DISSERTATION

BASIN-WIDE MULTI-RESERVOIR OPERATION USING REINFORCEMENT LEARNING

Submitted by:

Jin-Hee Lee

Department of Civil Engineering

In partial fulfillment of the requirements

for the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Summer 2005

COLORADO STATE UNIVERSITY

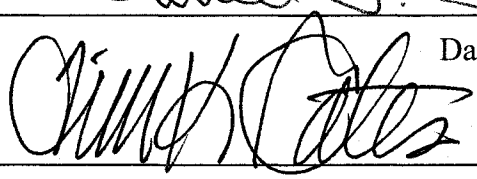
JULY 7, 2005

WE HEREBY RECOMMEND THAT THE DISSERTATION PREPARED UNDER OUR SUPERVISION BY JIN-HEE LEE ENTITLED "BASIN-WIDE MULTI-RESERVOIR OPERATION USING REINFORCEMENT LEARNING" BE ACCEPTED AS FULFILLING IN PART REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY.

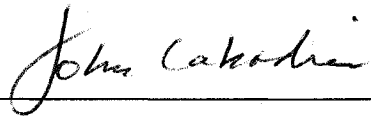
Committee on Graduate Work

Wade O. Troxell

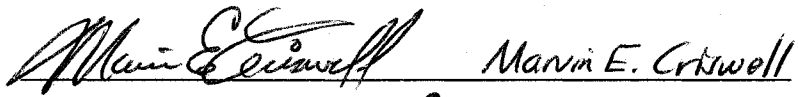
Darrell G. Fontane



Co adviser — T imothy K. Gates



Adviser — John W. Labadie



Department Head — for Sandra L. Woods

ABSTRACT OF DISSERTATION

BASIN-WIDE MULTI-RESERVOIR OPERATION USING REINFORCEMENT LEARNING

The analysis of large-scale water resources systems is often complicated by the presence of multiple reservoirs and diversions, the uncertainty of unregulated inflows and demands, and conflicting objectives. Reinforcement learning is presented herein as a new approach to solving the challenging problem of stochastic optimization of multi-reservoir systems. Conventional stochastic DP models have been applied to limited system representation requiring simplification and approximations that operators are unwilling to accept. The purpose of this study is to establish an optimization framework for realistic and reliable operation of multi-reservoir systems in order to reduce the gap between theoretical investigations and practical implementation.

Reinforcement learning is a simulation-based technique rooted in dynamic programming. One of the reinforcement learning approaches called Q-Learning avoiding requiring prior knowledge of the state transition probabilities in the system by direct use of historical data. The optimal control policy is learned based entirely on feedback mechanisms. In addition, reinforcement learning does not require synthetic streamflow generation and method for inferring rules required in implicit stochastic optimization approaches.

The Keum River basin in Korea was chosen as a case study to demonstrate the applicability of reinforcement learning for basin wide reservoir operation. The Keum River basin consists of 12 sub basins and two major reservoirs, and the operation of this river basin includes water supply, flood control, hydropower generation, and instream

flow requirements. These multiple objectives are combined into a single objective function for the dynamic programming optimization using the weighting method, assuming a unique performance measure for commensurating multi-objectives exists. A detailed simulation procedure for the Keum River basin is developed to accurately reflect the basin characteristics and consider all important component in the basin.

The Q-Learning method is used for generating integrated monthly operation rules for the Keum River basin. The Q-Learning model is evaluated by comparing with implicit stochastic dynamic programming and sampling stochastic dynamic programming approaches. Evaluation of the stochastic basin-wide operational models considered several options relating to the choice of hydrologic state and discount factors as well as various stochastic dynamic programming models. The performance of Q-Learning model outperforms the other models in handling of uncertainty of inflows.

Jin-Hee Lee
Department of Civil Engineering
Colorado State University
Fort Collins, CO 80523
Summer 2005

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and appreciation to my advisor, Dr. John W. Labadie, for providing me with unique opportunity to work in the research. I am very grateful to my co-advisor, Dr. Timothy K. Gates, for giving valuable advices. My deep gratitude also goes to committee members, Dr. Darell G. Fontane and Dr. Wade O. Troxell, for their helpful comment and suggestion.

I would also like to thank Mr. Kee-Uk Cha, Dr. Ick-Hwan Ko, Han-Goo Lee, Dr. Myung-Ki Park, Sang-Keun Park, and Jeong-Yup Kim for their help which made it possible for me to complete case study in the research. My deep gratitude is extended to Prof. Myung-Pil Shim, Prof. Byung-Ha Seo, and Prof. Gye-Woon Choi in Korea who made it possible for me to attend CSU and earn my degree.

Finally, I am grateful to my mother and father who have always been supportive of my graduate study. I am thankful to my friend, Chang-Uk Lim for his friendship and help. I also thank to Dr. Sung-Je Park and Dr. Kyung-Tak Kim who provided encouragement.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figures **Page**

1. Introduction

1.1. General Background and Motivation

As populations expand and economies develop, increasing competition for limited available water resources is occurring among both intrabasin and interbasin users. This has brought greater attention to integrated river basin management, requiring an extended scale of water management without losing model detail and accuracy. Constructing new infrastructure for increasing water supplies is increasingly difficult due to environmental concern clashing with classical development philosophies (1992, Yevjevich). Improving operational effectiveness and efficiency of existing reservoir systems for maximizing the beneficial uses is therefore receiving in increasing attention. Unfortunately, many existing reservoir operational policies do not reflect the need for fully integrated operations of multi-reservoir systems (Labadie, 2004). These realities force engineers to not only devise more precise and powerful tools for water managers and decision makers, but also manage larger-scale systems in a fully integrated manner.

The analysis of large-scale water resources systems is often complicated by the presence of (1) multiple reservoirs and diversions, (2) the uncertainty of unregulated inflows and demands, and (3) conflicting objectives (e.g. flood control vs. conservation purpose). In particular, the uncertainty of inflows makes it impossible to precisely identify future impacts of current decision-making. As a result, the efficient operation of multiple reservoir systems is a difficult and challenging task for water resources managers. Planning and management of multiple reservoir systems applies modeling techniques generally categorized as simulation and optimization methods. Both simulation and optimization techniques can be stochastic or deterministic, depending on

how uncertainties are treated. The deterministic way of handling uncertainties cannot estimate the reliability of the proposed solutions (Koutsoyiannnis et. al, 2002) since multiple reservoir systems are stochastic in most cases. The need for incorporating uncertainties in the planning and operation of multiple reservoir systems is important and necessary.

Simulation techniques are valuable for evaluating system performance under a given set of inputs and operating policies. The vast majority of reservoir planning studies and practical real-time operations are based on simulation techniques due to their versatility and capabilities of more detailed and realistic representations of water resources systems (Lund and Guzman, 1999).

Optimization techniques improve system performance by determining values for sets of decision variables that will map to extreme values of an objective function, subject to limitations, targets, and constraints on system management. These methods provide a means of systematically investigating alternative decisions and policies within a wide range of feasible solutions or scenarios. Optimization techniques used in planning and operations of multi-reservoir system include linear programming (LP; Watkins and McKinney, 1997), dynamic programming (DP; Tai and Goulter, 1987), non-linear programming (NLP; Peng and Buras, 2000), optimal control theory (Mizyed, 1992), and heuristic methods such as evolutionary algorithms (Oliveira and Loucks, 1997).

Obtaining optimal policies for multi-reservoir systems management using simulation is a difficult task because an infinite number of possible operation policies exist. This multi-reservoir system problem can be formulated as an optimization problem to search for the optimal policies automatically. However, this is also a difficult task since the

optimization problem depends on various unknown parameters. There is a trade-off between realistic model formulation of the system structure, which usually affects the quality of the solution, and the computational tractability in stochastic optimization problem. This has resulted in development of a variety stochastic optimization approaches.

In stochastic optimization, uncertainties can be dealt with in either an implicit or explicit manner. The implicit approach (Young, 1967) includes stream-flow synthesis, deterministic optimization, and regression analysis. It is relatively simple to implement and understand. However, the optimal operational policies are unique to the assumed hydrologic time series. Multiple-regression analysis applied to the optimization results for inferring feedback operating rules may result in invalid operation rules (Labadie, 2004). On the other hand, the explicit stochastic optimization approach (Braga Jr. et al., 1991; Tejada-Guibert et al., 1995) directly accounts for the uncertainties using statistical frequency analysis. Optimization is performed without the presumption of perfect foreknowledge of future events and optimal policies are determined without regression analysis or inference method on the optimization results (Labadie, 2004). The implicit and explicit stochastic optimization procedures represented in Figure 2-1 and Figure 2-2, respectively.

