using Bayesian decision theory (BDT) to capture uncertainty of forecast in reservoir operation. BDT allows new information to be incorporated in a systematic and flexible manner.

Kelman et al. (1990) developed a sampling stochastic dynamic programming (SSDP) model that generates operating policies capturing the temporal and spatial characteristic of inflows. They included the forecast for the current period as a state variable in determining optimal operating rules for reservoir operation.

Braga et al. (1991) developed a stochastic dynamic programming model for the optimization of hydropower production of a multiple storage reservoir system with correlated inflows. The model was applied to a subsystem of the Brazilian hydroelectric system.

Karamouz and Vasiliadis (1992) developed a Bayesian stochastic dynamic programming (BSDP) model, which generates optimal operating rules for real time reservoir operation. In BSDP the discrete Markov process is assumed to describe the transition of an inflow from one period to the next. The forecast for the next period's flow along with an actual inflow during the current period are state variables in generating operating policies. In addition, BSDP uses Bayesian decision theory (BDT) to develop and continuously update prior to posterior probabilities to capture the natural and forecast uncertainty. Different probability models, flow, and storage discretization schemes are also compared.

Tejada-Guibert et al. (1993) evaluated the effect on operation of using different state space discretizations and functional approximations within the SDP.

Vasiliadis and Karamouz (1994) presented a demand driven stochastic dynamic programming (DDSP). DDSP is an expanded version of BSDP developed by Karamouz and Vasiliadis (1992). In DDSP, not only the prior transition probabilities of flows (historical and forecasted) are updated to posterior probabilities but also posterior and other transition probabilities are updated for a given specific month. DDSP includes the uncertain demands for each month as an additional state variable.

Tejada-Guibert et al. (1995) formulated several SDP models, each employing a different set of hydrologic state variables. They compared use of seasonal and one-period-ahead flow forecasts with use of the current period's flow as hydrologic state variable. The performance of each formulation was examined with three different objectives, which placed different penalties on shortfalls associated with firm power and water targets of different magnitudes. The models were applied to Shasta-Trinity system in northern California and performance of each model was studied by simulation. For an objective with only moderate water and power targets, there was little difference among performance of policies that used different hydrologic state variables.

Talukdar (1999) developed a SDP model for optimal operation of multipurpose Sardar Sarovar Reservoir of Narmada River, India.

2.2.5 MODEL USED AS COMBINATION OF TWO OR MORE METHODS

In an attempt to utilize the advantages of different methods and to overcome the limitations of a particular method, researchers have also developed some hybrid optimization technique, which are combination of two or more algorithms.

Hall and Shepherd (1967) developed a DP-LP technique for a river reservoir optimization in which the multiple reservoir system is decomposed into a master problem and subproblems. The master problem could be seen as a system coordination agency and the subproblems as single reservoir managers. In their work subproblems were solved by DP. The schedule of releases and energy production were reported to the system coordination agency (master problem), which is LP.

Becker and Yeh (1974) suggested a combined solution methodology of LP-DP for determination of optimum real time reservoir operation associated with the California Central Valley Project (CVP). They considered a monthly operating model with a 22 decision variables and 48 constraints for each period. The LP was used with an objective of minimizing the loss in potential energy of the stored waters in the reservoirs resulting from any release policy. In the DP model, the energy constraint parameter was considered as single decision variable, the cumulative sum of the energy constraint parameter as the single state variable and maximization of the weighted sum of the reservoir end of month storages as the objective function.

Takeuchi and Moreau (1974) used a combination of LP with SDP models. In this model the objective function consists of two parts: immediate economic losses within the month and the expected present value of future losses as a function of end-of -storage levels in the reservoirs. The latter function is estimated by embedding the linear programming problem in a SDP problem.

Becker et al. (1976) used the same monthly model of Becker and Yeh (1974) and developed daily and hourly model for the CVP system. The monthly model output was used as an input to the daily model and the output of the daily model was used as an input to the hourly model.

Chaturvedi and Srivastava (1981) analyzed six major reservoirs of Narmada basin in India using deterministic LP with simulation model.

Palmer et al. (1982) developed simulation and LP models to determine the yield of the reservoir system when operated jointly with the Potomac River.

Marino and Mohammadi (1983) developed a methodology for the monthly operation of a system of two parallel multipurpose reservoirs. The model employed linear programming (LP) nested in dynamic programming (DP). At every stage of DP (i.e. months) a series of LP's are solved. The objective of LP is to minimize the total releases from the reservoir in each month. The objective of DP is to maximize the weighted sum of monthly water and power production.

Mohammadi and Marino (1984) developed one efficient algorithm for the real time monthly operation of a multipurpose reservoir. The model is a combination of LP (used for month by month optimization) and DP (used for annual optimization).

Kuo et al. (1990) developed a model for real time operation of Feitsui and Shihmen Reservoirs in the Tanshui River Basin, Taiwan. The model consists of a 10-day streamflow forecast model, a rule curve based simulation model and a DP optimization model. After getting an initial feasible operating policy by using the simulation model, the DP based optimization model was then used to determine an improved operating policy.

Vedula and Mohan (1990) developed a real time operational methodology for the Bhadra reservoir in the state of Karnataka (India). The algorithm have three phases of operation. The first phase determines the optimal release policy for a given initial storage and inflow using SDP. Second phase constitute the flow forecasting using ARIMA model and in the last phase a real time simulation model was developed. In the SDP model, the inflows were assumed to follow a discrete Markov process.

Jain et al. (1992) developed a model based on SDP formulation, which considers risk explicitly. The objective of the model was to maximize the reservoir storage at the end of flood season while ensuring that the risk of the overflow is within acceptable limit. Current period storage of the reservoir and an information variable, which can be used to determine the probabilistic properties of future inflows, was taken as state variables of the DP formulation. The model was applied to the Dharoi multipurpose reservoir of the Sabarmati River in Gujarat (India) and its performance was tested by simulation.

Vedula and Nagesh Kumar (1996) developed an integrated model to determine the optimal reservoir release policies and irrigation allocation to multiple crops. The model used LP-SDP combination as module 1 and module 2 respectively. Module 1 was an intraseasonal allocation model to maximize the sum of relative yields of all crops for a given state of the system using LP. Module 2 was a seasonal allocation model to derive operating policy using SDP. Reservoir storage, seasonal inflow, and seasonal rainfall were the three variables, which maximizes the expected sum of relative yield of all crops in a year. Seasonal inflow and seasonal rainfall were each assumed to constitute a stationary Markov process.

Ravi Kumar and Venugopal (1998) developed an integrated real time operation method for a large-scale south Indian irrigation system by using simulation and SDP. The irrigation demand pattern was determined by simulating the command area with historical data. The SDP was used to obtain an optimal release policy. The SDP model considered both the demand and inflow as stochastic and is assumed to follow first order Markov chain model. Finally another simulation model was used to study the degree of failure associated with adoption of the optimal operating policy for different reservoir storages at the start of the crop season.

2.2.6 ARTIFICIAL NEURAL NETWORK (ANN) MODEL

An artificial neural network (ANN) is a "computational paradigm inspired by the parallelism of the brain". An ANN is a parallel-distributed information processing system that has certain performance characteristics resembling biological neural networks of the human brain (Haykin, 1994). ANN consists of a number of interconnected computational elements called neurons that are arranged in a number of layers. The connection between each pair of neurons is called a link and is associated with a weight that is a numerical estimate of the connection strength. Every neuron in a layer receives and processes weighted inputs from neurons in the previous layer and transmits its output to neurons in the next layer. The weighted summation of inputs to a neuron is converted to an output according to transfer function (linear or sigmoidal function). The ANN is particularly valuable in performing classification and pattern recognition functions for processes governed by complex nonlinear interrelationship.

ANN models have been successfully applied in different field of water resources engineering such as river stage forecasting (Liong et al. 2000), reservoir operation (Raman and Chandramouli 1996, Cancelliere et al. 2002), rainfall disaggregation (Burian et al. 2001) and so on.

Raman and Chandramouli (1996) derived a general operating policy for a reservoir using ANN from the deterministic DP results. They used ANN for inferring optimal release rule conditioned on initial storage, inflows and demands. Operating rule derived by ANN model was compared with a stochastic dynamic programming model, standard operating policy and the operating policy produced by multiple linear regressions from the deterministic DP results. The performance of the ANN model was reported as better than that of the other models.

Chandramouli and Raman (2001) developed a multi-reservoir operation model using dynamic programming and a neural network. They used a three state variable dynamic programming algorithm in a constrained manner. In that dynamic programming model the three releases from the reservoirs considered were the decision variables. They compared the operating rule derived using ANN from the deterministic DP results with the operating rule derived using multiple linear regressions from the deterministic DP results. They found that operating rule derived using ANN give improve performance than operating rule derived using multiple linear regression in their case study of a multireservoir system called Parambikulam Aliyar Project system in India.

Cancelliere et al. (2002) used a neural networks approach for deriving irrigation reservoir operating rules. In their approach operating rules were determined as a two step process: first, a dynamic programming technique which determine the optimal releases with minimizing sum of squared deficit as objective function, subject to various constraint was applied. Then, the resulting releases from the reservoir are expressed as a function of significant variables by neural networks. Authors report improved performance of ANN approach.

2.2.7 FUZZY RULE BASED MODELLING

The concept of fuzzy logic was first introduced by Zadeh (Zadeh 1965). It is a superset of conventional (Boolean) logic that has been extended to handle imprecise data and the concept of partial truth.

Fuzzy rule base modeling is an alternative approach to inferring operating rules from historical operations or implicit stochastic optimization of reservoir systems (Labadie 2004).

Russell and Campbell (1996) proposed its application to find out reservoir operating rule by applying it on a single purpose hydroelectric project. But the authors concluded that although it is promising approach, it suffers from the "curse of dimensionality". It can supplement the conventional optimization techniques but cannot probably be a replacement.

Shrestha et al. (1996) proposed that inputs to reservoir operating policies (e.g., initial storage, inflows, and demands), as well as outputs (e.g., historical release policies or results from implicit stochastic optimization) can be described by fuzzy relations. Shrestha (1996) report excellent result in using a fuzzy rule based system to replicate historical operations for Ten killer Lake in Oklohoma.

Dubrovin et al. (2002) developed a fuzzy rule based control model for multipurpose real-time reservoir operation and found that it is better to fulfill the new multipurpose operational objectives determined by the experts.

2.2.8 GENETIC ALGORITHM (GA) MODEL

Genetic algorithm (GA) derives its concept from Darwin's Theory of survival of the fittest, was first envisaged in 1975 by John Holland (Holland, 1992). Genetic algorithms are search algorithms based on the mechanics of natural genetic and natural selection.

Esat and Hall (1994) used genetic algorithm (GA) in reservoir system optimization. They applied the GA technique for operating rule determination of a four-reservoir problem, which maximizes the benefits from power generation and irrigation water supply subject to constraints on storages and releases from the reservoirs. Advantages of GA over standard dynamic programming techniques in terms of computational requirements and the potential of GA in water resources system optimization was showed in their work

On the basis of a comparison of GA and dynamic programming, through their application in a reservoir system, Fahmy et al. (1994) concluded that GA had potential in application to large river basin systems.

Otero et al. (1995) applied a GA to determining minimum storm water detention storage capacities and optimal operating rules for managing freshwater runoff into the St. Lucie Estuary along the southeast coast of Florida.

Oliveira and Loucks (1997) used a GA to evaluate operating rules for multireservoir systems, demonstrating that GA can be used to identify effective operating policies. Significant benefits were perceived to lie in the freedom afforded by GA in the definition of operating policies and their evaluation. They used the real-valued chromosomes containing the coordinates of the points that define the piecewise linear operating rule functions. Their research suggests that genetic algorithms may be a practical and robust way of estimating operating policies for complex reservoir systems.

Wardlaw and Sharif (1999) used a GA to explore the potential of alternate GA formulation in application to reservoir systems, and to deterministic finite-horizon problems in particular. They considered four-reservoir and ten-reservoir problem. Their result demonstrated that a genetic algorithm could be satisfactorily used in real time operations with stochastically generated inflows. In their formulation binary, gray and real coded GA had been used. They concluded that a real-value representation, incorporating tournament selection, uniform crossover, and modified uniform mutation would operate most efficiently and produce the best results. They also mentioned that GA has potential as an alternative to stochastic dynamic programming.

Sharif and Wardlaw (2000) proposed application of a GA to direct optimization of period-of-record releases as an alternative to deterministic optimization approaches such as DDDP.

The advantage of GA lies not in its computational efficiency, but rather the robust ability to solve highly non-linear, nonconvex problems (Labadie 2004). Labadie (2004) in his state-of-the-art review recognized the ability of GA to be linked directly with trusted simulation models as a great advantage.

2.3 PREVIOUS WORKS ON SYNTHETIC STREAMFLOW GENERATION

2.3.1 INTRODUCTION

A long time series of streamflow data is generally required for the development of good operating policy of a reservoir. Non-availability of a long series of historic streamflow data for most of the reservoir project in the developing country calls for generation of streamflow data. Generated streamflows have been called synthetic to distinguish them from historic observations. Synthetic streamflow generation therefore can be regarded as an important component in the process of deriving reservoir-operating policy in case of non-availability of longer time series of streamflow data. Therefore a brief review of synthetic streamflow generation has been presented here.

2.3.2 LITERATURE REVIEW

Since the work of Thomas and Fiering (Thomas and Fiering 1962) many synthetic streamflow generation models have been developed and used in water resources planning. Thomas-Fiering model (Thomas and Fiering 1962) is a simple model of monthly flows, which has often been used in reservoir planning studies.

The Thomas-Fiering model is a simple example of very flexible family of autoregressive moving average (ARMA) time-series models (Box and Jenkins

1976) sometimes called Box-Jenkins models (Loucks et al. 1981). These models have been widely used in the modeling and forecasting of time series in many fields besides water resources. McLeod et al. (1977) advocated the use of advanced ARMA techniques for modeling hydrologic time series. One way of using these techniques to model monthly flow series is first to remove the seasonal nonstationarity of the monthly flows and then to develop an ARMA model for the resultant time series (Hipel et al., 1979). Lawrence and Kottegoda (1977) discussed the theoretical difficulties associated with this deseasonalization procedure. Of major importance, this approach assumes that the resultant deseasonalized series is stationary. That is, it assumes that the correlation between the flows in January and February is same as that flows in July and August. This assumption contradicts both the theoretical and empirical evidence (Moss and Bryson, 1974). Still this is comparatively simple model.

Young and Jettmar (1976) suggested fractional Gaussian noise (FGN) models of deseasonalized monthly flow time series. However Chaturvedi (1997) stated that these models are open to criticism on phenomenological grounds and are very difficult to use as well.

McLeod and Hipel (1978) on the basis of careful comparisons of the fits of ARMA and fractional Gaussian noise models to long streamflow sequences using Akaike information criteria (AIC) (Akaike 1978) indicated that the ARMA models are to be preferred.

Stedinger and Taylor (1982) illustrated synthetic streamflow generation model verification and validation. Verification consist of demonstrating that the statistics explicitly involved in model formulation are statistically the same for both generated and historical flows; validation of a streamflow model is the demonstration that such model is capable of reproducing statistics which are not explicitly included in its formulation.

Chaleeraktrakoon (1999) proposed stochastic procedure for generating seasonal flows at single or multiple sites. The procedure has the simple structure of the multivariate AR(1) (MAR(1)) model. He combined the MAR(1) with the singular value decomposition (SVD). The model was able to reproduce the basic statistics and drought related properties at monthly and annual time scales.

Raman and Sunilkumar (1995) employed an ANN to model a multivariate water resource time series and compared the results with those obtained by traditional autoregressive moving average (ARMA) models. The objective was to synthesize monthly inflow data for two reservoir sites in the Bharathapuzha basin in south India. A three layer feed forward ANN with backpropagation was used in the study. The consecutive normalized inflows to the reservoir for two previous months were chosen as inputs. The output was normalized inflow for the current month. They concluded that the results obtained using the ANN compared well with those obtained using statistical models.

Ochoa-Riveria et al. (2002) presented a hybrid model based on artificial neural networks for multivariate synthetic streamflow generation. The model consists of two components, the neural network (NN) deterministic component and a random component, which is assumed to be normally distributed. The NN component used two previous months inflow of two sites as input and current month's inflows as output. The performance of the NN based model was compared to that of a standard autoregressive AR(2) model. The basic statistics used to compare the model were mean, standard deviation, skew coefficient computed over the series. The authors report a better performance of ANN based approach.

2.4 CONCLUSION

The review of literature has revealed that different investigators have put their efforts to develop reservoir-operating policy using different methods. Efforts are still on for development of more efficient operating policy for maximizing beneficial use of the reservoir. Several investigators have claimed to observe better or competitive performance of reservoir operating policy derived by recent techniques, namely, ANN, fuzzy logic, and GA when compared with the traditional techniques. Although extensive researches have been done towards finding a better method of deriving optimal operating policy, superiority of a particular method in general could not be claimed.

LP has been used by many researchers in reservoir operation provided the objective function and constraints are linear. Although, linearization technique such as piece wise linearization can be used, in many problems it many lead to large error.

NLP has not enjoyed the popularity that LP has in water resources system analysis. NLP can not easily accommodate the stochastic nature of inflow.

DP has been found to be very popular in reservoir operation, because it is well suited for solution of multistage decision process. The popularity and success of DP technique can be attributed to the fact that non-linear and stochastic features, which characterize a large number of water resources systems, can be translated into DP formulation.

ANN, which has the capability to solve non-linear problem, may replace the traditional regression procedure of developing reservoir operating rule from deterministic DP results.

Fuzzy rule base modeling is an alternative approach to inferring operating rules from historical operations or implicit stochastic optimization of reservoir systems. It can supplement the conventional optimization techniques but cannot probably be a replacement.

GA has the robust ability to solve highly non-linear problems. It has potential as an alternative to stochastic dynamic programming. The ability of GA to be linked directly with trusted simulation model is a great advantage.

Necessity of deriving synthetic streamflow has also been felt as different investigators have used synthetic streamflow to have the advantage of using longer time series in deriving better reservoir-operating policy. For generation of synthetic streamflow, ARMA model is mostly used. ANN has also been attempted in generation of synthetic streamflow and reported to be compared well with statistical models.

Review of literature has also revealed that performance of different models varies from problem to problem and the potential of the recent techniques in deriving optimal operating policy are yet to be fully explored for different situations. Therefore author has decided to explore the potential of GA and ANN in deriving optimal operating policy through their application in the up coming Pagladia multipurpose reservoir. However, the efficiency of the policies derived by these recent techniques need to be assessed through their critical comparison with policies derived by different long-established traditional techniques. Therefore, stochastic dynamic programming model and deterministic dynamic programming with multiple linear regression have been chosen as the benchmark models.

CHAPTER 3

THE RIVER SYSTEM AND THE RESERVOIR

3.1 INTRODUCTION

A description of the Pagladia River system, the proposed reservoir and its salient features has been presented in this chapter. All the information were obtained from the detailed project report of Pagladia dam project (Brahmaputra Board, Government of India, 1997)

3.2 THE RIVER SYSTEM

3.2.1 BASIN DESCRIPTION

The Pagladia River is one of the most important north bank tributaries of the Brahmaputra. It causes heavy flood damages in its middle and lower reaches by over spilling banks and occasionally by changing courses. It originates at the southern slopes of Bhutan hills at an elevation of about 1300m. It flows for a total length of 196.8km up to its confluence with Brahmaputra near Lowpara village. It drains an area of 1820km^2 up to its confluence with the Brahmaputra. Its drainage area up to Indo- Bhutan border is 381km^2. The river has an annual average yield of 943.294Mm^3 at dam site. The River flows for a length of 19 km in hilly tracts of the Bhutan territory and for the rest 177.8 km, it flows through the Nalbari district of Assam, India. In the hilly portion, slope of the river bed is very steep, being 1 in 75, in the middle reach it is 1 in 200 and in the lower reach, i.e., from Hajo-Nalbari road to outfall it is 1 in 2600. Fig. 3.1 shows index plan of the Pagladia dam project.

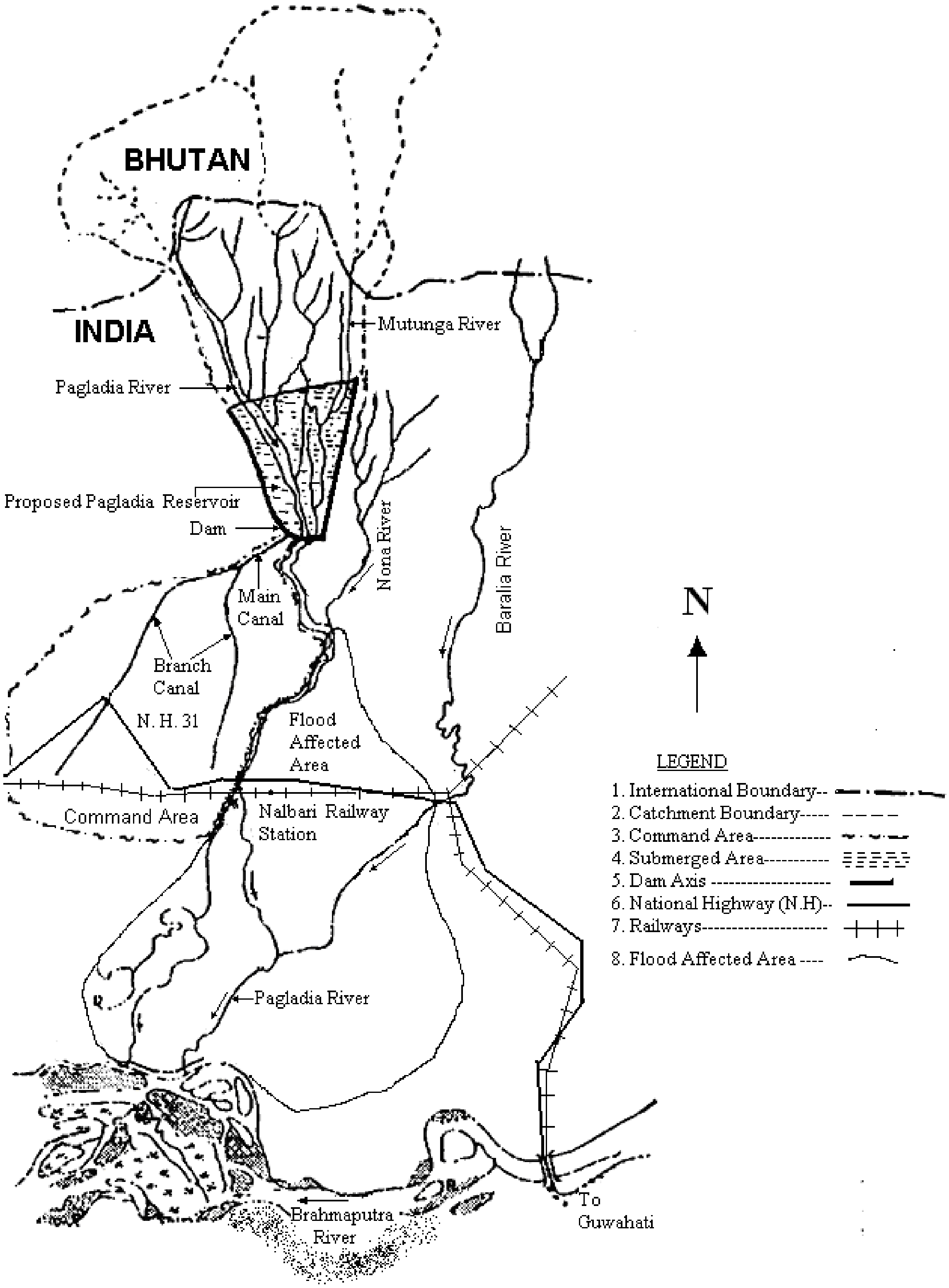


Fig.3.1. Index Plan of Pagladia Dam Project (Source: Brahmaputra Board, Government of India)

The main tributaries of the Pagladia River are (1) the Mutunga, (2) the Nona and (3) the Baralia. The tributaory Mutunga joins the Pagladia River at the upstream of the reservoir. The tributary Nona joins immediate downstream of the proposed Pagladia reservoir and the tributary Baralia joins further downstream from the reservoir. Due to the presence of these two tributaries on downstream of the reservoir, no mandatory release is considered from the proposed Pagladia multipurpose reservoir.

3.2.2 CLIMATE AND RAINFALL

Climate of the basin is similar to that of the other district in central Assam, India. The winter is cold and foggy while the summer is oppressively hot and humid. The rainfall is substantially high during the monsoon, which extends from May to September. Relative humidity in monsoon months (June to September) varies from 79% to 85%. The temperature in the winter goes down to about 11^0C and in the month of July-August it shoots up to about 37^0C-38^0C. The main rainfall season of the basin is from May to September during which 83% of annual rainfall occurs. The rainfall during October to April contributes only about 17% of the annual rainfall. The annual rainfall varies from 775mm to 3447 mm. Table 3.1 shows monthly mean rainfall in mm in the Pagladia river basin

Table3.1: Monthly mean rainfall

Month	Jan	Feb	Mar	April	May	June	July	Aug	Sept	Oct	Nov	Dec
Rainfall (mm)	15.27	21.98	54.76	186.30	345.21	482.35	567.14	340.17	299.73	107.04	16.52	16.75

3.2.3 AVAILABILITY OF STREAMFLOW DATA

A monthly streamflow series of 40 years in Mm^3 (million cubic meter) from 1957-1996 at dam site is available.

3.3 PAGLADIA RESERVOIR

3.3.1. LOCATION

The Pagladia reservoir is located near the village Thalkuchi (Latitude $26^0 37'30''$N, Longitude $91^0 30'45''$E) in the Nalbari district of Assam (India).

3.3.2. GEOLOGY OF THE DAM SITE

Along the centerline of the dam, the first upper layer of the area consists of silty clay. The river bed consists of fine to medium grain sand in the upper 6 m thick layer, below which there are fine sand with mica and clayey mix. Further below, a layer of coarse sand and pebbles of quartzite are available. No hard rock is available on the riverbed portion. The geology of the other parts of the dam alignment is similar to that of the riverbed portion.

3.3.3 PURPOSES OF THE RESERVOIR

The Pagladia reservoir, which is a multipurpose one, proposes to achieve three purposes, namely, flood control, irrigation and power generation.

3.3.3.1 FLOOD CONTROL

The area located on the downstream of the proposed dam site gets affected by floods almost every year. The loss due to flood include both tangible and intangible losses. While loss of lives and private properties such as livestock, crop, agricultural land, damage to public utilities like road, railways, power lines, telecommunications are the tangible losses, suffering of human being, set back in business, break out of epidemics, loss of educational environment etc. are the intangible losses.

Therefore the project proposes to mitigate the flood hazard in the downstream by lowering the flood peak by absorbing the same in the storage space provided in the reservoir for the purpose.

3.3.3.2 IRRIGATION

The project is estimated to provide irrigation facilities to 54160ha of gross command area on the right bank of Pagladia, downstream of the dam site with a net irrigation area of 34630ha.

3.3.3.3 HYDROPOWER GENERATION

The project is planned to generate maximum of 5.5 MW of hydropower. Canal head power house is proposed to generate hydropower.

3.3.4 PRINCIPAL FEATURES OF THE PAGLADIA PROJECT

The project is recommended for a full reservoir level (FRL) of EL 87.5 m with storage volume 312.64 Mm^3 and the maximum water level (MWL) of EL 91.0m with storage volume 472.64Mm^3. The reservoir is recommended for flood control reserve storage of 160 Mm^3, which is the storage volume between FRL and MWL. The minimum draw down level (MDDL) is fixed at EL 76.85m with storage volume 45.64Mm^3(dead storage volume). The crest level of canal intake is EL 76.85m. The elevation of the spillway crest is EL 77.0m and the maximum discharge through spillway is 2800cumecs. The spillway for the reservoir is a gated spillway. Safe carrying capacity downstream of the dam is 625cumecs.

Policy for water utilization is such that, the outflow from the canal head regulator will be regulated by a gate before leading to power house so that water requirement for 5.5MW power or so would be directed through turbine and remaining by a bypass, which will again join at the tail water reach so that combined discharges satisfy the irrigation requirement. However, when the power requirement is more than the irrigation requirement, the excess water will be let to the river by suitable gate adjustment in the tail water reach.

3.3.4.1 RESERVOIR CAPACITY VERSUS ELEVATION AND RESERVOIR CAPACITY VERSUS RESERVOIR AREA RELATIONSHIP

The polynomials representing relationship of capacity versus elevation and capacity versus area have been found from the curve fitting of the available data obtained from the detail project report, Pagladia dam project (Brahmaputra Board, Government of India 1997) as follows:

$$\text{Elevation (m)} = 2.5277E\text{-}013x^5 - 7.7446E\text{-}010x^4 + 8.7811E\text{-}007x^3 - 0.0004633x^2 + 0.13433x + 70.569 \quad (3.1)$$

$$\text{Area } (Mm^2) = 1.8131E\text{-}013x^5 - 6.3467E\text{-}010x^4 + 8.3992E\text{-}007x^3 - 0.00052245x^2 + 0.22165x + 1.839 \quad (3.2)$$

Where x is the reservoir capacity in Mm^3.

For fitting of the curve, norm of residual is used as the goodness of fit criteria . Norm of residual is calculated as

$$r = \left(\sum |r_i|^2\right)^{1/2}$$

Where r_i is the i^{th} residual of the point i.

3.3.4.2 THE PAGLADIA COMMAND

The gross command area (GCA) of the project is 54160ha with cultivable command area (CCA) of 40743.5ha. The net irrigable area is 34630ha.

3.3.4.3 BASIN NEEDS OR DEMANDS

3.3.4.3.1 IRRIGATION WATER REQUIREMENT

The project is planned to irrigate 54160ha of GCA, on the right bank of the Pagladia river downstream of dam site, with net irrigable area of 34630ha. The monthly irrigation water requirement is as given in Table 3.2.

Table3.2: Monthly irrigation water requirement:

Month	Jan	Feb	March	April	May	June	July	Aug	Sept	Oct	Nov	Dec
Demand (Mm^3)	0	0	30.98	50.55	33.56	17.06	34.29	43.72	42.19	54.65	0	0

3.3.4.3.2 POWER REQUIREMENT FROM POWER PLANT

Monthly power requirement from the power plant is as given in table 3.3.

Table 3.3 Monthly power requirements

Month	Jan	Feb	March	April	May	June	July	Aug	Sept	Oct	Nov	Dec
Power (10^3 MWh)	1.04	0.896	2.73	2.64	2.73	2.64	2.73	2.73	2.64	2.73	2.64	2.73

3.3.5 EVAPORATION

The monthly evaporation rate from Pagladia reservoir is stated in the Table 3.4 below.

Table3.4: Evaporation Rate from Pagladia Reservoir

Month	January	February	March	April	May	June	July	August	September	October	November	December
Evaporation (mm)	45.14	78.09	137.52	160.56	136.71	137.39	127.91	126.58	117.35	105.06	79.19	60.15

3.4 CONCLUSION

The Pagladia multipurpose reservoir on the river Pagladia is proposed to serve three purposes namely, flood control, irrigation and power generation. For the purpose of controlling flood the reservoir is proposed with flood control reserved storage. For optimum use of available water, the regulatory system is arranged in such a way that water released can be used for both irrigation and power generation.

The project has monthly irrigation demand and power demand. Monthly streamflow series for 40 years (1957-1996) at dam site along with monthly evaporation rate from Pagladia reservoir is available. Based on the purposes of the reservoir, water use policy, and available data, an optimization problem has been formulated to develop the optimal operating policy for the reservoir. Details of the problem formulation have been presented in the next chapter.

CHAPTER 4

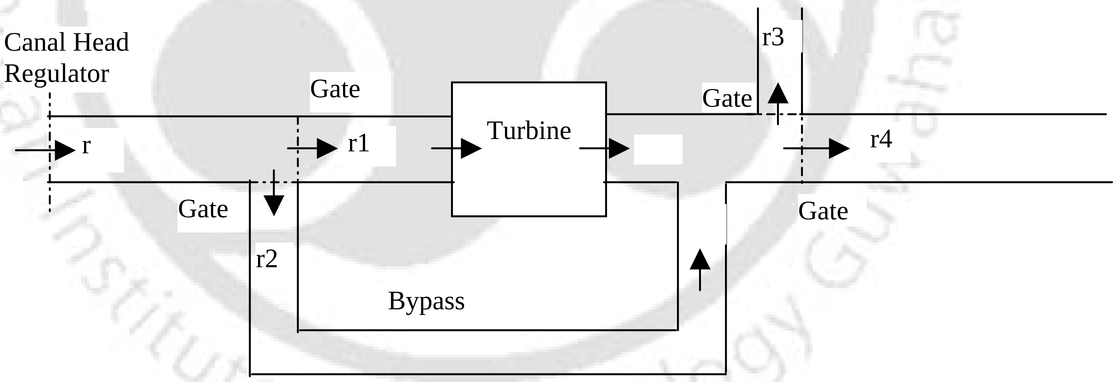
PROBLEM FORMULATION

4.1 INTRODUCTION:

In this chapter, on the basis of available monthly data, physical constraints of the reservoir and keeping in mind the purposes to be served by the reservoir, the optimization problem has been formulated for the Pagladia multipurpose reservoir. The formulation of an optimization problem means derivation of mathematical expressions for the objective function and the constraints of the problem.

4.2 PHYSICAL FEATURES OF THE RESERVOIR AND DISTRIBUTORY SYSTEM

As already mentioned in chapter 3, the Pagladia multipurpose reservoir is proposed to serve three purposes ,viz., irrigation, hydropower generation and flood control. The reservoir is proposed to operate from a storage volume of 45.64Mm^3 (MDDL) to 312.64Mm^3(FRL). Thus the conservative storage of the reservoir is 267.0Mm^3. For the purpose of controlling flood, the reservoir is planned with flood control reserve storage of 160.0 Mm^3. Including the flood control reserve storage, capacity of the reservoir at maximum water level (MWL) is 472.64 Mm^3. The reservoir needs to meet different irrigation water demand and different power demand in different months, as given in chapter 3. Fig.4.1 shows the schematic diagram of the regulating system for release to meet irrigation demand and water requirement for hydropower. The regulating system is proposed in such a way that when irrigation demand is more than the water requirement for power generation, the outflow from the head regulator will be regulated by a gate before leading to turbine so that water requirement for power demand would be directed through turbine and remaining by a bypass, which will again join at the tail water reach so that combined discharges satisfy the irrigation demand. However, when the water requirement for power is more than the irrigation demand, all the water will be led through the turbine and water in excess of irrigation demand will be led to the river by suitable gate adjustment in the tail water reach. Hence demand for each month of the year is the maximum of irrigation water demand and water requirement for hydropower. Again, release from the reservoir should be such that it does not violate the constraints of the upper limit (storage capacity at full reservoir level) and the lower limit (dead storage) at the end of each month.



r=total canal release, r1=maximum release through turbine, r2=release through bypass (r-r1), r3= water in excess of irrigation demand diverted to river, r4=irrigation supply

Fig. 4.1. Schematic diagram showing water distribution of the system

4.3 OBJECTIVE FUNCTION AND CONSTRAINTS

4.3.1 OBJECTIVE FUNCTION

The purpose of optimization is to choose the best of many acceptable designs or policies available. Therefore a criterion has to be chosen for comparing the different acceptable designs or policies and for selecting the best one. The criterion with respect to which a design or policy is optimized, when expressed as a function of the design variables, is known as the objective function (Rao, 1996). The choice of objective function is governed by the nature of the problem.

On the basis of the water distribution of the system, minimization of squared deficit of release from demand (demand is the maximum of irrigation water requirement and water requirement for hydropower) is considered as the objective function of the problem with release as the decision variables. Minimization of squared deficit of release from demand has the advantage that it tries to avoid large deficit in any particular month by distributing the deficit in more number of months. This is advantageous from management point of view. Mathematically the objective function or the system performance can be expressed as

$$\text{Minimize } f = \sum_{t=1} (D_t - R_t)^2 \tag{4.1}$$

Where t is the time period in month of a year. D_t is the demand in Mm^3 in time period t, R_t is the release in Mm^3 in time period t. Demand D_t in each time period t is the irrigation water requirement or the water requirement for hydropower whichever is maximum and is expressed as $D_t=\max(ID_t, PD_t)$, where ID_t is the irrigation water requirement in Mm^3 and PD_t is the hydropower water requirement in Mm^3. In the above equation of the objective function, if R_t is greater than D_t then (D_t-R_t) returns a zero value.

The irrigation requirement ID_t is as given in Table 3.2 (chapter 3). Water requirement for power generation (PD_t) is computed as follows:

The power generation in Kilowatt-hour (kWh) at any time t is given by the following equation of power generation

$$KWH_t = 2730 R_{pt} H_t \eta \quad \forall_t \qquad (4.2)$$

Where,

KWH_t = power generated (kWh) in time period t,

R_{pt} is the release in Mm^3 meant for power generation in any time period t,

H_t= average head (in meter) of the reservoir in time period t (the tail race of the turbine is EL 74.5m),

η = efficiency of the power plant (,i.e, turbine & generator efficiency) =0.85 (assumed)

PD_t in Mm^3 is calculated from the power demand, tabulated in the Table3.3 (chapter 3), by rewriting the above equation of power generation as,

$$PD_t = 2730 H_t \eta / \text{(Power demand in kWh in time period t)} \quad \forall_t \qquad (4.3)$$

4.3.2 CONSTRAINTS

(a) Continuity constraint of the reservoir:

$$S_{t+1} = S_t + Q_t - R_t - E_t \quad \forall_t \qquad (4.4)$$

Where,

S_t =storage in Mm^3 at the beginning of time period t;

S_{t+1} =storage in Mm^3 at the end of time period t or beginning of time period t+1;

Q_t = inflow in Mm^3 during time period t;

R_t = reservoir release in Mm^3 during time period t;

E_t = evaporation loss in Mm^3 during time period t.

The evaporation loss E_t is computed as the product of monthly mean reservoir area and monthly evaporation rate as shown in the equation below.

$$E_t = e_t A_{at} \qquad \forall_t \qquad (4.5)$$

Where e_t is the evaporation rate in meter per unit surface area of the reservoir during time period t and A_{at} is the mean reservoir area in Mm^2 during time period t. Mean reservoir area for time period t is calculated based on the average storage of the reservoir during time period t using the storage area relationship of the reservoir as given in equation 3.2 (chapter 3). The average storage of the reservoir during time period t is given as $(S_t+S_{t+1})/2$.

(b) Reservoir storage constraints:

The reservoir storage cannot be less than the dead storage volume of 45.64 Mm^3. The maximum storage of the reservoir at beginning or the end of any month cannot be more than the storage capacity of the reservoir at maximum water level (MWL) minus the flood control reserve storage. The maximum storage of the reservoir at the beginning or the end of any month = 312.64Mm^3(= 472.64 Mm^3 (storage at MWL) – 160.0Mm^3 (flood control reserve storage)) which is storage capacity of the reservoir at full reservoir level (FRL).

Thus the reservoir storage constraint is given as follows:

$$S^{min} \leq S_{t+1} \leq S^{max} \qquad \forall_t \qquad (4.6)$$

Where,

S^{min} = storage capacity of the reservoir at MDDL = dead storage volume =45.64Mm^3;

S^{max} = storage capacity of the reservoir at FRL = 312.64Mm^3.

(c) Constraints of release from the reservoir:

Release from the reservoir should be such that at the end of any month the reservoir storage level does not go above FRL and does not go below MDDL.