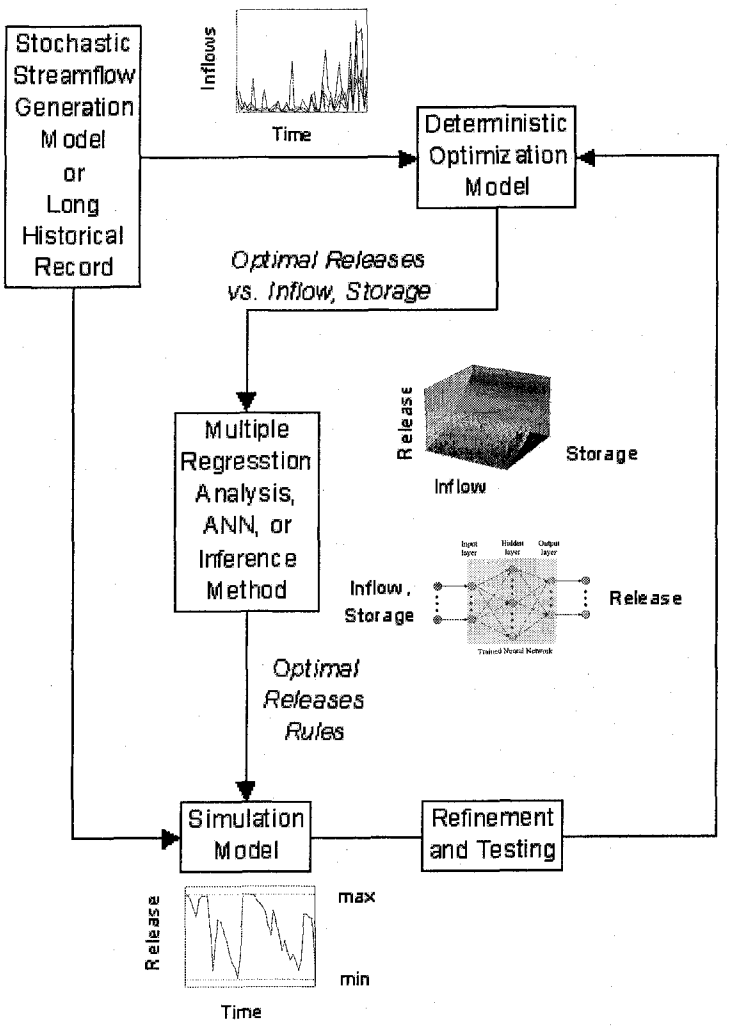


Figure 1.1 Implicit stochastic optimization procedure (Labadie, 2004)

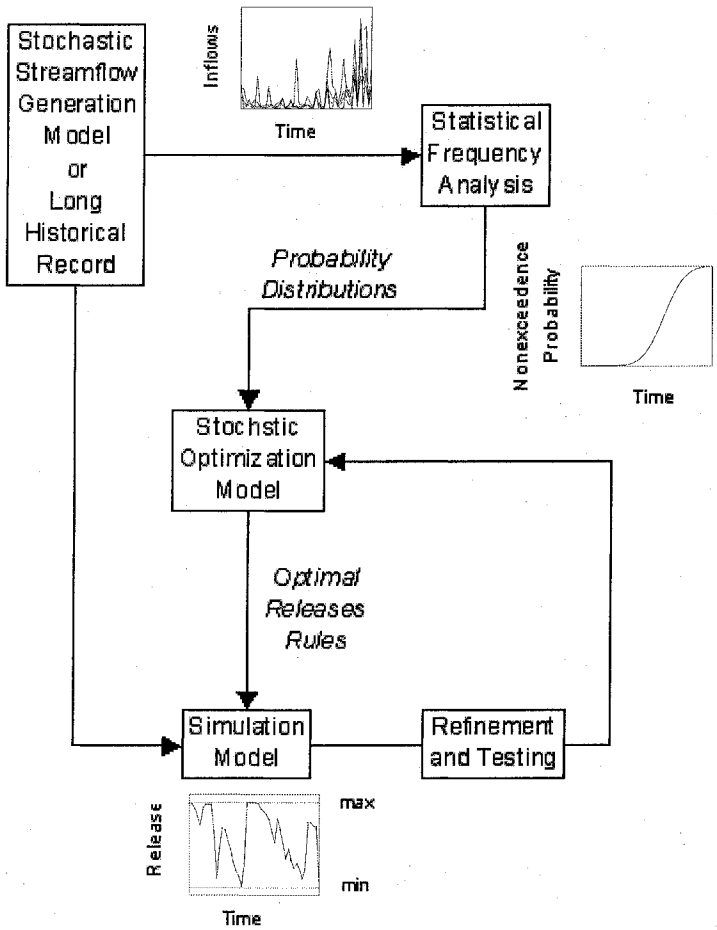


Figure 1.2 Explicit stochastic optimization procedure (Labadie, 2004)

Models using linear programming and dynamic programming have been successfully applied to stochastic reservoir operation. Both linear programming and dynamic programming models have their own advantages and disadvantages. Linear programming methods require all constraints and the objective functions to be linear. Dynamic programming methods have the ability to represent system dynamics and constraints realistically and directly incorporate stochastic inflows easily. In addition, dynamic programming can provide the optimal operating feedback policies for every possible state of the system once they are solved.

One of the most popular explicit stochastic optimization approaches is stochastic dynamic programming (SDP), which is based on the definition of state variables (usually reservoir storage and/or reservoir inflow), decision variables (usually releases from the reservoir), and probability distributions of random inflows. When correlation between successive inflows exists, the probability of inflow is assumed to govern by a simple Markov chain defining the inflow transition probability from current inflows to next period inflows instead assuming inflows are independently distributed. The computational burden associated with SDP is significantly greater than the implicit approach and the identification of transition probabilities is a difficult task when considering multiple inflows to multi-reservoir systems.

Markov chains can be used to describe the storage behavior of the system itself rather than analyzing the statistical structure of inflows as in SDP (Wang and Adams, 1986). Markov Decision Processes (MDP) are based on the Markov property, which assumes that the state of the system or environment embodies the information over the entire past

history of the system. This means the next state and immediate reward (benefit or return) depends only on the current state and action (decision). A multi-reservoir system can be formulated as an MDP by means of finite discretization of reservoir storage volumes and reservoir releases. The MDP is defined with a finite state matrix, a transition probability matrix, and a reward matrix. The transition probability matrix within a finite set of discrete states depends on the decision strategy or policy. A reward matrix is associated with a finite set of possible actions and is determined by feedback from the environment (system). The objective is to maximize the long-term expected reward by determining the policy in a multiple stage decision process. The policy specifies the actions to be taken as a function of the current state.

If a transition probability matrix and a reward matrix defining the MDP are completely known for a finite number of states, classical dynamic programming (DP) can be used to solve the MDP. Unfortunately, the DP algorithm suffers from two practical problems: the curse of dimensionality and information about the underlying MDP. For complex problems such as multi-reservoir system operations, the number of possible states grows exponentially with the number of state and decision variables, making the computational requirements of DP overwhelming. The DP algorithm also requires prior knowledge of the underlying MDP. The state transition probabilities and rewards due to actions should be known a priori. Unfortunately, this prior knowledge is difficult to access due to the complexities of the multi-reservoir system or insufficient data.

A possible method for overcoming the computational challenge of stochastic optimization of multireservoir systems is reinforcement learning. The search space over the range of the possible releases is reduced in reinforcement learning so the algorithm is

faster than SDP, yet it also finds better answers than the standard SDP since it is much easier to incorporate the stochastic nature of the inflows. That is to say, acquiring apriori knowledge of the stochastic structure of inflows in SDP is extremely difficult when complex spatial correlations exist among the system inflows. SDP approaches require access to a multivariate time series model to generate many synthetic inflow sequences to build when the available historical data are not sufficient. Contrast, reinforcement learning approaches do not require this process since they can find good models through a learning process regardless of the complex stochastic structure of the inflows.

Reinforcement learning was originally conceived by behavioral psychologists observing the ability of animals to learn appropriate actions in response to particular stimuli on the basis of associated rewards or punishments. Reinforcement learning can be considered as a computational framework for agents to operate a system and learn the system environment to achieve their long-term performance goals without any external supervisor. In reinforcement learning, the learning of an input-output mapping is performed through continual interaction with the environment on the basis of "punish and reward" to obtain information which can be processed to produce an optimal policy. When prior knowledge of the model (state transition function and reward function) due to insufficient data is not available, the reinforcement learning algorithm acquires the necessary knowledge about the model through learning while solving the optimal decision problem.

Over the past four decades, reinforcement learning has been found extensive use in the field of artificial intelligence (AI) and has been applied to many different domains such as adaptive control, robot navigation, economics, management science, and games. One

of the successful applications of reinforcement learning is that of the board game of 'backgammon' (Tesauro, 1994) known as TD-Gammon. TD-Gammon achieved to play at the level of the world's strongest player in backgammon. The practical applications of reinforcement learning to water resources systems are found in Damas et al., (2000), Damas et al., (2001), and Bhattacharya et al. (2002, 2003). These applications are limited to controls of water supply network system and water pumping system. The application to complex multi-reservoir systems in the study is new.

1.2. Research objective

The primary objective of this research is to investigate the applicability of reinforcement learning to operation of large scale, multi-purpose and multi-reservoir systems in a stochastic environment. The reinforcement learning framework is reviewed and compared to existing stochastic dynamic programming approaches in reservoir operation. This provides the extensive background on reinforcement learning and the standard application procedure of reinforcement learning for multi-reservoir systems.

Despite intensive research on the application of optimization models to reservoir systems, a continuing gap between theoretical developments and real-world implementations exists (Yeh, 1985; Wurbs, 1993). Possible reasons for this disparity are well summarized by Labadie (2004). "(1) many reservoir system operators are skeptical about models purporting to replace their judgment and prescribe solution strategies and feel more comfortable with use of existing simulation models; (2) computer hardware and software limitations in the past have required simplifications and approximations that operators are unwilling to accept; (3) optimization models are generally more

mathematically complex than simulation models, and therefore more difficult to comprehend; (4) many optimization models are not conducive to incorporating risk and uncertainty; (5) the enormous range and varieties of optimization methods create confusion as to which to select for a particular application; (6) some optimization methods, such as dynamic programming, often require customized program development; and (7) many optimization methods can only produce optimal period-of-record solutions rather than more useful conditional operating rules". The second objective of this study is to derive the more realistic and reliable operational rules for a multi-reservoir system to reduce the gap between theoretical developments and real-world implementations.

The problems of multi-reservoir system are multi-objective in nature because they consider two or more non-commensurable and conflicting objectives. This research assumes that a unique performance measure for multi-objective exists and uses it instead comparing tradeoffs between objectives. The reinforcement algorithm is used for determining long-term operational rules for a multi-reservoir system, which can also be applied for short-term or real-time (daily or less) operations.

The Keum River basin in Korea is chosen as a case study to demonstrate the applicability of the reinforcement learning algorithm as a real world problem. A detailed simulation procedure for Keum River basin is developed to accurately reflect the basin characteristics. This procedure is used within the optimization model for development of more precise operation rules. Results are compared with those of other stochastic optimization approaches such as an implicit stochastic dynamic programming and sampling stochastic dynamic programming (Kelmen et. al., 1990) as a demonstration of

the powerful characteristics of reinforcement learning to manage large-scale river basin systems.

1.3. Organization of Dissertation

Chapter 2 reviews optimization methods that can be applied to multi-reservoir systems in a stochastic environment, with chapter 3 providing the key ideas and various algorithms of reinforcement learning. Chapter 4 compares several stochastic dynamic programming models applied to reservoir operation. Chapter 5 describes the operation guidelines of the Keum River basin and the general framework of the modeling. The integrated operation policies obtained by implicit stochastic dynamic programming and sampling stochastic dynamic programming as well as a reinforcement learning are developed and evaluated. Chapter 6 presents the summary and conclusions.

2. Literature Review

2.1. Introduction

Dynamic programming has been used extensively and successfully in optimal operation of multiple reservoir system. The extensive and successful utilization of DP can be due to (1) the ease of handling non-linear formulation; (2) the formulation of stochastic feature by MDP; (3) the calculation of feedback or closed-loop policies; (4) the sequential decision nature of reservoir operation. In the DP models, state variables such as reservoir storage and hydrologic state are usually defined into finite number of states. When the large number of states variable are considered the application of DP suffers from "curse of dimensionality". The computation burden is increased exponentially as the state variables increase. Although the stochastic nature of inflows can be formulated easily in DP the statistical structure of inflows are difficult to obtain. These problems continued to present great challenges in the field of reservoir systems analysis.

Many DP algorithms have been developed in order to handle the stochastic nature of system and to decrease the state dimension. Instead of reviewing all previous works on reservoir operation, this review focuses on several DP algorithms related to these two topics. For more comprehensive and general information on reservoir operations, the following literature are available. Those reviews will provide good understanding and general insight into the application of reservoir operation problem and general idea about stochastic modeling approach.

Yakowitz (1982) reviewed dynamic programming (DP) models for water resource problems and examined computational techniques that have been used to obtain solutions to these problems. Problem areas included reservoir operations analysis related to the

topics in water quality management, water resource project development, irrigation management, and aqueduct design. In this review, various DP techniques such as discrete DP, differential DP, incremental DP, and policy iteration method in discrete DP were discussed and compared in both the deterministic and stochastic environment.

Yeh (1985) extensively reviewed state-of-the-art theories and applications of mathematical models applied to the operation of a reservoir systems. In this review, Yeh considered both deterministic and stochastic models and categorized them as LP, DP, NLP, and simulation. Critical reviews and in-depth analyses were made of each particular method, and merits and limitation of them were assessed. In the conclusion, Yeh recommended that future research in the field of reservoir management and operation should focus on development of stochastic models and solution techniques for the dimensionality problem, consideration of risk or reliability and multi-objectives, and incorporation of optimization and simulation model.

Reznicek and Cheng (1990) presented some ideas about the implementation of uncertainties in reservoir planning models. They reviewed different stochastic programming techniques including stochastic LP, stochastic LP with recourse, chance-constrained LP, stochastic DP, reliability-constrained DP, and reliability programming as an extension of the chance-constrained programming.

Wurbs (1993, 1996) reviewed computer models that have been developed for use in evaluating reservoir operations. Wurbs categorized reservoir-system-analysis models as reservoir-system-simulation models, optimization models, and system-analysis models based on a network-flow programming formulation and compared from a general overview perspective. In this review, Wurbs provided several key factors to be

considered in formulating a modeling and analysis approach for a particular application according to its characteristics.

More recently, Labadie (2004) assessed the state-of-the-art in reservoir system optimization models and considered future direction. In this review, Labadie reviewed a number of solution strategies using LP, NLP, DP, and optimal control theory. These strategies included implicit stochastic optimization, explicit stochastic optimization, real-time optimal control with forecasting, and heuristic programming methods. Finally, Labadie concluded the keys to success in implementation for reservoir system optimization models are involvement of decision makers in system development, and improved linkage with simulation models.

The reviews of stochastic programming and uncertain programming in other field can also be found in the literature. Sen (2001) discussed several LP based stochastic programming methods and computational issues and challenges. Liu (2001) presented a review of the broad classes of uncertain programming including expected value models, chance-constrained programming, dependent-chance programming, and a simulation-based genetic algorithm.

2.2. Deterministic DP and Implicit stochastic optimization

Dynamic Programming (DP) is a useful mathematical technique for solving sequential decision-making problems. The main idea of DP is to decompose the highly complex problem into sub-problems which are solved recursively. DP has been intensively used in optimal operation of multiple reservoir system. It's popularity and success is due to the ease of nonlinear handling and calculation of feedback or closed-policies. However, DP

has a major hindrance to the multiple reservoir system which is well known "curse of dimensionality", a term originally coined by Bellman (1961). This describes the exponential growth in computer time and storage requirement with increase in not only the state–space dimension, but also the number of discretized state and decision variables. The computational burden is increased when hydrologic uncertainty in the models is incorporated explicitly. This results in the use of deterministic DP models where inflows over the entire time assumed to be known. Deterministic models often use historical mean flows, critical period flows, and forecasted flows as well as historical flows.

Bellman and Dreyfus (1962) introduced the dynamic programming successive approximations (DPSA) and then Larson (1968) generalized it. Instead of considering all possible combinations of the state vector, DPSA only optimizes over one component at a time until the original problem converges. Unfortunately, convergence to local optima is only guaranteed for special cases of convex problems (Korsak and Larson, 1970). Collins (1977) presented the "two at a time" DP approach which considers the reservoirs in overlapping pairs. The practical application to multiple reservoir systems is found in several studies in Trott and Yeh (1971), Yeh and Trott (1972), Giles and Wunderlick (1981), and Yi et al. (1997).