$R_t^{min} \le R_t \le R_t^{max} \quad \forall_t$ (4.7)

Where,

$R_t^{min} = \max [0, (S_t + Q_t - E_t - S^{max})] \quad \forall_t$ (4.8)

$R_t^{max} = S_t + Q_t - E_t - S^{min} \quad \forall_t$ (4.9)

In the above equations

R_t^{min} = minimum release in time period t;

R_t^{max} = maximum release in time period t;

4.4 CONCLUSION

The optimization problem for the development of monthly operating policy for the proposed Pagladia multipurpose reservoir has been formulated in this chapter. Minimization of squared deficit of the monthly release from the monthly demand has been considered as the most suitable objective function as it prevents large deficit in any particular month and tends to distribute the deficits uniformly in more number of months. This is generally advantageous for both agricultural and power management point of view. The flood control has been set as the constraints of the optimization problem by keeping flood control reserve storage empty at the beginning of any month. For the development of good operating policy a longer series of streamflow data is generally preferred. Therefore in the next chapter (chapter 5) synthetic streamflow data for the Pagladia River at dam site is generated.

CHAPTER 5

SYNTHETIC STREAMFLOW GENERATION

5.1 INTRODUCTION

A long time series of streamflow data is generally required for development of good operating policy for a reservoir. In the case of the Pagladia multipurpose reservoir considered in this study, 40 years of monthly streamflow data is available. However all the available 40 years of data cannot be used for developing the operating policy, as the performance of the developed operating policy need to be evaluated through simulation of the reservoir for a sequence of streamflow not used in the development of the policy. Therefore to have longer time series, synthetic monthly stremmflow data for Pagladia River at dam site has been generated. In this chapter 100 years of synthetic monthly streamflow for the Pagladia River has been generated using three different models namely, Thomas-Fiering model (Thomas and Fiering 1962), Autoregressive Moving Average (ARMA) model (Box and Jenkins 1976) and an artificial neural network (ANN) based model. All the synthetic streamflow generation models in this study have been developed based on the observed monthly streamflow data of 40 years (1957-996). A statistical comparison among the series generated by different models has been carried out to select the best series for its application in the development of optimal operating policy for Pagladia multipurpose reservoir.

5.2 THOMAS-FIERING MODEL

The Thomas Fiering model (Thomas and Fiering, 1962) is a simple model of monthly flows, which is often used for reservoir planning studies (Stedinger and Taylor 1982). In this study log transformed historic monthly streamflow data has been used to generate synthetic monthly streamflow using the Thomas-Fiering model, as the log-transformed data were found to be normally distributed. The use of log transformed streamflow data has the added advantage of eliminating the negative flows that occur occasionally when untransformed streamflows are used in the model (Maass et al. 1970)

The transformation of the historic streamflow $Q_{m,y}$ in month *m* and year *y* is given by

$$q_{m,y} = \ln(Q_{m,y}) \qquad (5.1)$$

where, $q_{m,y}$ is the transformed streamflow in month *m* and year *y*.

For Thomas-Fiering model, synthetic monthly $q_{m,y}$ series is generated with the following recursive relationship

$$q_{m+1,y} = \bar{q}_{m+1} + r_m (s_{m+1} / s_m)(q_{m,y} - \bar{q}_m) + s_{m+1}(1 - r_m^2)^{1/2} \zeta_{m,y} \qquad (5.2)$$

Where

$\bar{q}_m$ = observed average of the historic monthly streamflow series for month *m*;

$s_m{}^2$ = observed variance of the historic monthly streamflow series for month *m*

$\zeta_{m,y}$ = independent standard normal random variables;

r_m = observed correlations between month *m* and *m*+1 of the historic streamflow series

In the above equation, $q_{m+1,y}$ is understood to be $q_{1,y+1}$ when *m*=12

The $q_{m,y}$ values thus generated is then transformed to synthetic monthly flows by using the relation,

$Q_{m,y} = \exp(q_{m,y})$ (5.3)

Using the Thomas-Fiering model a 100 years of synthetic monthly streamflow series has been generated in this study for the Pagladia river at dam site.

5.3 AUTOREGRESSIVE MOVING AVERAGE (ARMA) MODELS

Autoregressive moving average (ARMA) models are linear stochastic models. Log transformed monthly streamflow data has been used for developing the ARMA model for synthetic streamflow generation. Before applying ARMA model for synthetic streamflow generation of the Pagladia River, the time series of monthly streamflow has been tested for its stationarity. Stationarity has been tested with respect to mean and variance. To test the statistics of stationarity, the available monthly data set (1957-1996) has been divided into 4 sub series each of 10 years. Table 5.1 shows mean, variance, and standard deviation of the total series along with the sub series for periods 1957-1966, 1967-1976, 1977-1986, and 1987-1996 of the log transformed historic monthly streamflow data.

Table 5.1. Mean, variance, standard deviation of sub series for period 1957-1966, 1967-1976, 1977-1986, and 1987-1996 and of the total series for the period 1957-1996 of log transformed historic monthly streamflow data.

Statistics	Period 1957-1966	Period 1967-1976	Period 1977-1986	Period 1987-1996	Period 1957-1996
Mean	3.69395	3.8075	3.53362	3.59583	3.65773
Variance	1.69213	1.3077	1.479	1.57468	1.52412
Std. Dev.	1.30082	1.14355	1.21614	1.25486	1.23455

Test of mean:

The 95% confidence interval for mean of the total series (1957-1996) is 3.436-3.878. From Table 5.1 it is observed that mean of all sub series lies within 95% confidence interval of the total series (1957-1996).

Test of variance:

To examine stationarity of the series in terms of variance it has been decided to check whether the sub series variances can be regarded as equivalent to the variance of the total series. Therefore test of null hypothesis H_0: $\sigma^2 = \sigma_0^2$ has been conducted.

The test statistic (Montgomery and Runger, 1994) is

$$Z_0 = \frac{S - \sigma_0}{\sigma_0 / \sqrt{2n}} \tag{5.4}$$

where, σ_0^2 is the variance of the population, and S is the standard deviation of sample of size n

and we would reject H_0 if $z_0 > z_{\alpha/2}$ or if $z_0 < -z_{\alpha/2}$.

Where α is significance level and is considered as 0.05 in this study

With $\alpha = 0.05$

$z_{0.025} = 1.96$ and $-z_{0.025} = -1.96$

Period 1957-1966

$z_0 = 0.83$

since $z_0 < z_{0.025}$ and $z_0 > -z_{0.025}$ we can not reject H_0

Period 1967-1976

$z_0 = -1.14$

since $z_0 < z_{0.025}$ and $z_0 > -z_{0.025}$ we can not reject H_0

Period 1977-1986

$z_0 = -0.23$

since $z_0 < z_{0.025}$ and $z_0 > -z_{0.025}$ we can not reject H_0

Period 1987-1996

z_0=0.25

since $z_0 < z_{0.025}$ and $z_0 > -z_{0.025}$ we can not reject H_0

From the test statistics of mean and variance it can be said that the historic streamflow series is stationary.

Modeling streamflow series:

For the purpose of modeling, the stationary time series has been decomposed into four additive components (Das 2000) as follows:

$$Z_t=T_t+P_t+D_t+V_t \qquad \forall_t \qquad (5.5)$$

Where Z_t=time series variable

T_t=trend component (a deterministic component)

P_t=periodic component (a deterministic component)

D_t=dependent stochastic component

V_t= independent stochastic component

t= time index in month

Deterministic components:

The plot of transformed streamflow series (Fig.5.1) has clearly shown that the series does not have any trend component but do have a periodic component with periodicity of 12 months.

Stochastic components:

The dependent stochastic component D_t and independent stochastic component V_t together comprises stochastic part S_t of the time series. The stochastic component of the streamflow series has been obtained by removing the periodic component.

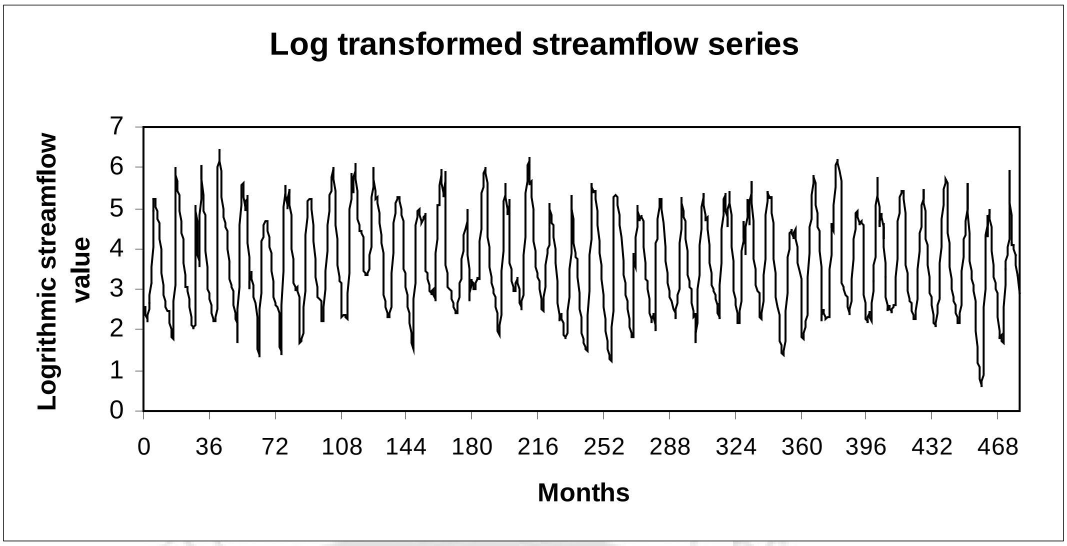


Fig. 5.1. Plot of logtransformed streamflow series of the Pagladia River at Dam site

Removal of Periodic Component:

Periodic component of the series has been removed by representing it as a Fourier series of the form

$$P_t = \mu + \sum_{i=1}^{h} A_i \cos(2\pi it/m) + B_i \sin(2\pi it/m)] \qquad \forall_t \qquad (5.6)$$

Where P_t =the harmonically fitted means at period t (t=1, 2, …, m),

m = number of period in the year,

μ =the population mean,

h=total number of harmonics(=m/2 or(m+1)/2 depending on whether m is even or odd),

A_i and B_i are Fourier coefficients, which are defined as

$$A_i = (2/m)\sum_{t=1}^{m} \overline{Z}_t \cos(2\pi it/m) \qquad i=1,2…h \qquad (5.7)$$

$$B_i = (2/m)\sum_{t=1}^{m} \overline{Z}_t \sin(2\pi it/m) \qquad i=1,2…h \qquad (5.8)$$

Where $\overline{Z}_t = (m/N)\sum_{j=1}^{N/m} Z_{t+m(j-1)}$

For monthly data m=12 and h=6. Though it is not necessary to expand the series up to the maximum number of harmonics, in this study all 6 harmonics have been included.

After removal of the periodic component, the stochastic part S_t of the transformed historical monthly streamflow series of the Pagladia River has been obtained.

Modelling Stochastic Component using ARMA Model:

The dependent stochastic component D_t of the S_t series is modeled by ARMA (p,q) family of models with p number of autoregressive terms and q number of moving average terms as follows:

$$S_{t+1} = \sum_{i=1}^{p} \Phi_1 S_{t=1-i} + V_t - \sum_{j=1}^{q} \theta_j V_{t-j} \qquad \forall\, t \qquad (5.9)$$

Where Φ and θ are the autoregressive and moving average parameters respectively.

The development of an ARMA model for a particular time series is often described as consisting of three steps (Loucks et al. 1981): (1) identification of reseanoable values of p and q; (2) estimation of parameters $\Phi_1,\ldots\Phi_p$, $\theta_1,\ldots,\theta_q$ for given values of p and q; and (3) diagnostic checks of the fitted model to ensure that it is an adequate model for the time series(Box and Jenkins,1976, Hipel et al, 1977). However, in practice, these first two steps are often combined. To identify values of p and q that may produce a reasonable model of a time series, the autocorrelations r_k and partial autocorrelations $PC_{k,,k}$ of a time series has been computed.

The autocorrelation r_k of lag k has been computed by the formula given by Kottegoda (1980) as follows:

$$r_k = \frac{\sum_{t=1}^{N-k} (Z_t - \overline{Z})(Z_{t+k} - \overline{Z})}{\sum_{t=1}^{N} (Z_t - \overline{Z})} \qquad \forall\, t \qquad (5.10)$$

where

r_k =autocorrelation function of time series Z_t at lag k

$\overline{Z}$ = mean value of the time series Z_t

N = total number of time series data

The partial autocorrelation function $PC_{k,,k}$ of lag k, has been computed by the formula given by Durbin (1960) as follows:

$$PC_{k,k} = \frac{r_k - \sum_{j=1}^{k-1} PC_{k-1,j}\, r_{k-j}}{1 - \sum_{j=1}^{k-1} PC_{k-1,j}\, r_j} \qquad \forall\, t \qquad (5.11)$$

where,

$PC_{k,k}$ =partial autocorrelation function at lag k

r_k =autocorrelation function at lag k

$PC_{k,j} = PC_{k-1,j} - PC_{k,k}\, PC_{k-1,k-j}$

$j = 1, 2, \ldots, k-1$

The computed values of autocorrelation and partial autocorrelation have been plotted against various lags k to select tentative order of the model. While the autocorrelation function is useful for identifying the value of q for an ARMA(0,q) model, the partial autocorrelation function is particularly useful for identifying the value of p for an ARMA(p,0) model.

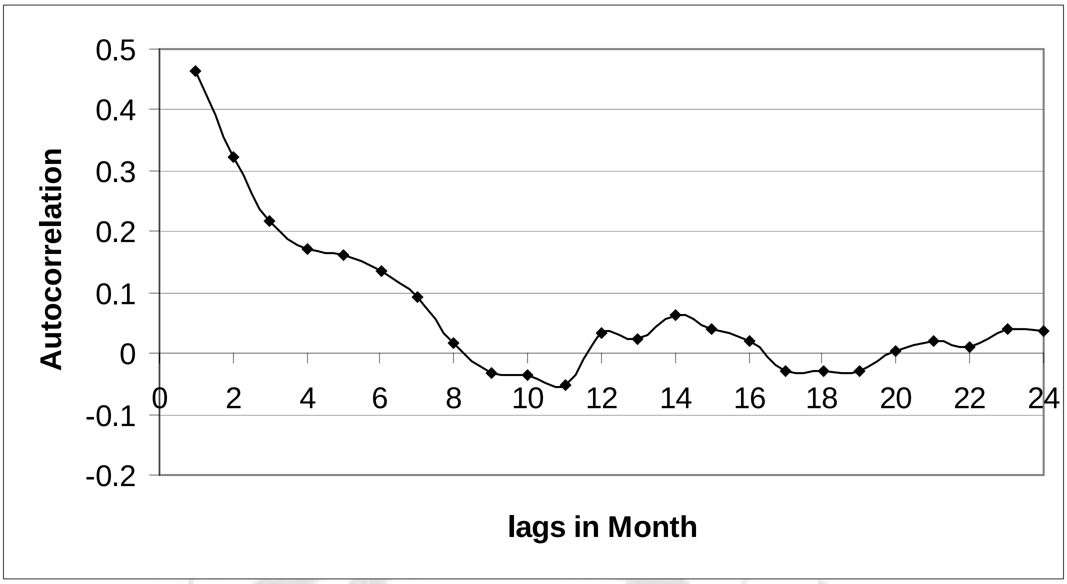


Fig.5.2 Autocorrelation of inflows to Pagladia dam site station for various lags.

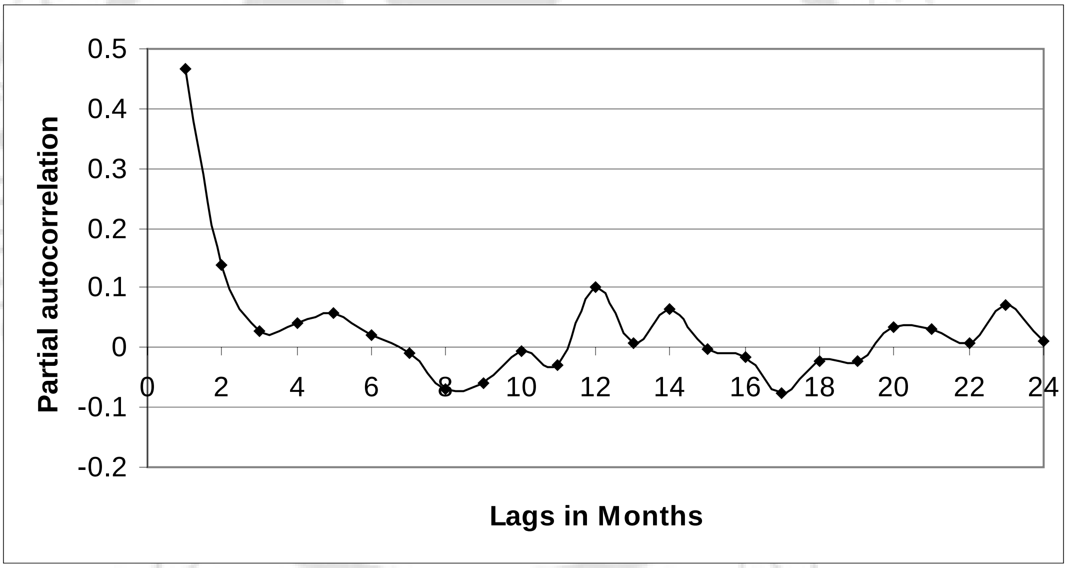


Fig.5.3. Partial autocorrelation of inflows to Pagladia dam site station for various lags

Fig. 5.2 shows the autocorrelation with various lags in months and Fig. 5.3 shows partial autocorrelation with various lags in months. Fig. 5.2 shows a dampened sine curve of the autocorrelation and Fig 5.3 shows that only first two partial autocorrelations have non zero value. Based on these two behaviors of the autocorrelation and partial autocorrelation, ARMA (2,0) model can be considered as the tentative model (Box and Jenkins 1976) for modeling the stochastic component. However, to ascertain this as the best model, 8 different models with different combinations of p and q values varying from 0 to 2 have been tested in this study.

Parameter Estimation:

Autoregressive parameters have been computed using following recursive formula given by Durbin (1980):

$$\Phi_{p,p} = \frac{r_p - \sum_{j=1}^{p-1} \Phi_{p-1,j} r_{p-j}}{1 - \sum_{j=1}^{p-1} \Phi_{p-1,j} r_j}$$

$$\Phi_{p,j} = \Phi_{p-1,j} - \Phi_{p,p} \Phi_{p-1,p-j} \qquad (5.12)$$

$j \quad =1, 2,3,\ldots, p\text{-}1.$

Where Φ is the autoregressive parameter, and p is the order of the autoregressive process and r_p is the autocorrelation of the time series at lag p.

The moving average parameter θ is computed by the following formula of Anderson (1976):

$$r_k = \frac{\theta_k + \sum_{j}^{q-k} (\theta_j)(\theta_{j+k})}{1 + \sum_{j=1}^{q} \theta_j^2} \qquad (5.13)$$

where k= 1, 2, …,q

The best ARMA model has been selected as the one, which minimizes the Akaike's information criteria (AIC) (Akaike, 1978) as given below:

$$AIC=2\ln(ML^*)+2(p+q+1) \qquad (5.14)$$

Where ML* is the maximum value of the likelihood function. The value of ML^* is given by the likelihood function as given below:

$$f_{V_1,...,V_n}(v_{1,...,}v_n) = (2\pi\sigma_v^2)^{-n/2} \exp\left(-\frac{1}{2\sigma_v^2}\sum_{t=1}^{n} v_t^2\right) \quad (5.15)$$

where σ_v^2 = variance of V_t series.

The Akaike information criterion (AIC) for various ARMA models is presented in Table 5.2. Among eight different ARMA models the AIC criteria has been found to be minimum for ARMA (2,0) model as shown in Table 5.2.

Table 5.2: Value of Akaike information criteria (AIC) for alternate ARMA (*p*,*q*) Models

q \ *p*	0	1	2
0		599.71	592.65
1	862.76	641.47	641.50
2	674.79	622.34	616.74

To check adequacy of the ARMA (2, 0) model, the residual series V_t generated by the model has then been subjected to diagnostic checking by modified Ljung-Box_Pierce (Ljung and Box, 1978) statistic. The modified Ljung-Box-Pierce statistic is given by:

$$Q = N(N+2)\sum_{k=1}^{K}(N-k)^{-1} r_k^2(v) \quad (5.16)$$

Where

N is the number of observations, *K* is total number of autocorrelation considered and $r_k(v)$ is the k^{th} autocorrelation of the V_t series.

In this study the value of Q for the first 30 autocorrelations (i.e. K=30) has been computed and found to be 23.05.

The 10% and 5% points for χ^2, with 28 (i.e. $K-p-q$) degrees of freedom are 37.9 and 41.3 respectively. Since the value of the statistic Q is less than the 10% and 5% points for the χ^2 distributions, the model ARMA (2,0) has been accepted.

The autoregressive parameters for the ARMA (2,0) model have been found to be Φ1=0.48 and Φ_2= 0.12. Using these values of $\Phi_1$and Φ_2, the stochastic part of the streamflow series has been generated using the following recursive equation starting from any month

$$S_t = 0.48 S_{t-1} + 0.12 S_{t-2} + \mu_v + \sigma_v \zeta_t \qquad \forall\, t \qquad (5.17)$$

Where

t = 1,2…***n***

n is the number of months for synthetic streamflow generation,

μ_v = mean of the residual series V_t,

σ_v = standard deviation of the residual series V_t,

ζ_t = independent standard normal random variables.

To start the recursive process, the first two S_t values (for t<1) are taken from the known transformed streamflow series.

The generated stochastic part for a month is then added to the deterministic component (periodic component) of the respective month and the anti-logarithm of this sum total gives the synthetic streamflow of that month.

Using this ARMA (2,0) model a synthetic streamflow series of 100 years has been generated.

5.4 ARTIFICIAL NEURAL NETWORK (ANN) APPROACH

5.4.1. FUNDAMENTALS OF ANN

Artificial neural networks (ANNs) are information processing system based on the present understanding of the biological nervous systems. The processing units of an artificial neural network are called neurons, which are arranged in layers. Neurons between layers are connected by links of variable weights. The most commonly used architecture of neural network (NN) is the feed forward neural network and is used in this study. Feed forward network has no feed back connections. The network has one input layer with neurons or nodes where data are introduced, one or more hidden layers with number of nodes or neurons, where the data are processed, and the output layer, where the results for given inputs are produced. The neuron in hidden layers and output layers are called activation function, which produces output based on the weighted sum of input signals entering into the neuron. The neurons in each layer receive as their inputs only the output of all neurons in the preceding layer, i.e. the input signal propagates through the network in a forward direction, on a layer-by-layer basis. The architecture of a three layer feed forward ANN is shown in Fig. 5.4. The information passing through the connections from one neuron to another are manipulated by weights that control the strength of a passing signal. When these weights are modified, the information transferred through the network changes and the network output alters.

The weights are adjusted to obtain the desired response from an ANN. This process is called learning. The learning process can either be supervised or unsupervised. Several learning examples are presented to the ANN, and when it has learned enough examples, it is considered trained. After the learning cycles, the weights are frozen. A data set that the ANN has not encountered before is presented to validate its performance. Depending on the outcome, either the ANN has to be re trained or it can be implemented for its designated use.

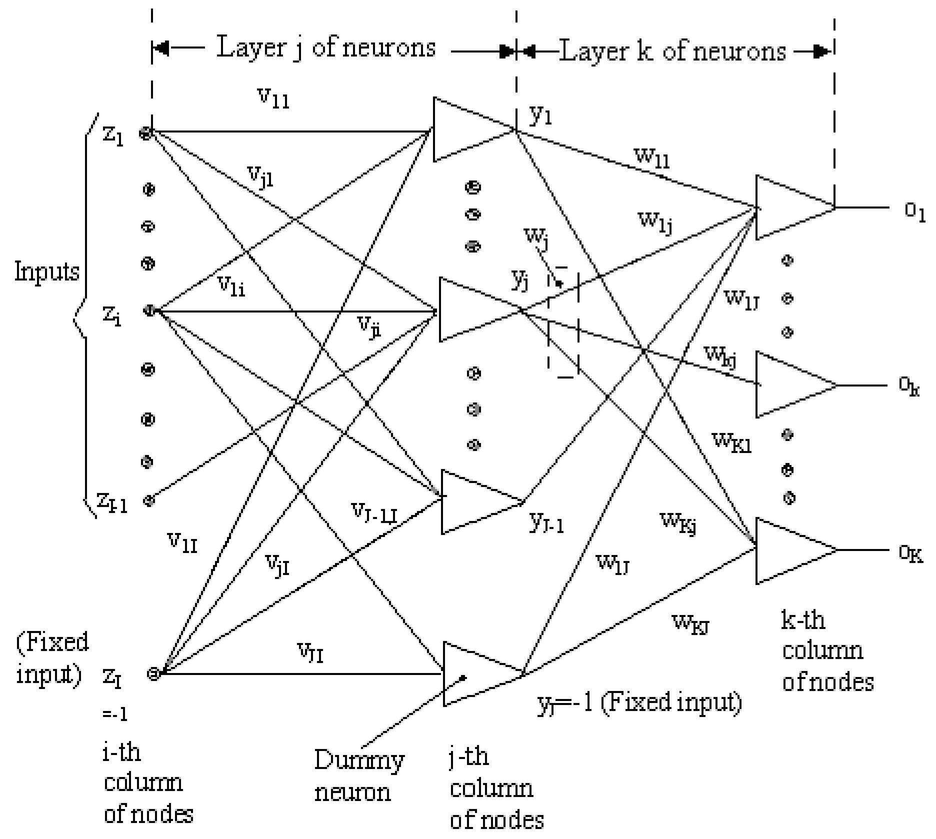


Fig. 5.4 Layered feedforward neural network with two continuous perceptron layers

A feedforward error backpropagation (BP) network may be adopted under a supervised learning mode. Neurons in the input layer act as buffers through which input data are sent. The dummy neurons act as a bias factor. The number of neurons in a hidden layer is decided after a rigorous course of training and testing. The sigmoidal function or linear function may be used for activation function. The basic characteristics of the sigmoidal funtion are that it is continuous, differentiable elsewhere, and is monotonically increasing. The input to the function is not restricted. For a unipolar sigmoidal function, the output is always bound between 0 and 1. In this study the unipolar sigmoidal activation function has been used. Using the unipolar sigmoidal function, the output y_j from a neuron in the hidden layer becomes

$$y_j = f\left(\sum w_{ji} x_i + b_j\right) = \frac{1}{1+e^{-\left(\sum w_{ji} x_i + b_j\right)}} \tag{5.18}$$

Where,

w_{ji} = the weight of the connection joining the j^{th} neuron in the hidden layer with the i^{th} neuron in the input layer,

x_i = the value of the i^{th} neuron in the input layer, and

y_j = the output from the j^{th} neuron in the hidden layer.

b_j = bias for the j^{th} neuron in the hidden layer

The output of neurons in the output layer is computed similarly.

The working principle of feed forward network is available elsewhere (Zurada 1999).

5.4.1.1 TRAINING OF ANN

ANN models are not inherently deterministic, but instead learn examples presented to them. The learning, or training of ANNs consists of showing example inputs and target outputs to the network and iteratively adjusting internal parameters until the network can produce meaningful results. After the ANN is trained, the relationship between inputs and outputs, which may be nonlinear and extremely complicated, is encoded in the network. The ANN is then able to produce output based on input fed to the input nodes. Training of the network is usually carried out using the back-propagation algorithm. Multi-layered network, trained by back-propagation are currently the most popular and proven (Hagan et al. 1996) and has been used in this study. The generalized delta rule, popularly known as backpropagation algorithm (Rumelhart et al. 1986), for training the network is embodied in the following steps:

(1) Start with an assumed set of weights. The initial weights are initialized with the help of a random number generator and they are very close to zero.

(2) Apply an input vector to the network and calculate the corresponding output values.

(3) Compare the computed outputs with the correct outputs and determine a measure of the error.

(4) Determine the amount by which each weight needs to be changed. In the bacpropagation algorithm, the weight associated with a neuron is adjusted by an amount proportional to the strength of the signal in the connection and the total measure of the error.

(5) Apply the corrections to the weights. The total error at the output layer is then reduced by redistributing this error backward through the hidden layers until the input layer is reached.

(6) Repeat the item 1-5 with all the training vectors until the error for all vectors in the training set is reduced to an acceptable value.

In this study training has been carried out using the backpropagation algorithm, which is a gradient descent procedure. The algorithms updates the interconnection weights w_{ji} using the derivatives δ_j as follows:

$$\Delta w_{ji}(s)=-\eta\delta_j x_i+\alpha\Delta w_{ji}(s-1) \tag{5.19}$$

where, η= learning rate; α= the momentum factor; s=epoch/training iteration number, δ=a factor depending on whether neuron j is an output neuron or a hidden neuron (Rumelhart and Mcclelled 1987). A training iteration is defined as one cycle of training using the considered pattern set. For the j^{th} neuron in the output layer

$$\delta_j = \left(\frac{d\,f}{d\,net_j}\right)[y_j^{(t)} - y_j] \qquad (5.20)$$

in which $y_j^{(t)}$ = desired response; y_j=output response from NN; f =activation fuction; and $net_j = \sum w_{ji} x_i$ (,i.e., weighted sum of input into the neuron j).

For the j^{th} neuron in the hidden layer

$$\delta_j = \left(\frac{df}{dnet_j}\right)\sum_q w_{qj}\delta_q \qquad (5.21)$$

in which q= number of neurons in the output layer; δ_q = already computed for the q^{th} neuron in the output layer.

Similarly changes in the bias for j^{th} neuron is given by:

$$\Delta b_j(s) = \eta\delta_j + \alpha\Delta b_j(s-1) \qquad (5.22)$$

The error backpropagation training algorithm flow chart (Zurada 1999) for a three-layered network is given in Fig. 5.5. In this study packaged ANN software has been used.

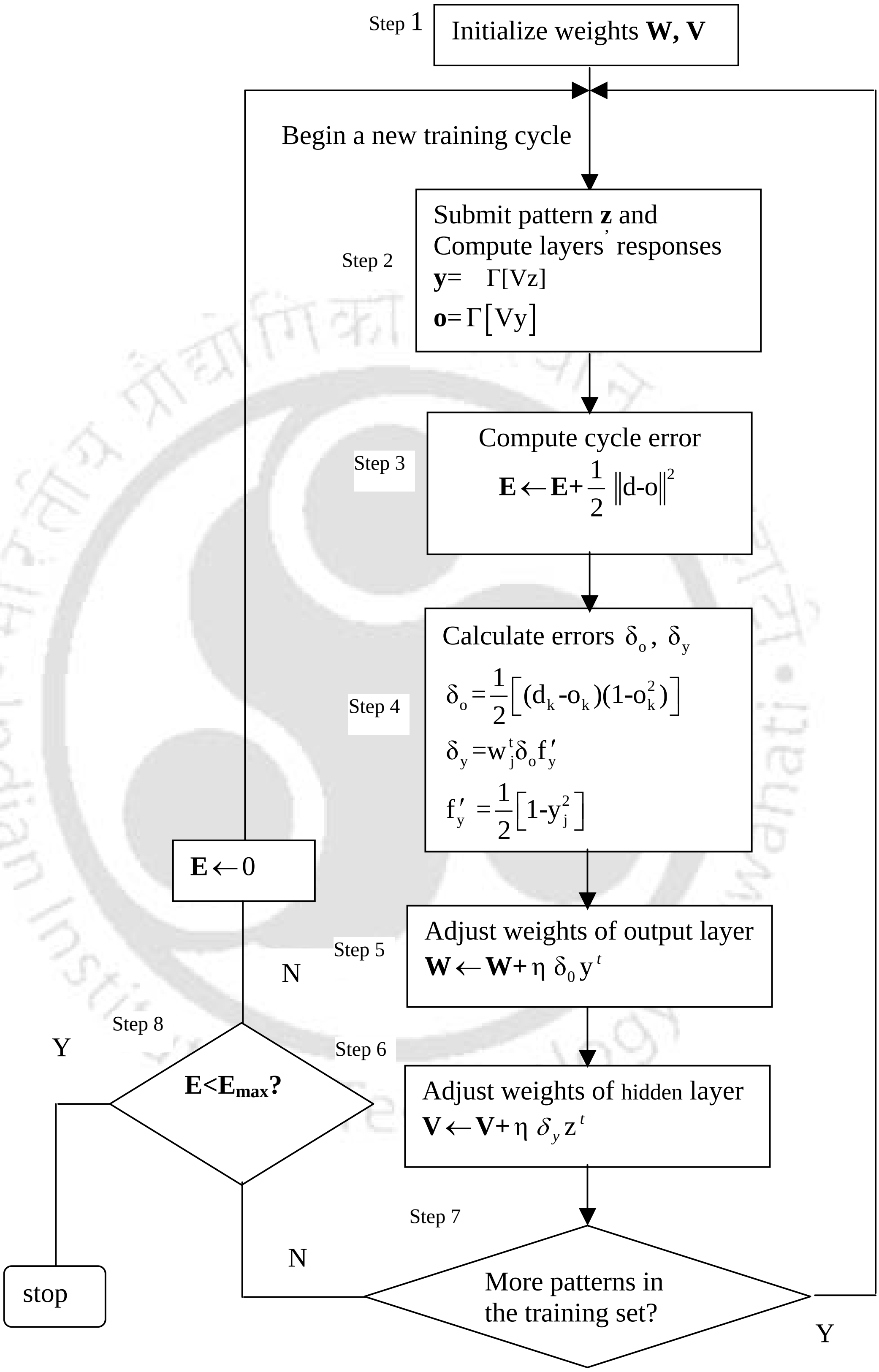


Fig.5.5 Error back-propagation training algorithm flowchart

The performance measures used in this study for the evaluation of neural network are mean square error (MSE) and mean relative error (MRE). They are defined by:

$$MSE = \frac{1}{pq}\sum_{q}\sum_{j=1}^{p}[y_j^{(t)} - y_j]^2 \qquad (5.23)$$

$$MRE = \frac{1}{pq}\sum_{q}\sum_{i=1}^{p}\left|\frac{[y_j^{(t)} - y_j]}{y_j^{(t)}}\right| \times 100 \qquad (5.24)$$

Where, $y_j^{(t)}$ = standardized target value for pattern j,

y_j = output response from the network for pattern j,

p= total number of training pattern.

q=number of output nodes

The MSE and MRE are good measures for indicating the goodness of fit at high and moderate output values, respectively (Karunanithi et al, 1994).

5.4.1.2 CHOOSING THE BEST NETWORK

It is indeed a very difficult task to choose the best network. It requires trial and error procedure. The network architecture, learning rate η(eta) and momentum factor α (alpha) values are finalized after examining various combinations. The effectiveness and convergence of training depends significantly on the value of learning rate. If it is too high, then the search may miss a valley in the error surface. On the other hand if it is too small, the convergence will be very slow (Chandramouli and Raman 2001). A momentum factor α (alpha) is generally used to accelerate the convergence. The learning rate η and the momentum factor α values are decided after examining different combinations. The number of neurons in the hidden layer of the neural network is finalized after a trail an error procedure using different combinations of learning rate and momentum factor. Burian et al. (2001)

stated that typically the generalization of prediction and accuracy of an application increases as the number of hidden neurons decreases; as the number of hidden neurons increases, there is a corresponding increase in the number of parameters describing the approximating functions. Consequently more the hidden neurons employed the more specific the trained ANN is to the training data. Often in an ANN application the number of neurons in the hidden layer are selected based on the particular application and only after testing various number of neurons in the hidden layer.

5.4.2 ANN BASED MODEL FOR SYNTHETIC STREAMFLOW GENERATION

An ANN based model has been developed in this section to generate synthetic streamflow for the Pagladia River at dam site. The ANN based model in this study consist of two components, namely an ANN component and a random component. While the ANN component has been used to generate a streamflow value, the addition of a random component to the ANN generated value is necessary to eliminate the chances of generating same sequence of streamflow repeatedly.

5.4.2.1 DVELOPING THE ANN COMPONENT

In this study, a three-layered feed forward network has been used. The activation function used in the hidden and output neuron is sigmoidal which is unipolar. Selection of input variables for the ANN model has been done after rigorous experimentations requiring extensive trials. Finally, three variables, namely, streamflow of current month, mean of historic streamflow of the current month and standard deviation of historic streamflow of the current month has been found to be the best combinations of input variables. The output of the network is the streamflow of the succeeding month. Thus the input pattern to the network is consisting of streamflow of current month, mean and standard deviation of historic streamflow for the current month. The output pattern is the streamflow of the succeeding month of the time series. The architecture of the neural network is shown in Fig.5.6.

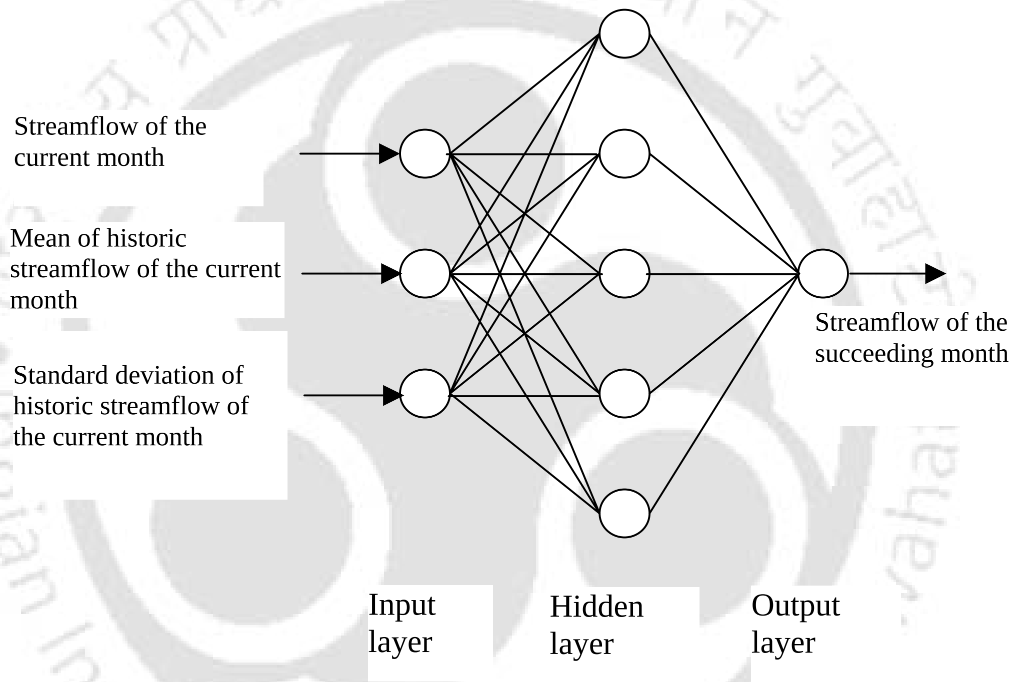


Fig.5.6 Artificial Neural Network Streamflow Generation Model

Out of the 40 years (1957-1996) of historic monthly streamflow data, the streamflow data pertaining to 35 years (1957-1991) has been used to train the neural network and the remaining 5 years streamflow data (1992-1996) has been used to validate the efficiency of the ANN model. Since there are twelve mean flows and twelve standard deviation of flows for twelve months ,i.e., January through

December, therefore, in the input pattern the mean and standard deviation of the historic monthly streamflow repeats after each twelve time steps. Since the output response of the sigmoidal activation function is between 0 and 1, the target output value of monthly streamflow is standardized by dividing them with the maximum value among the target value of streamflow.