Larson (1968) first proposed incremental dynamic programming (IDP), also known as discrete differential dynamic programming (DDDP). In IDP or DDDP, initial feasible solution trajectories for system states are determined along with narrow corridors defining the trajectory boundaries. The problem is solved successively until convergence to a single trajectory is reached. Convergence to global optimum is not guaranteed, although at least local optimum is assured. The IDP or DDDP approach is sensitive to the

dicretization interval of state variables and initial feasible trajectory. The reservoir operation studies using IDP is in the literature by Trott and Yeh (1971), and Heidari et al. (1971).

Nonmongcol and Askew (1976) proposed the combination approach of "two at a time" DPSA and IDP or DDDP which initially solves the problem with DPSA and refines or confirm the solution with IDP or DDDP. The approach is adapted in the generalized dynamic programming software package CSUDP (Labadie, 2003).

To address the stochastic problem in reservoir operations, Young (1967) introduced implicit stochastic optimization, which consists of stream-flow synthesis, deterministic optimization, and regression analysis. First, a large number of equally probable sequences of inflow are synthetically generated if a long continuous series of historical data is not available. Next, these inflow sequences are optimized through any deterministic dynamic programming method resulting in optimal release trajectories. Finally, a multiple linear or nonlinear regression analysis is used to determine a final optimal reservoir feedback operating policy based on the state variables.

Schweig and Cole (1968) consider the control rules for a linked two reservoir systems for allocating water to meet a common demand. The objective function minimizes the total long-term cost of transmission of water from one reservoir to the other and shortage of water supply. The inflows are treated as random variables in the study and the operation rules are defined by multivariate analysis of the deterministic optimization results. Roefs and Bodin (1970) use implicit stochastic DP technique to define operational rules for a simplified three reservoir system.

Bhaskar and Whitalatch (1980) derived both linear and nonlinear release rules for a single reservoir using implicit stochastic DP. Performance of these policies was verified and compared through simulation, indicating that the simple linear policies were better than the nonlinear policies. Willis et al. (1984) derive the probabilistic reservoir release policy for a single reservoir system by using Monte Carlo optimization. They used reservoir storage and inflow as the observed conditioning hydrologic variables.

Karamouz et al. (1992) extended an implicit stochastic optimization to consider a multiple reservoir system. The derived operation rules were evaluated in a simulation model using different sizes of reservoirs and various sets of release boundaries conditions. A fully dynamic network flow model HEC-PRM was applied to the main stem of Missouri River reservoir system in implicit manner (Lund and Ferreira, 1996).

Neural networks (NN) can be used to derive the functional relationship between observable hydrologic states and release policies. The applications of deterministic DP and NN were found in Raman and Chandramouli (1996) for a single reservoir system and (Chandramouli and Raman (2001) for multiple reservoir system.

The implicit stochastic optimization has several advantages over explicit stochastic methods. First, it is relatively simple to implement and understand. Second, it can reduce the computational effort and uses deterministic optimization. Third, it doesn't need to build a stochastic model. However, as Labadie (2004) indicated regression analysis may result in poor correlations that invalidate the operating rules.

2.3. Explicit stochastic optimization

The deterministic dynamic programming assumes perfect knowledge of all input parameters, which may result in overestimated system benefits and underestimated costs and losses. The need for direct consideration of the uncertainty has led to many researches for applying several variants of stochastic dynamic programming. The classification of these techniques can be made by how to handle the characteristics of stochastic inflow. Various available stochastic dynamic programming techniques in the field of reservoir system analysis are reviewed in here according to the consideration of uncertainties associated with reservoir management and operations.

2.3.1. Stochastic Dynamic Programming

Little (1955) introduced a stochastic form of dynamic programming for optimal hydropower scheduling at Grand Coulee reservoir on the Columbia River. States of the system were defined as the current water levels and inflows in the previous period using the Markov property. On the other hand, Gesford and Karlin (1958) simply used current reservoir levels as the states under the independent assumptions for the inflows and derived the optimal reservoir operational policies. Russell (1972) extended the work of Gesford and Karlin to multi-purpose reservoir operation and derived some analytic properties of an optimal solution.

Buras (1965) applied SDP to a serial two-reservoir system in series for irrigation water supply. The states are defined as the current reservoir levels and the current flow between them. Butcher (1971) adopted Little's stochastic DP approach to a single multi-purpose reservoir case. The states are defined as the current reservoir levels and the current flow

between reservoirs. Arunkumar and Chon (1978) modified the work of Buras allowing random inflows into both reservoirs. Releases from the higher level reservoir to the lower level one is not included as a state.

General solution techniques such as value iteration and policy iteration by Howard (1960) have been developed for obtaining solutions to stationary Markov processes. He introduced the rewards concept corresponding to the probability transition matrix in the Markov process. The value iteration updates the value of cost-to-go function to yield a better value using Bellman's Principle of Optimality. Cost-to-go function is defined as value of all costs from next period to the time horizon. Policy iteration continually improves a policy using cost-to-go function to yield a better policy until it finds the optimal policy. White (1963) and Mawer and Thorn (1974) have considered a variation of Howard's (1960) policy iteration procedure to avoid the computational difficulty in applying policy iteration.

Yakowitz (1982) assumed that policy iteration was not suitable for solution of large-scale stochastic optimal control problem such as reservoir system operation due to the slow convergence to the optimal policy. Instead, the value iteration method has been widely used in those problems since it more directly represents Bellman's optimality theory. However, Wang and Adams (1986) mentioned the policy iteration procedure still got the potential advantage for reservoir system problems if a more efficient algorithm developed for policy iteration procedure.

SDP for multiple reservoir system also suffers from "curse of dimensionality" because discretization of inflows as well as discretization of states is required for application of SDP. This has motivated research efforts to reduce the number of state variables and the

number of discrete values for each state variable, and adapt more efficient algorithm for solving SDP.

Turgeon (1980) proposed an aggregation-decomposition method in which broke up the n-state variable stochastic optimization problem into n stochastic optimization subproblems of two-state variables. He applied this method to a network of six reservoir-hydroplant complexes to determine the weekly operating policy. Turgeon and Charbonneau (1998) derived the monthly operating policy of Hydro-Quebec's 26 large reservoirs using the same method. The marginal energy production cost in the objective function was considered instead of fixed production cost.

Valdes et al. (1992) suggested a space-time aggregation-disaggregation procedure for the daily real-time operation of a multi-reservoir system. Saad et al (1994) proposed nonlinear disaggregation technique for the operation of long-term multiple hydropower reservoir systems. Implicit stochastic dynamic programming approach and a neural network were used to disaggregate the optimal policy derived by SDP.

Archibald et al. (1997) presented a variant of the aggregation-decomposition method that allows solution time increases linearly with the number of reservoirs in the system and practical for larger reservoir system. Archibald et al. (1999) compared the computational comparison of nested Benders decomposition and DP decomposition approach for hydropower generation multiple reservoir cases. Although aggregation-disaggregation methods reduce the computational cost the detail of the original problem can be lost and disaggregation of aggregate policy may result in unrealistic operation policies.

Saad and Turgeon (1988) proposed a principal component SDP for multi-reservoir hydoropower system. Major components of the system state were determined by implicit stochastic optimization. The major components of the system state were used for stochastic multiresevoir hydropower system operation optimization. Saad et al. (1992) improved this approach by employing censored-data statistical methods for the better estimation of the storage's covariance matrix. Although principal component SDP is practical approach the main disadvantage lies in the assumption that any relationships among the system states are linear, or at least that any non-linear contribution is small

Coarser grid approach with more accurate approximation of the cost-to-go function can reduce the computation burden. Instead of using nearest-neighbor interpolation to cost-to-go function, more accurate higher order interpolation approaches have been developed (Foufoula-Georgiou and Kitanidis, 1988; Foufoula-Georgiou, 1991; Johnson et al., 1993; Chen et al., 1999, Philbrick Jr. and Kitanidis, 2001). On the other hand Gal (1979) proposed parameters iteration method, which uses predefined cost-to-go function. However, Doran (1975) warned that coarse grid approach might give incorrect and even nonsense answer, and results in the sub-optimal reservoir operation policies.

Stochastic dual DP was proposed by Pereira and Pinto (1985, 1991), which is based on SDP and Benders decomposition (Benders, 1962). The dimensionality problem was avoided by approximating the expected cost-to-go function obtained from dual solution of the optimization problem at each stage. The expected cost-to-function were estimated by a piecewise linear interpolation. This approach applied to 37 reservoirs system (Pereira and Pinto, 1985) and 39 reservoirs system in Brazil (Pereira and Pinto, 1991).

However, stochastic dual DP used deterministic equivalent methods which assume that the multiple inflow sequences are known along the whole planning period.

2.3.2. Forecast-Based Approach

Stedinger et al. (1984) suggest the use of best inflow forecast of the current period as a hydrologic state variable such as snowmelt model. Multiple inflow scenario based solution approaches were proposed by Grygier and Stedinger (1985). Kelman et al. (1990) develop sampling stochastic dynamic programming (SSDP). Unlike SDP, the stochastic structure of stream-flows process is not explicitly considered in SSDP. Instead, the features of the process are implicitly captured with a large number of stream-flow sequences that are assumed to be the realizations of the annual stream-flow process. The stream-flow sequences observed or stochastically generated are called stream-flow scenarios. SSDP requires the transition probability of the remainder of scenario starting in the next period given the current scenario instead the Markov transition probabilities. Recently, Faber and Stedinger (2001) used ensemble snowmelt-season streamflow prediction (ESP) by the National Weather Service to develop a series of likely streamflow realizations. They compared the SSDP model's performance with ESP forecasts to its performance when using historical stream-flow time series and recommended the combined SSDP/ESP algorithm for a real-time reservoir operation.

Karamouz and Vasiliadis (1992) proposed a model, called Bayesian stochastic dynamic programming (BSDP), which revised the classical SDP in order to capture the uncertainty of the forecast. The forecast uncertainty is captured by the conditional probabilities of forecasts with discrete lag 1 Markov process and Bayesian decision

theory (BDT). Three state variables are defined including reservoir storage at the beginning of the time period, reservoir inflow during the period, and inflow forecasts for the next period. Kim and Palmer (1997) replaced one period ahead inflow forecasts with seasonal flow forecasts using a snowmelt runoff-forecasting model as a state variable.

The performance of the forecast based stochastic model can be highly depends on how to handle the degree of uncertainty in future flow conditions. The forecast based stochastic models are also necessary to derive conditional probabilities of scenarios for SSDP and prior and posterior transition probabilities for BSDP. Unfortunately, deriving conditional probabilities or prior and posterior transition probabilities are difficult when many inflows are presence in reservoir system.

2.3.3. Chance-constraint DP and Stochastic Differential DP

Chance-constrained dynamic programming explicitly accounts for uncertainty by replacing the constraints having uncertain parameters with their deterministic equivalent forms using probability of violation of stochastic constraints. Askew (1974a, 1974b) first presented the chance-constraint dynamic programming model to overcome the lack of control over system failure associated with optimum operating policies. The objective functions included penalty term for system failure defined as when a certain target release cannot be satisfied. Sneidovich and David (1975) in their comments on Askew's paper (1974b) suggested the introduction of additional state variables representing the expected number of system failures up to the current stage. Askew (1975) examined variations on a penalty for failure and a risk premium added to the discount rate and recommended the use of the risk premium.

Rossman (1977) developed a reliability-constrained dynamic programming model for determining randomized reservoir release rules within the context of the Lagrangian duality theory of nonlinear programming. However, all of the chance-constrained or reliability-constrained models were only used in annual operation planning for single reservoir and the application of these models were limited to one or two reliability constraints (Askew, 1978). In addition, these models required prior knowledge of reliability associated with the system performance such as defining a level of system failure.

Stochastic differential dynamic programming (SDDP) is an SDP approach without discretization of the state vector in order to avoid the curse of dimensionality. Jacobson and Mayne (1970) originally presented differential dynamic programming (DDP). Instead of discretizing the state vector, DDP transforms the DP problem into a dynamic-quadratic performance problem, which can be solved by numerical quadratic programming techniques. Murray and Yakowitz (1979) proposed a linear constrained differential dynamic programming formulation and applied to multi-reservoirs problem. Gjelsvic (1982) applied DDP to the operation of parallel multi-reservoir hydropower systems with random inflows.

Trezos and Yeh (1989) extended DDP to consider stochastic inflows to general multi-reservoir hydropower systems. El-Awar et al. (1998) modified Trezos and Yeh's approach (1989) to determine optimal feedback release policies for both four reservoir hydropower system operations with unconditional. He also tested the SDDP model for single reservoir with multi-lag inflow. However, the primary disadvantage of SDDP is that convergence to global optima is not guaranteed except for convex programming

problems (El-awar et. al, 1998). In addition, the nonlinear structure is ignored in operation rules since SDDP formulation utilizes a linear feedback control law.

3. Reinforcement Learning

3.1. Introduction

Reinforcement learning was originally conceived by behavioral psychologists observing the ability of animals to learn appropriate actions in response to particular stimuli on the basis of associated rewards or punishments. An animal discovers the link between a specific stimulus or action and a reward (or punishment), and then chooses the preferred action on the basis of the reward and punishment. This learning process is called reinforcement learning. It is different from supervised learning because animals are not told explicitly what action they have to take in a particular situation. Instead, they need to find out what action to take for themselves according to their reinforcement that can be strengthened by an appropriate action resulting in a satisfactory or improved state of affairs.

In the field of artificial intelligence, reinforcement learning (Kaelbling et al., 1996) represents a framework for a computational learning agent to learn behavior through trial-and-error interactions with an environment in order to improve its rewards over time. Sutton and Barto (1998) described reinforcement learning as, "Reinforcement learning is learning what to do - how to map situations to actions---so as to maximize a numerical reward signal. The learner is not told which actions to take, as in most forms of machine learning, but instead must discover which actions yield the most reward by trying them. In the most interesting and challenging cases, actions may affect not only the immediate reward, but also the next situation and, through that all subsequent rewards. These two characteristics - trial-and-error search and delayed reward---are the two most important distinguishing features of reinforcement learning."

The first application of reinforcement learning was the checker player developed by Samuel (1959) which is recognized as a significant achievement in AI and machine learning. Over the past four decades, reinforcement learning has found extensive use in the field of artificial intelligence (AI) and been applied to many different domains such as adaptive control, robot navigation, economics and management science, and games. The remainder of this chapter focuses on reinforcement learning algorithms and the theory behind them. This summary is based on the reinforcement learning and the neuro-dynamic programming literature (Sutton and Barto (1998), Kaelbling et al. (1996), and Bertsekas and Tsitsiklis, 1996).

3.2. Reinforcement learning system and Markov decision Process

A reinforcement learning system consists of the agent, environment, and their interactions. The learner or decision maker is called the agent and everything except the agent is called the environment. The agent is connected to its environment via action, reward, and state. Reinforcement learning differs from the more popular supervised learning in several aspects. First, reinforcement learning system is not provided with expected output for a given input. Instead, the agent is provided the immediate reward and subsequent state for an action taken, but is not informed which action would have resulted in optimal long-term rewards. It is necessary for the agent to gather useful experience in order to act optimally. Second, the agent initially does not know anything about a model of environment and is usually unable to observe all aspects of the environment. That is reason why evaluation of the system is often concurrent with learning in reinforcement learning. It explores its environment as much as possible and

takes into account not only previous data, but also current sensed data. Exploration is also necessary to discover such an actions that has not selected before in order to make better action selections in the future. However, the agent also needs to exploit some information by taking actions that has tried in the past and found to be effective in producing reward in order to obtain a lot of reward. This indicates that reinforcement learning is formulated in terms of the problem, not the solution. Figure 3.1 depicts the components of a reinforcement learning system and the agent-environment interaction.

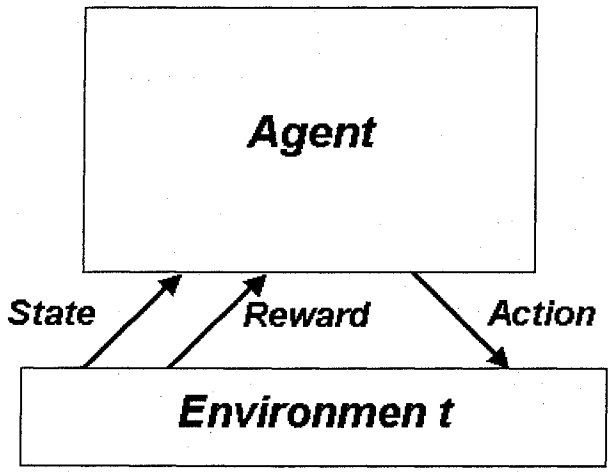


Figure 3. 1 Reinforcement Learning System

Reinforcement learning generally consists of (1) a finite number of state $S = \{s_1, s_2, \cdots, s_n\}$, (2) a finite set of actions $A = \{a_1, a_2, \cdots, a_n\}$ available to an agent (3) a reward given by the environment to the agent and (4) a state transition probability which determines the probability that the environment will make a transition to one state to another when the agent performs an action *a*. At each stage *t*, the environment is represented by some state $s_t \in S = \{s_1, s_2, \cdots, s_n\}$ which summarizes the entire past experience of the learning system gained from its interaction with the environment. The agent selects and performs an action a_t from the available action set $A = \{a_1, a_2, \cdots, a_n\}$.