Training of the network has been carried out using error backpropagation training algorithm. The learning rate η and the momentum factor α values are decided after examining different combinations. Learning rate η =0.01, 0.02, 0.05, 0.1, 0.5 and 0.9, and momentum factor α = 0.2, 0.3, 0.5, and 0.9 have been considered to find the best combinations. The number of neurons in the hidden layer of the neural network has been finalized after a trail an error procedure using different combinations of learning rate and momentum factor. Each combination of learning rate and momentum factor has been tested for different numbers of hidden neurons. The network has been trained for 2000 epochs (iterations) as there is very negligible improvement of the MSE value after 2000 epochs.

In this investigation, number of hidden neurons considers are 3, 4, 5, 6 and 10. Fig. 5.7 shows reduction of MSE for a network having 5 neurons in the hidden layer and for some of the combinations of learning rate η and momentum factor α. It is observed that for a network having 5 neurons in the hidden layer, the best value of η=0.9 and α =0.5.

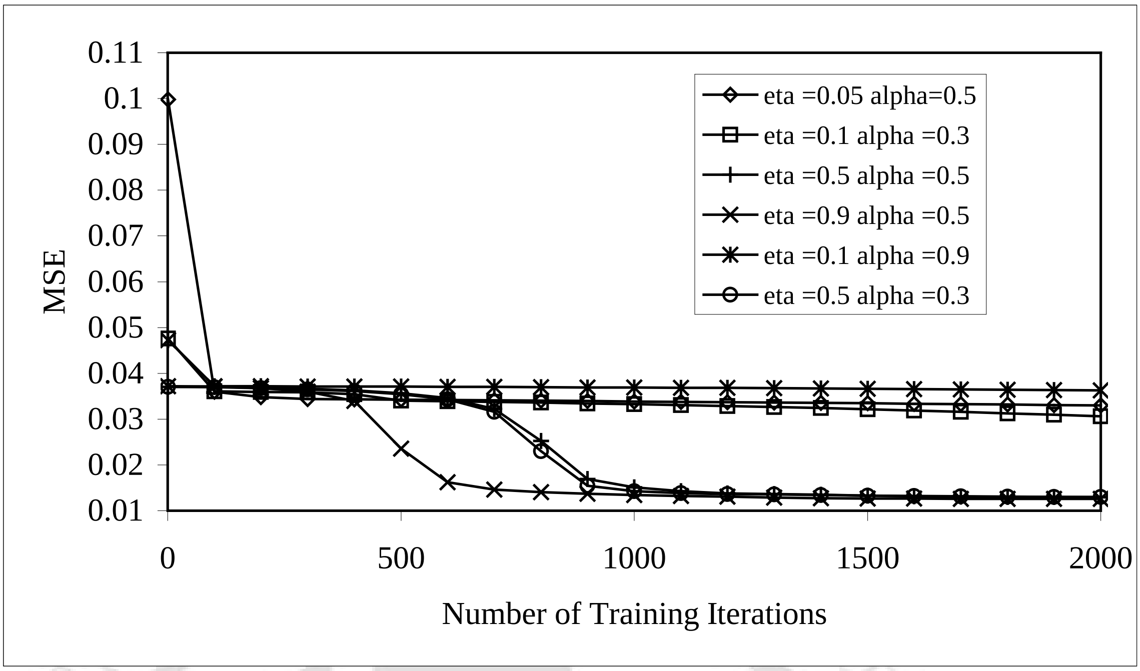


Fig.5.7 Reduction of MSE for different combinations of η(eta) and α (alpha)

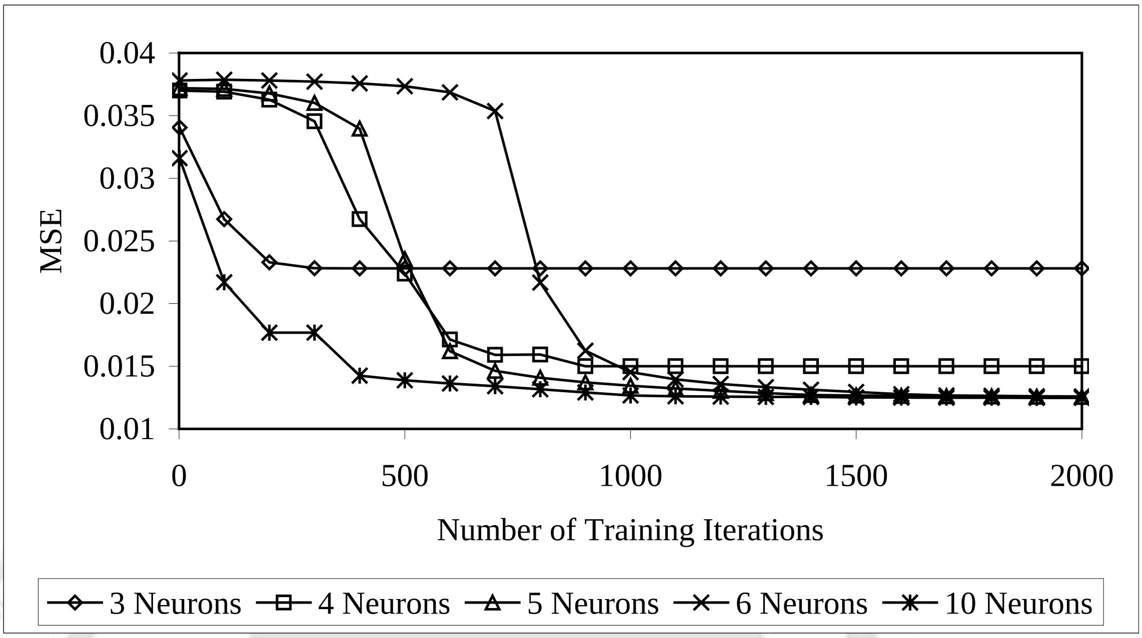
For each network having a definite number of neurons in the hidden layer, the best combination of η(eta) and α (alpha) values have been chosen after 2000 training iterations. Table 5.3 shows the best combinations of η and α values for network with different number of neurons in the hidden layer.

Table 5.3: Best combinations of η and α values for network with different number of neurons in the hidden layer

Hidden neurons	η	α
3	0.5	0.5
4	0.9	0.5
5	0.9	0.5
6	0.9	0.5
10	0.9	0.5

Fig.5.8 shows reduction of MSE for networks with different numbers of neuron in the hidden layer. It has been found that the networks having 5, 6 and 10

neurons in the hidden layer give MSE value very close to each other after 2000 training iteration.



.

Fig.5.8 Reduction of MSE for network with different numbers of neurons in the hidden layer.

All the trained networks have then been tested using 5 years of historic monthly streamflow data (1992-1996). Table 5.4 shows MSE and MRE values of training and testing for network having different number of neurons in the hidden layer.

Table 5.4: MSE and MRE values of training and testing for network having different number of neurons in the hidden layer.

Hidden Neurons	Training		Testing	
	MSE	MRE	MSE	MRE
3	0.0228	59.6742	0.0166	52.1387
4	0.01480	41.5578	0.0114	30.7612
5	0.01255	32.7815	0.0101	20.4316
6	0.01259	32.9261	0.0101	20.6007
10	0.01247	32.6416	0.0102	20.8858

From Table 5.4 it has been observed that MSE and MRE of training is minimum for the network having 10 neurons in the hidden layer. MRE and MSE values of training for the network having 5 neurons in the hidden layer are also very close to those of the network having 10 hidden neurons. However, during testing, minimum value of MSE has been found in networks having 5 and 6 neurons in the hidden layer. Both the networks having 5 and 6 neurons in the hidden layer give equal amount of MSE value equal to 0.0101. But the network having 5 neurons in the hidden layer gives the least value of MRE. Therefore network having 5 neurons in the hidden layer has been considered to be the best network in this study.

5.4.2.2 GENERATING SYNTHETIC STREAMFLOW

In this investigation when the trained ANN was simulated to generate streamflow sequences, it was found that after generating streamflow sequence for a few years, the sequence was repeated. This happened when the network generated a value for a particular month, which the network had generated for that particular month in a previous year. Therefore to prevent the network from repeatedly generating the same sequence of streamflow, a small random component calculated on the basis of the standard deviation of the observed streamflow is added to the output produced by the network. The random component is chosen as $\xi_t\sigma_t$, where, ξ_t is independent standard normal random variable with mean zero and variance unity, σ_t is the standard deviation of observed streamflow of the corresponding month.

Thus the synthetic streamflow model is consisting of two parts, namely, neural network and a random component. The following procedure has been adopted in this study to generate synthetic streamflow.

Initially a known streamflow of any month (say, January) along with the mean and standard deviation of historic streamflow for that month are fed to the network input nodes and the output from the network is computed. The output produced by the network is the streamflow of the succeeding month. The output of the network is then modified by adding a small random component to it as stated above. If the streamflow so modified becomes negative, then it is replaced with the minimum observed streamflow of the month. Now the corrected streamflow along with the mean streamflow and standard deviation of streamflow of the month is used as input to the network to generate a streamflow for the next succeeding month. The procedure is repeated for required number of times to get the required length of synthetic streamflow series. In this study 100 years of synthetic streamflow has been generated using the ANN based model.

5.5 COMPARISON OF SYNTHETIC STREAMFLOW GENERATED BY DIFFERENT MODELS

Three synthetic streamflow generation models, namely, Thomas-Fiering model, ARMA (2,0) model and ANN-based model have been used to generate 100 years of synthetic monthly streamflow series. The synthetic monthly streamflow series have been compared with the observed streamflow series of 40 years (1957-1996). The statistics considered to compare the synthetic series are mean value of streamflow of each month, standard deviation of streamflow of each month, mean of the streamflow series, standard deviation of the streamflow series and skew coefficient of the streamflow series. Fig.5.9 shows a comparative plot of mean value of streamflow of each month for the observed and synthetics series. Among all the synthetically generated series, the series generated by ANN based model has been found to be quite closer to the observed series in respect of the mean value of streamflow of each month. Fig.5.10 shows a comparative plot of standard deviation of streamflow of each month for the observed and synthetics series. In respect of this statistic also, among all the synthetically generated series, the series generated by ANN based model has been found to be quite closer to the observed series. Table 5.5 shows mean, standard deviation and skew coefficient of the observed and synthetic monthly streamflow series. The mean of the synthetic series generated by ANN-based model is 82.2Mm^3 against observed mean of 78.5Mm^3. On the other hand, mean of the synthetic series generated by Thomas-fiering model and ARMA (2,0) model are 87.3Mm^3 and 86.3Mm^3 respectively. The standard deviation of the synthetic series generated by ANN-based model, Thomas-Fiering model and ARMA (2,0) model are 99.03Mm^3, 61.1Mm^3 and 108.0 Mm^3 respectively as against the standard deviation of 96.2 Mm^3 of the observed streamflow series. Among all the synthetic streamflow series, the skew coefficient of the synthetic series generated by the ANN-based model has been found to be closer to that of the observed streamflow series. From the above comparison it has been found that synthetic streamflow series generated by ANN based model is better than the synthetic streamflow series generated by the Thomas-Fiering model and ARMA (2, 0) model in case of Pagladia River.

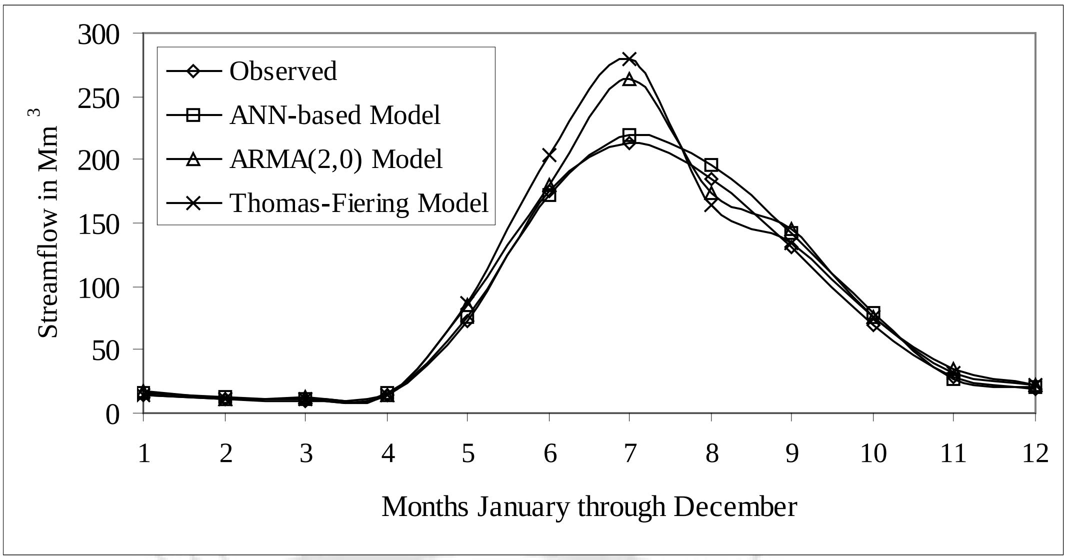


Fig.5.9 Comparison of synthetic monthly mean streamflow for various models with observed monthly mean streamflow

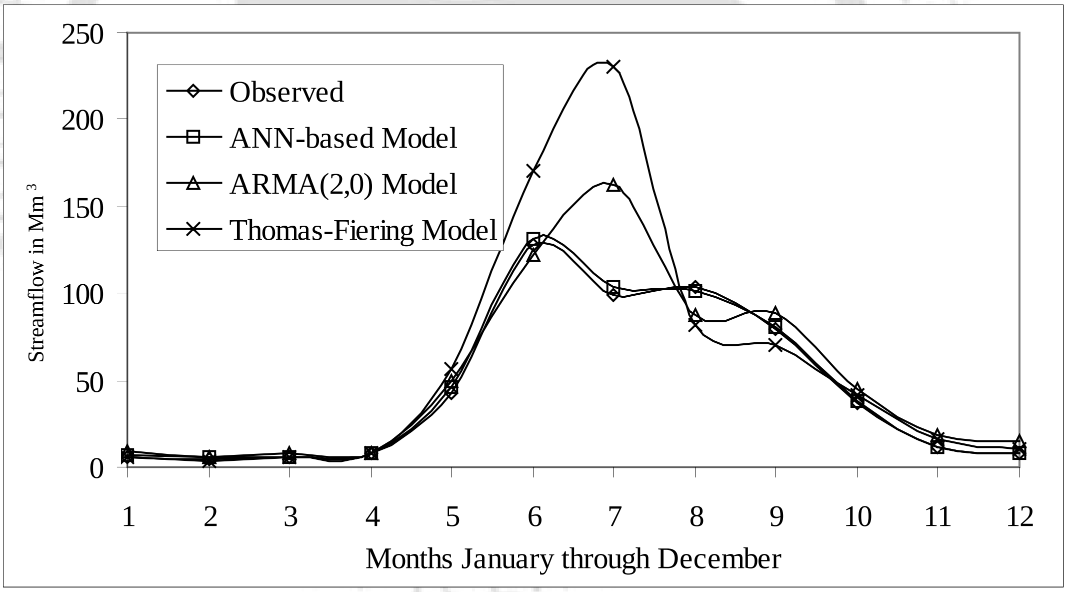


Fig. 5.10. Comparison of synthetic monthly standard deviation of streamflow for various models with observed monthly standard deviation of streamflow

Table 5.5. Mean, standard deviation and skew coefficient of 100-year synthetic monthly streamflow series by various models and of 40 years observed monthly streamflow series.

	Mean	Standard deviation	Skew coefficient
Observed	78.5	96.2	2.101
ARMA(2,0)	86.3	108.0	3.832
Thomas-Fiering	87.3	61.1	2.672
ANN-based	82.2	99.03	1.743

5.6 CONCLUSION

Three different models have been used to generate synthetic monthly streamflow series for Pagladia River at dam site. The models used for synthetic streamflow generation are Thomas-Fiering model, ARMA (2,0) model and ANN-based model. Streamflow series generated have been compared with the observed series on the basis of five statistics, namely, mean value of streamflow of each month, standard deviation of streamflow of each month, mean of the streamflow series, standard deviation of the streamflow series and skew coefficient of the streamflow series. From the comparison of the statistics it has been found that the synthetic monthly streamflow series generated by ANN-based model is better than that of the Thomas-Fiering model and ARMA (2,0) model. Therefore in this study 100 years of synthetic monthly streamflow series generated by the ANN based model has been chosen for the purpose of developing reservoir operating policy for the Pagladia multipurpose reservoir.

CHAPTER 6

RESERVOIR OPERATION MODEL USING DETERMINISTIC DYNAMIC PROGRAMMING

6.1 INTRODUCTION

Dynamic programming (DP) is a mathematical technique dealing with the optimization of multistage decision process. A multistage decision process is one in which a number of single-stage processes are connected in series so that the output of one stage is the input to the succeeding stage. Since reservoir operation requires a multistage decision over the planning horizon, dynamic programming is well suited for solution of such problem. DP can handle non-linear objective function and constraints. Deterministic DP model uses a specific sequence of streamflows, either historical or synthetically generated, in determining reservoir-operating rules (Karamouz and Houck 1987). In this study 100 years of synthetic monthly streamflow series generated by artificial neural network (ANN) based model has been used in the deterministic DP model. As deterministic DP model solves a specific sequence of streamflow for reservoir operation, it cannot be regarded as a general operating policy. Therefore, for practical application, a general operating policy must be determined. Two approaches have been attempted to obtain general operating policy from the deterministic DP results; one using multiple linear regression called DPR, and another using ANN called DPN model in this study.

6.2 MODEL FORMULATION USING DETERMINISTIC DYNAMIC PROGRAMMING

6.2.1 MODEL FORMULATION

A backward moving discrete dynamic programming has been formulated for the Pagladia multipurpose reservoir. The objective function considered in this study is minimization of squared deficit of release from demand, as given in the chapter 4. Time period is considered as the stage of the model and reservoir storage is considered as the state of the model. The decision variable is the release from the reservoir. As the model has been formulated to develop monthly operating policy, therefore, the stage of the model is time period in month.

6.2.1.1 DISCRETIZATION

Dynamic programming considered in this study is a discrete dynamic programming. Therefore storage volume is restricted to finite sets of characteristic storages volumes.

6.2.1.2 RECURSIVE EQUATION

The objective function for the deterministic dynamic programming algorithm is written as follows:

$$\text{Minimize } f = \sum_{t=1}^{T} (D_t - R_t)^2 \tag{6.1}$$

The detail of the objective function has already been explained in chapter 4.

The recursive equation for any time period t is

$$f_t^n(S_t) = \underset{S_t \in \varphi_t}{\text{minimum}}[Z_t + f_{t+1}^{n-1}(S_{t+1})] \tag{6.2}$$

where,

$Z_t = (D_t - R_t)^2$ for $(D_t - R_t) > 0$, and returns a zero value otherwise.

T=total number of optimization stages (in this study T=1200),

t = index of time period (in month),

S_t = reservoir storage (a state variable) at the beginning of time period t,

Q_t = inflow during time period t,

D_t =demand during time period t,

R_t = release during time period t,

ϕ_t = discrete set of characteristic storage volumes considered at the beginning of time period t,

n = total number of time periods remaining including the current period before the reservoir operation terminates.

The above recursive equation is solved subject to the following constraints, details of mathematical formulation of which have already been discussed in chapter 4.

Continuity constraint of the reservoir:

$$S_{t+1} = S_t + Q_t - E_t - R_t \tag{6.3}$$

Reservoir storage constraint:

$$S^{min} \leq S_{t+1} \leq S^{max} \tag{6.4}$$

Where,

S^{min} =45.64Mm^3 and S^{max} =312.64Mm^3.

Constraints of release from the reservoir:

$$R_t^{min} \leq R_t \leq R_t^{max} \tag{6.5}$$

Where,

$R_t^{min} = \max[0, (S_t + Q_t - E_t - S^{max})]$

$R_t^{max} = S_t + Q_t - E_t - S^{min}$

6.3 DETERMINISTIC DYNAMIC PROGRAMMING MODEL APPLICATION

6.3.1 DISCRETIZATION OF STORAGE

The Pagladia Reservoir is proposed to operate from a storage volume of 45.64 Mm^3 (MDDL) to 312.64Mm^3(FRL). Therefore, conservative storage of the reservoir is 267.0 Mm^3, which is 29.26% of the mean annual flow (943.294Mm^3) to the reservoir. Three different discretization of the conservative storage have been used in the deterministic dynamic programming model. Table 6.1 shows number of characteristic storages and discretization steps.

Table 6.1 Number of characteristic storages and discretization steps

Number of characteristic storages	Discretization step (Mm^3)
20	14.05
30	9.21
40	6.85

6.3.2 INPUT DATA

The data used for developing the deterministic dynamic programming model are:

(1) 100 years of synthetic monthly streamflow (i.e. T=1200 months).

(2) Monthly evaporation rate

(3) Monthly irrigation water requirement

(4) Monthly power requirement.

Monthly evaporation is computed as the product of monthly mean reservoir area and monthly evaporation rate.

6.3.3 MODEL RESULTS

Three deterministic dynamic programming models with different storage discretization have been solved for same set of input data. The solution of the deterministic dynamic programming gives the optimal releases against the initial storages and inflows into the reservoir for 1200 months. Therefore a large number of patterns consisting of initial storage, inflow and optimal release are obtained from the solution of deterministic dynamic programming. Table 6.2 shows the number of characteristic storagse, discretization steps, and the number of patterns consisting of initial storage, inflow and optimal release.

Table 6.2 Number of characteristic storages, discretization steps, and number of patterns

Number of characteristic storages	Discretization step (Mm^3)	Number of Pattern obtained
20	14.05	24000
30	9.21	36000
40	6.85	48000

The pattern obtained from deterministic dynamic programming result is optimal only for the flow series considered and cannot be used as a general operating policy. Therefore general operating policy must be determined. Two approaches have been used in this study to derive the general operating policy from the patterns obtained from deterministic dynamic programming results. They are multiple linear regression and artificial neural network approach. The general operating policy derived using multiple linear regression from deterministic dynamic programming result has been called DPR model and the general operating policy derived using artificial neural network from deterministic dynamic programming result has been called DPN model in this study.

6.4 MULTIPLE LINEAR REGRESSION MODEL

Bhaskar and Whitlach (1980), after examining different forms, suggested the simple linear form to express the optimal release as a function of initial storage and inflow during the period. Karamouz and Houck (1982) used such a form in their DPR model. Raman and Chandramouli (1996) used linear form to express the optimal release in their DPR model. In this study, the optimal release is expressed as a linear function of initial storage and inflow during the time period as shown in the equation 6.6.

$$R_t = aS_t + bQ_t + c \qquad \forall\ t \qquad (6.\ 6)$$

Where a, b, c are the coefficients of the general operating policy.

The patterns obtained from the deterministic dynamic programming result are regressed using the least square method. In this study multiple linear regression model that have used the patterns obtained from deterministic dynamic programming results with 20 characteristic storages of the reservoir is called DPR1 model. Similarly the multiple linear regression models those have used patterns obtained from the deterministic dynamic programming results with 30 and 40 characteristic storages are called DPR2 and DPR3 respectively. Table 6.3 shows the value of the coefficients of the multiple linear regression equation 6.6 of the DPR models.

Table 6.3 Coefficients of Multiple Linear Regression for DPR models

Model	Coefficients		
	a	b	c
DPR1	0.2779	0.5553	-32.8653
DPR2	0.2678	0.595	-44.2306
DPR3	0.2742	0.579	-40.3128

6.4.1 PERFORMANCE EVALUTATION OF DPR MODELS

The performance evaluation of the policies developed by DPR models has been carried out through simulation of the reservoir for 228 months of historic streamflow series (1977 - 1996). The starting month of the simulation was October 1977 with initial storage of the reservoir as 45.64Mm^3 (dead storage volume). The final month of the simulation period was September 1996.

Following criteria are considered in the reservoir simulation to evaluate the performance of the models in this study.

(1) Total squared deficit.

(2) Total irrigation deficit.

(3) Number irrigation deficit month.

(4) Total power generation.

(5) Number power deficit month.

(6) Total water deficit.

(7) Number of times the reservoir is full.

(8) Total spill.

6.4.1.1 SIMULATION AND RESULTS

Table 6.4 shows the simulation results of the reservoir for 228 months of historic streamflow (1977-1996) using DPR models.

Table 6.4 Simulation results of DPR models for 228 months of historic streamflow data (1977-1996)

Model	Total squared deficit (10^5)	Total irrigation deficit (Mm^3)	Number of irrigation deficit Month	Total power generation (10^8 kWh)	Number of power deficit month	Total water deficit (Mm^3)	Number of times the reservoir is full	Total spill (Mm^3)
DPR1	9.67	916.4	44	3.22	182	9474.9	18	2535.75
DPR2	6.83	1012.87	49	3.25	180	8043.21	27	2832.03
DPR3	7.65	972.43	48	3.25	179	8436.21	21	2766.5

From the Table 6.4 it has been observed that among three DPR models, total squared deficit has been found minimum in DPR2 model for the simulation period. In terms of total irrigation deficit, number of irrigation deficit months, and total spill, the DPR1 model has been found to be better than other two models; however it gives maximum squared deficit, less power generation, and maximum water deficit and keeps the reservoir full for less number of times. DPR2 and DPR3 models have been found to give equal power generation. All the models give almost equal numbers of power deficit months. Total water deficit have been found minimum in DPR2 model. The DPR2 model has also been found to keep the reservoir full for maximum number of times than other two models. From the analysis it has been seen that the models perform differently with respect to different performance criteria.

6.5 ARTIFICIAL NEURAL NETWORK (ANN) MODEL

The details of the artificial neural network procedure have already been discussed in chapter 5. In this section the procedure of deriving general reservoir operating rule using neural network has been presented. The patterns obtained from the solution of deterministic dynamic programming have been used to train the neural network. The input pattern consists of initial storage and inflow into the reservoir during the period. The output pattern considered is the optimal release during the time period corresponding to initial storage and inflow. Three different DPN models are developed in this section, they are DPN1, DPN2 and DPN3 based on the number of patterns obtained from deterministic dynamic programming result. The DPN1 model uses 24000 patterns obtained from the solution of deterministic dynamic programming that have used 20 characteristic storages. The DPN2 model uses 36000 patterns obtained from the solution of deterministic dynamic programming that have used 30 characteristic storages. The DPN3 model uses 48000 patterns obtained from the solution of deterministic dynamic programming that have used 40 characteristic storages. The network considered here is a feed forward neural network and training is carried out using backpropagation algorithm with gradient descent search technique. The activation function used for neurons is sigmoid activation function, which is unipolar. The network has three layers; one input layer, one hidden layer and one-output layer. Since the output response of a unipolar sigmoid activation function is between 0.0 and 1.0 therefore standardization of output patterns are essential. Output patterns are standardized by dividing them by the maximum value in output patterns. Fig. 6.1 shows the architecture of the neural network reservoir operation model.

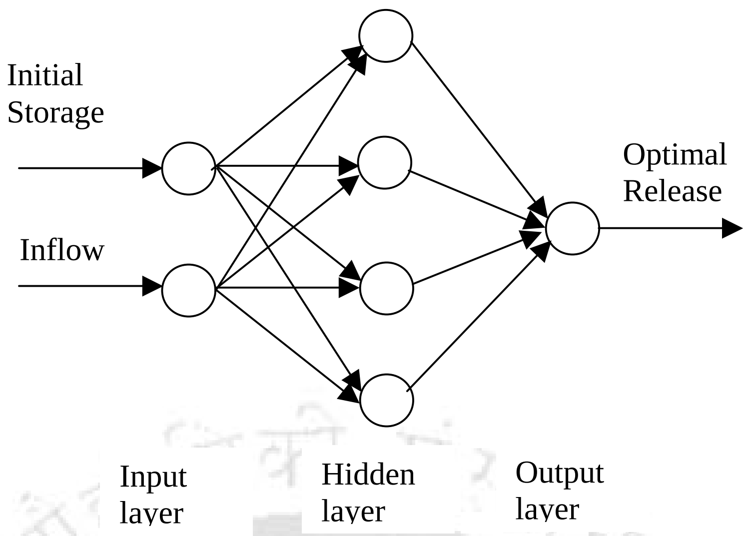


Fig. 6.1. Architecture of neural network reservoir operation model

6.5.1 TRAINING AND TESTING OF ANN

6.5.1.1 TRAINING OF ANN

To train the neural network some performance measures are necessary. The performance measures used in this study for the evaluation of neural network are mean square error (MSE) and mean relative error (MRE), which were explained in chapter 5.

As stated in chapter 5, the effectiveness and convergence of the neural network training depends significantly on the value of learning rate η(eta). A momentum factor α (alpha) is used to accelerate the convergence of training. The learning rate η and the momentum factor α values are decided after examining different combinations. Learning rate η =0.01, 0.02, 0.05, 0.1, 0.5, 0.9 and momentum factor α = 0.2, 0.3, 0.5, 0.9 have been considered to find the best combinations. The number of neurons in the hidden layer of the neural network has been finalized after carrying a trail an error procedure using different combinations of learning rate and momentum factor. For each combination of learning rate and momentum factor the network with different numbers of hidden neurons has been trained for 2000 training iterations/ epoch, as there is very negligible improvement of performance function after 2000 training iterations. In this investigation, number of hidden neurons considers are 3, 4, 5, 6 and 10. This procedure has been used to train all the three DPN models, namely, DPN1, DPN2, and DPN3.

Fig. 6.2 shows reduction of MSE for a network having 4 neurons in the hidden layer and with some of the combinations of learning rate η and momentum factor α for DPN1 model. It has been observed that the best combination is η=0.02 and α =0.9 for a network having 4 neurons for DPN1 model.

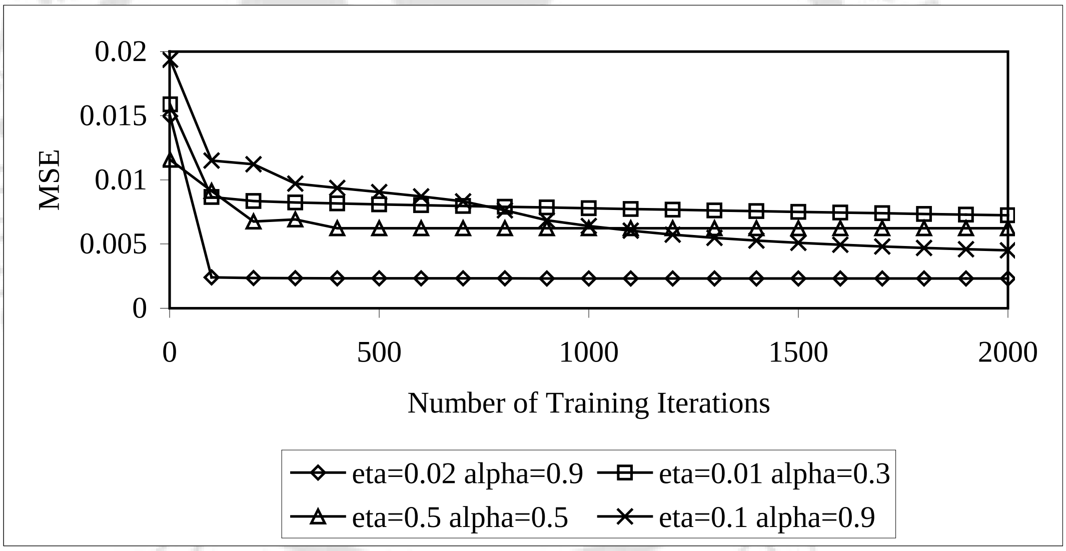


Fig. 6.2. Reduction of MSE of training for different combinations of learning rate (eta) and momentum factor (alpha) in a network having 4 neurons in hidden layer

Fig. 6.3 shows the reduction of MSE for networks having different number of neurons in the hidden layer with their best combinations of η and α values for DPN1 model. It is observed that except the network with 3 neurons in the hidden layer, all other network give MSE value very close to each other after 2000 training iterations.

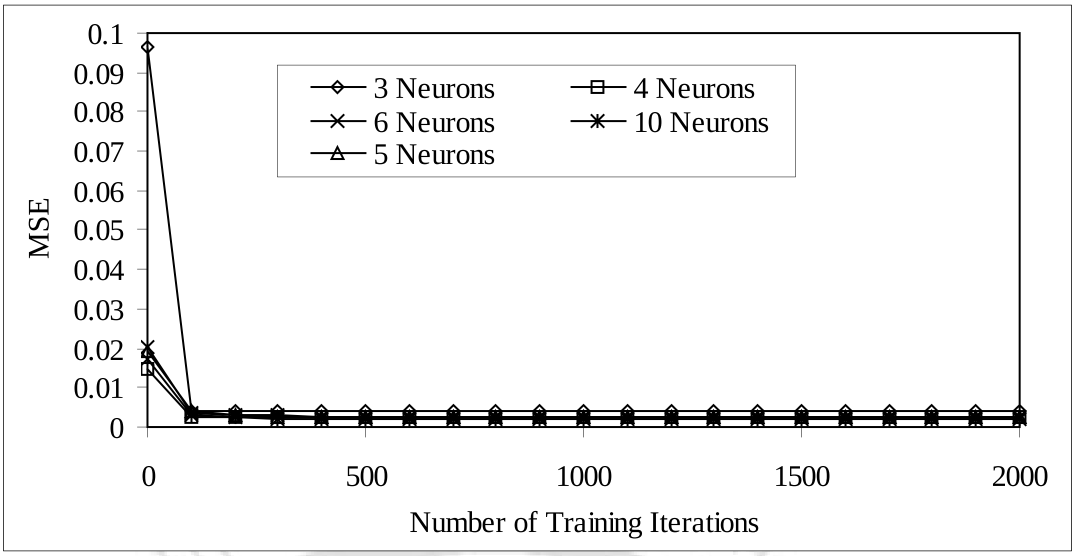


Figure 6.3: Reduction of MSE in the networks having different number of neurons in the hidden layer with their best combinations of learning rate (eta) and momentum factor (alpha)

Table 6.5 shows the MSE values of training after 2000 iterations along with MRE values for networks having different number of neurons in the hidden layer with their best combination of η and α values for DPN1 models. The minimum MSE and MRE values have been found for the network having 10 neurons in the hidden layer.

Table 6.5: MSE and MRE of training for DPN1 model with different number of neurons in hidden layer and best combinations of η and α values

Number of Hidden Neurons	η	α	Training Iterations	MSE	MRE
3	0.01	0.3	2000	0.00392502	18.5452
4	0.02	0.9	2000	0.00223228	10.7148
5	0.01	0.3	2000	0.00225473	14.9964
6	0.01	0.3	2000	0.00239248	10.0677
10	0.01	0.3	2000	0.00184637	8.3385

Table 6.6 shows the MSE values of training after 2000 iterations along with MRE values for networks having different number of neurons in the hidden layer with their best combination of η and α values for DPN2 models. The minimum MSE and MRE values have been found from the network having 10 neurons in the hidden layer. The MSE and MRE values of the network having 4 neurons in the hidden layer are close to those of the network with 10 neurons in the hidden layer.

Table 6.6. MSE and MRE of training for DPN2 model with different number of neurons in hidden layer and best combinations of η and α values

Number of Hidden Neurons	η	α	Training Iterations	MSE	MRE
3	0.01	0.3	2000	0.00374303	22.4534
4	0.02	0.9	2000	0.00222165	9.953
5	0.01	0.3	2000	0.00292979	16.0302
6	0.01	0.3	2000	0.00283897	13.1186
10	0.01	0.3	2000	0.00215883	8.9314

Table 6.7 shows the MSE values of training after 2000 iterations along with MRE values for networks having different number of neurons in the hidden layer with their best combination of η and α values for DPN3 models. The minimum MSE and MRE values have been found for the network having 10 neurons in the hidden layer. However, the MSE and MRE values for the network having 4 neurons in the hidden layer are also small and close to those of the network having 10 neurons in the hidden layer.

Table 6.7. MSE and MRE of training for DPN3 model with different number of neurons in hidden layer and best combinations of η and α values

Number of Hidden Neurons	η	α	Training Iterations	MSE	MRE
3	0.01	0.3	2000	0.00354409	18.4794
4	0.02	0.9	2000	0.00206611	9.7473
5	0.01	0.3	2000	0.00260617	15.3498
6	0.01	0.3	2000	0.00248827	11.5272
10	0.01	0.3	2000	0.00182739	8.4593

6.5.1.2 TESTING OF ANN

After training, testing of the networks has been carried out. The test patterns were obtained from solution of deterministic dynamic programming using 20 years of historic streamflow data (1957-1976). The test patterns for DPN1, DPN2 and DPN3 model were found from solution of deterministic dynamic programming using 20, 30 and 40 characteristic storages of the reservoir respectively. The performances were measured on the basis of three criteria, namely, MSE, MRE and sum of squared deficit of release from demand. While MSE and MRE have been used to test the mapping capability of the network, the squared deficit value has been used to test its performance in reservoir simulation for the period 1957-1976. Since the objective function in the present study is the minimization of squared deficit of release from demand, therefore sum of squared deficit for the test period is an important measure to select the optimal network for each of the DPN models.

Table 6.8 shows the testing results obtained from the simulation of the trained networks for DPN1 model. It has been found that MSE and MRE values are minimum for the network having 10 neurons in the hidden layer. However the MSE and MRE values of the network having 4 neurons in the hidden layer are very close to those of the network having 10 neurons in the hidden layer. On the other hand the simulated sum of squared deficit has been found minimum for the network having 4 neurons in the hidden layer. Therefore the network having 4 neurons in the hidden layer is considered as the optimal network for DPN1 model.

Table 6.8. MSE and MRE for test pattern and simulated sum of squared deficit for test period (1957-1976) for DPN1 model

Number of hidden neuron	MSE	MRE	Simulated sum of squared deficit for test period (10^5)
3	0.0044	21.3133	7.8023
4	0.0023	8.4162	6.9489
5	0.0032	19.8411	23.365
6	0.0025	10.2111	29.969
10	0.0022	8.3118	7.9505

Table 6.9 shows the testing results obtained from the simulation of the trained networks for DPN2 model. It has been found that MSE value is minimum for network having 4 and 10 hidden neurons. The MRE value has been found minimum for the network having 10 neurons in the hidden layer. The simulated sum of squared deficit on the other hand has been found minimum for the network having 4 neurons in the hidden layer. Therefore the network having 4 neurons in the hidden layer is considered as the optimal network for DPN2 model.

Table 6.9: MSE and MRE for test pattern and simulated sum of squared deficit for test period (1957-1976) for DPN2 model

Number of hidden neuron	MSE	MRE	Simulated sum of squared deficit for test period (10^5)
3	0.0043	22.4537	7.5864
4	0.0022	8.7193	6.6293
5	0.0037	21.2450	22.638
6	0.0032	16.1729	27.973
10	0.0022	7.9935	6.9277

Table 6.10 shows the testing results obtained from the simulation of the trained networks for DPN3 model. It has been found that MSE values for the networks having 4 and 10 hidden neurons are equal and minimum. The MRE value has been found minimum in the network having 10 neurons in the hidden layer. MRE value for the networks having 4 hidden neurons has also been found to be very close to that of the network having 10 hidden neurons. On the other hand simulated sum of squared deficit has been found minimum for the network having 4 neurons in the hidden layer. Therefore the network having 4 neurons in the hidden layer is considered as the optimal network for DPN3 model.

Table 6.10. MSE and MRE for test pattern and simulated sum of squared deficit for test period (1957-1976) for DPN3 model

Number of Hidden neuron	MSE	MRE	Simulated sum of squared deficit for test period (10^5)
3	0.004	21.2763	7.6572
4	0.002	7.8853	6.7113
5	0.0026	15.3351	22.432
6	0.0027	13.5163	28.222
10	0.002	7.5212	7.0567

6.5.2 PERFORMANCE EVALUTATION OF DPN MODELS

The performance evaluation of DPN models has been carried out through simulation of the reservoir for 228 months of historic streamflow series (1977 – 1996). The starting month of the simulation was October 1977 with initial storage of the reservoir as 45.64Mm^3 (dead storage volume). The final month of the simulation period was September 1996.