One time step later, the agent receives an immediate reward $r_{t+1}(s_t, s_{t+1}, a_t) \rightarrow \Re$, and the environment transitions to a new state s_{t+1} as a result of action a_t in state s_t. The agent evaluates its performance by the immediate reward that the environment provides. Since the environment is stochastic and uncertain, taking the same action in the same state on different occasions may result in different new states and/or different reward values (Kaelbling et al., 1996). Therefore, the agent is supposed to find a policy $\pi_t(s,a): S \times A \rightarrow [0,1]$, mapping from the state to probabilities of selecting each possible action. A policy is denoted by $\pi_t(s,a)$ which is the probability of taking action a in state s. However, it can be assumed that the environment is stationary for simplicity, so the probabilities of making state transitions or the immediate rewards do not change over time. As a result, the objective of the agent in stationary case is to determine a deterministic policy $\pi(s): S \rightarrow A$, mapping from the state to action.

It is important for the agent to decide how to take the future into account in the decision and three models have been suggested for this problem. The finite-horizon model optimizes expected reward over the next T time steps when the agent-environment interaction breaks naturally into subsequences. It is assumed there is no consider what will happen after T time steps. The expected reward after time step t in the finite-horizon model is as follows.

$$R_t = E(\sum_{k=0}^{T} r_{t+k+1}) \tag{3.1}$$

In many cases, when the agent-environment interaction does not break naturally into identifiable episodes, the infinite-horizon discounted model takes the long-term reward of the agent into account with a discount factor γ $(0 \leq \gamma \leq 1)$. Here, the discount factor γ can be interpreted as an interest rate, a probability of living another step, or as a mathematical trick to bound the infinite sum (Kaelbling et al., 1996).

$$R_t = E(\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}) \tag{3.2}$$

The average-reward model optimizes its long-term average reward, which can be seen as the limiting case of the infinite-horizon discounted model as the discount factor approaches 1. This model is referred to as a gain optimal policy model (Bertsekas, 1995).

$$R_t = \lim_{T \to \infty} E(\frac{1}{T} \sum_{t=0}^{T} r_t) \tag{3.3}$$

One problem with this model is there is no way to distinguish between one policy with a large amount of reward in the initial phases and another with large amount in the later stages when their long-term average-reward are the same (Kaelbling et al., 1996).

In general, the response of the environment at the next time step $t+1$ depends on everything that has happened earlier, which can be formulated as follows;

$$\Pr\{s_{t+1} = s', r_{t+1} = r \mid s_t, a_t, r_t, s_{t-1} a_{t-1}, \cdots, s_0, a_0, r_0\} \quad \forall\ s', s_t \in S,\ a_t \in A(s_t) \tag{3.4}$$

On the other hand, if the response of the environment has the Markovian property, then the response of the environment at the next time step $t+1$ depends only on the state and action at the current time step t, which can be formulated as follows;

$$\Pr\{s_{t+1} = s', r_{t+1} = r \mid s_t, a_t\} \quad \forall\ s', s_t \in S,\ a_t \in A(s_t) \tag{3.5}$$

Reinforcement learning systems assume that the system dynamics displays the Markovian property (Sutton and Barto, 1998). When given any state s and action a, the dynamics of a learning system can be characterized by the probability of each possible next state s' and the expected value of the immediate reward described in equations (3.6) and (3.7). The objective of agents in a reinforcement learning system is to choose the optimal actions for the given state. The decision process for doing this is referred to as a Markov decision process (MDP). The state transition probability and immediate reward are defined as follows when the agent performs an action a_t in a given state s_t:

$$P^{a_t}_{s_t s_t'} = \Pr\{s_{t+1} = s' \mid s_t = s, a_t = a\} \quad \forall\ s', s_t \in S,\ a_t \in A(s_t) \tag{3.6}$$

$$R^{a_t}_{s_t s_t'} = E\{r_{t+1} \mid s_t = s, a_t = a, s_{t+1} = s'\} \quad \forall\ s', s_t \in S,\ a_t \in A(s_t) \tag{3.7}$$

3.3. Value functions and optimal policy for an MDP

Value functions, either as functions of states or state-action pairs, play an essential role in reinforcement learning algorithms. Value functions estimating how good it is to be in a given state are called state value functions. The state value function for policy π is defined as follows:

$$V^{\pi}(s) = E_{\pi}\{R_t \mid s_t = s\} = E_{\pi}\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s\} \tag{3.8}$$

Value functions estimating how good it is to take a given action a in a given state s are called state-action value function. The state-action value function for policy π is defined as follows:

$$Q^{\pi}(s,a) = E_{\pi}\{R_t \mid s_t = s, a_t = a\} = E_{\pi}\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a\} \tag{3.9}$$

The relationship between the state value function $V^{\pi}(s)$ and the state-action value function $Q^{\pi}(s,a)$ is defined as follows:

$$V^{\pi}(s) = \sum_{a} \pi(s,a) Q^{\pi}(s,a) \tag{3.10}$$

A fundamental property of the value functions is that of recursion. From equation (3.8),

$$V^{\pi}(s) = E_{\pi}\{R_t \mid s_t = s\} \tag{3.11}$$

$$= E_{\pi}\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s\} \tag{3.12}$$

$$= E_{\pi}\{r_{t+1} + \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \mid s_t = s\} \tag{3.13}$$

$$= \sum_a \pi(s,a) \sum_{s'} P^a_{ss'} \left[R^a_{ss'} + \gamma \, E_\pi \left(\sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \mid s_{t+1} = s' \right) \right] \tag{3.14}$$

$$= \sum_a \pi(s,a) \sum_{s'} P^a_{ss'} \left[R^a_{ss'} + \gamma \, V^\pi(s') \right] \tag{3.15}$$

Equation (3.15) represents the relationship between the value of a state and that of subsequent states.

The goal of reinforcement learning algorithm is to find a policy that achieves optimal long-term rewards. For finite MDPs, value functions are useful since they provide a partial ordering over policies. A policy π is defined to be better than a policy π' if its expected reward is greater than that of π' for all states. The expected reward of optimal policy π^* is greater than all other available policies. The optimal state value function $V^*(s)$ is defined as follows.

$$V^*(s) = \max_\pi V^\pi(s) \quad \forall \; s \in S \tag{3.16}$$

The optimal action-value function $Q^*(s,a)$ is defined as follows.

$$Q^*(s,a) = \max_\pi Q^\pi(s,a) \quad \forall \; s \in S, \; a \in A(s) \tag{3.17}$$

This represents the expected return of action a in state s, assuming as optimal policy is followed thereafter. Therefore, the optimal action-value function $Q^*(s,a)$ can be defined as follows in terms of $V^*(s)$:

$$Q^*(s,a) = E(r_{t+1} + \gamma V^*(s_{t+1}) \mid s_t = s, a_t = a) \tag{3.18}$$

The optimal state-value function and the optimal action-value function satisfy the Bellman's optimality principle (Bellman, 1957). Bellman's optimality equation for $V^*(s)$ is defined as

$$V^*(s) = \max_a Q^{\pi^*}(s,a) \tag{3.19}$$

$$= \max_a E_{\pi^*}\left(R_t \mid s_t = s,\ a_t = a\right) \tag{3.20}$$

$$= \max_{a \in A(s)} E_{\pi^*}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s,\ a_t = a\right\} \tag{3.21}$$

$$= \max_{a \in A(s)} E_{\pi^*}\left\{r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \mid s_t = s,\ a_t = a\right\} \tag{3.22}$$

$$= \max_{a \in A(s)} E_{\pi^*}\left(r_{t+1} + \gamma\ V^*(s_{t+1}) \mid s_t = s,\ a_t = a\right) \tag{3.23}$$

$$= \max_{a \in A(s)} E_{\pi^*} \sum_{s'} P^a_{ss'}\left[R^a_{ss'} + \gamma\ V^*(s')\right] \tag{3.24}$$

and Bellman's optimality equation for $Q^*(s,a)$ is defined as

$$Q^{\pi^*}(s,a) = E\left(r_{t+1} + \gamma\ \max_{a'} Q^*(s_{t+1}, a') \mid s_t = s,\ a_t = a\right) \tag{3.25}$$

$$= \sum_{s'} P^a_{ss'}\left[R^a_{ss'} + \gamma\ \max_{a'} Q(s', a')\right] \tag{3.26}$$

A deterministic optimal policy $\pi(s)$ mapping from states to actions can be easily obtained by one-step look-ahead search when the optimal value function $V^*(s)$, including immediate reward function and state transition probabilities, is available. The optimal policy π^* is specified as.

$$\pi(s) = \pi^*(s,a) = \arg\max_{a \in A(s)} \sum_{s'} P^a_{ss'} \left[R^a_{ss'} + \gamma V^*(s') \right] \tag{3.27}$$

Using $Q^\pi(s,a)$ makes it much easier to obtain the optimal policy by assigning zero probabilities to non-optimal policies and non-zero probabilities to the optimal policies:

$$\begin{aligned} &\pi^*(s,a) = 0 \quad \forall \ a \ \text{ such that } Q^*(s,a) \neq \max_{a'} Q^*(s,a') \\ &\therefore \pi(s) = \pi^*(s,a) \neq 0 \quad \forall \ a \ \text{ such that } Q^*(s,a) = \max_{a'} Q^*(s,a') \end{aligned} \tag{3.28}$$

Dynamic programming (DP) methods can compute optimal policies for MDPs, given perfect model of the environment. One approach to find optimal policy is to find the optimal value function It can be determined by a simple iterative algorithm called value iteration. Another approach is policy iteration, which manipulates the policy directly rather than finding it indirectly via the optimal value function. In practice, value iteration is considerably faster per iteration, but policy iteration requires fewer iterations (Kaelbling et al., 1996). These two algorithms are defined by pseudo-code as in Figure 3.2 and 3.3. Figure 3.4 graphically compares the update procedures for the value functions $V^*(s)$ and $Q^*(s,a)$ in a system with two states and three actions. The black

nodes represent situations where the agent has chosen an action and the white nodes represent states where the agent has not yet chosen an action.

Initialize $V(s)$ arbitrarily, eg. $V(s) = 0$, for all $s \in S$
Loop until converged to optimal value function V^*
  Loop for $s \in S$
    Loop for $a \in A(s)$
      $Q(s,a) := \sum_{s'} P^a_{ss'} \left[R^a_{ss'} + \gamma V(s') \right]$
    Loop end
    $V(s) := \max_a Q(s,a)$
  Loop end
Loop end

Figure 3. 2 Value iteration algorithm

Initialize $V^\pi(s)$ arbitrarily, eg. $V^\pi(s) = 0$, for $\forall s \in S$
Choose an arbitrary policy $\pi'(s)$, for $\forall$ s $\in$ S
  Loop (until policy does not change: $\pi(s) = \pi'(s)$)
    $\pi(s) := \pi'(s)$
    evaluate the value function of policy $\pi(s)$:
      $V^\pi(s) = \sum_{s'} P^a_{ss'} \left[R^a_{ss'} + \gamma V^\pi(s') \right]$
    improve the policy at each stae
      $\pi'(s) := \arg\max_a \sum_{s'} P^a_{ss'} \left[R^a_{ss'} + \gamma V(s') \right]$
  Loop end

Figure 3. 3 Policy iteration algorithm

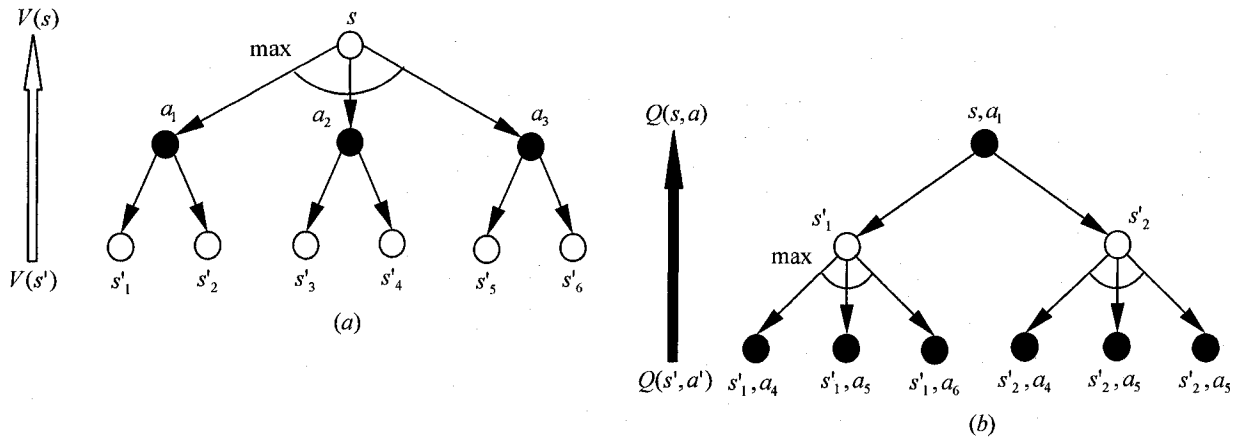


Figure 3.4 Value function update diagrams for a) $V^*(s)$ and b) $Q^*(s,a)$

3.4. Reinforcement learning algorithm

In the previous section, the value iteration and policy iteration algorithms were introduced to obtain an optimal policy for an MDP assuming that the model of environment such as transition probability is available. Various reinforcement learning algorithms are now considered for solving an MDP problem when the model of environment is not available. To obtain an optimal policy, the agent must interact with its environment directly to obtain information and knowledge about model. Barto et al. (1995), Bertsekas and Tsitsiklis (1996), and Sutton and Barto (1998) discuss various algorithms for solving for MDP in computer science and the control system engineering fields. These methods are variously known as real-time dynamic programming (RTDP), neuro-dynamic programming, or reinforcement learning. Broadly speaking these can be referred to as reinforcement learning, which is a simulation-based technique rooted in DP. Reinforcement learning can be divided into two main categories: model-based (or indirect) and model-free (or direct) approaches.

3.4.1. Model-based algorithms

The model-based approaches perform simulation to obtain state transition probability and rewards and then solve the MDP to determine a control policy. RTDP (Barto et al., 1995) and Dyna (Barto et al., 1995) are the two primary class of the model-based approach.

3.4.1.1. Real Time Dynamic Programming

The Real Time Dynamic Programming (RTDP) algorithm is proposed by Barto et al. (1995) uses the history or past experience to build a model while interacting with the environment. The algorithm is implemented in the following steps:

1) The state transition probability at time step t is estimated based on the history or past experience.

$$P_{ss'}^{a}(t) := \frac{n_{ss'}^{a}(t)}{N_{s}^{a}(t)} \tag{3.29}$$

where, $n_{ss'}^{a}(t)$ is the observed number of actual transitions made from state s to state s' performing an action $a(\in A(s))$ until time step t and $N_{s}^{a}(t)\left(=\sum_{s'\in S} n_{ss'}^{a}(t)\right)$ is the total number of times that action a was executed in state s.

2) Immediate reward r_{t+1} can also be learned as they are observed.

3) At each time step t, determine the value function of a state s using the value iteration algorithm with the estimates of transition probability and immediate reward.

$$Q_t(s,a) := \sum_{s'} P^a_{ss'}(t)\left[R^a_{ss'}(t) + \gamma\, V_{t-1}(s')\right] \tag{3.30}$$

$$V_t(s) := \max_a Q_t(s,a) \tag{3.31}$$

4) After the value iteration routine has converged, perform additional updates of value function by assigning execution probability to each admissible action to the current state according to the value of action. This routine is called exploration, which is necessary to find a true optimal policy because the current model (transition probability and immediate reward) is not necessarily correct. Use of the Boltzmann Distribution (Bareto et al., 1995) is the one way to implement exploration.

$$\Pr(a') = \frac{\exp\left(\frac{Q(s,a')}{T}\right)}{\sum_{a \in A(s)} \exp\left(\frac{Q(s,a)}{T}\right)} \tag{3.32}$$

where T is a positive parameter controlling the probability of choosing an action a'.

3.4.1.2. Dyna

The Dyna algorithm is first introduced by Sutton (1990, 1991) and extended by Barto et al. (1995). In Dyna, simulations were performed not only to build a model but also to adjust the policies and then use the learned model to further adjust the policies. The following steps explain the Dyna algorithm.

1) Update incrementing statistics for the state transition probability and immediate reward using the same routine as in RTDP. The updated models are $\hat{P}^a_{ss'}$ and $\hat{R}^a_{ss'}$.

2) Update the action-value function at state s using the state-value iteration rule:

$$Q(s,a) := \hat{R}^a_{ss'} + \gamma \sum_{s'} \hat{P}^a_{ss'} \left[\max_{a'} Q(s',a') \right] \tag{3.33}$$

3) Perform k additional updates with k state-action pairs at random, and then update them according to the same algorithm as before:

$$Q_t(s_k,a_k) = \hat{R}^{a_k}_{s_k s'} + \gamma \sum_{s'} \hat{P}^{a_k}_{s_k s'} \left[\max_{a'} Q_t(s',a') \right] \tag{3.34}$$

4) Choose an action a' to perform in state s' based on the action-value function. The exploration strategy used in RTDP can be applied to choose an action.