The performance criteria considered for evaluation of DPN models are same as used in the DPR models.

6.5.2.1 SIMULATION AND RESULTS

Table 6.11 shows the simulation results of the reservoir using DPN models for 228 months of historic streamflow (1977-1996).

Table 6.11 Simulation results of DPN models for 228 months historic streamflow data (1977-1996)

Model	Total squared deficit (10^5)	Total irrigation deficit (Mm^3)	Number of irrigation deficit Month	Total power generation (10^8 kWh)	Number of power deficit month	Total deficit (Mm^3)	Number of times the reservoir is full	Total spill (Mm^3)
DPN1	8.32	1178.0	45	3.22	170	8416.90	33	2679.7
DPN2	7.98	1203.6	47	3.19	166	8213.40	27	3219.2
DPN3	8.09	1182	47	3.20	168	8272.20	36	2878.4

From the Table 6.11 it has been observed that among three DPN models, total squared deficit is minimum in DPN2 model for the simulation period. In terms of total irrigation deficit, number of irrigation deficit months, total power generation, and total spill, the DPN1 model have been found to be better than other two models although it has been found to give maximum squared deficit than other two models.

Number of deficit months for power generation and total water deficit have been found minimum in DPN2 model. DPN3 model have been found to keep the reservoir full for maximum numbers of times than other two models. Although different DPN models, considered in this study, perform differently with respect to different performance criteria, no significant difference have been observed among them in any of the criteria considered.

6.6 CONCLUSION

The general operating policy from the deterministic DP result for the Pagladia multipurpose reservoir has been derived in this chapter. Two approaches have been used to derive the general operating policy from deterministic DP result; one using multiple linear regression and another using ANN and the corresponding models are called DPR and DPN models respectively. Three DPR and three DPN models have been developed in this chapter. While development of DPR model is straight forward, the development of DPN model requires extensive experimentation for finding a better-generalized network. Considering the similarity in simulation result, DPR model could be preferred to DPN model because of simplicity. The DPN model, on the other hand, cannot be ignored because many authors found better performance of DPN model than DPR model (, e.g, Raman and Chandramouli 1996).

Operating rules in this chapter have been derived using implicit stochastic procedure. In the next chapter an explicit stochastic method, namely, stochastic dynamic programming (SDP) has been attempted to derive the optimal operating policy for the Pagladia multipurpose reservoir.

CHAPTER 7

RESERVOIR OPERATION MODEL USING STOCHASTIC DYNAMIC PROGRAMMING

7.1 INTRODUCTION

Stochastic dynamic programming (SDP) is an explicit stochastic optimization technique. The SDP model for reservoir operation uses a statistical description of the streamflow process instead of a specific streamflow sequence. The SDP models are very useful for long term operating policy. They can handle non-linear objective function and constraints and are well suited for sequential decision process. In this chapter optimal monthly operating policy for the Pagladia multipurpose reservoir has been developed using SDP model. The SDP describes streamflow with a discrete lag-one Markov process. The model developed in this study incorporates transition probability matrices derived from 100 years of synthetic monthly streamflow data generated by artificial neural network based model.

7.2 MODEL FORMULATION USING STOCHASTIC DYNAMIC PROGRAMMING (SDP)

7.2.1 ASSUMPTIONS

The following assumptions have been made in the development of stochastic dynamic programming model

(1) The stochastic process is stationary and the probability distribution is not changing over time.

(2) Only inflows to the reservoir is considered as stochastic variable and it is assumed to follow first order Markov Chain.

(3) The model is discrete since continuous variables of time, storage and inflow are approximated by discrete units.

7.2.2 STATE VARIABLES

In the reservoir operation model in this study, the reservoir release is a function of initial storage of the reservoir and current month inflow to the reservoir. Therefore, initial storage and current month inflow are chosen to be the state variables in this study.

7.2.3 DISCRETIZATION

The discretization of inflow and storage volumes to a set of predetermined intervals is the first and most important step in the discrete SDP model formulation. The characteristic or representative inflow and storage are found out by partitioning the range of inflow and storage into suitable intervals.

7.2.4 TRANSITION PROBABILITY MATRIX

A common assumption in SDP model is that, the streamflow follows a first order Markov process. A Markov chain is a special kind of Markov process whose state X(t) can take only discrete values. The first order Markov chain has the property that the value X_t of the process at time t, depends only on its value X_{t-1} at time t-1.

Mathematically,

$$\text{Prob}(X_t = Q_j \mid X_{t-1} = Q_i,\ X_{t-2}, = Q_a,\ X_{t-3} = Q_b, \ldots, X_0 = Q_q)$$

$$= \text{Prob}(X_t = Q_j \mid X_{t-1} = Q_i) \qquad (7.1)$$

which is called the transition probability and it gives the probability that the process at time t will be in state 'j' given that at time t-1 the process was in state 'i'. The transition probability is denoted by P_{ij}^{t}, where

$$P_{ij}^{t} = \text{prob}(X_t = Q_j \mid X_{t-1} = Q_i) \qquad (7.2)$$

The transition probability satisfy the condition

$$\sum_j P_{ij} = 1, \text{ for all 'i'} \qquad (7.3)$$

To estimate the transition probability matrix of inflows, the range of possible inflow is discretised into convenient number of class intervals.

7.2.5 STATE TRANSFORMATION

The reservoir storage state transformation is governed by the mass balance (continuity equation) as given in chapter 4.

7.2.6 SDP RECURSIVE EQUATION

Stochastic dynamic programming model developed in this study is a backward moving dynamic programming algorithm. Stage of the model is time period in month. At each stage or time period t, the optimal reservoir release or equivalently final storage volume depends on two state variables: the initial storage volume and current inflow.

The objective function of the optimization problem is minimization of squared deficit of release from demands as explained in chapter 4.

The recursive relationship for backward moving SDP is

$$f_t^n(k,i) = \underset{l}{\text{minimum}}[B_{kilt} + \sum_j P_{ij}^t f_{t+1}^{n-1}(l,j)] \quad \forall k,i;\ l \text{ feasible} \tag{7.4}$$

where,

n = total number of period remaining including the current period before reservoir operation terminates ;

t = index of time periods or stages (in this study t is in month);

i/j = index of characteristic inflow volume in time period t/t+1;

k/l = index of characteristic storages volumes at the beginning of time period t/t+1 ;

P_{ij}^t = probability that inflow Q_j occurs in time period t+1, given a known inflow of Q_i in period t;

B_{kilt} = system performance (squared deficit of release from demand) in the period t for initial storage volume corresponding to index k, inflow volume corresponding to index i, and final storage volume corresponding to index l.

The recursive equation 7.4 is solved subject to the following constraints as discussed in chapter 4.

Continuity constraint of the reservoir:

$$S_{t+1} = S_t + Q_t - E_t - R_t \tag{7.5}$$

Reservoir storage constraint:

$$S^{min} \leq S_{t+1} \leq S^{max} \tag{7.6}$$

Where,

S^{min} =45.64Mm^3 and S^{max} =312.64Mm^3.

Constraints of release from the reservoir:

$$R_t^{min} \leq R_t \leq R_t^{max} \tag{7.7}$$

Where,

$R_t^{min} = \max [0, (S_t + Q_t - E_t - S^{max})]$

$R_t^{max} = S_t + Q_t - E_t - S^{min}$

7.2.7 TERMINATION CRITERIA

As the recursive equations are solved for each period in successive years, the policy in each particular period t will repeat in each successive year after a few years. When this condition is satisfied, and when the expected annual performance $f_t^{n+T}(k,i) - f_t^{n}(k,i)$ is constant for all states k, i and for all periods t (total period in a year being *T*) within a year, the policy is said to reach a steady state. This condition is treated as the termination criteria.

7.3 FINAL PRESENTATION OF OPERATING RULE

The SDP model presented here provides a steady state operating policy. The operating policy designated by the SDP model is a set of rules specifying the final storage volume at the end of each month for each combination of storage volume at the beginning of the month and inflow during the month.

7.4 SDP MODEL APPLICATION

7.4.1 DISCRETIZATION OF STATE VARIABLES

7.4.1.1 DISCRETIZATION OF STORAGE

The Pagladia Reservoir is proposed to operate from a storage volume of 45.64 Mm^3 (MDDL) to 312.64Mm^3(FRL). Therefore, conservative storage of the reservoir is 267.0 Mm^3, which is 29.26% of the mean annual flow

(943.294Mm^3) to the reservoir. For small and medium reservoir (capacity 20-50 percent of mean annual flow), 20-30 characteristic storages are sufficient to generate a good operating rule (Karamouz and Houck, 1987). However in this study 20, 30 and 40 characteristic storages for the conservative storages of the reservoir has been considered in an attempt to find good operating policy for the Pagladia multipurpose reservoir. Three different discretization of the conservative storage have been used in the SDP model. Moran's (1954) scheme of storage discretization is used to find the characteristic storages. Table 7.1 shows number of characteristic storages and discretization steps.

Table 7.1 Number of characteristic storages and discretization steps

Number of characteristic storages	Discretization steps (Mm^3)
20	14.05
30	9.21
40	6.85

The SDP models with 20, 30 and 40 numbers of characteristic storages are called SDP1, SDP2, and SDP3 respectively in this study.

7.4.1.2 DISCRETIZATION OF INFLOW

In this study 100 years of synthetic monthly streamflow generated by artificial neural network based model has been used. All the 100 streamflow values of each 12 time periods were arranged individually in ascending order and for each period eight values (class boundaries) were selected such that almost equal numbers of values fall in each class interval, to have seven class intervals for the streamflow values of every months. The inflow boundaries for each of the

12 time periods are given in Table 7.2. The characteristic inflow values within each class boundary are presented in Table 7.3. The characteristic inflow values are the average values within an interval.

Table 7.2 Selected values of class boundary for different months

Month	Class boundary							
	1	2	3	4	5	6	7	8
January	1.2	8.04	11.35	13.13	15.7	18.44	22.88	34.89
February	1.07	5.91	8.49	9.91	11.99	14.4	17.71	30.34
March	1.04	4.39	7.09	9.92	11.71	13.98	16.77	28.65
April	1.02	5.58	9.96	13.11	15.67	20.53	24.16	35.69
May	2.34	23.61	43.99	60.97	73.75	91.25	110.52	179.47
June	10.67	20.38	87.99	132	158.72	226.03	305	648.06
July	6.02	71.92	126.58	171.99	201.74	239.84	314.44	520.94
August	10.53	68.56	127.63	166.6	194.58	241.67	306.17	454.69
September	8	48.85	86.11	109.58	137.43	166.28	205.43	370.16
October	1.25	22.08	45.1	57.09	72.18	87.15	104.87	179.79
November	2.1	13.42	19.31	22.35	25.68	32.29	36.74	82.46
December	1.25	10.53	15.04	17.93	20.21	22.61	28.84	56.87

Table 7.3 Characteristic Inflow of different Months

Month	Inflow Index						
	1	2	3	4	5	6	7
January	4.92	10	12.3	14.8	17	20.8	27
February	3.9	7	9.3	11.1	13.1	16.1	21.5
March	2.44	5.89	8.73	10.7	12.8	15.2	20.4
April	3.01	7.64	11.5	14.1	18.2	22.2	28.7
May	13.4	33.4	52.6	67.3	82.3	101	142
June	18.3	61.3	109	144	193	264	404
July	43.7	105	148	189	225	275	373
August	45.4	109	152	184	216	270	367
September	34.6	69.1	98.8	122	152	189	258
October	15	36.6	50.8	66	81.6	96.9	128
November	10.4	17.3	21	23.8	29.3	34.3	44.6
December	6.44	13.1	16.9	19.3	21.6	25.9	33.9

7.4.2 TRANSITION PROBABILITY MATRIX (TPM)

The Transition probability of inflow of a particular class interval (Table 7.2) in a particular time period is estimated by counting how many times the next period inflow interval (j) followed the current period inflow interval (i). This counting is done for all the next period inflow intervals (all j's) for a particular current period inflow interval (i) and all the frequencies are summed up to have the total number of occurrences. Then frequency of each interval is divided by the total number of occurrences of all the next period interval to get the probability that the next period flow is 'j' given that current period flow is 'i'. This procedure is repeated for all i's and for all time periods. All the twelve TPM are presented in table 7.4 (a) through 7.4(l).

Table 7.4(a). Transition probability matrix for January to February

February j / January i	1.07-5.91	5.91-8.49	8.49-9.91	9.91-11.99	11.99-14.4	14.4-17.71	17.71-30.34
1.2-8.04	0.118	0.176	0.118	0.0588	0.176	0.0588	0.294
8.04-11.35	0.0588	0.235	0	0.0588	0.294	0.294	0.0588
11.35-13.13	0.235	0.176	0.235	0.118	0.118	0	0.118
13.13-15.7	0.118	0.118	0.294	0.235	0.0588	0.0588	0.118
15.7-18.44	0.176	0.0588	0.0588	0.353	0.235	0.118	0
18.44-22.88	0.176	0	0.118	0.0588	0.0588	0.353	0.235
22.88-34.89	0.111	0.222	0.167	0.111	0.0556	0.111	0.222

Table 7.4(b). Transition probability matrix for February to March

March j / February i	1.04-4.39	4.39-7.09	7.09-9.92	9.92-11.71	11.71-13.98	13.98-16.77	16.77-28.65
1.07-5.91	0.235	0.412	0.118	0.118	0.0588	0	0.0588
5.91-8.49	0.235	0.235	0.0588	0.118	0.176	0.118	0.0588
8.49-9.91	0.118	0.118	0.294	0.118	0.176	0.118	0.0588
9.91-11.99	0.118	0.176	0.294	0.176	0.118	0.0588	0.0588
11.99-14.4	0.176	0.0588	0.0588	0.118	0.176	0.294	0.118
14.4-17.71	0.0588	0	0	0.176	0.235	0.176	0.353
17.71-30.34	0.0556	0	0.167	0.167	0.0556	0.222	0.333

Table 7.4(c). Transition probability matrix for March to April

April j / March i	1.02-5.58	5.58-9.96	9.96-13.11	13.11-15.67	15.67-20.53	20.53-24.16	24.16-35.69
1.04-4.39	0.353	0.176	0.235	0.0588	0.0588	0.118	0
4.39-7.09	0.235	0.294	0.235	0.0588	0.118	0.0588	0
7.09-9.92	0.235	0.176	0.118	0.176	0.176	0.118	0
9.92-11.71	0.0588	0.118	0.176	0.235	0.0588	0.235	0.118
11.71-13.98	0.0588	0.0588	0.176	0.176	0.235	0.118	0.176
13.98-16.77	0.0588	0.118	0.0588	0.176	0.176	0.235	0.176
16.77-28.65	0	0.0556	0	0.111	0.167	0.111	0.556

Table 7.4(d). Transition probability matrix for April to May

May j / April i	2.34-23.61	23.61-43.99	43.99-60.97	60.97-73.75	73.75-91.25	91.25-110.52	110.52-179.47
1.02-5.58	0.294	0.176	0.118	0.294	0.0588	0.0588	0
5.58-9.96	0.118	0.353	0.235	0.118	0.0588	0.118	0
9.96-13.11	0.235	0.0588	0.118	0.0588	0.235	0.118	0.176
13.11-15.67	0.176	0	0.118	0.235	0.176	0.118	0.176
15.67-20.53	0.0588	0.0588	0.176	0.176	0.0588	0.235	0.235
20.53-24.16	0.0588	0.0588	0.118	0	0.235	0.235	0.294
24.16-35.69	0.0556	0.278	0.111	0.111	0.167	0.111	0.167

Table 7.4(e). Transition probability matrix for May to June

June j / May i	10.67-20.38	20.38-87.99	87.99-132	132-158.72	158.72-226.03	226.03-305	305-648.06
2.34-23.61	0.235	0.176	0.294	0	0.0588	0.235	0
23.61-43.99	0.294	0.176	0.118	0.118	0.176	0.118	0
43.99-60.97	0.176	0.118	0.235	0.0588	0.118	0.176	0.118
60.97-73.75	0.235	0.0588	0.0588	0.235	0.176	0.176	0.0588
73.75-91.25	0.0588	0.0588	0.176	0.176	0.176	0.176	0.176
91.25-110.52	0.0588	0.176	0.118	0.176	0.118	0.0588	0.294
110.52-179.47	0.0556	0.111	0	0.222	0.167	0.0556	0.389

Table 7.4(f). Transition probability matrix for June to July

July j / June i	6.02-71.92	71.92-126.58	126.58-171.99	171.99-201.74	201.74-239.84	239.84-314.44	314.44-520.94
10.67-20.38	0.474	0.158	0.158	0.105	0.105	0	0
20.38-87.99	0.2	0.267	0.2	0.0667	0.2	0.0667	0
87.99-132	0.176	0.118	0.235	0.118	0.118	0.0588	0.176
132-158.72	0.0588	0.118	0.235	0.294	0.0588	0.118	0.118
158.72-226.03	0.0588	0.176	0.0588	0.176	0.118	0.235	0.176
226.03-305	0	0.118	0.118	0.0588	0.235	0.235	0.235
305-648.06	0	0.0556	0	0.167	0.167	0.278	0.333

Table 7.4(g). Transition probability matrix for July to August

August j / July i	10.53-68.56	68.56-127.63	127.63-166.6	166.6-194.58	194.58-241.67	241.67-306.17	306.17-454.69
6.02-71.92	0.176	0.235	0.118	0.118	0.118	0.118	0.118
71.92-126.58	0.235	0.118	0.235	0.0588	0.176	0.118	0.0588
126.58-171.99	0.118	0.294	0.0588	0.118	0.118	0.118	0.176
171.99-201.74	0.0588	0.176	0.235	0.176	0.176	0.118	0.0588
201.74-239.84	0.176	0.118	0.0588	0.235	0.0588	0.176	0.176
239.84-314.44	0.118	0	0.118	0.235	0.118	0.176	0.235
314.44-520.94	0.0556	0.111	0.167	0.0556	0.222	0.167	0.222

Table 7.4(h). Transition probability matrix for August to September

September j / August i	8-48.85	48.85-86.11	86.11-109.58	109.58-137.43	137.43-166.28	166.28-205.43	205.43-370.16
10.53-68.56	0.294	0.176	0.118	0.235	0.0588	0.0588	0.0588
68.56-127.63	0.0588	0.294	0.294	0.235	0	0.118	0
127.63-166.6	0.176	0.118	0.118	0.235	0.176	0.118	0.0588
166.6-194.58	0.235	0.235	0.0588	0.0588	0.118	0.118	0.176
194.58-241.67	0.118	0.0588	0.235	0.118	0.294	0.118	0.0588
241.67-306.17	0.0588	0.118	0.0588	0.118	0.118	0.176	0.353
306.17-454.69	0.0556	0	0.111	0	0.222	0.278	0.333

Table 7.4(i). Transition probability matrix for September to October

October j / September i	1.25-22.08	22.08-45.1	45.1-57.09	57.09-72.18	72.18-87.15	87.15-104.87	104.87-179.79
8-48.85	0.353	0.176	0.118	0.235	0	0	0.118
48.85-86.11	0.0588	0.118	0.294	0.176	0.235	0.0588	0.0588
86.11-109.58	0.176	0.353	0.0588	0.0588	0.235	0.118	0
109.58-137.43	0.118	0.118	0.294	0	0.118	0.0588	0.294
137.43-166.28	0.118	0.118	0.0588	0.176	0.176	0.176	0.176
166.28-205.43	0.0588	0.0588	0.176	0.235	0.176	0.176	0.118
205.43-370.16	0.111	0.0556	0	0.111	0.0556	0.389	0.278

Table 7.4(j) Transition probability matrix for October to November

November j / October i	2.1-13.42	13.42-19.31	19.31-22.35	22.35-25.68	25.68-32.29	32.29-36.74	36.74-82.46
1.25-22.08	0.176	0.235	0.176	0.176	0.176	0.0588	0
22.08-45.1	0.294	0.235	0.176	0.0588	0.0588	0.118	0.0588
45.1-57.09	0.294	0.118	0.294	0.0588	0.118	0.118	0
57.09-72.18	0.0588	0.118	0.0588	0.235	0.235	0.118	0.176
72.18-87.15	0.0588	0.0588	0.118	0.176	0.0588	0.294	0.235
87.15-104.87	0.118	0.176	0.118	0.176	0.0588	0.0588	0.294
104.87-179.79	0	0.0556	0.0556	0.111	0.278	0.222	0.278

Table 7.4(k). Transition probability matrix for November to December

December j / November i	1.25-10.53	10.53-15.04	15.04-17.93	17.93-20.21	20.21-22.61	22.61-28.84	28.84-56.87
2.1-13.42	0.353	0.235	0.294	0	0.0588	0.0588	0
13.42-19.31	0.294	0.118	0.176	0.294	0.0588	0.0588	0
19.31-22.35	0.0588	0.294	0.118	0.176	0.118	0.118	0.118
22.35-25.68	0.176	0.176	0.235	0.176	0.176	0.0588	0
25.68-32.29	0.118	0.176	0	0.412	0.0588	0.118	0.118
32.29-36.74	0	0	0.176	0	0.294	0.235	0.294
36.74-82.46	0	0	0	0	0.167	0.333	0.5

Table 7.4(l). Transition probability matrix for December to January

January j / December i	1.2-8.04	8.04-11.35	11.35-13.13	13.13-15.7	15.7-18.44	18.44-22.88	22.88-34.89
1.25-10.53	0.294	0.118	0.118	0.176	0.235	0	0.0588
10.53-15.04	0.235	0.176	0.235	0.176	0.176	0	0
15.04-17.93	0.235	0.235	0.176	0.176	0.118	0	0.0588
17.93-20.21	0.0556	0.0556	0.167	0.111	0.111	0.333	0.167
20.21-22.61	0.125	0.313	0.188	0.125	0.125	0	0.125
22.61-28.84	0	0	0.118	0	0.176	0.412	0.294
28.84-56.87	0.0556	0.111	0	0.222	0.0556	0.222	0.333

7.5 PERFORMANCE EVALUTATION OF SDP OPERATING POLICIES

The performance evaluation of operating policies developed by SDP models has been carried out through simulation of the reservoir for 228 months of historic streamflow series (1977 –1996). The starting month of the simulation was October 1977 with initial storage of the reservoir as 45.64Mm^3 (dead storage volume). The final month of the simulation period was September 1996.

7.5.1 CRITERIA FOR RESERVOIR OPERATION PERFORMANCE

The performance criteria as mentioned in section 6.4.1 of chapter 6, have been used for the evaluation of operating policies developed by three SDP models.

7.5.2 SIMULATION AND RESULTS

Table 7.5 shows the simulation results of the reservoir for 228 months of historic streamflow (1977-1996).

Table 7.5 Simulation results of SDP models for 228 months of historic streamflow data (1977-1996)

Model	Total squared deficit (10^5)	Total irrigation deficit (Mm^3)	Number of irrigation deficit Month	Total power generation (10^8 kWh)	Number of power deficit month	Total water deficit (Mm^3)	Number of times the reservoir is full	Total spill (Mm^3)
SDP1	6.48	272.17	24	3.44	184	7917.79	113	3272.82
SDP2	5.82	261.9	21	3.41	185	7801.65	76	3233.23
SDP3	6.07	217.59	26	3.39	184	7981.99	84	3696.0

Simulation results have shown that total squared deficit is minimum in SDP2 model i.e. the model based on 30 characteristic storages. SDP1 model, i.e. the model based on 20 characteristic storages has been found to give total squared deficit more than those of the SDP2 and SDP3 models. While total irrigation deficit has been found minimum in SDP3 model, the number of irrigation deficit month has been found minimum in SDP2 model. SDP1 model produces more power than SDP2 and SDP3 models. SDP2 model comes second in terms of power generation. Number of deficit months for power is equal for SDP1 and SDP3 model. It is important to mention that when there is deficit for irrigation there is also deficit for power generation, which is not shown explicitly in the Table 7.5. Total water deficit has been found minimum in SDP2 model. SDP1 model keeps the reservoir full for maximum number of times than the other two models. SDP2 spills a lesser quantity of water than SDP1 and SDP3 models. Considering overall performance, SDP2 model has slight edge over the other two SDP models.

7.6 CONCLUSION

The operating policy for the Pagladia multipurpose reservoir has been developed in this chapter using SDP model. Three different SDP models, namely SDP1, SDP2 and SDP3 have been developed using 20, 30 and 40 characteristic storages of the reservoir. Seven characteristic values of inflow have been considered for each month to develop the operating policy by all the three SDP models. From the simulation result it has been observed that although squared deficit is minimum in the SDP2 model other SDP models also show better performance in respect of some of the criteria considered in this study for performance evaluation. However, considering minimum total squared deficit, minimum number of irrigation deficit month, and lesser spill in SDP2 model, the operating policy developed by SDP2 model can be preferred. Minimum squared deficit in SDP2 model indicates that deficit is distributed more uniformly among the deficit months. This is advantageous from the management point of view. The operating policy derived by SDP2 model has been presented in Fig. 7.1a through Fig.7.12g for all months January through December and for all the seven inflow indices. Stochastic dynamic programming used in this chapter is a conventional optimization technique. In the next chapter a non-conventional optimization technique, namely, genetic algorithm is used to develop the optimal operating policy for the Pagladia multipurpose reservoir.

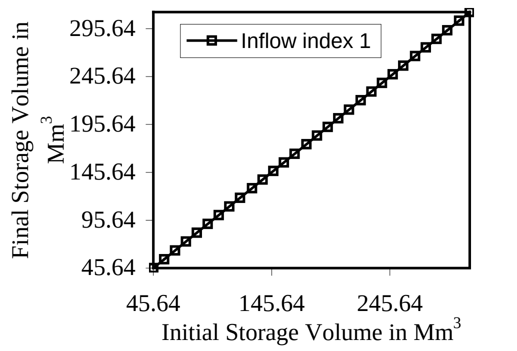


Fig. 7.1a: Optimal operating policy for January

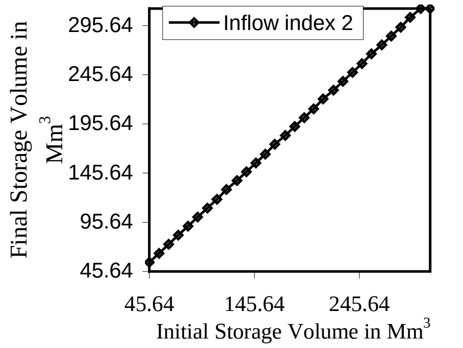


Fig 7.1b: Optimal operating policy for January

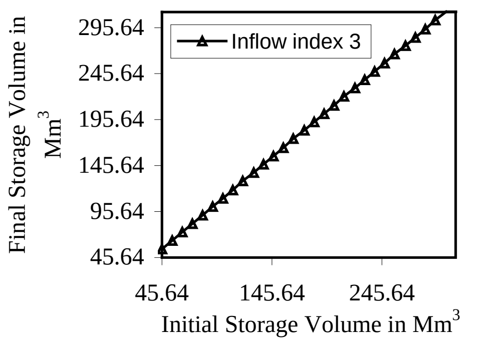


Fig.7.1c: Optimal operating policy for January

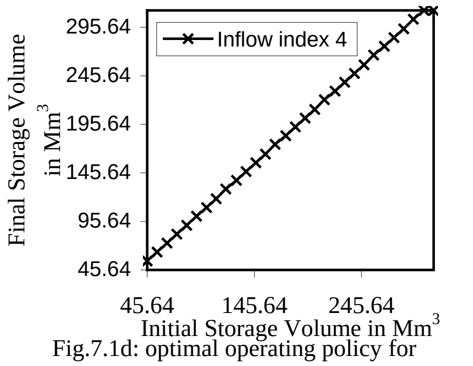


Fig.7.1d: optimal operating policy for January

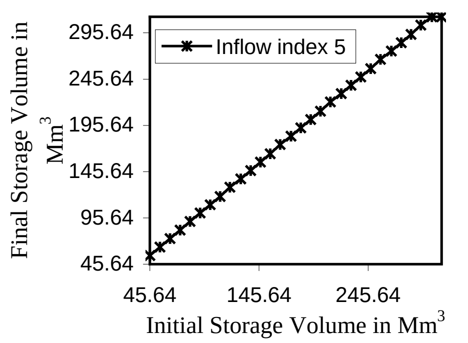


Fig.7.1e: Optimal operating policy for January

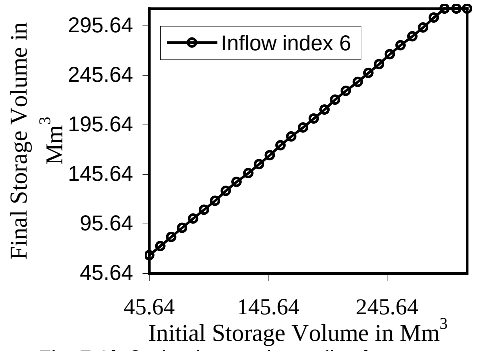


Fig. 7.1f: Optimal operating policy for January

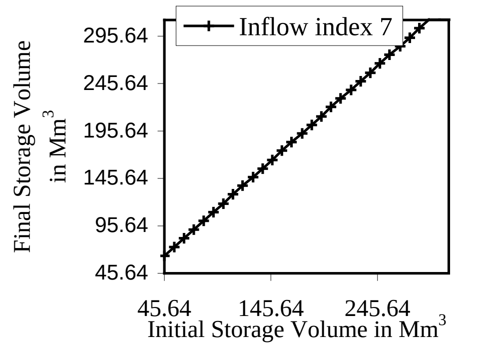


Fig 7.1g: Optimal operating policy for January

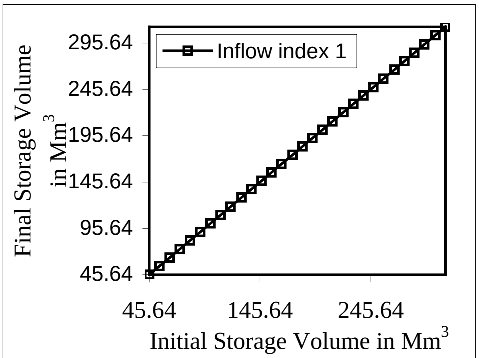


Fig. 7.2a: Optimal operating policy for February

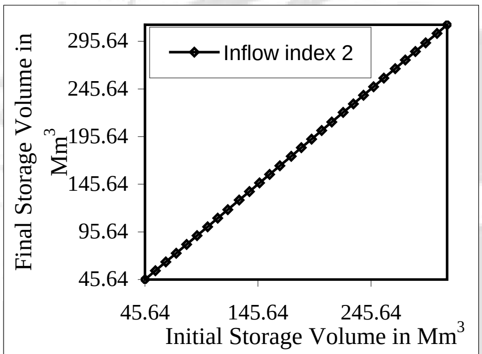


Fig. 7.2b: Optimal operating policy for February

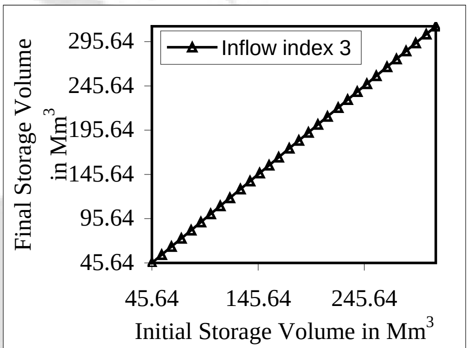


Fig. 7.2c: Optimal operating policy for February

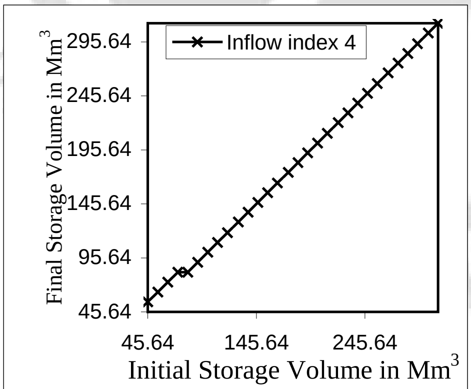


Fig. 7.2d: Optimal operating policy for February

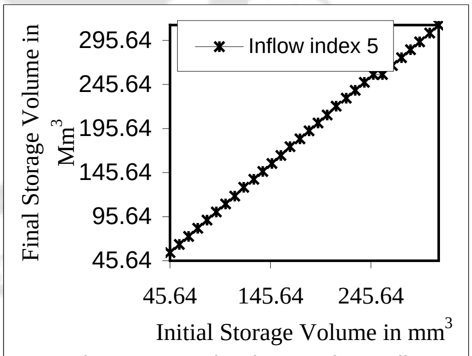


Fig.7.2 e: Optimal operating policy for February

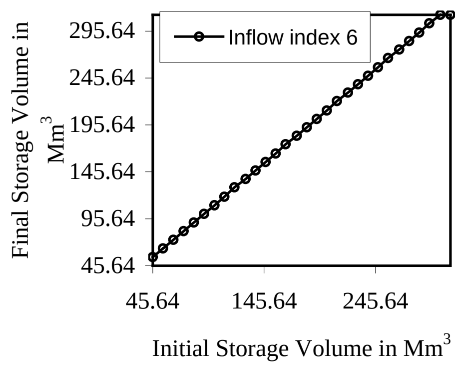


Fig. 7.2f: Optimal operating policy for February

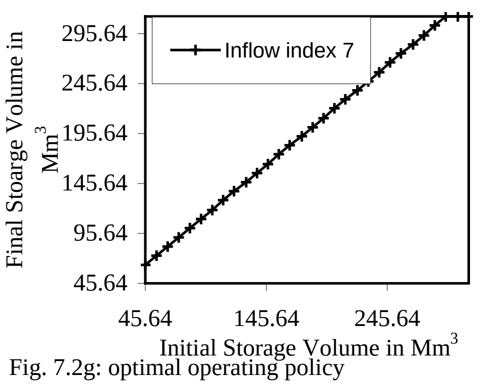


Fig. 7.2g: optimal operating policy forFebruary

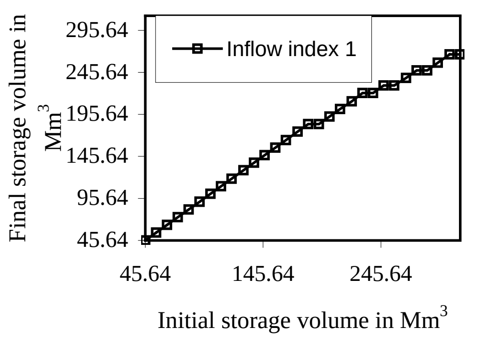


Fig. 7.3a: Optimal operating policy for March

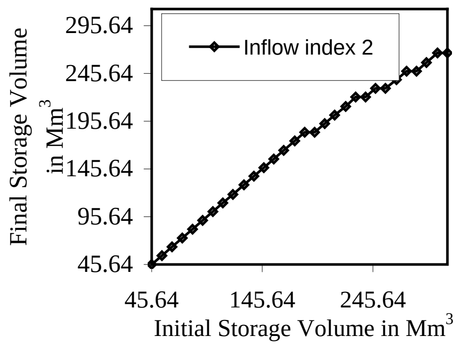


Fig. 7.3b: Optimal operating policy for March

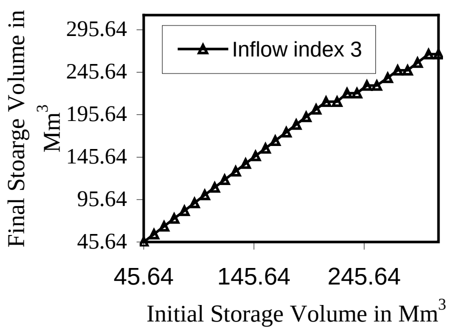


Fig. 7.3c: Optimal operating policy for March

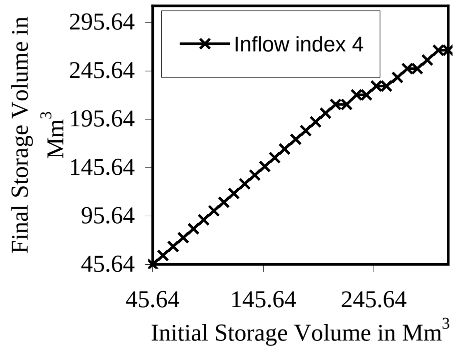


Fig7.3d: Optimal operating policy for March

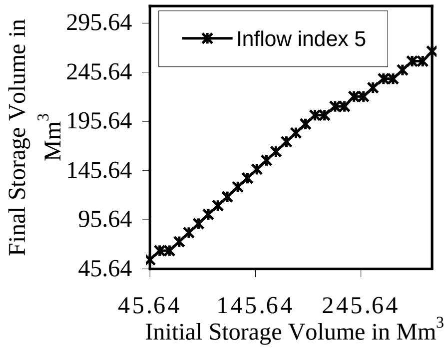


Fig.7.3e: Optimal operating policy for March

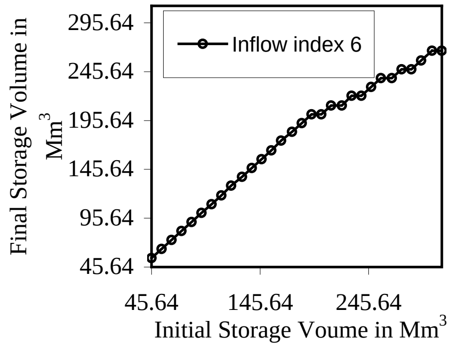


Fig. 7.3f: Optimal operating policy for March

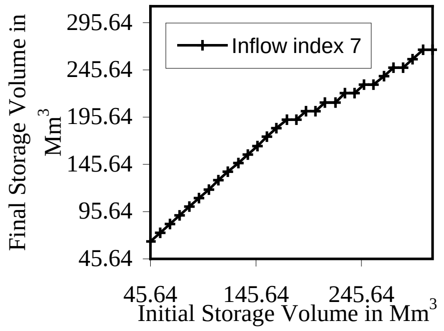


Fig.7.3g: Optimal operating policy for March

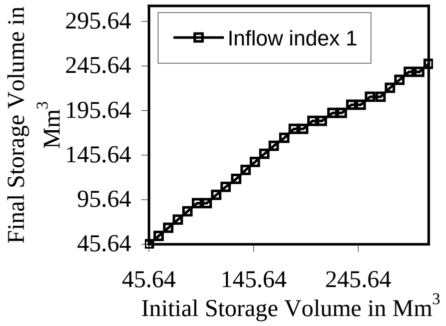


Fig. 7.4a: Optimal operating policy for April

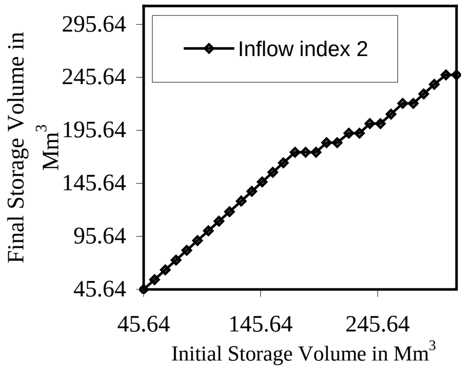


Fig. 7.4b: Optimal operating policy for April

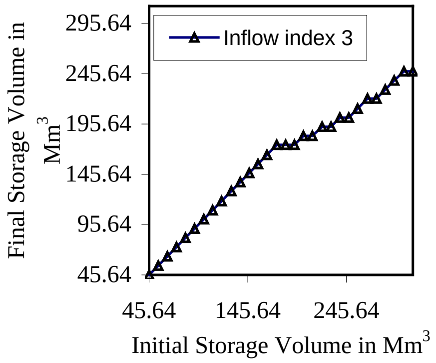


Fig.7.4c: Optimal operating policy for April

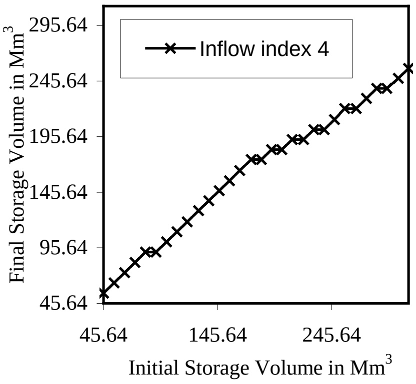


Fig. 7.4d: Optimal operating policy for April

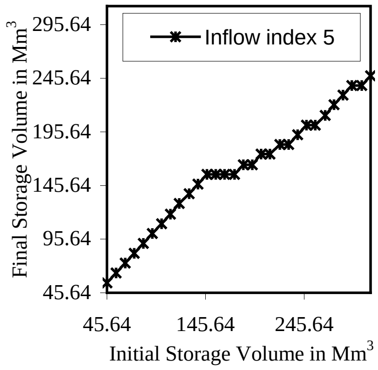


Fig.7.4e: Optimal operating policy for April

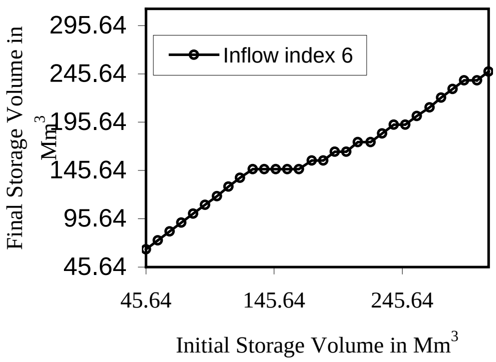


Fig. 7.4f: Optimal operating policy for April

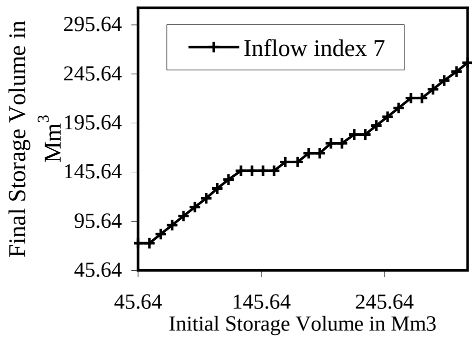


Fig.7.4 g: Optimal operating policy for April

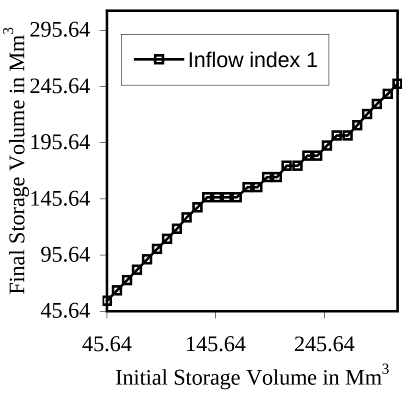


Fig. 7.5a: Optimal operating policy for May

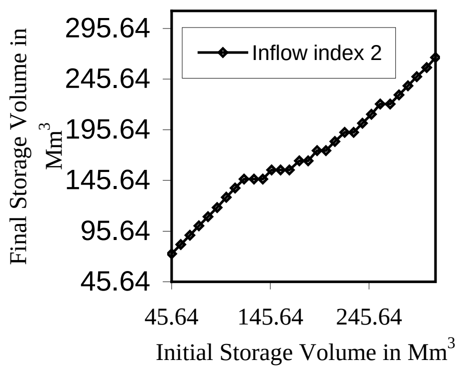


Fig. 7.5b: Optimal operating policy for May

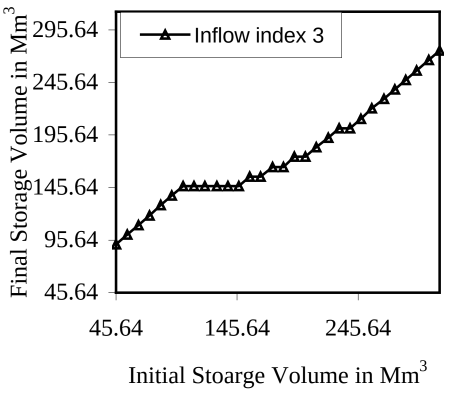


Fig.7.5c: Optimal operating policy for May

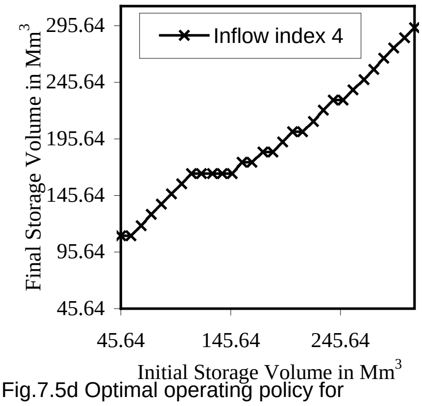


Fig.7.5d Optimal operating policy for May

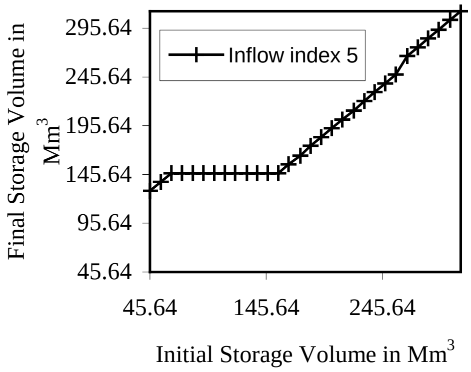


Fig.7.5e: Optimal operating policy for May

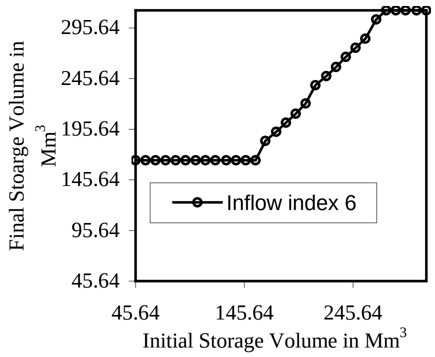


Fig. 7.5f: Optimal operating policy for May

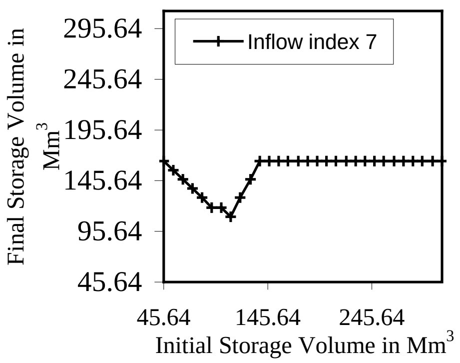


Fig.7.5g: Optimal operating policy forMay

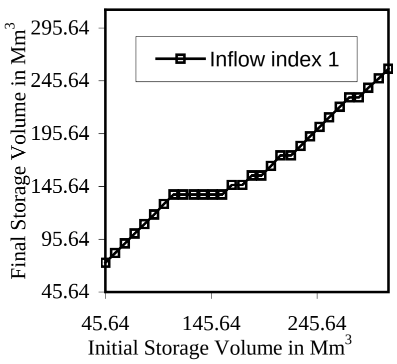


Fig. 7.6a: Optimal operating policy for June

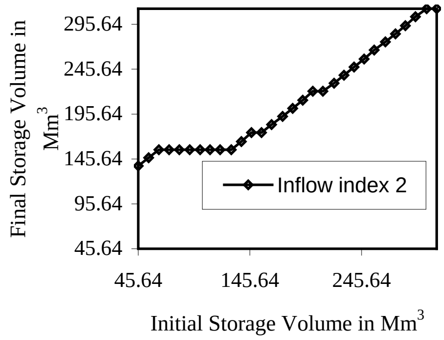


Fig.7.6b: Optimal operating policy for June

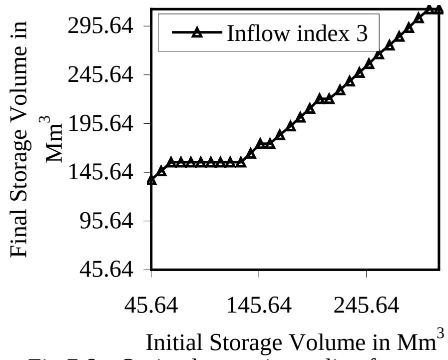


Fig.7.6c: Optimal operating policy for June

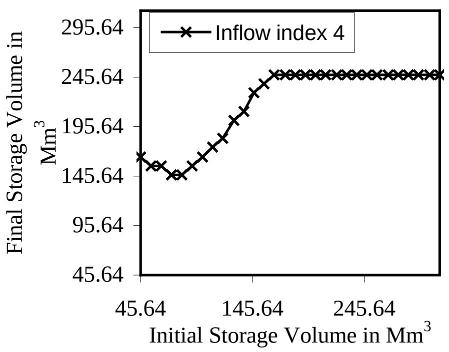


Fig. 7.6d: Optimal operating policy for June

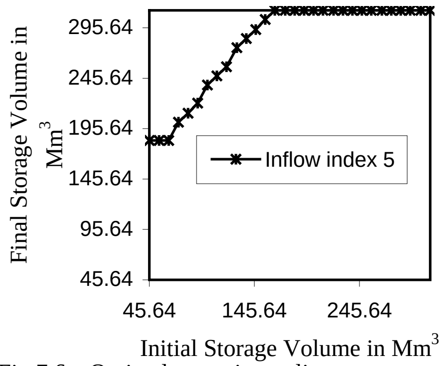


Fig.7.6e: Optimal operating policy for June

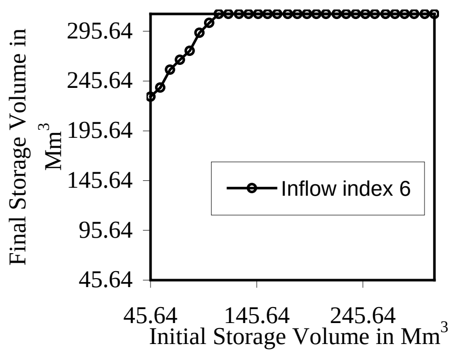


Fig.7.6f: Optimal operating policy for June

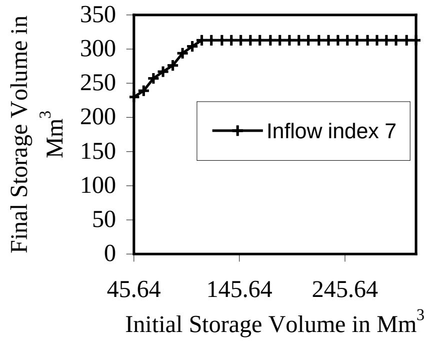


Fig. 7.6.g: Optimal operating policy for June

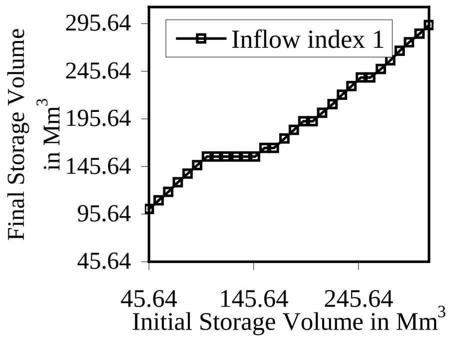


Fig.7.7a: Optimal operating policy for July

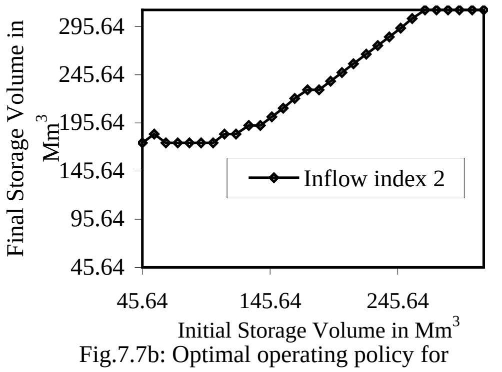


Fig.7.7b: Optimal operating policy for July

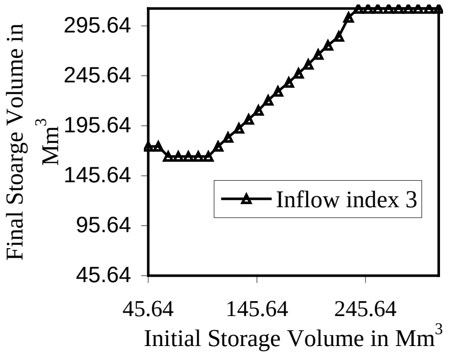


Fig.7.7c: Optimal operating policy for July

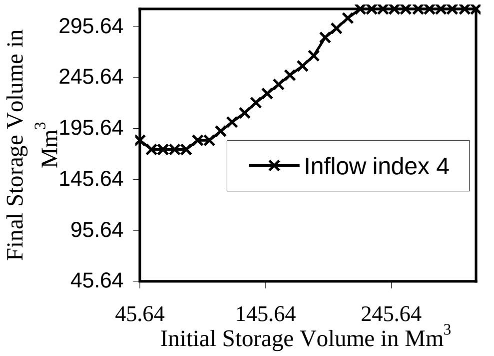


Fig.7.7d: Optimal operating policy for July

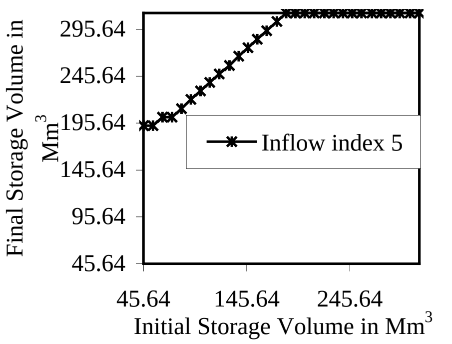


Fig.7.7e: Optimal operating policy for July

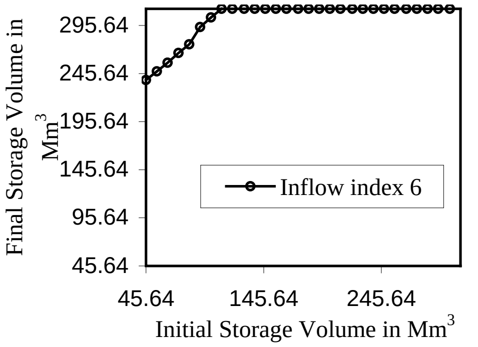


Fig.7.7f: Optimal operating policy for July

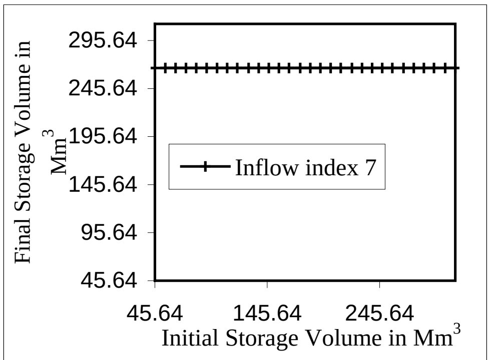


Fig.7.7g: Optimal operating policy for July

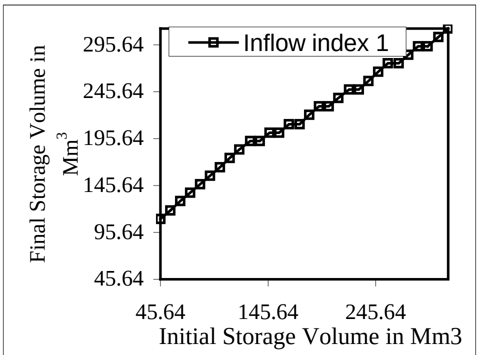


Fig.7.8a: Optimal operating policy for August

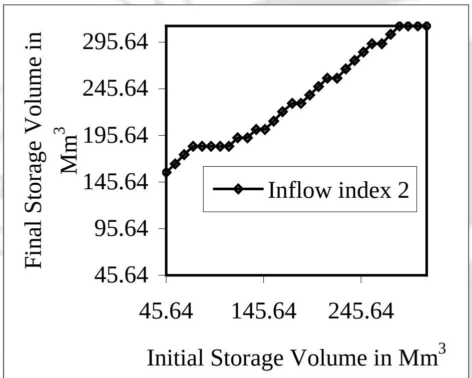


Fig. 7.8b: Optimal operating policy for August

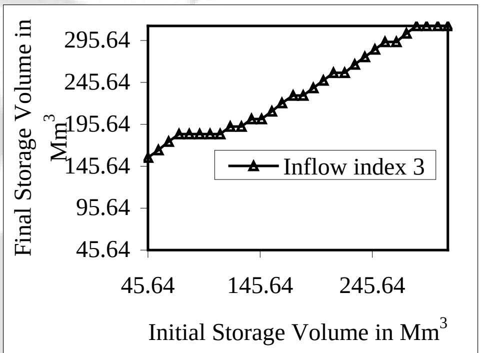


Fig.7.8c: Optimal operating policy for August

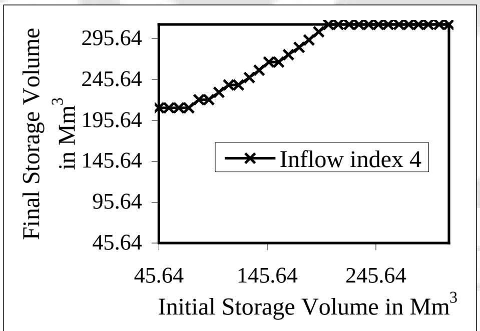


Fig.7.8d: Optimal operating policy for August

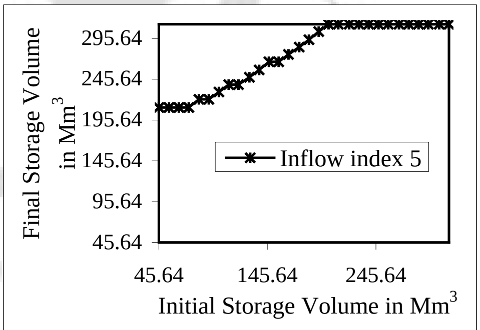


Fig.7.8e: Optimal operating policy for August

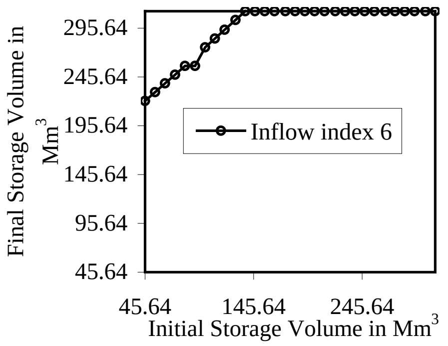


Fig.7.8f: Optimal operating policy for August

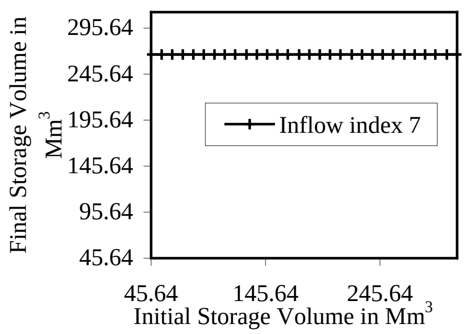


Fig.7.8g: Optimal operating policy for August

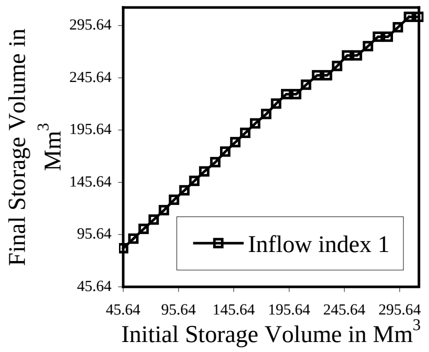


Fig.7.9a: Optimal operating policy for September

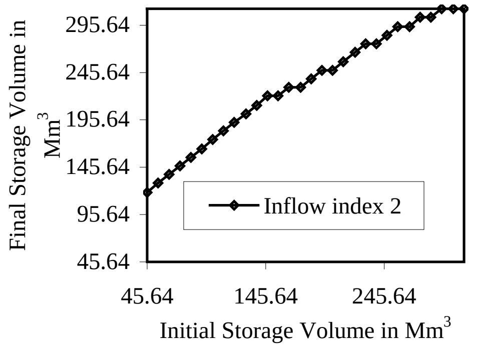


Fig. 7.9b: Optimal operating policy for September

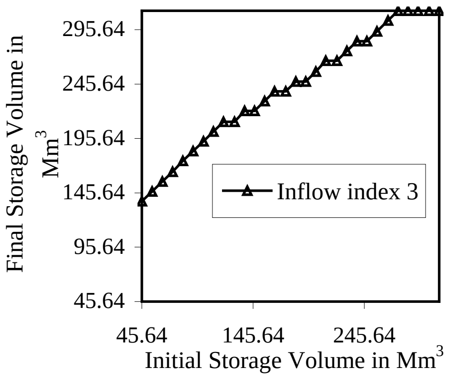


Fig.7.9c: Optimal operating policy for September

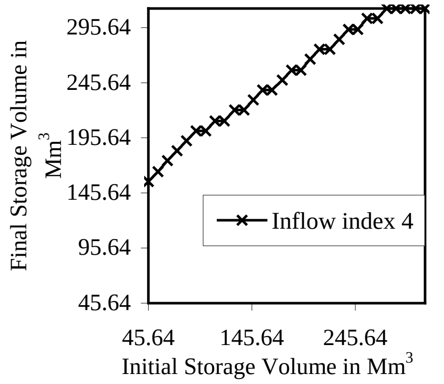


Fig.7.9d: Optimal operating policy for September

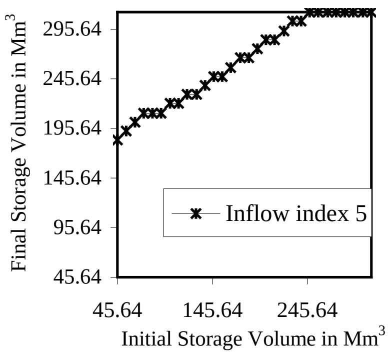


Fig.7.9e: Optimal operating policy for September

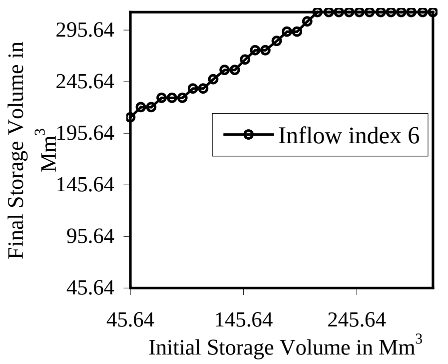


Fig.7.9f: Optimal operating policy for September

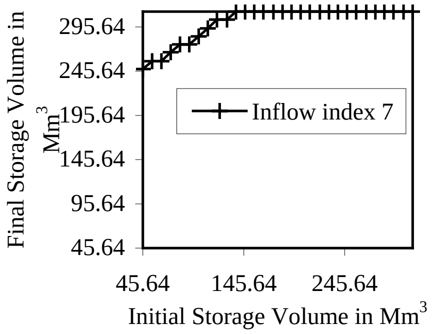


Fig.7.9g: Optimal operating policy for September

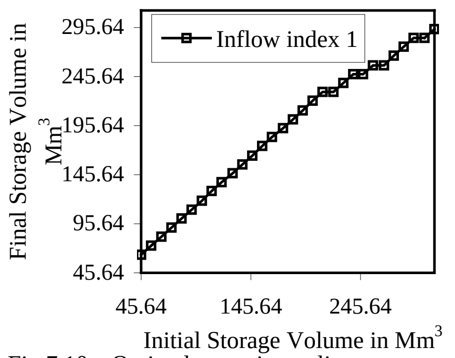


Fig.7.10a: Optimal operating policy for October

…..

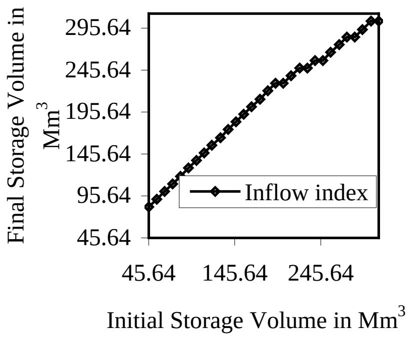


Fig.7.10b: Optimal operating policy for October

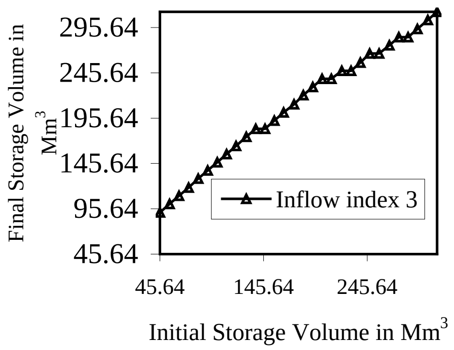


Fig.7.10c: Optimal operating policy for October

……

….

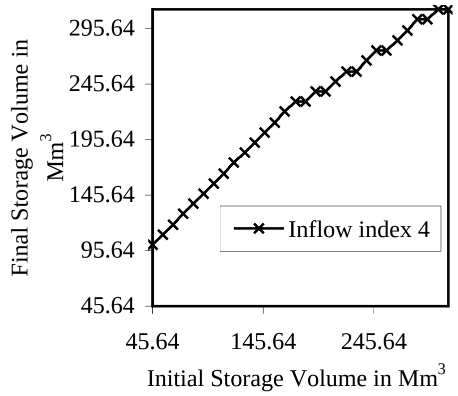


Fig.7.10d: Optimal operating policy for October

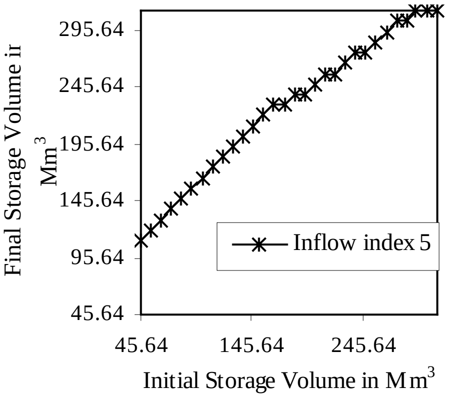


Fig.7.10e: Optimal operating policy October

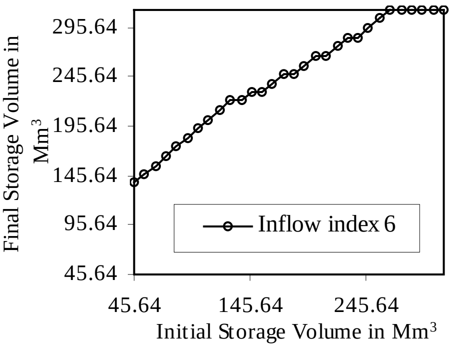


Fig.7.10f: Optimal operating policy for october

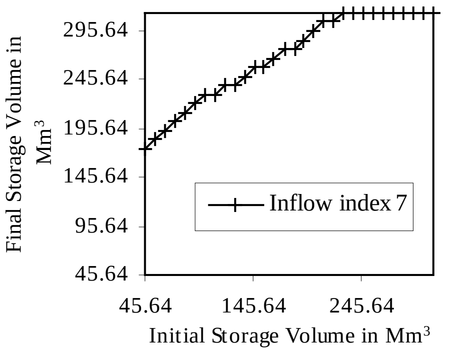


Fig.7.10g: Optimal operating policy for October

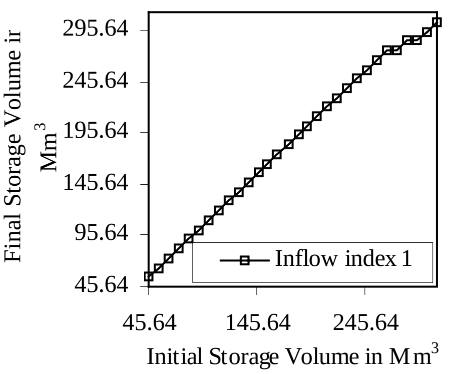


Fig.7.11a: Optimal operating policy November

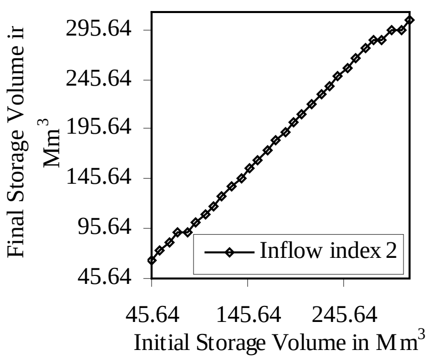


Fig.7.11b. Optimal operating policy for November

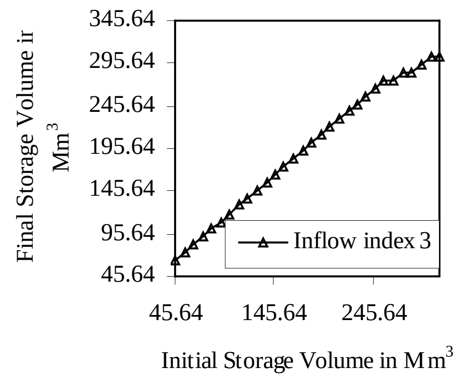


Fig.7.11c: Optimal operating policy for November

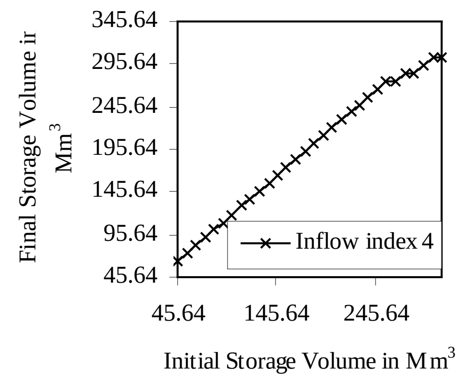


Fig.7.11d: Optimal operating policy for November

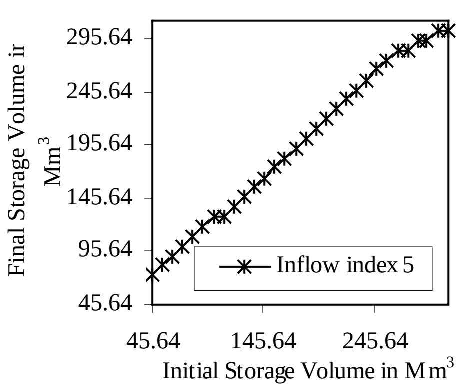


Fig.7.11e: Optimal operating policy for November

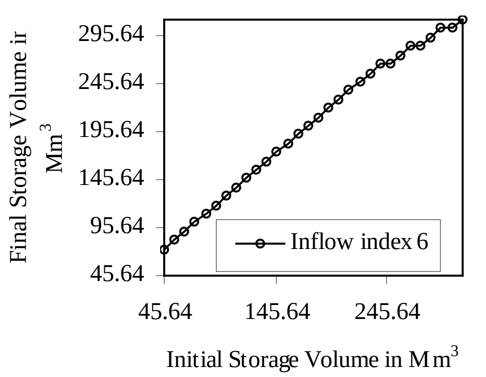


Fig.7.11f: Optimal operating policy for November

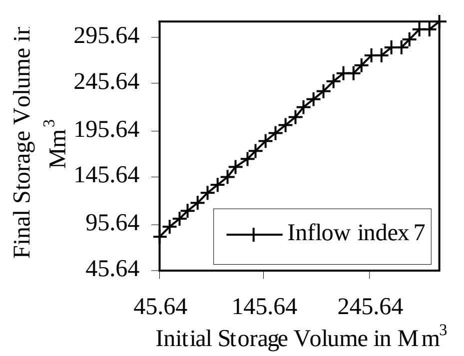


Fig.7.11g: Optimal operating policy November

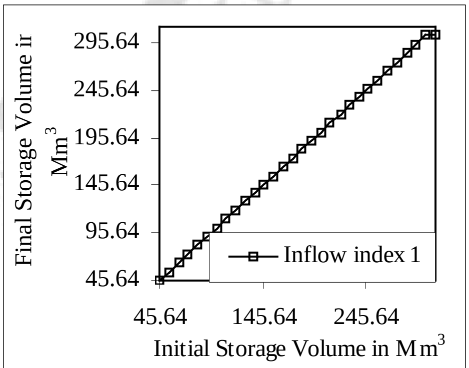


Fig. 7.12a: Optimal operating policy for December

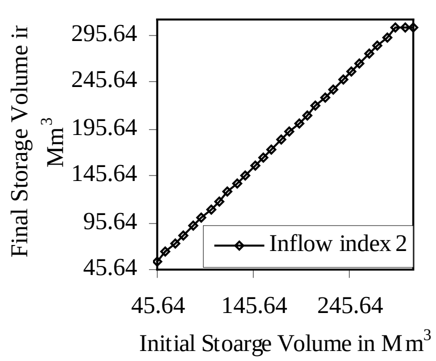


Fig.7.12b: Optimal operating policy for December

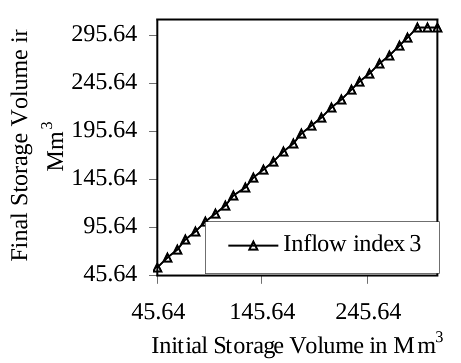


Fig.7.12c: Optimal operating policy for December

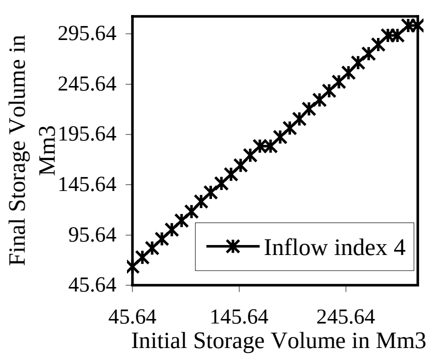


Fig.7.12d: Optimal operating policy for December

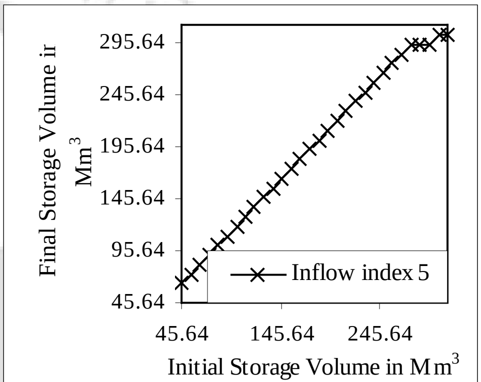


Fig.7.12e: Optimal operating policy for December

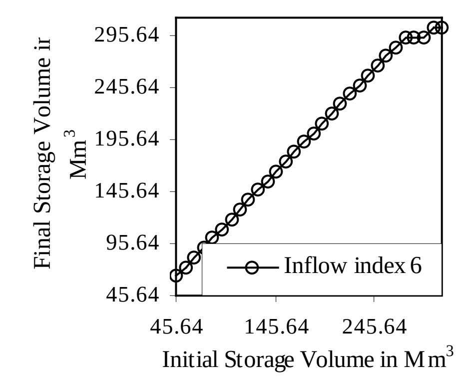


Fig.7.12f: Optimal operating policy for December

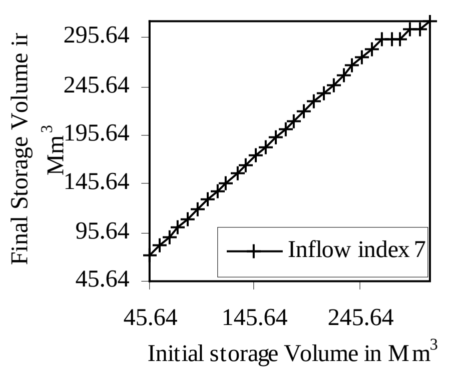


Fig.7.12g: Optimal operating policy for December

CHAPTER 8

GENETIC ALGORITHM MODEL FOR RESERVOIR OPERATION

8.1 INTRODUCTION

In spite of extensive research in reservoir optimization, researchers are still in search of some optimization techniques, which can derive more efficient reservoir operating policy for better management of reservoir. Genetic algorithm (GA) is a robust optimization technique and in this chapter an attempt has been made to derive optimal operating policy for the Pagladia multipurpose reservoir using GA. The synthetic monthly streamflow data of 100 years generated by the ANN based model have been used in the GA model to derive the operating policy.

8.2 GENETIC ALGORITHMS (GAs)

8.2.1 INTRODUCTION

Genetic algorithms (GAs), which derives its concept from Darwin's Theory of survival of the fittest, was first envisaged in 1975 by John Holland (Holland, 1992). Genetic algorithms are search algorithms based on the mechanics of natural genetic and natural selection. Genetic algorithms are heuristic techniques for searching over the solution space of a given problem in an attempt to find the best solution or set of solutions (Forrest, 1993). The basic elements of natural genetics-reproduction, crossover, and mutation- are used in the genetic search procedure.

Goldberg (1989) identifies the following as fundamental differences between GAs and traditional optimization methods:

1. GAs work with a coding of the parameter set, not the parameter themselves.
2. GAs search from a population of points, not a single point.
3. GAs use payoff (objective function) information, not derivatives or other auxiliary knowledge.
4. GAs use probabilistic transition rule, not deterministic rules.

A genetic algorithm can be considered to consist of the following steps (Burn and Yulianti 2001):

1. Select an initial population of strings.
2. Evaluate the fitness of each string.
3. Select strings from the current population to mate.
4. Perform crossover (mating) for the selected strings.
5. Perform mutation for selected string elements.
6. Repeat steps 2-5 for the required number of generations.

A flowchart for GAs is given in Fig. 8.1.

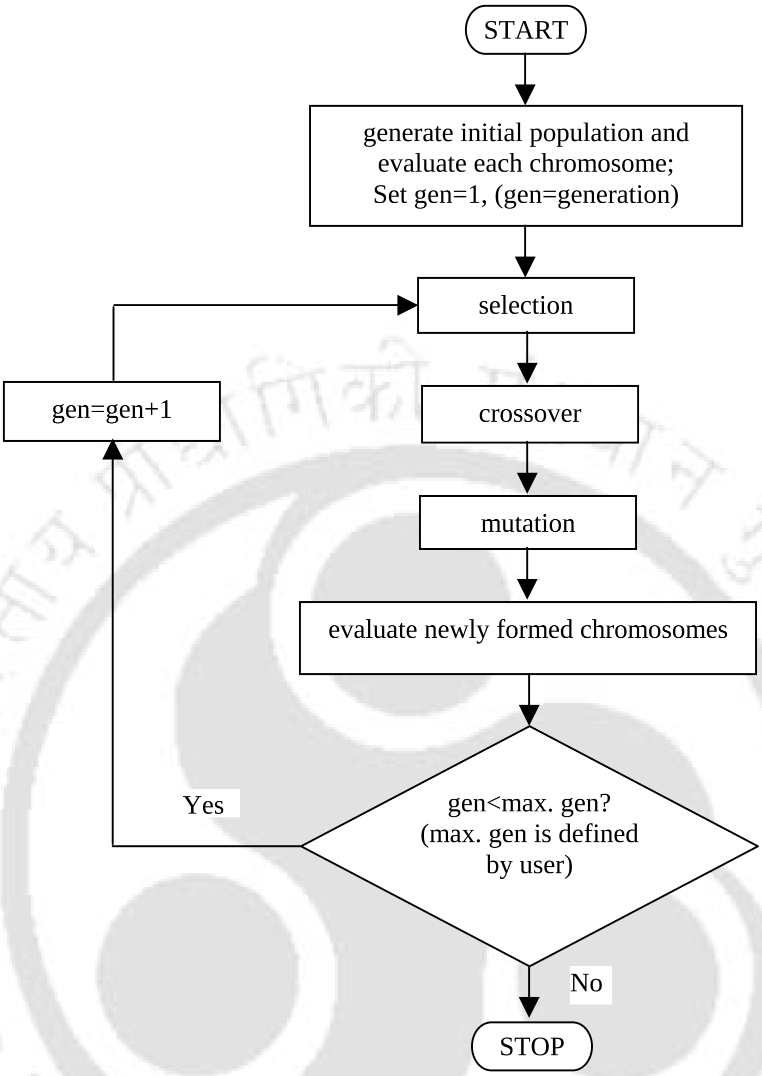


Fig. 8.1. Flowchart for GA

8.2.2 WORKING PRINCIPLE

The primary concept behind the use of GAs is the representation of solutions to a problem in an encoded format. These encoded parameters (alleles) are referred to as genes. The genes are joined to build strings, which represent a potential solution to the problem. These strings of variables are called chromosomes. The random interaction of the genes in population under different GA operators constitutes the GA technique.

8.2.3 FORMULATION OF GAs

The first and foremost importance of a GA formulation is to represent the decision or design variables of a problem in a string (also referred to as chromosome). Each chromosome is a potential solution and is comprised of a series of sub-strings or genes, representing components or variables that either form or can be used to evaluate the objective function of the problem. Each gene can be represented by a binary string mapped to the range of permissible value of the variable or by the real value of the variable. The fitness of a chromosome as a candidate solution to a problem is an expression of the value of the objective function represented by it. The fitness is also a function of the problem constraints and may be modified through the use of penalties when constraints are not satisfied.

8.2.4 REPRESENTATION SCHEMES

The representation schemes normally used in GAs for parameter coding are:

1. Binary coding
2. Gray coding
3. Real coding

In binary coding, a chromosome is represented by a string of binary bits that encode integers, real numbers, or anything else appropriate to a problem. Standard binary coding of variables permits large jumps in variable values between generations, which can lead to difficulty in converging to a good solution. To overcome this limitation Goldberg (1989) proposed the concept of Gray coding in which the binary representation of two adjacent variable values changes by only one binary digit. Gray coded representation has the property that any two points next to each other in the problem space differ by one bit only. In other words, an increase of one step in the parameter value corresponds to a change of a single bit only. In binary and gray coding discretization of the decision variable space is required.

An alternative technique of formulating GA is to use real value coding to represent the chromosome. Real valued chromosomes have been used with success by different investigators (,e.g., Oliveira and Loucks 1997, Wardlaw and Sharif 1999).

Real-value coding appears to offer several advantages over both binary and gray coding (Wardlaw and Sharif 1999).

1. A higher maximum fitness value is achieved.
2. A smoother convergence to a solution is achieved.
3. The average fitness of the population generated is much higher.

In real value representation, individual genes of a chromosome are initially allocated values randomly within the feasible limits of the variable represented. With a sufficiently large population of chromosomes, adequate representation can be achieved. There is a significant advantage in not wasting computer time on decoding for objective function evaluation, although a more careful approach to mutation is required. In real-value coding there is no discretization of the decision variable space. This is another advantage of this approach. Considering these, real coded GA has been used in this study.

8.2.5 FITNESS FUNCTION

GAs mimic the survival-of-the-fittest principle of nature to make a search process. Therefore, GAs are naturally suitable for solving maximization problems. Minimization problems are usually transformed into maximization problems by some suitable transformation. The fitness function value of a string is known as the string's fitness.

The operation of GAs begins with a population of random strings representing design or decision variables. Thereafter, each string is evaluated to find the fitness value. The population is then operated by three main operators- reproduction, crossover, and mutation- to create new population of points. The new population is further evaluated and tested for termination. If the termination criteria is not met, the population is iteratively operated by the above three operators and evaluated. The procedure is continued until the termination criterion is met. One cycle of these operations and the subsequent evaluation procedure is known as a generation in GA's terminology. The operators are described next.

8.2.6 GA OPERATORS

GA basically simulates the natural process of reproduction to produce successive fitter chromosomes. Three genetic operators used in the reproductive process are:

(1) Selection

(2) Crossover

(3) Mutation

Selection is a systematic process, which ensure that chromosomes in the population with high fitness values have a higher probability of being selected for combination with other chromosomes of high fitness. Combination is achieved through the crossover of pieces of genetic material between selected chromosomes. Mutation allows for the random mutations of bits of information in individual genes. Through successive generations, fitness should progressively improve.

8.2.6.1 SELECTION APPROACHES

The procedure by which chromosomes are chosen for participation in the reproduction process is called selection. Following selection procedures are generally used.

1. Roulette wheel selection,
2. Stochastic remainder roulette wheel selection,
3. Tournament selection.

Roulette wheel selection is a fitness proportionate selection in which each member of the population is allocated one slot on the wheel. The width of each member's slot varies in direct proportion to its fitness. Assuming the population size is n, the wheel is spun n times creating n/2 couples to mate to provide for the next generation.

A more stable version of roulette wheel selection operator is sometimes used. After the expected count for each individual string is calculated, the strings are first assigned copies exactly equal to the mantissa of the expected count. Thereafter, the regular roulette-wheel selection is implemented using the decimal part of the expected count as the probability selection. This selection method is less noisy and is known as the stochastic remainder roulette wheel selection.

Tournament selection is another selection procedure used widely in GA.

Tournament selection operator satisfies the following three criteria:

(1) Any feasible solution is preferred to any infeasible solution,

(2) Between two feasible solutions, the one having better objective function value is preferred,

(3) Between two infeasible solutions, one having smallest constraint violation is preferred.

Goldberg and Deb (1990) compared various selection schemes and indicated a preference for the tournament selection scheme. In this study tournament selection scheme has therefore been used. In tournament selection a group of individuals are chosen at random from the population, and the individual with the best fitness is selected for inclusion in the next generation. The procedure is repeated until the appropriate numbers of individuals are selected for the new generation.

Selection alone cannot introduce any new individuals into the population, ,i.e., it cannot find new points in the search space. These are generated by genetically inspired operators, of which the most well known are crossover and mutation.

8.2.6.2 CROSSOVER APPROACHES

Exchange of important building blocks between two strings, known as parent string, is termed as cross over in GA. GA attempts to create new strings that preserve the best material from the two parent strings. The number of strings in which material is exchanged is controlled by the crossover probability forming part of the parametric data. This operator tends to enable the evolutionary process to move toward promising regions of the search space. Goldberg (1989) and Michalewicz (1992) describe the following methods of crossover:

1. One-point crossover.

2. Two-point crossover.
3. Uniform crossover.

Crossover occurs between two selected chromosomes with some specified probability, usually in the range of 0.5-1.00. Fig. 8.2 shows the crossover approaches for one point, two point and uniform crossover. In one-point crossover, a crossover point is selected at random at some point c in the chromosome length L. Swapping all genes between positions c and L creates two new individuals. In two-point crossover, genetic material between two positions chosen at random along the length of the chromosomes is exchanged. Uniform crossover operates on individual genes of the selected chromosomes, rather than blocks of genetic material, and each gene is considered in turn for crossover or exchange.

An important aspect of crossover in application to multivariate problem in binary coding is that crossover should occur only at gene boundaries, because each gene consists of alleles, or bits, and crossover may split the genes. This is not an issue for real-value representations. In real-value coding the gene comprises a single allele and is itself the parameter value.

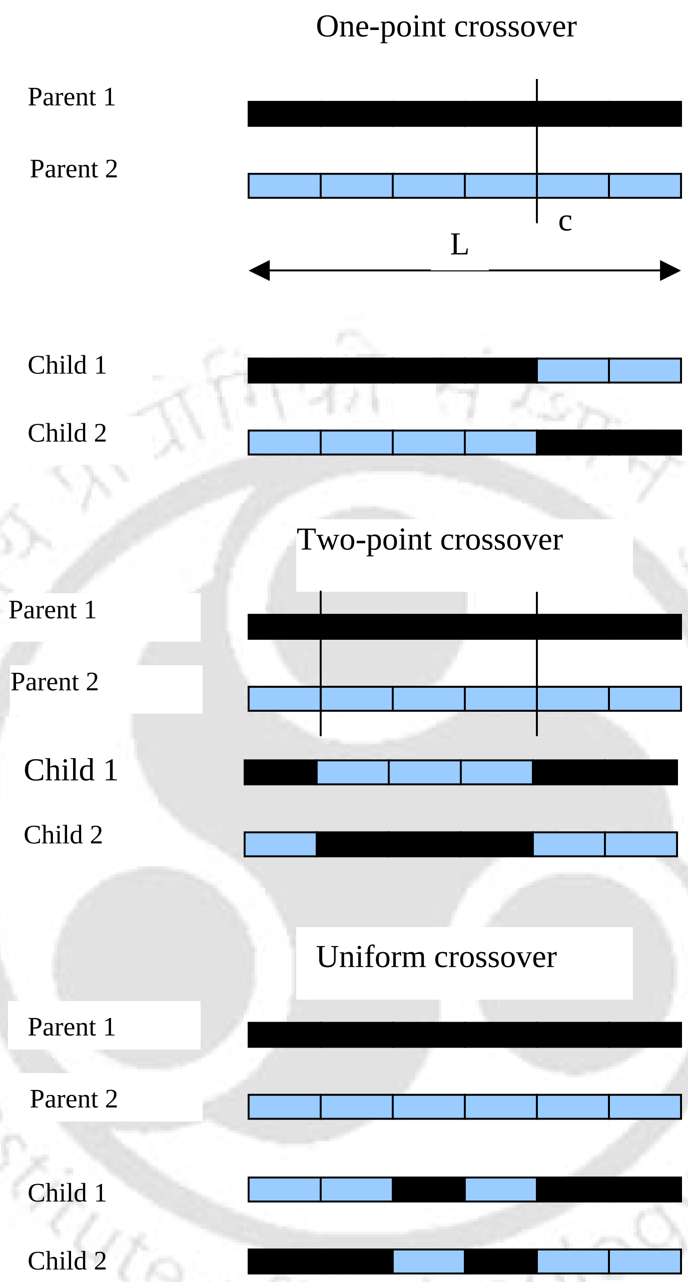


Fig. 8.2 Approaches to Crossover

Deb and Agrawal (1995) developed a crossover technique for use in the real coded GA. They called their technique as simulated binary crossover (SBX). They found that the real coded GA with the SBX operator was able to perform as good as or better than binary coded GA with single point crossover. They also found that

SBX operator performs better than other crossover operator used in real coded GA. In this study SBX operator has been used. The SBX operator works with two parent solutions and creates two children solutions.

The procedure of computing children solutions c_1 and c_2 from two-parent solutions y_1 and y_2 using SBX operator is as follows:

Step 1. A random number u is created between 0 and 1.

Step 2. A parameter β_q is determined as follows:

$$\beta_q = (u\alpha)^{\frac{1}{\eta_c+1}}, \qquad \text{if } u \le \frac{1}{\alpha}$$

$$= \left(\frac{1}{2-u\alpha}\right)^{\frac{1}{\eta_c+1}}, \qquad \text{otherwise} \tag{8.1}$$

Where $\alpha = 2 - \beta^{-(\eta_c+1)}$ and β is calculated as follows:

$$\beta = 1 + \frac{2}{y_2 - y_1}\min[(y_1 - y_l),(y_u - y_2)] \tag{8.2}$$

Here, the parameter y is assumed to vary in [y_l, y_u]. y_l and y_u are the lower and upper boundary of y respectively. The parameter η_c is the distribution index and can take any non-negative value.

Step 3. The children solutions are then calculated as follows:

$$c_1 = 0.5[(y_1 + y_2) - \beta_q|y_2 - y_1|] \tag{8.3}$$

$$c_2 = 0.5[(y_1 + y_2) + \beta_q|y_2 - y_1|] \tag{8.4}$$

For handling multiple variables, each variable is chosen with a probability 0.5 and the above operator is applied variable by variable.

8.2.6.3 MUTATION APPROACHES

Mutation is the process that facilitates introduction of new genetic material in to a population. A mutation probability is specified that permits random mutations to be made to individual genes. The mutation operator is introduced to prevent premature convergence to local optima by randomly sampling new points in the search space. It is carried out by randomly flipping bits. The three basic approaches to mutation in GAs are

1. Uniform mutation
2. Modified uniform mutation, and
3. Non-uniform mutation.

Uniform mutation and non-uniform mutation are two basic approaches for real-value representations (Michalewicz 1992). Uniform mutation permits the value of a gene to be mutated randomly within its feasible range of values, possibly resulting in significant modifications of otherwise good solutions. Modified uniform mutation permits modification of a gene by a specified amount, which can either be positive or negative. In non-uniform mutation, the amount by which genes are mutated can be reduced as a run progress, and can therefore help in the later generations to fine tune the solutions.

Deb and Goyal (1996) developed a mutation approach using a polynomial probability distribution to create a solution c in the vicinity of parent solution y. The technique restricts its search only to the permissible values of the variables, thereby reducing the search effort in converging to the optimum solution. The procedure is used for a parameter $y \in [y_l, y_u]$ as follows and is used in this study:

1. Create a random number u between 0 and 1

2. Calculate the parameter δ_q as follows:

$$\delta_q = [2u + (1-2u)(1-\delta)^{\eta_m+1}]^{\frac{1}{\eta_m+1}} - 1, \text{ if } u \le 0.5,$$

$$= 1 - [2(1-u) + 2(u-0.5)(1-\delta)^{\eta_m+1}]^{\frac{1}{\eta_m+1}}, \text{ otherwise} \quad (8.5)$$

where, $\delta = \min[(y - y_l), (y_u - y)]/(y_u - y_l)$. The parameter η_m is the distribution index for mutation and takes any non-negative value.

3. Calculate the mutated child as follows:

$$c = y + \delta_q (y_u - y_l). \quad (8.6)$$

8.2.7 GAs FOR CONSTRAINED OPTIMIZATION

Genetic algorithms can be used to solve constrained optimization problems. Although different methods for handling constraints have been suggested, penalty function methods have been mostly used (Deb 1991; Goldberg 1983). In penalty function method, a penalty term corresponding to the constraint violation is added to the objective function. The penalty parameter must be set right in penalty function method.

Setting appropriate value of penalty parameter or penalty weight is an important task in constraint handling problem. Deb (2000) developed a constraint handling method based on penalty function approach that does not require any penalty parameter. Constraints are handled by using suitable fitness function, which depends on the current population. Solutions in a population are assigned fitness so that feasible solutions are emphasized more than infeasible solutions. The method developed by Deb (2000) used tournament selection operator where two solutions are chosen from the population and better one is selected. In the method developed by Deb (2000), since solutions are not compared in terms of both objective function value and constraint violation information, there is no need for any explicit penalty parameter. However, to avoid any bias from any particular constraint, all constraints are normalized. Also in this method all constraints are converted to greater-than-equal to type. In this study constraint-handling technique developed by Deb (2000) has been used.

The fitness function is computed as follows (Deb 2000):

$$F(x) = f(x), \qquad \text{if } g_j(x) \geq 0, \forall j, j = 1, 2, \ldots J$$

$$= f_{max} + \sum_{j=1}^{J} \langle g_j(x) \rangle, \text{ otherwise} \qquad (8.7)$$

where 〈〉 denotes absolute value of the operand if the operand is negative and returns a zero value otherwise. $g_j(x)$ is the normalized constraints. ‘f(x)’ is the objective function value. The parameter f_{max} is the objective function value of the worst feasible solution in the population; J is the number of constraints.

8.3 FORMULATION OF GA FOR RESERVOIR OPERATING POLICY

The objective function for the optimization problem is minimization of squared deficit of release from demand. The objective function, which is explained in chapter 4, is given below:

$$\text{Minimize } f = \sum_{t=1}^{T} (D_t - R_t)^2 \qquad (8.8)$$

Where f is the objective function value and T is the number of time period in month considered to calculate the objective function value.

The constraints of the reservoir operation problem, as explained in chapter 4 are:

Continuity constraint of the reservoir:

$S_{t+1} = S_t + Q_t - E_t - R_t$ (8.9)

Reservoir storage constraint:

$S^{min} \leq S_{t+1} \leq S^{max}$ (8.10)

Where,

$S^{min} = 45.64 Mm^3$ and $S^{max} = 312.64 Mm^3$.

Constraints of release from the reservoir:

$R_t^{min} \leq R_t \leq R_t^{max}$ (8.11)

Where,

$R_t^{min} = \max [0, (S_t + Q_t - E_t - S^{max})]$

$R_t^{max} = S_t + Q_t - E_t - S^{min}$

The first step to solve a problem using a GA is to construct a string that can represent the decision variables those need to be determined. In reservoir optimization, release is the decision variable. In this study the release of a particular month has been defined as a function of water availability (initial storage + inflow during the current month) in the reservoir during that month. In this GA formulation reservoir release rule has been assumed to be connected piecewise linear functions. Also it is assumed that release rule is a non-decreasing function. Fig. 8.3 shows the way of representing reservoir release rule as connected piecewise linear functions of water availablility for a month.

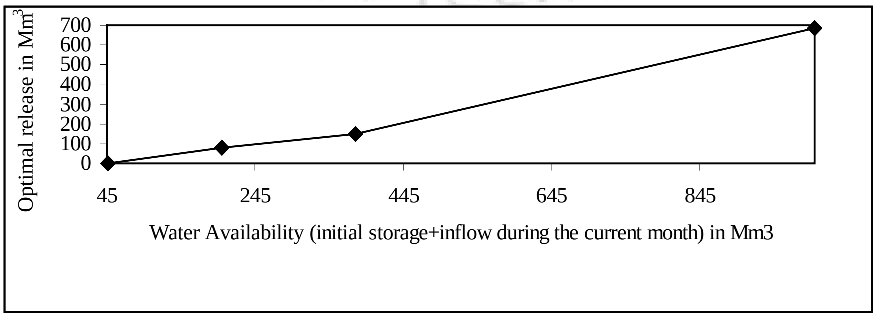


Fig.8.3. Reservoir release rule for a month

Since the release rule has been assumed to be connected piecewise linear functions, therefore, it is necessary to find the coordinates of the end points of the connected piecewise linear functions. Each endpoint consists of two coordinate ,viz., water availablity and release, which are unknown variable of the optimization problem in the GA formulation in this study. The problem is to find the coordinate of all the end points of the connected piecewise linear release rule functions that define the operating policy, which maximizes the system performance. However, in this formulation coordinate values of the extreme end points of the release rule function are known. Coordinates of the first point correspond to the condition, where no release can be made as water available become just equal to dead storage. The coordinates of the last point refer to the situation of having maximum possible water available. At this point water available in excess of storage at full reservoir level must be released to avoid violating constraints of upper limit of reservoir storage. The following paragraphs elaborate the procedure adopted for determining the release rule in this study for the reservoir problem defined in chapter 4.

Let the number of piecewise linear release rule functions considered are 3 for each month as shown in Fig 8.3. This means we have 4 end points and we have to determine 8 coordinates because each end point consist of two coordinates which are water available and release. Among these four end points coordinates of the first and the last points are known. At the first point, water availability corresponds to dead storage ,i.e., 45.64 Mm^3 and the release is zero. At the last point, the coordinate of water available has been considered as 1000 Mm^3, which is slightly higher than the sum of full reservoir storage and maximum monthly streamflow found in the observed as well as synthetically generated series. This value of maximum available water has been considered to ensure feasibility of the solution space for any inflow to the reservoir. The release coordinate of the last point is considered as the release needed to avoid violation of full reservoir storage constraint, because neither irrigation demand nor hydropower demand can exceed this necessary release for such extreme water available condition. Thus we need to determine the coordinates of two intermediate end points only. In general for N number of connected piecewise linear release rule functions there are N+1 end points and the number of unknown coordinates required to be determined are 2(N-1). In the GA formulation in this study, for each of the unknown points we need to set the upper and lower boundary of water availability and release. Convention used in this study to select the upper and lower boundary of the water availability and release for any of the unknown points in any month is as follows:

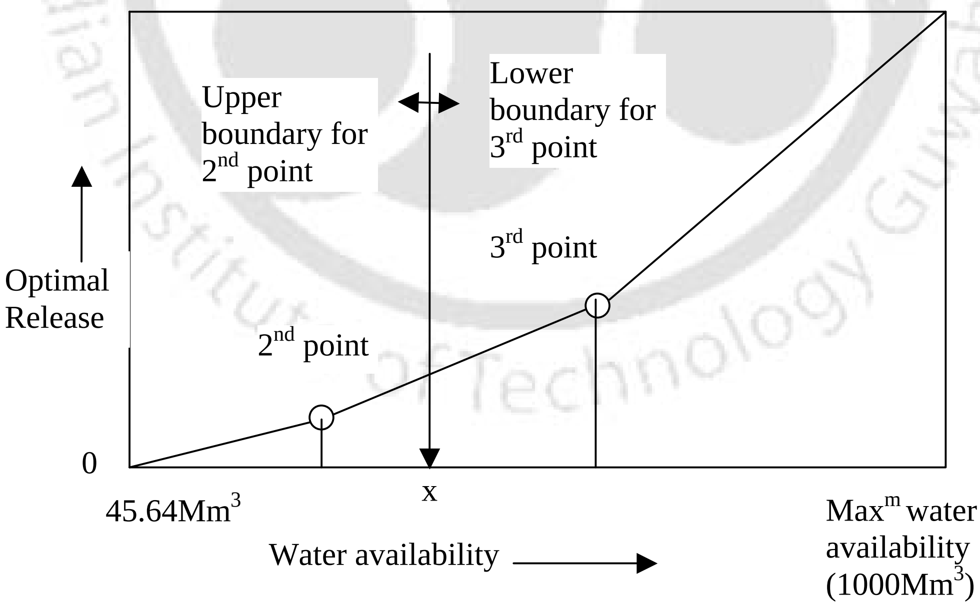


Fig. 8.3a. Boundary for coordinate point of water availability

If the upper boundary of the coordinate of water availability for second point (say) is x then coordinate of the lower boundary of the water availability for third point is also x (Fig. 8.3a). Lower boundary for release is 0 for water availability within the storage up to full reservoir level. For water availability exceeding the capacity at full reservoir level, the lower boundary of release is lower boundary of water available minus storage at full reservoir level.

The upper boundary of release coordinate for a point is set based on upper boundary of water availability for the point considered and dead storage of the reservoir.

For N number of connected piecewise linear release rule function, we need to determine 2(N-1) coordinates for each month. Hence for 12 month of the year we need to determine 2(N-1)12 number of coordinates. This means, the string of the GA will comprise 2(N-1)12 genes to derive the monthly operating policy for 12 months.

A string (say P1) of the population is a vector containing the coordinates of all the points that define the operating policy of the reservoir and can be represented as:

$P1=[W_1^1\ R_1^1\ W_1^2\ R_1^2 --- W_1^{N-1}\ R_1^{N-1} -------- W_{12}^1\ R_{12}^1\ W_{12}^2\ R_{12}^2 --- W_{12}^{N-1}\ R_{12}^{N-1}]$

In the above string, W represents total quantity of water availability (initial storage + current inflow) during a month of the year and R represent release during that month of the year. The superscript denotes end point and the subscript denote month of the year. N is the maximum number of piecewise linear release rule function considered to define the operating policy. In the above string for each month superscript begins from 1 and ends at N-1, this is because we want to determine coordinates of N-1 intermediate end points only. In GA, coordinate values for release and water availability are randomly assigned within their boundary. Therefore it may generate some infeasible strings. To handle such infeasible situation a penalty function method developed by Deb (2000) has been used for any constraint violation in this study. In all cases release and water available coordinate of a point should be such that final storage for each month satisfies the storage constraints ,i.e., final storage value for each month is greater than or equal to 45.64 Mm^3 (dead storage) and smaller than or equal to 312.64 Mm^3, which is the capacity at full reservoir level.

8.3.1 FITNESS EVALUATION OF STRINGS

In the GA model developed in this study, strings (operating policies) are randomly generated and these are simulated using a long sequence of synthetically generated streamflow and other relevant data to evaluate their fitness function values. Fitness function in this study is a minimization function, which represents summation of squared deficit of release from demand. Fitness function is computed based on the objective function value and the constraint violation information. Constraint violation information has been derived using the following logic for this problem.

Let S_{t+1} be the final storage value of any month t of the simulation period obtained by simulating the reservoir operation using a string (operating policy) generated by GA. If for any month t of the simulation period, the value of S_{t+1} is smaller than 45.64 Mm^3 (dead storage) or greater than 312.64 Mm^3 (storage at full reservoir level) the string is infeasible. To handle the infeasible string i.e. the policy that violate the constraints, the constraint-handling technique developed by Deb (2000) has been used.

The strings generated by GA must have to satisfy the following normalized constraints:

$$g_t = (S_{t+1} - 45.64)/S_{t+1} \geq 0 \qquad (8.12)$$

$$g_t = (312.64 - S_{t+1})/ S_{t+1} \geq 0 \qquad (8.13)$$

If, S_{t+1} becomes less than 45.64, then constraint violation is computed by

$g_t = (S_{t+1} - 45.64)/S_{t+1}$ (8.14)

And if S_{t+1} becomes greater than 312.64, then constraint violation is computed by

$g_t = (312.64 - S_{t+1})/ S_{t+1}$ (8.15)

In addition to the above physical constraints of the reservoir, following constraints have been considered to ensure that the release rule remains a non-decreasing function.

$G_{i,j} = (R_j^{i+1} - R_j^{i})/ R_j^{i} \geq 0$ (8.16)

where i= 1, 2,…N-2, and j= 1,2, …12.

If in a situation, $G_{i,j}$ becomes less than 0, the constraint violation is computed as

$G_{i,j} = (R_j^{i+1} - R_j^{i})/ R_j^{i}$ (8.17)

The fitness function is as follows:

$$F = f, \quad \text{if } g_t \geq 0, \forall t; \text{ and if } G_{i,j} \geq 0, \text{ for } \forall\, i,j$$

$$= f_{max} + \sum_{t=1}^{T} < g_t > + \sum_{j=1}^{12} \sum_{i=1}^{N-2} < G_{i,j} >, \text{ otherwise} \quad (8.18)$$

where, 〈〉 denotes absolute value of the operand if the operand is negative and returns a zero value otherwise. ‘f’ is the objective function value given by equation (8.8). The parameter f_{max} is the objective function value of the worst feasible solution in the population; t is the month of time series. T is the number of time periods in months (in this study T =1200) considered to calculate the fitness of strings (string represents the operating policies for 12 months) generated by GA. Synthetic monthly streamflow data for 1200 months along with other relevant data have been used in this study to calculate the fitness of strings.

8.3.2 SENSITIVITY ANALYSIS

8.3.2.1 SENSITIVITY TO CROSSOVER AND MUTATION PROBABILITY

Sensitivity analysis has been carried out to fix the best parameter setting viz. crossover probability and mutation probability for different GA models. Four GA models have been considered in this study for deriving optimal operating policy. They are GA1, GA2, GA3, and GA4 with 4, 6, 9, and 11 piecewise linear release rule function respectively. Therefore the length of string for GA1, GA2, GA3 and GA4 models are 72, 120, 192 and 240 respectively. For all GA models, fitness of strings has been computed using 1200 months of synthetic streamflow series. As mentioned earlier, fitness function in this study is a minimization function, which represents summation of squared deficit of release from demand. Goldberg (1989) suggested that good performance might be achieved from a GA using a high crossover probability and low mutation probability. Therefore, crossover probability in the range 0.6-1.0 and mutation probability in the range 0.005-0.2 have been tested to find the best combination of crossover and mutation probability with respect to fitness function values. Sensitivity to crossover has been analyzed against mutation probability of 0.005, .05, 0.1,0.15 and 0.2 and sensitivity to mutation has been analyzed against crossover probability of 0.6, 0.7, 0.75, 0.8, 0.85, 0.9, and 1.0. Population size used to carry out sensitivity analysis is equal to three times the length of the string in real value coding used in this study. The number of generation has been restricted to 3000 as there is very negligible improvement of fitness function after 3000 generation. To carryout the analysis in a systematic way, crossover probability were first varied assigning a fix value for the mutation probability (say 0.05). All the GA models were run for these values of crossover probabilities and with a fixed value of mutation probability. Fig 8.4 shows the plot of sensitivity towards crossover for a mutation probability of 0.05. Following the above procedure sensitivity to crossover with all the values of mutation probability considered in this study were carried out for all the models. Similarly, to carryout the analysis for sensitivity towards mutation probability, mutation probabilities were first varied assigning a fix value for the crossover probability (say 0.8). All the GA models were run for these values of mutation probabilities and with a fixed value of crossover probability. Fig 8.5 shows the plot of sensitivity towards mutation for a crossover probability of 0.8. Following the above procedure, sensitivity to mutation with all the values of crossover probability considered in this study were carried out for all the models. The best combination of crossover and mutation probability obtained for each of the model on the basis of above analysis has been presented in the Table 8.1.

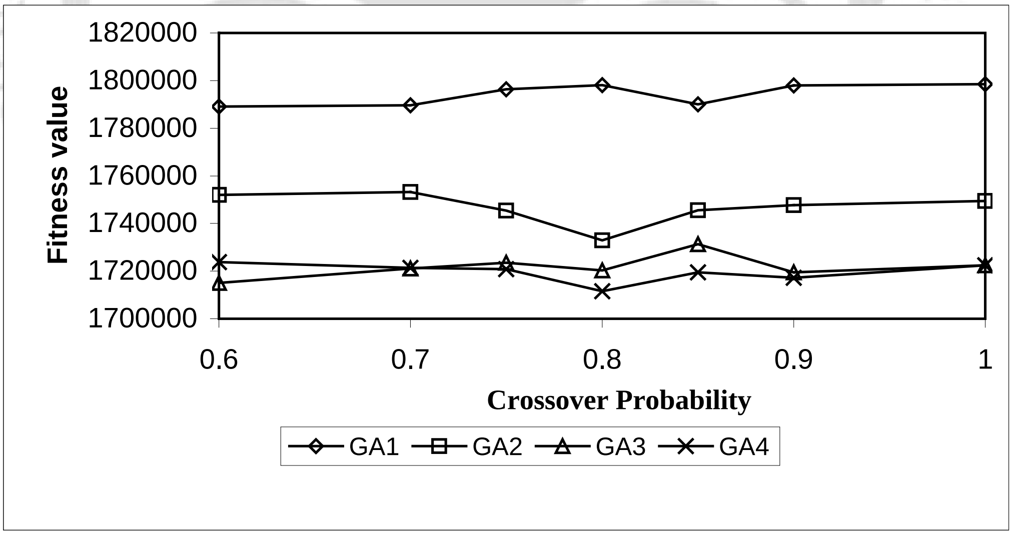


Fig.8.4. Sensitivity to crossover probability with mutation probability of 0.05.

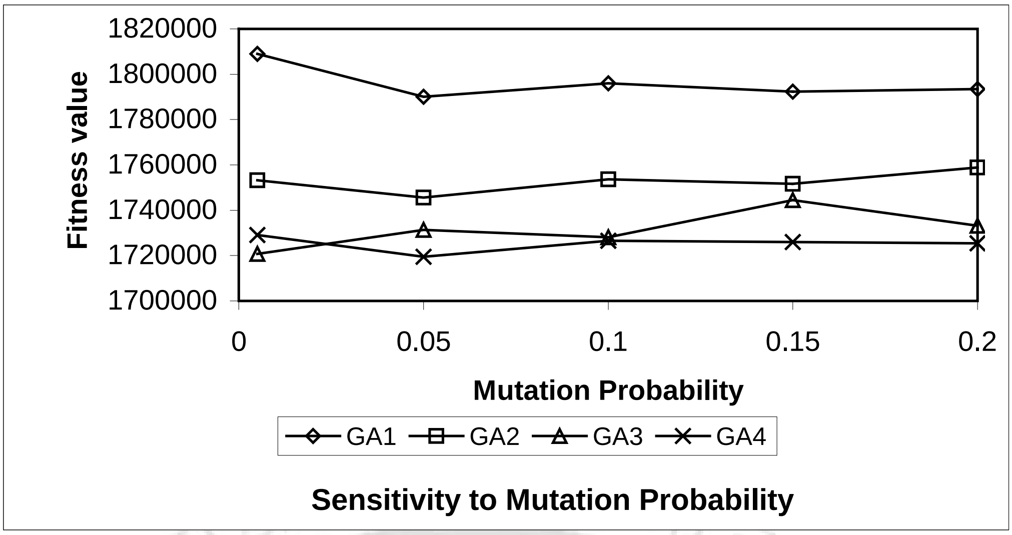


Fig.8.5. Sensitivity to mutation probability with crossover probability 0.8

Table 8.1. Best combinations of crossover and mutation probability for GA1, GA2, GA3 and GA4 models.

GA Model	GA1	GA2	GA3	GA4
Cross over Probability	0.9	0.8	0.8	0.8
Mutation Probability	0.005	0.05	0.05	0.05

8.3.2.2 SENSITIVITY TO POPULATION SIZE

Performance of GA is also influenced by population size. With small population size diversity in a population cannot be maintained. Usually the population size is taken as 2 to 4 times the length of the string (Rao 1996). However, to have a more accurate analysis, sensitivity of all the GA models to population size has been carried out with population size of 1, 2,3,and 5 times of the string length. The best combinations of crossover and mutation probability as presented in Table 8.1 have been used for analysis of sensitivity to population size. Fig. 8.6a through

8.6d shows the influence of population size on fitness value produced by the different GA models after 3000 generation. Fig.8.6a shows effect of population size on fitness value for GA1 model. For GA1, there is large improvement of fitness value when population size increases from 72 (equal to string length) to 144 (2 times of string length), but there is very negligible improvement of fitness value thereafter specially when population size increases from 216 (three times of string length) to 360 (5 times of string length). Fig 8.6b shows effect of population size on fitness value for GA2 model. For GA2, there is no large improvement of fitness value when population size increases from a value equal to the string length to a value 2 times that of the string length, but there is large improvement of fitness value when population size increases from 2 times of string length to 3 times of string length. But there is no improvement of fitness value when population size increases from 360 (3 times of string length) to 600 (5 times of string length). Fig.8.6c shows the effect of population size on fitness value for GA3 model. Here it has been observed that there is moderate improvement of fitness value when population size increases from 192 (equal to string length) to 576 (3 times of string length), but there is very negligible improvement of fitness when population size increases from 576 to 960 (5 times of string length). Fig.8.6d shows the effect of population size on fitness value for GA4 model. It has been observed that there is improvement of fitness values when population size increases from 240 (equal to string length) to 480 (2 times of string length) then from 480 to 720 (3 times of string length), but there is no improvement of fitness when population size increases from 720 to 1200 (5 times of string length).

From the above analysis of sensitivity to population it has been observed that fitness value invariably improve in all the GA models when population size was increased up to 3 times of the string length. However, rate of improvement of fitness differ in each model. In some cases fitness function has been found to deteriorate when population size was increased beyond 3 times of the string length. Considering the above, a population size of three times the length of the string can be considered as most preferred one for deriving optimal operating policy for the Pagladia multipurpose reservoir.

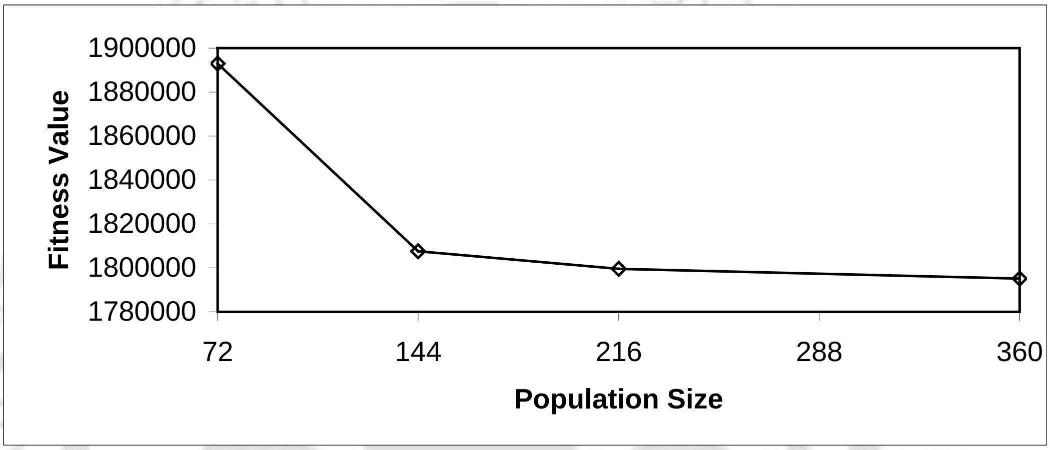


Fig. 8.6a. Influence of Population size on Fitness for GA1 model

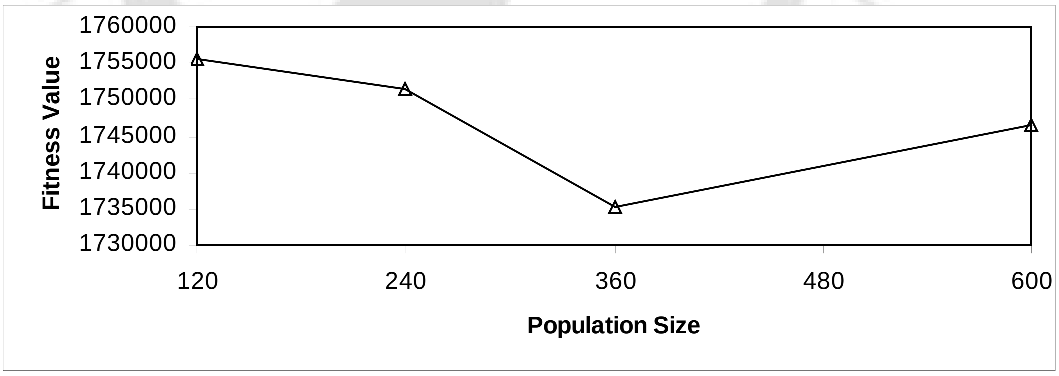


Fig. 8.6b Influence of population size on Fitness for GA2 model

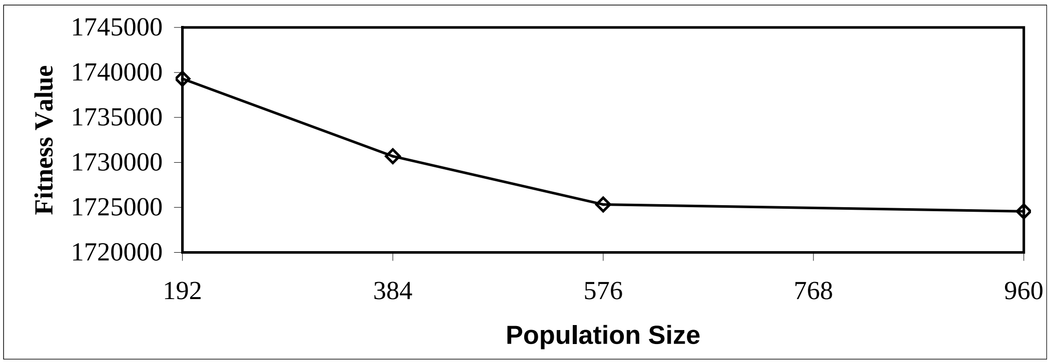


Fig. 8.6c: Influence of Population size on Fitness for GA3 Model

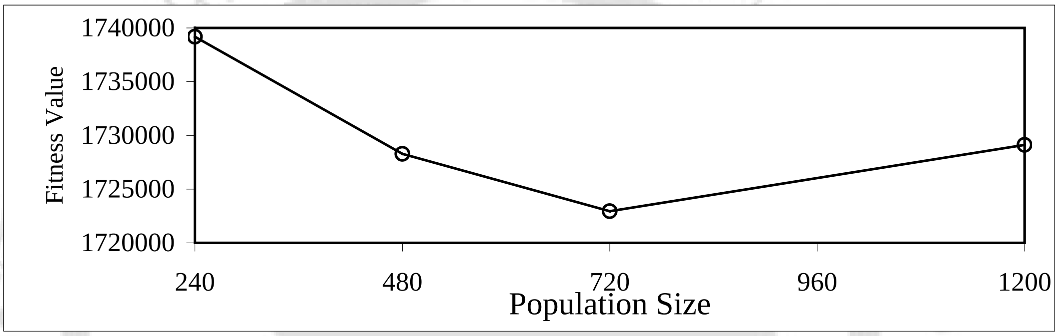


Fig. 8.6d. Influence of Population Size on Fitness for GA4 model

8.4. FINDING OPTIMAL OPERATING POLICY

After carrying out the sensitivity analyses, optimal operating policies are derived by each of the four GA models using the synthetic monthly streamflow series of 100 years. A population size equal to three times the length of the string has been used for all he GA models. With the best combination of crossover and mutation probability, each of the GA models has been run for 3 times. For each run of the GA, 3000 generations has been considered, as there is only negligible improvement of fitness function value after 3000 generations. After 3 runs of the GA, the best string (,i.e., the best operating policies for 12 months) that gives the best fitness function value (,i.e., minimum sum of squared deficit of release from demand) has been chosen as the optimal operating policy. In this way optimal operating policies have been found using GA1, GA2, GA3 and GA4 models. The optimal operating policies derived using GA3 model has been presented in section 8.6.

8.5 PERFORMANCE EVALUTATION OF OPERATING POLICIES DEVELOP BY GA MODELS

The performance criteria as mentioned in section 6.4.1 of chapter 6 have been used for the evaluation of operating policy developed by four GA models. The performance evaluation of operating policies developed by GA models has been carried out through simulation of the reservoir for 228 months of historic streamflow series from (1977 –1996). The starting month of the simulation was October 1977 with initial storage of the reservoir as 45.64Mm^3 (dead storage volume). The final month of the simulation period was September 1996

8.5.1 SIMULATION AND RESULTS

Table 8.2 shows the reservoir simulation results obtained by using the operating policies developed by GA1, GA2, GA3 and GA4. Standard operating policy (SOP) results have also been shown in the table 8.2. In SOP, if the available water is less than or equal to demand then available water is released, on the other hand if the water availability is higher than demand, the quantity equal to demand is released and the remaining quantity is stored if possible, otherwise it is spilled.

Table 8.2: Simulation results of GA models for 228 months of historic streamflow data (1977-1996) along with SOP results.

Model	Total squared deficit (10^5)	Total irrigation deficit (Mm^3)	Number of irrigation deficit month	Total power generation (10^8 KWH)	Number of power deficit month	Total water deficit (Mm^3)	Number of times the reservoir is full	Total Spill (Mm^3)
GA1	5.06	201.96	14	3.32	182	7208.30	79	3710.4
GA2	5.60	259.7	14	3.30	187	7594.76	70	3569.06
GA3	5.04	172.67	12	3.33	182	7277.66	72	3542.6
GA4	5.09	257.14	14	3.32	188	7290.71	66	3501.0
SOP	432.0	1099.65	38	2.22	164	70166.6	10	1159.13

From Table 8.2 it has been observed that minimum squared deficit for the simulation period is obtained with the operating policy derived by GA3 model. GA3 model has also been found to give less total irrigation deficit and less number of irrigation deficit months as compared to the other three GA models. In terms of power generation also GA3 model has been found to be marginally better than the other three GA models. It is important to mention that, in a month, when there is deficit for irrigation there is also deficit for power generation, which has not been shown explicitly in the Table 8.2. Number of power deficit month has been found more or less the same in all the GA models. Total water deficit has been found minimum in GA1 model. Next to GA1 model total water deficit has been found minimum in GA3 model. Total water deficit includes deficit for both irrigation and power generation. The operating policy developed by GA1 model keeps the reservoir full for maximum number of times, i.e., 77 times. Next to GA1 model, GA3 model keeps the reservoir full for maximum number of times (72). The total spill has been found minimum with the operating policy derived by GA4 model. Next to GA4 model it is the GA3 model that spills lesser quantity of water.

Above comparisons have revealed that operating policy derived by all the GA models are competitive. However GA3 model has been found to be better than other GA models in respect of several important performance criteria. Considering better performance of GA3 model in terms total squared deficit, total irrigation deficit, number of irrigation deficit month and total power generation, the operating policy derived by GA3 model can be regarded as the most efficient one among all the operating policies derived by different GA models in this study.

8.6 CONCLUSION

The operating policy for the Pagladia multipurpose reservoir has been developed in this chapter using GA technique. Four different GA models have been developed to derive the operating policy for the Pagladia multipurpose reservoir. The GA models developed are GA1, GA2, GA3 and GA4 with 4, 6, 9 and 11 piecewise release rule functions respectively. Among the four models, operating policy developed by GA3 model is found better than other three models in terms of total squared deficit, total irrigation deficit, number of irrigation deficit month and total power generation. Therefore operating policy derived by GA3 model has been considered as the most efficient one among all the operating policies derived by different GA models in this study. The operating policy for January through December derived by GA3 model is presented in Fig.8.7 through Fig.8.18.

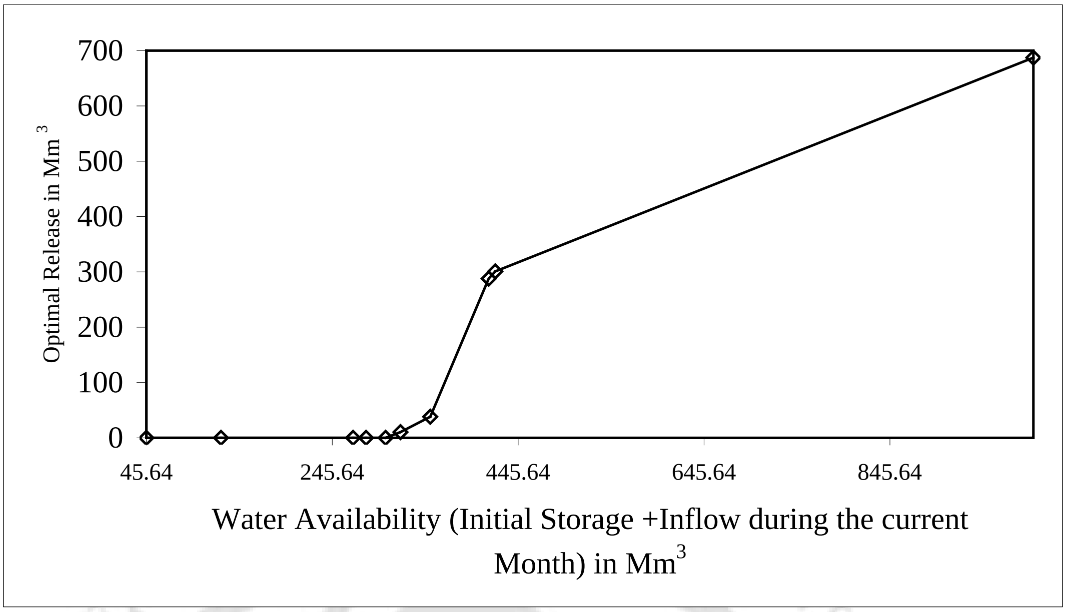


Fig.8.7. Optimal Operating Policy for the Month of January

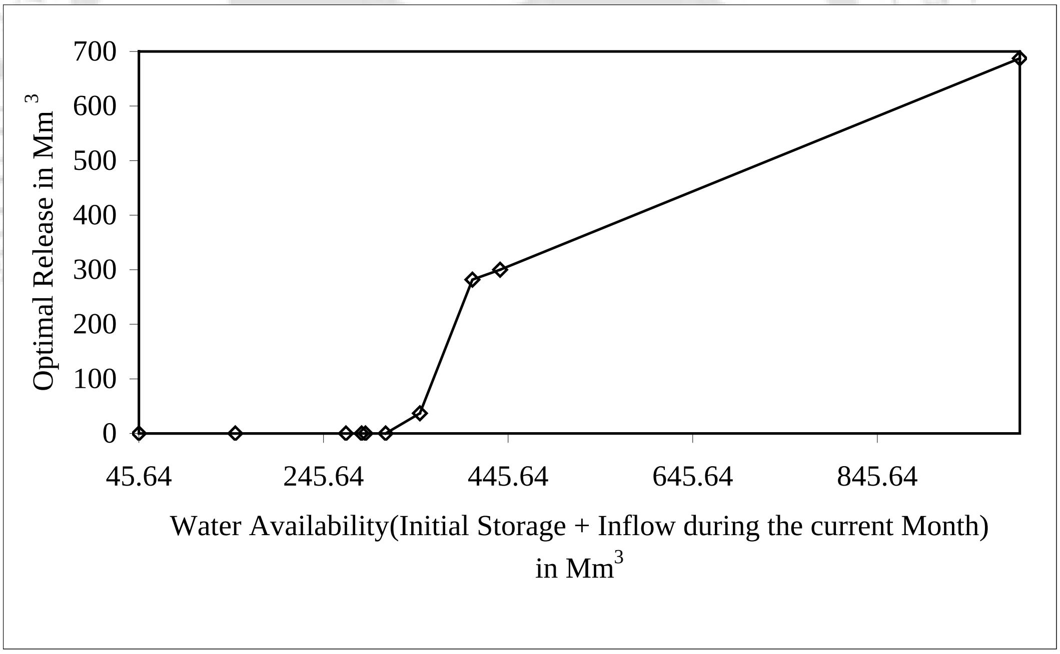


Fig. 8.8: Optimal Operating Policy for the Month of February

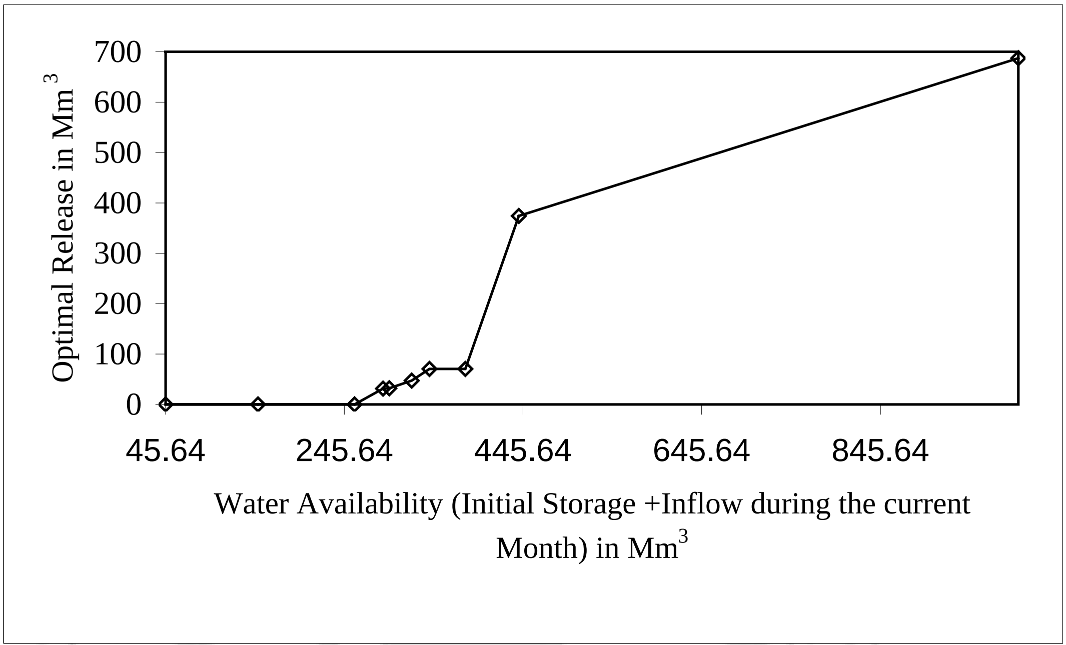


Fig. 8.9. Optimal Operating Policy for the Month of March

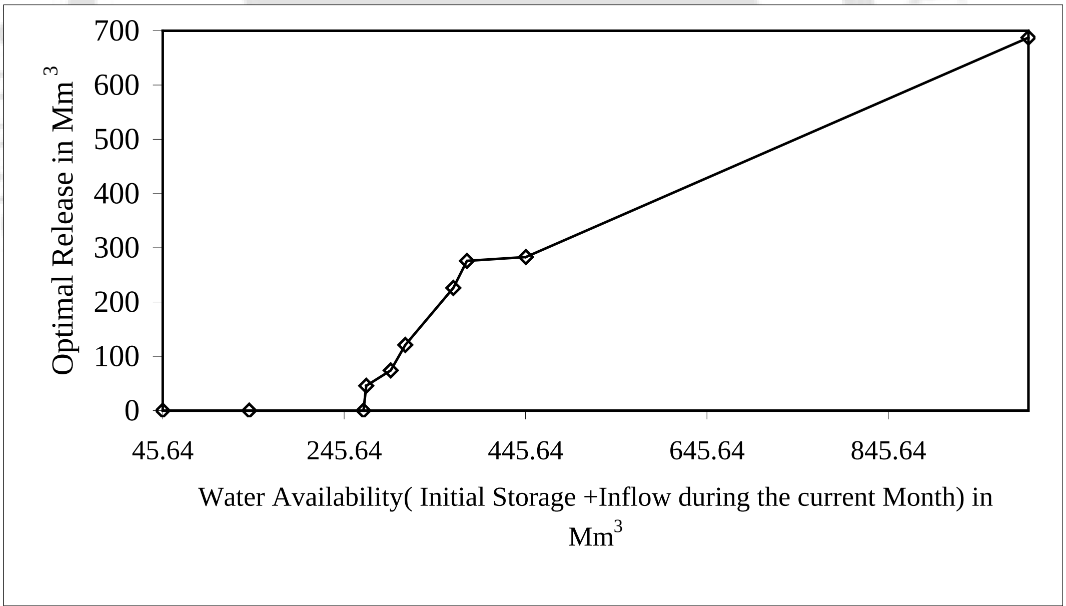


Fig. 8.10. Optimal Operating Policy for the Month of April

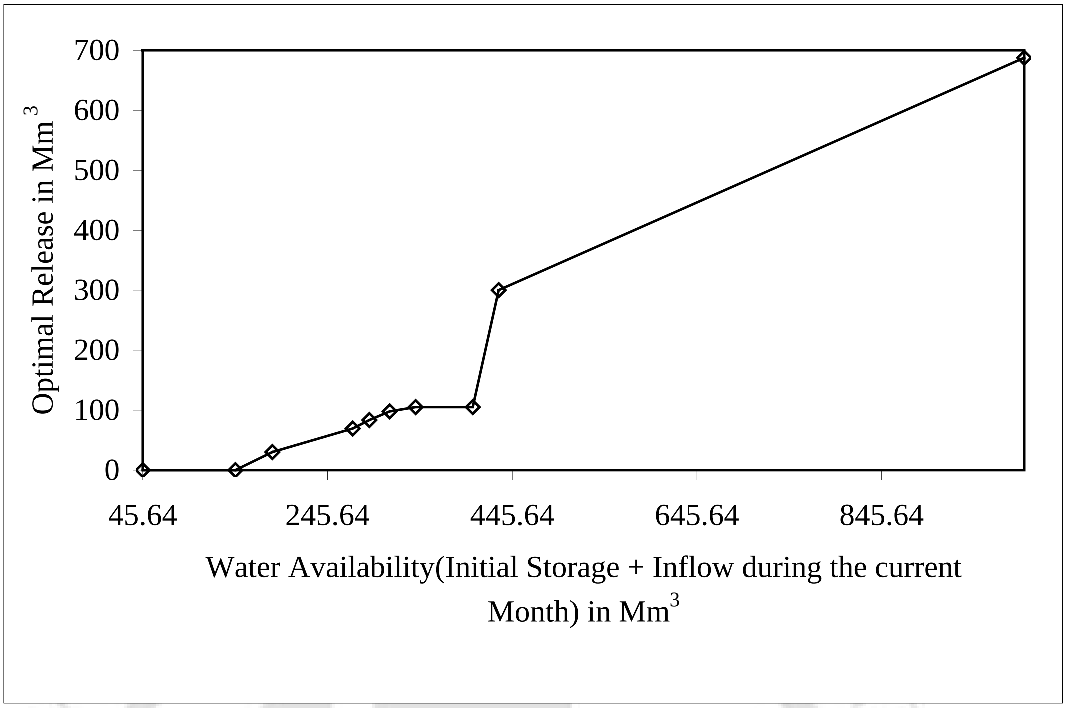


Fig. 8.11. Optimal Operating Policy for the Month of May

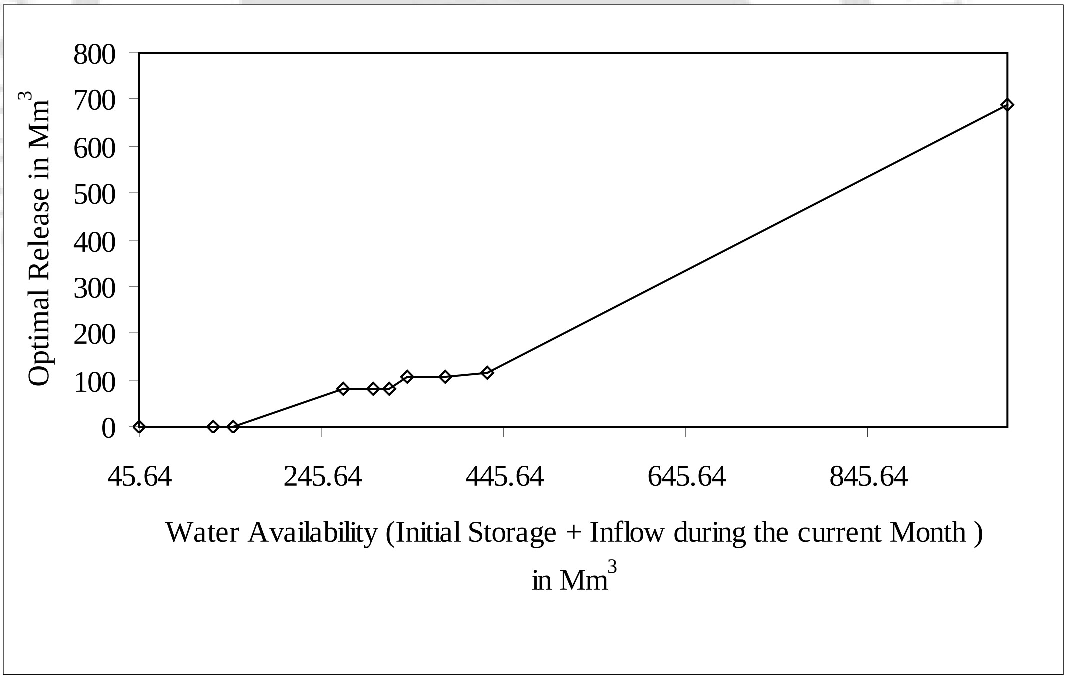


Fig. 8.12. Optimal Operating Policy for the Month of June

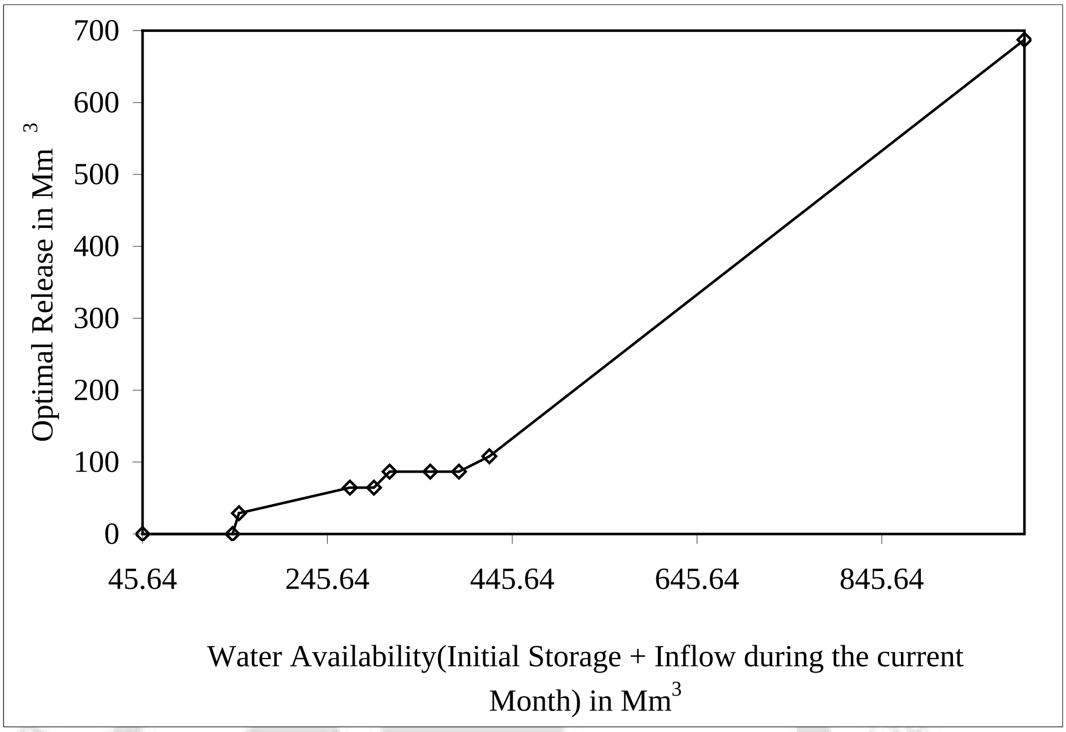


Fig.8.13. Optimal Operating Policy for the Month of July

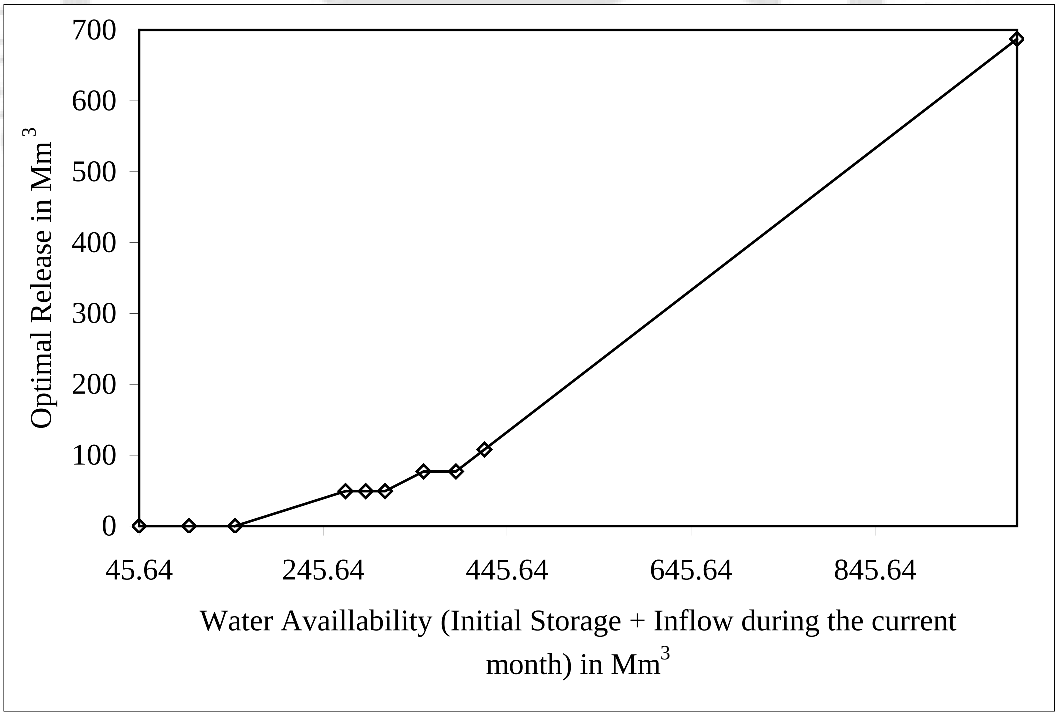


Fig.8.14. Optimal Operating Policy for the Month of August

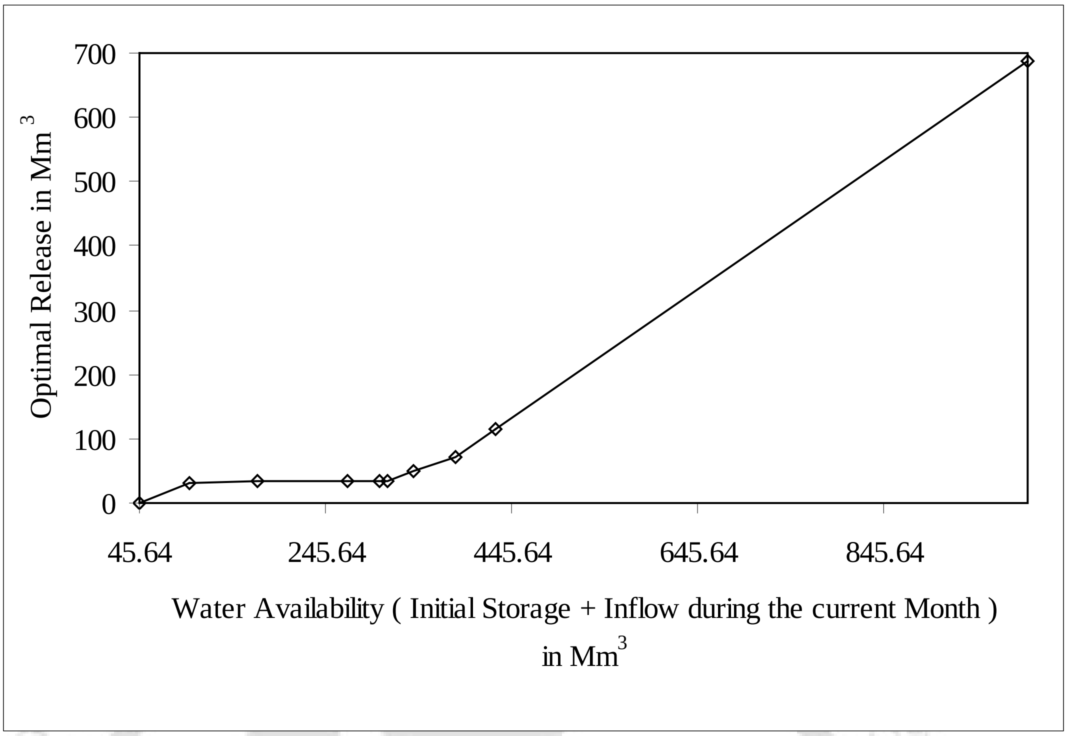


Fig.8.15. Optimal Operating Policy for the Month of September

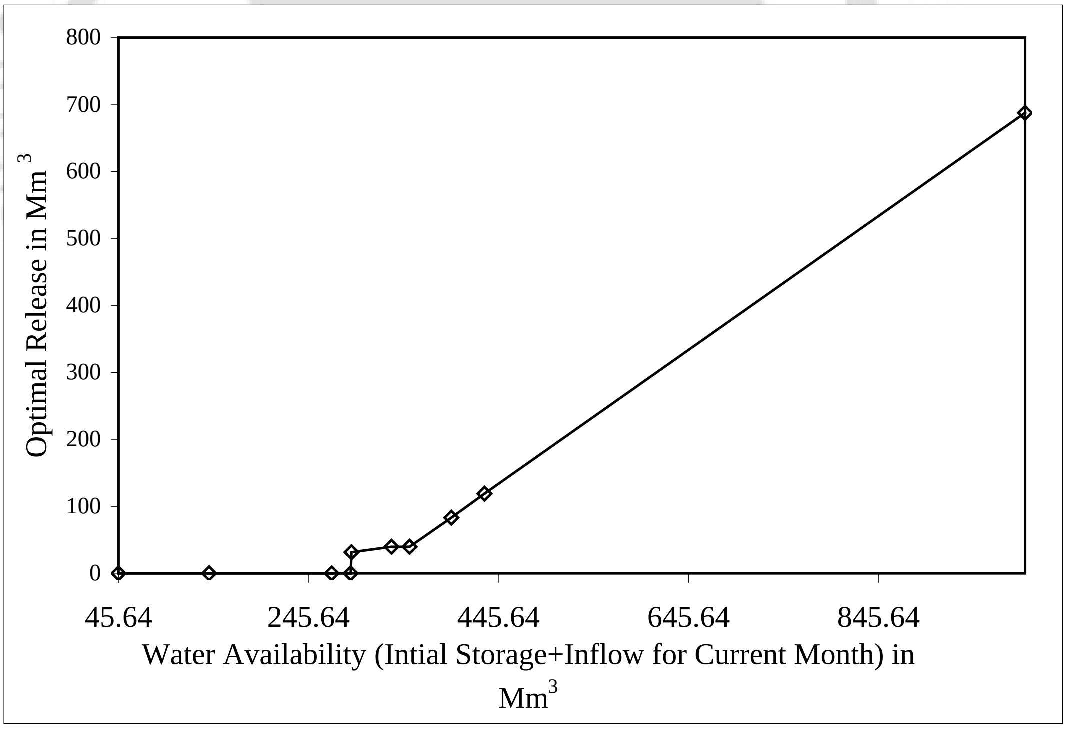


Fig.8.16. Optimal Operating Policy for the Month of October

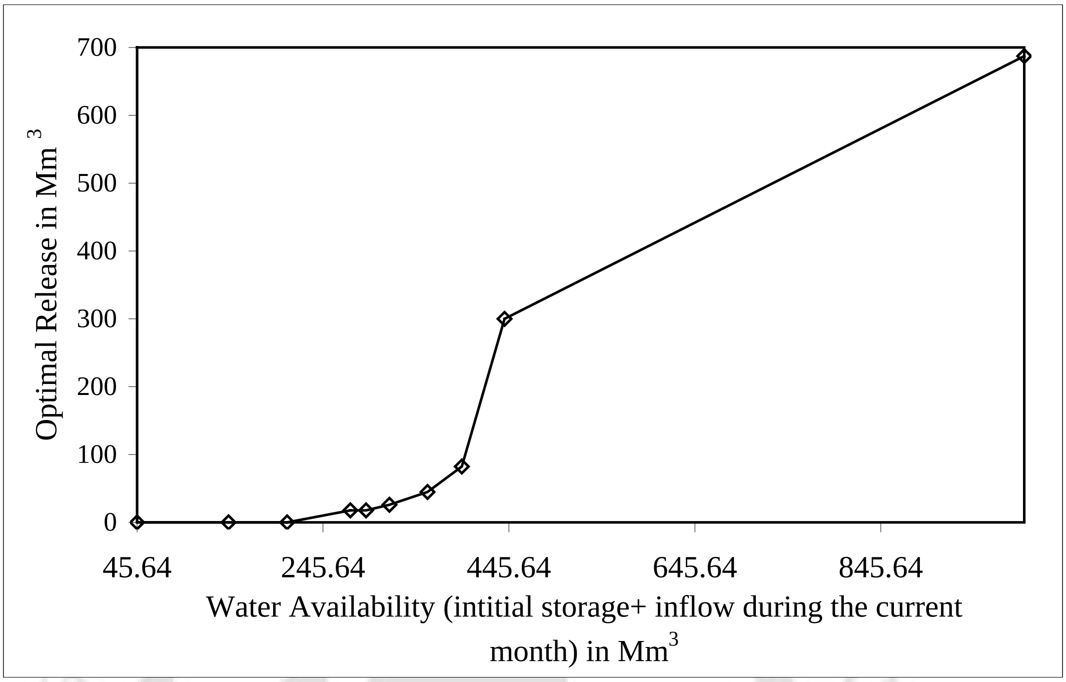


Fig.8.17. Optimal Operating Policy for the Month of November

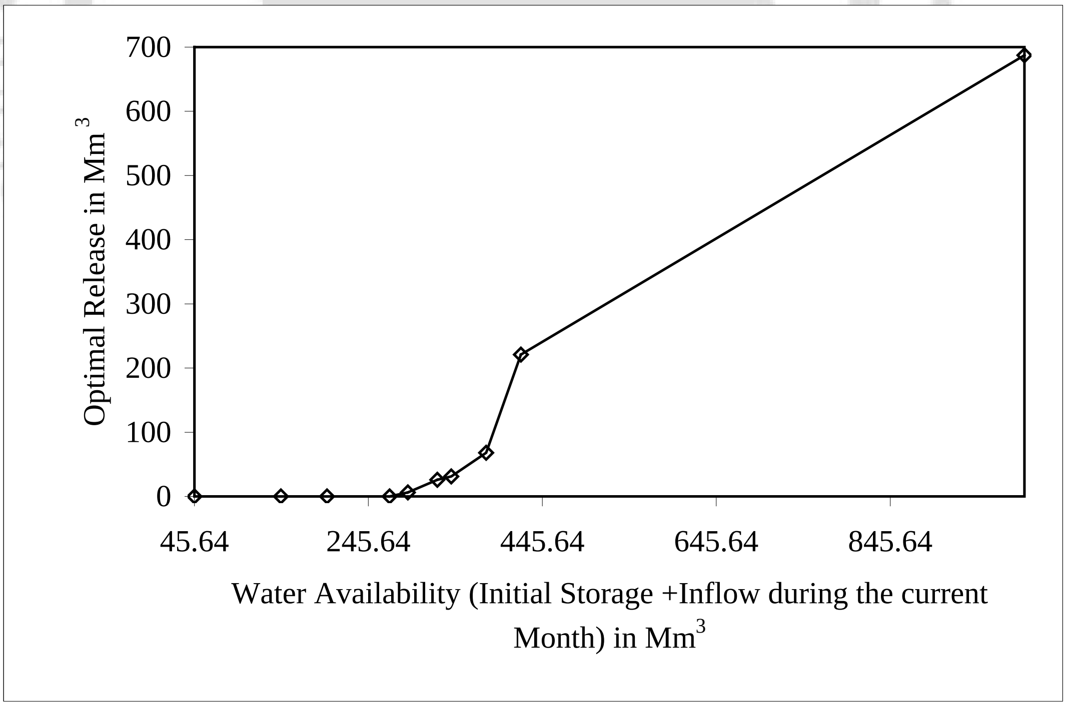


Fig.8.18. Optimal Operating Policy for the Month of December

CHAPTER 9

COMPARISION OF RESERVOIR OPERATION MODELS

9.1 INTRODUCTION

In this chapter a comparison of all the reservoir operation models developed in this study has been carried out based on the reservoir simulation results obtained in the previous chapters (chapter 6,7 and 8). Models developed in this study are: three DPR models, namely DPR1, DPR2, and DPR3; three DPN models, namely DPN1, DPN2, and DPN3; three SDP models, namely SDP1, SDP2, and SDP3; and four GA models, namely, GA1, GA2, GA3 and GA4. Although performances of all the DPR, DPN, SDP and GA models were evaluated in the previous chapters, an overall comparison and comprehensive evaluation of all these models has been carried out in this chapter so as to suggest an efficient operating policy for the Pagladia multipurpose reservoir.

9.2 COMPARISON OF RESERVOIR OPERATION MODELS

Performances of all the operating policies have been analyzed on the basis of reservoir simulation results. All the data required to carryout the simulation run have been kept same for all the models. For making a fair comparison among all the models developed in this study, all the eight performance criteria that were used to evaluate each of the DPR, DPN, SDP and GA models have been used to make a comparison across the models. The performance criteria used are:

(1) Total squared deficit.

(2) Total irrigation deficit.

(3) Number of irrigation deficit month.

(4) Total power generation.

(5) Number of power deficit month.

(6) Total water deficit.

(7) Number of times the reservoir is full.

(8) Total spill.

Table 9.1 shows the comparative performances of different models in respect of the above-mentioned criteria when reservoir simulation was carried out for 228 months of historic streamflow (1977-1998) series.

From table 9.1 it has been observed that minimum squared deficit is found in GA3 model. All the GA models considered in this study give less squared deficit than DPR, DPN and SDP model. This means GA model is efficient in terms of minimizing total squared deficit, which is the objective function of the reservoir optimization model in this study.

In terms of total irrigation deficit also GA3 model is found efficient than other models. However except GA3 model, irrigation deficit is more or less same in SDP and GA models. All the DPR and DPN models have been found to give large irrigation deficit. All GA models are found efficient in terms of number of deficit month for irrigation. Among all the GA models, GA3 model gives the minimum number of irrigation deficit months. In all the SDP models number of deficit months for irrigation is almost double of the GA3 model. In all the DPR and DPN models, number of irrigation deficit month is almost four times than that of the GA3 model.

Table 9.1: Comparisons of reservoir simulation results for 228 months of historic streamflow(1977-1996).

Model	Total squared deficit (10^5)	Total irrigation deficit (Mm^3)	Number of Irrigat-ion deficit month	Total power generat-ion (10^8 KWH)	Number of power deficit month	Total water deficit (Mm^3)	Number of times the reservo-ir is full	Total spill (Mm^3)	Minimum occurrence for Irrigation deficit		Maximum occurrence for Irrigation deficit		Minimum occurrence for power deficit		Maximum occurrence for power deficit		Minimum Deficit		Maximum Deficit		Minimum occurrence for reservoir full		Maximum occurrence for reservoir full		Minimum occurrence for reservoir empty		Maximum occurrence for reservoir empty		Annual reliability %	
									Months	No of times	Months	No. of times	Months	No. of times	Months	No of times	Months	Qty Mm^3	Months	Qty Mm^3	Months	No. of times	Months	No. of times	Months	No. of times	Months	No. of times	Irrigat-ion	Power
DPR1	9.67	1916.4	44	3.22	182	9474.92	18	2535.75	June-Sept	0	March, April	19	July	5	Dec, Feb-May	19	Sept.	89.8	April	2176.4	Dec-July	0	Sept	8	Nil	Nil	Nil	Nil	80.7	20.17
DPR2	6.83	1012.87	49	3.25	180	8043.21	27	2832.03	June-Sept	0	March, April	19	July	3	Nov-May	19	August	81.95	March	1842.4	Dec-July	0	Sept.	10	Nil	Nil	Nil	Nil	78.5	21.05
DPR3	7.65	972.44	48	3.25	179	8436.21	21	2766.5	June-Sept.	0	March, April	19	July	4	Nov-May	19	Sept.	78.64	April	1936.8	Dec-July	0	Sept.	9	Nil	Nil	Nil	Nil	78.94	21.49
DPN1	8.32	1178.0	45	3.22	170	8416.90	33	2679.7	July-Sept.	0	March, April	19	July, August	4	Dec-May	19	August	50.69	March	1958.2	Nov-May	0	July	11	Nil	Nil	Nil	Nil	80.26	25.44
DPN2	7.98	1203.6	47	3.19	166	82134.0	27	3219.2	July-Sept.	0	March, April	19	July	2	Dec-May	19	July	37.07	March	1863.8	Nov-August	0	July	11	Nil	Nil	Nil	Nil	79.39	27.19
DPN3	8.09	1182	47	3.20	168	8272.20	36	2878.4	July-Sept.	0	March, April	19	July	2	Dec-May	19	July	41.46	March	1892.0	Nov-August	0	July	11	Nil	Nil	Nil	Nil	79.39	26.31
SDP1	6.48	272.17	24	3.44	184	7917.79	113	3272.82	May, July	0	Oct	10	July, August	8	Nov-April	19	July	191.59	March	1100.13	March-May	0	Oct, Nov	15	Nil	Nil	Nil	Nil	89.47	19.29
SDP2	5.82	261.9	21	3.41	185	7801.65	76	3233.23	May, July	0	Oct	9	July, August	8	Nov-April	19	July	180.25	March	1102.8	Jan,April-June	0	Oct	15	Nil	Nil	Nil	Nil	90.78	18.85
SDP3	6.07	217.59	26	3.39	184	7981.99	84	3696.0	May, Sept	0	Oct	9	July, August	6	Nov-May	19	August	123.07	March	1102.48	Jan,April-June	0	Oct	16	Nil	Nil	Nil	Nil	88.59	19.29
GA1	5.06	201.96	14	3.32	182	7208.30	79	3710.4	May-Aug	0	Oct	7	August	6	Nov-May	19	August	140.98	March	1021.36	Dec-Feb April-July	0	Sept	18	Nil	Nil	Nil	Nil	93.86	20.18
GA2	5.60	259.7	14	3.30	187	7594.76	70	3569.06	May-July	0	Oct	7	August	7	Nov-May	19	Sept.	237.85	April	907.66	Dec-Feb April-July	0	Oct	17	Nil	Nil	Nil	Nil	93.86	17.98
GA3	5.04	172.67	12	3.33	182	7277.66	72	3542.6	May-Aug	0	Oct	7	July	6	Nov-May	19	August	159.7	March	950.3	Dec-Feb April-July	0	Oct	17	Nil	Nil	Nil	Nil	94.74	20.18
GA4	5.09	257.14	14	3.32	188	7290.71	66	3501.0	June, July	0	Oct	8	July	7	Nov-April	19	July	158.12	March	959.08	Dec-Feb April-May	0	Oct	13	Nil	Nil	Nil	Nil	93.86	17.54
SOP	432.0	1099.65	38	2.22	164	70166.6	10	1159.13	June-Sept	0	April	19	Oct, August, Sept	9	March-May	19	August	1101.8	April	16112.1	Dec-July	0	Oct, Sept.	4	Oct, June	1	April, May	16	83.33	28.07

In terms of power generation, SDP model has been found efficient. Among the SDP models, SDP1 gives maximum power but it gives maximum total irrigation deficit. Among GA models GA3 give maximum power. All DPR and DPN models generate less power than SDP and GA models although number of deficit month for power is found less in DPN models. DPN3 model gives least number of power deficit months. Apart from DPN models all other models give almost equal number of power deficit month.

In terms of total water deficit, all the GA models perform efficiently. Among the GA models, total water deficit is found minimum in GA1 model. Next to GA1 model total water deficit is found minimum in GA3 model. SDP models can be placed next to GA models in respect of total water deficit. DPR and DPN models perform poorly in this regard.

SDP model is found to keep the reservoir full for maximum number of times. Performance of GA models in keeping the reservoir full is also close to that of SDP models. DPR and DPN models keep the reservoir full for least number of times. In respect of spill SOP model spill less quantity of water as compared to other models. Among the different models, it appears that water utilization is better in GA and SDP models.

Between the irrigation and power generation, Pagladia multipurpose reservoir has the first priority of irrigation. From the simulation result it has been observed that GA and SDP models are competitive. For the present study DPR and DPN models has been found less efficient than GA and SDP models. Considering the fact that GA models give minimum squared deficit, minimum irrigation deficit and less number of irrigation deficit months than other models, operating policy developed by GA3 model, which is found best among the GA models, is recommended for Pagladia multipurpose reservoir.

9.3 CONCLUSION

Comparison among different models developed in this study has been carried out to explore their efficiency and hence to select an efficient operating policy for the Pagladia multipurpose reservoir. For having a fair comparison, a total of eight different criteria covering different aspects of the reservoir have been chosen. SDP models and the GA models have been found to be competitive in respect of some of the performance criteria. Although performance of different models varies with different performance criteria, considering overall performance and giving priority to irrigation the operating policy derived by GA3 model has been selected as the most efficient operating policy and is suggested for practical application in the Pagladia multipurpose reservoir.

CHAPTER 10

CONCLUSION, GENERAL DISCUSSION AND RECOMMENDATIONS FOR FURTHER STUDIES

10.1 INTRODUCTION

Although brief conclusions have been presented at the end of each of the previous chapters, a comprehensive conclusion followed by an overall discussion on the works performed in this study is presented in this chapter. Scope for future extension of this work is also suggested.

10.2 CONCLUSION AND GENERAL DISCUSSION

Review of literature has revealed that different investigators have put their efforts to develop reservoir-operating policy using different methods. Efforts are still on for developing more efficient operating policy for maximizing beneficial use of reservoirs. Besides traditional techniques, some researchers have also attempted application of non-traditional techniques like, ANN, Fuzzy logic, and GA. Although extensive research has been done towards finding a better method for deriving optimal operating policy, superiority of a particular method in general could not be claimed. This has led the author to investigate the potential of different techniques including GA and ANN in developing reservoir-operating policy through their application in the Pagladia multipurpose reservoir so that an efficient operating policy can be suggested for implementation.

The three basic purposes for which the Pagladia multipurpose reservoir is proposed are flood control, irrigation and power generation. The flood in the downstream area is proposed to be controlled by providing a flood control storage space in the reservoir. The regulatory system is such that, the water released for irrigation is led through the turbine to produce power. Irrigation water required in excess of power demand is led to the field through a bypass. Similarly, if the water requirement for power is more than the irrigation demand, then the water in excess of irrigation demand is led directly to the river from the tail race of the power house. The water resources system optimization problem for the Pagladia dam project has been formulated considering all these physical aspects. Due importance has also been given to the climatic aspects. Minimization of squared deficit of the monthly release from the monthly demand has been considered as the most suitable objective function as it prevents large deficit in any particular month and tends to distribute the deficits uniformly over more numbers of months. This is generally advantageous from both agricultural and power management point of view.

For the development of a good operating policy, a longer series of streamflow data is generally required. Therefore 100 years of synthetic monthly streamflow of the Pagladia River at dam site have been generated using three models, namely, Thomas-Fiering model, ARMA model and an ANN based model. The generated series have been compared with the observed series on the basis of some statistics. The statistics considered to compare the synthetic series are mean value of streamflow of each month, standard deviation of streamflow of each month, mean of the streamflow series, standard deviation of the streamflow series and skew coefficient of the streamflow series. The statistics of the synthetic monthly streamflow series generated by the ANN based model have been found better than those of the Thomas-Fiering model and the ARMA model when compared with the statistics of historic monthly streamflow series. Therefore, 100 years of synthetic monthly streamflow data generated by ANN based model have been used to develop operating policy of the Pagladia multipurpose reservoir. The comparative study of all the above methods of synthetic streamflow generation have revealed that, in general, ANN based model developed in this study can be considered as a potential alternative to the other conventional methods of synthetic streamflow generation.

Using 100 years of synthetic monthly streamflow series, the deterministic dynamic programming (DP) has been solved considering different numbers of characteristic storages of the reservoir. Two approaches have been used to derive the general operating rule from the deterministic DP result: one using multiple linear regression called DPR, and another using artificial neural network called DPN. The deterministic DP have been solved using 20, 30 and 40 numbers of characteristic storages of the reservoir, and the corresponding DPR models are called DPR1, DPR2, and DPR3 respectively. Similarly, the corresponding DPN models are called DPN1, DPN2, and DPN3 respectively.

Optimal operating policies have also been developed using SDP models. The SDP models have been developed considering 20, 30 and 40 characteristic storages of the reservoir and the corresponding models called SDP1, SDP2 and SDP3 respectively. For all the three SDP models, seven numbers of characteristic inflows have been considered. It has been observed that SDP2 model ,i.e,. the model based on 30 characteristic storages of the reservoir, performed better than other two SDP models in terms of squared deficit as well as some other performance criteria considered in this study.

Four GA models have been developed considering different numbers of connected piecewise linear release rule functions. The GA models developed are GA1, GA2, GA3 and GA4 with 4, 6, 9 and 11 connected piecewise linear release rule functions respectively. From the reservoir simulation result, it has been found that GA models are not very sensitive with the number of connected piecewise linear release rule functions considered in this study. However, among all the four GA models, GA3 model has been found to give better performance in terms of squared deficit, irrigation deficit, number of irrigation deficit months, and total power generation.

A comprehensive comparison of all the operating policies developed by different DPR, DPN, SDP, and GA models has been carried out on the basis of their performance in reservoir simulation. For having a fair comparison among all the models, a total of eight different criteria covering different aspects of the reservoir have been chosen so as to suggest the most efficient operating policy. The criteria considered for comparison are total squared deficit, total irrigation deficit, number of irrigation deficit month, total power generation, number of power deficit month, total water deficit, number of times the reservoir is full, and total spill. The comparison has revealed that all the GA models have been found to give less squared deficit than DPR, DPN and SDP models. In terms of total irrigation deficit, GA models have been found to have a slight edge over SDP models and they outperform the DPR and DPN models developed in this study. All the GA models have been found efficient in terms of number of deficit months for irrigation. Among all the GA models, GA3 model has been found to give the minimum number of irrigation deficit month. In all the SDP models, number of deficit months for irrigation has been found almost double of that of the GA3 model. In all the DPR and DPN models, numbers of irrigation deficit months have been found almost four times than that of the GA3 model. In terms of power generation, SDP models have been found efficient. Performance of GA models in this regard is also close with that of the SDP models. All DPR and DPN models have been found to generate less power than SDP and GA models. However, in respect of power deficit months, DPN models, in general, have been found to give marginally better performance than the other models. Total water deficit has been found minimum in GA models than in all other models. Among GA models, total water deficit has been found minimum in GA1 model. Next to GA1 model, total water deficit has been found minimum in GA3 model. SDP1 model has been found to keep the reservoir full for maximum number of times. However, performance of other SDP models in this regard has been found more or less same as that of the GA models. DPR and DPN models keep the reservoir full for least number of times. All the models spill more quantity of water than SOP model.

The study has shown that GA and SDP models are competitive. GA outperforms the DPR and DPN models developed in this study. Although performance of different models varies with different performance criteria, considering overall performance and giving priority to irrigation, the operating policy derived by GA3 model has been considered most appropriate for practical application in the Pagladia multipurpose reservoir.

10.3 RECOMMENDATIONS FOR FURTHER STUDIES

Following possibilities are suggested for further extensions of the present works.

(1) In the present study, operating policies have been developed to decide a release to be made in a month on the basis of initial storage and inflow of the current month. However, in real time operation, inflow value of the current month is not known in advance. Therefore, an efficient streamflow forecasting model for Pagladia River at dam site may be developed for better implementation of the

policy developed in this study. ANN may be used to develop good forecast model, as potential of ANN in forecasting river stage or flow has already been established (,e.g., Liong et al. 2000).

(2) In this study optimal monthly operating policy for the Pagladia multipurpose reservoir has been developed based on the available monthly data. Future study may include development of within month short period operating policy for more efficient operation of the Pagladia multipurpose reservoir, particularly for flood management. Daily data at dam site will of course be necessary for this purpose.

(3) Operating policy for the reservoir has been developed using the objective function as minimization of squared deficit of release from demand. This objective function has been considered as the most appropriate one in the present scenario. With the change in the Government policy, and with the change in the socio-economic condition of the people, preference may go towards other objective functions such as maximization of irrigation benefit, maximization of power benefit, and maximization of flood benefit. Derivation of optimal operating policy with these objective functions can also be considered in order to provide more options to the administrators to choose an appropriate operating policy commensurate with the changed scenario.

REFERENCES

Akaike, H. (1978), “On the likelihood of a time series model”, *Statistician*, 27(3-4), 217-236.

Alarcon, L., and Marks, D. H. (1979). “A stochastic dynamic programming model for the operation of High Aswan Dam.” Tech. Rep. 246, Ralph M. Parsons Lab., Dept. of Civil Engg., Mass. Inst. of Technol., Cambridge.

Anderson, O. D. (1976). *Time Series Analysis and Forecasting, the Box-Jenkins Approach*, Butterworth and Co., London.

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

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

Becker, L., Yeh, W., W-G., Fults, D.M., and Sparks, D. (1976). “Operation models for Central Valley projects.” *J. Water Resour. Plan. Manage.*, *ASCE* 102(WR1), 101-115.

Bellman, R. (1957). *Dynamic Programming*. Princeton University Press, Princeton, New Jersey.

Bellman, R., and Dreyfus, S.E. (1962). *Applied Dynamic Programming*. Princeton University Press, Princeton, New Jersey.

Bhaskar, N. R., and Whitlatch, Jr., E.E. (1980). “Derivation of monthly reservoir release policies.” *Water Resour. Res.*, 16(6), 987-993.

Bogle, M.G.V., and O'Sullivan, M.J. (1979). “Stochastic optimization of a water supply system.” *Water Resour. Res.*, 15(4), 778-786.

Box, G.E.P., and G.M. Jenkins(1976), *Time Series Analysis: Forecasting and Control*, Holden-Day, san Francisco, Calif

Braga, B.P.F., Yeh, W.W-G., Becker, L., and Barros, T. L. (1991). "Stochastic optimization of multiple reservoir system operation." *J. Water Resour. Plan. Manage.*, ASCE , 117(4), 471-481.

Brahmaputra Board, Government of India. (1997). *Detail project report, Pagladia dam project,* Brahmaputra Board, Government of India.

Bras, R. L., Buchanan, R.B., and Curry, K.C. (1983). "Real time adaptive closed loop control of reservoirs with the High Aswan Dam as a case study." *Water Resour. Res.*, 19(1), 33-52.

Buras, N. (1985). "An application of mathematical programming in planning surface water storage." *Water Resour. Bull.,* 21(6), 1013-1020.

Burian, J.S., Durrans, S.R., Nix, S.J. and Pitt, R.E.(2001). " Training Artificial Neural Networks to perform rainfall disaggregation", *J. Hydrologic Engg,* ASCE, 6(1), 43-51.

Burn, D. H. and Yulianti, J. S. (2001). "Waste -load allocation using genetic algorithms" *J. Water Resour. Plan. Manage.*, ASCE, 127(2), 121-129.

Butcher, W.S. (1971). "Stochastic dynamic programming for optimum reservoir operation." *Water Resour. Bull.*, 7(1), 115-123.

Cancelliere, A., Giuliano, G., Ancarani, A. and Rossi, G. (2002). "A neural network approach for deriving irrigation reservoir operating rules." *J. Water Resour. Manage.*, 16, 71-88.

Chaleeraktrakoon, C. (1999), "Stochastic procedure for generating seasonal flows", *J. Hydrologic Engg*, ASCE, 4(4),337-343.

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

Chaturvedi, M. C., and Srivastava, D. K. (1981). “Study of a complex water resources system with screening and simulation models.” *Water Resour. Res.*, 8(3), 311-328.

Chaturvedi, M.C. (1987). *Water Resources Systems Planning and Management*. Tata McGraw Hill, New Delhi, India.

Chow, V. T., Maidment, D. R., and Tauxe, G.W. (1975). “Computer time and memory requirements for DP and DDDP in water reservoir system analysis.” *Water Resour. Res.*, 11(5), 621-628.

Das G.(2000), *Hydrology and Soil Conservation Engineering*, Prentice Hall of India, New Delhi.

Datta, B., and Houck, M. H. (1984). “A stochastic optimization model for real time operation of reservoirs using uncertain forecasts.” *Water Resour. Res.*, 20(8) ,1039-1046.

Datta, B., and Burges, S.J. (1984). “Short term, single, multiple purpose reservoir operation : Importance of loss function and forecast errors.” *Water Resour. Res.*, 20(9), 1167-1176.

Deb, K. (1991). “Optimal design of a welded beam structure via genetic algorithms.” AIAA Journal, 29(11), 2013-2015.

Deb, K. (1998). Optimization for engineering design: Algorithms and example, New Delhi, Prentice Hall of India.

Deb, K. (2000). “An efficient constraint handling method for genetic algorithms.” *Comput. Methods Appl. Mech. Engrg., Elsevier,* 186, 311-338.

Deb, K., and Agrawal, R. B. (1995). "Simulated binary crossover for continuous search space." *Complex Systems*, 9, 115-148.

Deb, K. and Goyal, M. (1996). "A combined genetic adaptive search (GeneAS) for engineering Design." *Computer Science and Informatics*, 6(4), 30-45.

Doran, D. G. (1975). "An efficient transition definition for discrete state reservoir analysis : The divided interval technique." *Water Resour. Res.*, 11(6), 867-873.

Dorfman, R. (1962). Mathematical models: The multistructure approach in design of water resources systems. Edited by A. Maass et al., Harvard Univ. Press, Mass., Cambridge.

Druce, D. J. (1990). "Incorporating daily flood control objectives into a monthly stochastic dynamic programming model for a hydroelectric complex." *Water Resour. Res.*, 26(1), 5-11.

Dubrovin, T., Joima, A., Turunen, E. (2002). " Fuzzy model for real time reservoir operation." *J. Water Resour. Plan. Manage.,* ASCE, 128(1), 66-73.

Durbin, J. (1960). "The fitting of time series models." Rev. Internat. Statist. Inst., 28, 233-244.

Esat, V., and Hall, M.J. (1994). "Water resources system optimization using genetic algorithms." *Proc. Ist Int. Conference on Hydroinformatics*, Balkema, Rotterdam, The Netherlands, 225-231.

Esmaeil-Beik, S., and Yu, Y.-S. (1984). "Optimal operation of multipurpose pool of Elk City Lake." *J. Water Resour. Plan. Manage.,* ASCE, 110(WR1), 1-14.

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, ASAE, St. Joseph, Mich.

Forrest, S.(1993). "Genetic algorithms: Principles of natural selection applied to computaion." *Science,* 261(13), 872-878.

Foufoula-Georgiou, E., and Kitanidis, P.K. (1988). "Gradient dynamic programming for stochastic optimal control of multidimensional water resources systems." *Water Resour. Res*., 24(8), 1345-1359.

Fults, D.M., and Hancock, L. F. (1972). "Optimum operations model for Shastra-trinity system." *J. Hydraulic Div.*, ASCE, 98 (HY9), 1497-1514.

Fults, D. M., Hancock, L. F., and Logan, G. R. (1976). "A practical monthly optimum operations model." *J. Water Resour. Plan. Manage.*, ASCE, 102 (WR1), 63-76.

Giles, J.E., and Wunderlich, W.E. (1981). "Weekly multipurpose planning model for TVA reservoir system." *J. Water Resour. Plan. Manage.*, ASCE, 107 (WR2), 495-511.

Goldberg, D. E. (1983). "Computer aided gas-pipeline operation using genetic algorithms and rule learning." *Dissertation abstracts international*, 44, 10, 3174B (University Microfilms No. 8402282).

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning.* Addison-Wesley, Reading, Mass.

Goldberg, D. E., and K. Deb. (1990). ''A Comparative analysis of selection schemes used in genetic algorithms.'' *Foundations of genetic algorithms*, Morgan Kaufman, San Mateo, Calif., 69–93.

Goulter, I.C., and Tai, F. K. (1985). "Practical implications in the use of stochastic dynamic programming for reservoir operation." *Water Resour. Bull.*, 21(1), 65-74.

Hagan, M. T., Demuth, H. B., and Beale, M. (1996). *Neural network design*, PWS/KENT Publishing, Co., Boston.

Hall, W.A., Butcher, W.S., and Esogbue, A. (1968). "Optimization of the operation of a multipurpose reservoir by dynamic programming." *Water Resour. Res.,* 4(3), 471-477.

Hall, W.A., and Shepherd, R.W. (1967). "Optimum operation for planning of a complex water system." *UCLA Engg. report no. 67-56*, Water Research Centre.

Hall, W.A., Tauxe, G.W., and Yeh, W.W-G. (1969). "An alternative procedure for the optimization of operations for planning with multiple river, multiple purpose systems." *Water Resour. Res*., 5(6), 1367-1372.

Haykin, S. (1994). *Neural networks: a comprehensive foundation*. Mac-Millan, New York.

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

Hipel, K. W., Mcbean, E. A., and McLeod, A. I. (1979). " Hydrologic generating model selection." *J. Water Resour. Plan. Manage.*, ASCE, 105(WR2), 223-242.

Hipel, K. W., McLeod, A. I., and Lennox, W. C. (1977). "Advances in Box-Jenkins Modeling. 1. Model construction." *Water Resour. Res.,* 13(3), 567-576.

Holland, J.H. (1992). *Adaptation in Natural and Artificial Systems*. 2nd ed. Mass. Inst. of Tecnol. M.I.T. Press, Cambridge, Mass.

Houck, M.H., and Cohon, J.L. (1978). "Sequential stochastic linear programming models: A proposed method for design and management of multipurpose reservoir system. *Water Resour. Res.,* 14(2), 161-168.

Houck, M. H., and Datta, B. (1981). "Performance evaluation of a stochastic optimization model for reservoir design and management with explicit reliability criterion." *Water Resour. Res.*, 17(4), 827-832.

Howard, R. A. (1960). *Dynamic programming and Markov process*, MIT Press, Cambridge,Mass.

Jacobson, D. and Myne, D. (1970). Differential Dynamic Programming. Academic Press, New York.

Jain, S.K., Yoganarasimhan, G.N., and Seth., S. M. (1992). "A risk based approach for flood control operation of a multipurpose reservoir." *Water Resour. Bull.*, 28(6), 1037-1043.

Jones, L., Willis, R. and Finney, B. (1986). " Water resources systems planning: Differential dynamic programming models." *Proc., Water Forum '86*, ASCE, Reston, VA, 1033-1040.

Jores, E., Liebman, J.C., and Revelle, C. (1971). "Operating rules for joint operation of raw water sources." *Water Resour. Res.*, Vol. 7(2): 225-235.

Karamouz, M. (1990). Bayesian decision theory and fuzzy sets theory in system operation. In : Optimizing the Resources for Water Management, *Proc. of the 17th Annual Water Resources Conference*, ASCE, New York.

Karamouz, M. and Houck, M. H. (1982). "Annual and monthly reservoir operating rules generated by deterministic optimization." *Water Resour. Res.*, 18(5), 1337-1344.

Karamouz, M., and Houck, M.H. (1987). "Comparison of stochastic and deterministic dynamic programming for reservoir operating rule generation." *Water Resour. Bull.*, 23(1),1-8.

Karamouz, M., Houck, M. H., and Delleur, J. W. (1990). "Optimization and simulation of multiple reservoir systems." *J. Water Resour. Plan. Manage.*, ASCE, 118(1), 71-81

Karamouz, M., and Vasiliadis, H.V. (1992). "Bayesian stochastic optimization of reservoir operation using uncertain forecasts." *Water Resour. Res.*, 28(5), 1221-1232.

Karunanithi, N., Grenney, W. J., Whitley, D. and Bovee, K. (1994). "Neural networks for river flow prediction." *J. Computing in Civ. Engg.*, ASCE, 8(2), 201-220.

Kelman, J., Stedinger, J. R., Cooper, L.A., Hsu, E., and Yuan, S. Q. (1990). "Sampling stochastic dynamic programming applied to reservoir operation." *Water Resour. Res.*, 26(3), 447-454.

Klemes, V. (1977). "Discrete representation of storage for stochastic reservoir optimization." *Water Resour. Res.*, 13(1), 149-158.

Kottegoda, N. T. (1980). *Stochastic water Resources Technology*. MacMillan Press, London.

Kuo, J. T., Hsu, N.S., Chu, W., Wan, S., and Lin, Y.J. (1990). "Real time operation of Tanshui river reservoirs." *J. Water Resour. Plan. Manage.*, ASCE, 116(3), 349-361.

Labadie, J. W. (2004). "Optimal operation of multireservoir systems: State-of-the-art review." *J. Water Resour. Plan. Manage.*, ASCE, 130(2), 93-110.

Larson, R.E. (1968). *State incremental dynamic programming*. Elsevier, New York.

Lawrence, A.J., and N.T. Kottegoda(1977), "Stochastic modeling of riverflow time series", *J .R. Stat. Soc., Ser. A*, 140(1), 1-47.

Liong, S. Y., Lim, W. H., and Paudyal, G. N. (2000). "River stage forecasting in Bangladesh: neural network approach." *J. Comput. Civ. Engg.*, ASCE, 14(1), 1-7.

Little, J. D. C. (1955). "The use of storage water in a hydroelectric system*.*" *J. Oper. Res.*, 3(2), 187-197.

Liu, C.S., and Tedrow, A.C. (1973). "Multi-lake river system operation rules." *J. Hydraul. Div.*, ASCE, 99 (HY9), 1369 - 1381.

Ljung, G.M., and Box G.E.P. (1978), "On a measure of lack of fit in time series models." *Biometrika*, 65, 297-303.

Loucks, D.P., (1968). "Computer models for reservoir regulations." *J. Sanit. Eng. Div.*, ASCE, 94 (SA4), 657-669.

Loucks, D.P., 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.

Loucks, D.P., and Falkson, L.M. (1970). "A comparison of some dynamic, linear and policy iteration methods for reservoir operation." *Water Resour. Bull.*, 6(3), 384-400.

Loucks, D. P., Stedinger, J. R., and Haith, D. A. (1981). *Water Resources System Planning and Analysis*. Prentice Hall, Englewood Cliffs, New Jersey.

Maass, A., Hufschmidt, M. M., Dorfman, R., Thomas, Jr., H. A., Marglin, S. A., and Fair, G. M. (1970). *Design of Water Resources Systems*. Harvard University Press, Cambridge.

Marino, M. A., and Loaiciga, H. A. (1985). "Quadratic model for reservoir management: application to the Central Valley Project." *Water Resour. Res.*, 21(5), 631-641

Mujumdar, P.P., and Narulkar, S. (1993). "Optimization models for multireservoir planning and operation." Jal Vigyan Sameeksha, Publication of INC on Hydrology, VIII(1) : 29-52.

Mujumdar, P.P., and Ramesh, T.V.S. (1997). "Real time reservoir operation for irrigation." *Water Resour. Res*., 33(5), 1157-1164.

Mujumdar, P.P., and Ramesh, T.V.S. (1998). "A short term reservoir operation for multicrop irrigation." *Hydrol. Sc. J.*, 43(3), 479-494.

Manne, A.S. (1962). "Product mix alternatives: flood control, electric power and irrigation." *Int. Econ. Rev*., 8(1), 30-54.

Marino, M.A., and Mohammadi, B. (1983). "Reservoir operation by linear and dynamic programming." *J. Water Resour. Plan. Manage.* , ASCE , 109(4), 303-319.

Martin, Q.W. (1983). "Optimal operation of a multiple reservoir systems." *J. Water Resour. Plan. Manage.* , ASCE, 109(1), 58-73.

McLeod, A.I., Hipel, K.W., and Lennox, W.C. (1977), "Advances in Box-Jenkins modeling", *Water Resour. Res.,* 13(3), 567-586.

McLeod, A. I., Hipel, K. W. (1978). "Preservation of the rescaled adjusted range. 1. A reassessment of the Hurst phenomena." *Water Resour. Res.,*14(3), 491-508.

Meier, W. L., Jr., and C. S. Beightler. (1967). "An optimization method for branching multi-stage water resources systems." *Water Resour. Res*., 3(3), 645-652.

Mejia, J. M., I. Rodriguez-Iturbe, and D. R. Dawday (1972). "Streamflow simulation. 2. The broken line process as a potential model for hydrologic simulation." *Water Resources Research,* 8(4), 931-941

Michalewicz, Z. (1996). Genetic algorithms + data structures = evolutionary programs, Springer, New York.

Mohammadi, B., and Marino, M. A. (1984). "Reservoir operation: choice of objective functions." *J. Water Resour. Plan. Manage* , ASCE , 110(1), 15-29.

Montgomery, D. C. and Runger, G. C. (1994), *Applied statistics and probability for engineers*, John Wiley and Sons, Inc.

Moran, P. A. P. (1954). "A probability theory of dams and storage systems." Aust. *J. Appl. Sc.*, 5, 116-124.

Moss, M.E. and M.C. Bryson (1974), "Autocorrelation structure of monthly streamflows", *Water Resou. Res.*, 10(4), 737-744.

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

Nagesh Kumar, D. and Baliarsingh, F. (2003). " Folded dynamic programming for optimal operation of multireservoir system." *J. Water Resour. Manage.*, 17, 337-353.

Nopmongcol, P., and Askew, A. J. (1976). "Multi-level incremental dynamic programming." *Water Resour. Res*., Vol. 12(6), 1291-1297.

Ochoa-Riveria, J.C., Garcia-bartual, R. and Andreu, J. (2002). "Multivariate synthetic streamflow generation using a hybrid model based on artificial neural networks." *Hydrology and Earth System Science*, 6(4), 641-654.

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

Otero, J., Labadie, J., and Haunert, D. (1995). " Optimization of managed runoff to the St. Lucie estuary." *Proc., First Int. Conf.*, W Espey Jr. and P. Combs, eds., Water Resources Engineering Division, ASCE, San Antonio.

Oven-Thomson, K., Alarcon, L., and Marks, D. H. (1982). "Agricultural versus hydropower trade-offs in operation of the High Aswan Dam." *Water Resour. Res.*, 18 (6), 1605-1614.

Ozden, M. (1984). "A binary state DP algorithm for operational problem of multireservoir systems." *Water Resour. Res.*, 20(1), 9-14.

Palmer, R. N., Smith, J.A., Cohon, J.L., and Revelle, C.S. (1982). "Reservoir management in Potornac river basin." *J. Water Resour. Plan. Manage.*, ASCE , 108 (WR1), 47-66.

Parikh, S.C. (1966). "Linear decomposition programming of optimal long range operation of a multipurpose reservoir system." *Rep. ORC 66-68,* Oper. Res. Cent., Univ. of Calif., Berkeley.

Paundyal, G., Shrestha, D. L., and Bogardi, J. J. (1990). "Optimum hydropower system configuration based on operational analysis." *J. Water Resour. Plan. Manage* , ASCE , 116(2), 233-246.

Peng, C. -S., and Buras, N. (2000). "Practical estimation of inflows into multireservoir systems." *J. Water Resour. Plan. Manage.*, ASCE , 126 (5),331-334.

Philbrick Jr., C.R., and Kitanidis, P. K. (1999). "Limitations of deterministic optimization applied to reservoir operations." *J. Water Resour. Plan. Manage*, ASCE , 125(3), 135-142.

Raman, H., and Chandramouli, V. (1996). "Deriving a general operating policy for reservoir using neural network." *J. Water Resour. Plan. Manage, ASCE* , 122(5), 342-347

Raman , H. and Sunilkumar, N. (1995). “ Multivariate modeling of water resources time series using artificial neural networks.” *Hydrolog. Sci. J.*, 40, 145-163.

Rao, S.S. (1996). *Engineering Optimization: Theory and Practice*, New Age International (P) ltd., Publishers, New Delhi.

Ravi Kumar, V., and Venugopal, K. (1998). “Optimal operation of south Indian irrigation systems.” *J. Water Resour. Plan. Manage, ASCE,* 124(5), 264-271.

Revelle, C., Jores, E., and Kirby, W. (1969). “The linear decision rule in reservoir management and design. 1, Development of stochastic model.” *Water Resour. Res.*, 5(4), 767-777.

Reznicek, K.K., and Simonovic, S.P. (1990). “An improved algorithm for hydropower optimization.” *Water Resour. Res.*, 26(2), 189-198.

Roefs, T.G., and Bodin, L.D. (1970). “Multireservoir operation studies.” *Water Resour. Res.*, 6(2), 410-420.

Roefs, T.G., and Guitron, R.A. (1975). “Stochastic reservoir models: Relative computational effort.” *Water Resour. Res.*, 11(6), 801-804.

Rumelhart, D. E., Hinton, G. E., and Williams R. J. (1986). “Learning internal representations by error propagation.” Parallel distributed processing, Vol. 1. MIT Press. Cambridge, Mass., 318-362.

Rumelhart, D.E., and Mcclelland, J. L. (1987). Parallel distributed processing: explorations in the microstructure of cognition. MIT Press, Cambridge, Mass., Vol. I and II.

Russell, S.O., and Campbell, P.F. (1996). “Reservoir operating rules with fuzzy programming.” *J. Water Resour. Plan. Manage*, ASCE, 122 (3), 165-170.

Sharif, M., and Wardlaw, R. (2000). “Multireservoir systems optimization using genetic algorithms: Case study." *J. Comput. Civ. Engg.*, 14(4), 225-263.

Shrestha, B.P., Duckstein, L., and Stakhiv, E.Z. (1996). "Fuzzy rule-based modelling of reservoir operation." *J. Water Resour. Plan. Manage*, ASCE, 122(4), 262-269.

Simonovic, S. P., and Marino, M. A. (1980). "Reliability programming in reservoir management, 1, single multipurpose reservoir." *Water Resour. Res*.,16(5), 844-848.

Simonovic, S. P.(1992). "Reservoir system analysis: Closing gap between theory and practice." *J. Water Resour. Plan. Manage, ASCE*, 118(3), 262-280.

Stedinger, J.R. (1984). "Performance of LDR models for preliminary design and reservoir operation." *Water Resour. Res*., 20(2), 215-224.

Stedinger, J.R. and Taylor, M.R. (1982), "Synthetic streamflow generation 1. model verification and validation", *Water Resour. Res.*, 18(4), 909-918.

Stedinger, J.R., Sule, B.F., and Loucks, D. P. (1984). "Stochastic dynamic programming models for reservoir operation optimization." *Water Resour. Res.*, 20(11), 1499-1505.

Su, S.Y., and Deininger, R.A. (1974). "Modeling the regulation of Lake Superior under uncertainty of future water supplies." *Water Resour. Res*., 10(1), 11-25.

Takeuchi, K., and Morreau, D. H. (1974). "Optimal control of multi-unit inter basin water resources system." *Water Resour. Res*., 10(3), 407-414.

Talukdar, B. (1999). "Operation of a multipurpose reservoir using stochastic dynamic programming." PhD Dissertation, University of Roorkee, Roorke, India.

Tejada-Guibert, J. A. Johnson, Stedinger, J. R. (1993). "Comparison of two approaches for implementing multireservoir operating policies derived using stochastic dynamic programming.” *Water Resour. Res.*, 29(12), 3969-3980.

Tejada-Guibert, J. A., Johnson, S. A., and Stedinger, J. R. (1995). “The value of hydrologic information in stochastic dynamic programming models of a multireservoir system.” *Water Resour. Res*., Vol. 31(10) : 2571-2579.

Tejada-Guibert, J., Stedinger, J., and Staschus, K. (1990). “ Optimization of the value of CVP’s hydropower production.” *J. Water Resour. Plan. Manage.*, ASCE, 117(3), 340-351.

Thomas, H.A., and Fiering M. B. (1962), Mathematical synthesis of streamflow sequences for the analysis of river basins by simulation, in *Design of Water Resource Systems*, edited by A. Maass, M.M. Humfschmidt, R. Dorfman, H.A. Thomas, Jr., S.A. Marglin, and G.M. Fair, Harvard University Press, Cambridge, Mass.

Thomas, H.A., and Watermeyer, P.W. (1962). Mathematical models: a stochastic sequential approach, in *Design of Water Resources Systems*, edited by A. Maass, Harvard Univ. Press, Cambridge, Mass., pp. 540-564.

Torabi, M., and Mobasheri, F. (1973). “A stochastic dynamic programming model for the optimum operation of a multi-purpose reservoir.” *Water Resour. Bull*., 9(6), 1089-1099.

Trezos, T., and Yeh., W. W.-G. (1987). “Use of stochastic dynamic programming for reservoir management.” *Water Resour. Res*., 23(6), 983-996.

Trott, W. M., and Yeh, W. W-G. (1971). “Multi-level optimization of a reservoir system.” Paper presented at the Annual and National Environmental Engineering Meeting, ASCE, St. Louis, Mo., Oct, 18-21.

Trott, W.J., and Yeh, W., W-G. (1973). “Optimization of multiple reservoir system.”

J. Hydraul. Div., ASCE, 99 (HY10) : 1865-1884.

Turgeon, A. (1982). " Incremental dynamic programming may yield non optimal solutions." *Water Resour. Res*., 18(6), 1599-1604.

Vasiliadis, H., and Karamouz, M. (1994). " Demand driven operation of reservoirs using uncertainty-based optimal operating policies." *J. Water Resour. Plan. Manage*, ASCE, 120(1), 101-114.

Vedula, S. and Mujumdar, P.P. (1992). "Optimal reservoir operation for irrigation of multiple crops." *Water Resour. Res.*, 28(1), 1-9.

Vedula, S., and Mohan, S. (1990). "Real-time multipurpose reservoir operation: A case study." *Hydrol. Sc. J.*, 35(4), 447-462.

Vedula, S., and Nagesh Kumar, D. (1996). "An integrated model for optimal reservoir operation for irrigation of multiple crops." *Water Resour. Res*., 32(4), 1101-1108.

Wardlaw, R., and Sharif, M. (1999). "Evaluation of genetic algorithms for optimal reservoir system operation." *J. Water Resour. Plan. Manage*, ASCE, 125(1), 25-33.

Windsor, J. S. (1973). "Optimization model for the operation of flood control systems." *Water Resour. Res*., 9(5), 1219-1226.

Wurbs, R.A. (1993). "Reservoir system simulation and optimization models." *J. Water Resour. Plan. Manage.*, ASCE, 119(4), 455-472.

Wurbs, R. A., Tibbets, M. N., Cabezas, L. M., and Roy, L. C. (1985). " State-of – the-art review and annotated bibliography of systems analysis techniques applied to reservoir operation." *Tech. Rep. 136.* Texas Water Resources Institute, College Station, Tex.

Yakowitz, S. J. (1982). "Dynamic programming applications in water resources."

Water Resour. Res.,. 18(4), 673-696.

Yakowitz, S. J. (1983). " Convergence rate analysis of the state incremental dynamic programming methods." *Automatica*, 19(1), 53-60.

Yang, J. P., and Soh, C. K. (1997). "Structural optimization by genetic algorithms with tournament selection." *J. Comp. in Civ. Engg.*, ASCE, 11(3), 195-200.

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

Young, G. K. (1967). "Finding reservoir operation rules." *J. Hydraul. Div.*, ASCE, 93 (HY6), 297-321.

Young, G.K., and R.U. Jettmar (1976), "Modeling monthly hydrologic persistence." *Water Resources Research*, 12(5), 829-835

Zadeh, L.A. (1965). "Fuzzy sets." *Information and Control*, 8(3), 38-53.

Zurada, J. M. (1999). *Inrtoduction to artificial neural systems.* West Publishing co., St. Paul, Minn.

PUBLICATIONS BASED ON THE RESEARCH WORK
(Published and communicated)

International Journal

Ahmed J. A. and Sarma A.K. (2005), " Genetic Algorithm for Optimal Operating Policy of a Multipurpose Reservoir" *International Journal of Water Resources Management, Springer*, 19(2), 145-161

International Conference

Sarma A.K. and Ahmed J.A. (2004), "Optimal Operating Policy for a Multipurpose Reservoir using Genetic Algorithm" *Proceedings of the Fifteenth IASTED International Conference on Modelling and Simulation*, Marina Del Rey, California, USA, 480-484.

Ahmed J.A. and Sarma A.K. (2004), " Neural Network-based Stochastic Dynamic Programming for Reservoir Operation", *Proceedings of International Conference on Advanced Modelling Techniques for Sustainable Management of Water Resources (AMTSMW)*, Warangal, India, 112-114.

Ahmed J.A. and Sarma A. K. (2002), "Stochastic Model for Synthetic Streamflow Generation" *Proceedings of International Conference on Recent Trends in Probability and Statistics: Theory and Applications*, Guwahati, India.

Communicated to international journal

Ahmed J. A. and Sarma A.K. (2005) "Artificial Neural Network Model for Synthetic streamflow generation". Communicated to *International Journal of Water Resources Management, Springer*.
(Manuscript No. WARM -05-00058)