3.4.2. Model free algorithm

In the model-based algorithm, an agent learns a model of environment by simulation and then solves the MDP to determine a control policy. On the other hand, the model-free approach learns a control policy based purely on the environment feedback without attempting to learn a model. Monte Carlo methods (Sutton and Barto, 1998), temporal difference learning (Bertsekas and Tsitsiklis, 1996), and Q-learning (Bertsekas and Tsitsiklis, 1996) are examples of the model-free approach.

3.4.2.1. Monte Carlo Methods

Monte Carlo (MC) methods use actual or simulated experience in order to estimate value functions and determine optimal policies. MC methods estimate state or state-action values by averaging the observed sample returns from interaction with an environment. For well-defined returns, MC methods are generally defined for episodic tasks (or finite tasks) which include the terminal state. The value estimates and policies are changed after the completion of episodes. As more episodes are tried, the value of returns converges to the true values of the returns under the policy.

MC methods use generalized policy iteration (GPI) which involves interacting processes of policy evaluation and policy improvement. MC methods require sufficient exploration because choosing the currently estimated best actions provides the returns from alternative actions that are actually better. One solution is to use a stochastic policy with nonzero probabilities of equally selecting all actions from all states ($\pi(s,a) > 0$) when it interacts with the environment to generate experience and evaluates its performance under this policy. However, they are unlikely in learning from real experience. Instead, two general approaches called on-policy method and off-policy method can be used.

In the case of on-policy methods, the agent can use a soft stochastic policy such as an ε-greedy policy, which selects with probability ($1-\varepsilon$) the action that is greedy with respect to the current estimate of the action-value function and with probability ε (a small positive value) any other action. The off-policy methods use two policies called the behavior policy used to generate experience and the estimation policy used to estimate

the value function. The pseudo-codes for these two algorithms are shown in Figure 3.5 and 3.6.

Initialize, for all $s \in S, a \in A(s)$

$Q(s,a) :=$ arbitarary

Returns$(s,a) :=$ empty list

$\pi(s,a) :=$ an arbitarary ε - soft policy

Loop:

Generate an episode using $\pi(s,a)$

For each pair s,a appearing in the episode

$R :=$ return following the first occurrence of s,a

Append R to Returns(s,a)

$Q(s,a) :=$ average(Returns(s,a))

Loop for $s \in S$ in episode

$a^* := \arg\max_a Q(s,a)$

Loop for all $a \in A(s)$

$$\pi(s,a) := \begin{cases} 1-\varepsilon+\dfrac{\varepsilon}{|A(s)|} & \text{if } a = a^* \\ \dfrac{\varepsilon}{|A(s)|} & \text{if } a \neq a^* \end{cases}$$

Loop end

Loop end

Loop end

Figure 3.5 ε-soft on-policy MC algorithm

Initialize, for all $s \in S$, $a \in A(s)$

$Q(s,a) :=$ arbitarary

$\mathrm{N}(s,a) := 0$

$\mathrm{D}(s,a) := 0$

$\pi :=$ arbitarary deterministic policy

Loop :

Select a policy π' and generate an episode using π':

$$s_0, a_0, r_1, s_1, a_1, r_2, \cdots, s_{T-1}, a_{T-1}, r_T, s_T$$

$\tau :=$ latest time at which $a_\tau \neq \pi(s_\tau)$

For each pair s,a appearing in the episode after τ:

$t :=$ the time of first occurrence of s,a after τ

$$\omega := \prod_{k=t+1}^{T-1} \frac{1}{\pi'(s_k, a_k)}$$

$$\mathrm{N}(s,a) := \mathrm{N}(s,a) + \omega R_t$$

$$\mathrm{D}(s,a) := \mathrm{D}(s,a) + \omega$$

$$Q(s,a) := \frac{\mathrm{N}(s,a)}{\mathrm{D}(s,a)}$$

Loop for $s \in S$

$$\pi(\mathrm{s}) := \arg\max_a Q(s,a)$$

Loop end

Loop end

Figure 3.6 off-policy MC algorithm

3.4.2.2. Temporal Difference Learning

Temporal difference (TD) learning is a combination of Monte Carlo ideas and dynamic programming (DP) ideas. As with the MC methods, TD learns directly from actual or simulated experiences without prior knowledge of a model of the dynamics of the environment. Similar to DP, however, TD updates the value functions based on previously estimated values without waiting for a final return. This feature is called bootstrapping.

The simplest TD method, known as one-step TD (TD(0)), updates the value functions based on the immediate reward and the estimated value of the next state, whereas MC methods must wait until the end of episode to update the value functions. In between are the n-step backups (updates), based on the n steps of received reward and the estimated value of the n^{th} next state. N-step TD method are rarely used because they require waiting n steps to observe the reward and state. Equation (3.35) explains the update rules from TD(0) to MC methods.

$$
\begin{aligned}
&V(s_t) := V(s_t) + \alpha\left[r_{t+1} + \gamma\, V(s_{t+1}) - V(s_t)\right] && \text{TD(0)} \\
&V(s_t) := V(s_t) + \alpha\left[r_{t+1} + \gamma\, r_{t+2} + \gamma^2\, V(s_{t+3}) - V(s_t)\right] && \Uparrow \\
&V(s_t) := V(s_t) + \alpha\left[r_{t+1} + \gamma\, r_{t+2} + \gamma^2\, r_{t+3} + \gamma^3\, V(s_{t+4}) - V(s_t)\right] && \text{TD}(\lambda) \\
&\quad\vdots && \Downarrow \\
&V(s_t) := V(s_t) + \alpha\left[R_t - V(s_t)\right] && \text{MC} = \text{TD(1)}
\end{aligned}
\qquad (3.35)
$$

where, α is learning rate parameter. The backup dimensions of reinforcement algorithms are depicted in figure 3.7.

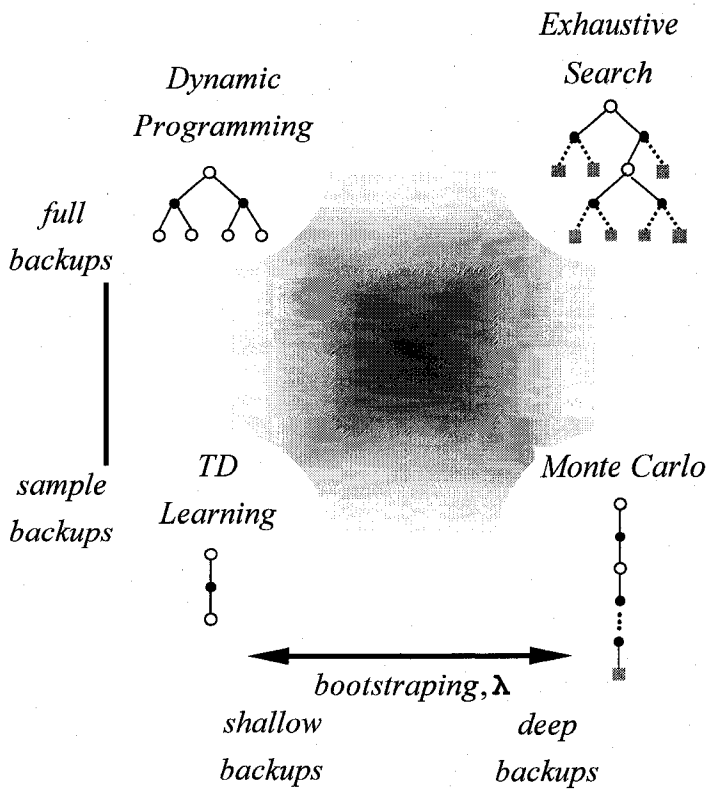


Figure 3.7 Backup dimensions of reinforcement learning methods

The unique advantage of the TD methods over the DP methods is that a model of the environment is not required, such as its reward and transition probability. In addition, the TD methods uses sample updates instead of full updates as in the case of DP. The next advantage of TD methods over MC methods is that TD methods require only one time step of simulation in order to update the value functions instead of wait until the end of an episode. In practice, the TD methods have usually been found to converge faster than MC methods on stochastic tasks (Sutton and Barto, 1998).

The TD method can be used for the policy iteration algorithm. It is particularly useful to estimate state-action values rather than state values for the policy iteration. Thus,

another primary goal for the TD method is to estimate Q^* by policy evaluation with state-action values. As with MC methods, TD methods require sufficient exploration in the generated experience in order to find the optimal policy. One can use either on-policy or off-policy approaches to ensure an adequate exploration in TD method.

Examples of on-policy and off-policy TD control methods are called SARSA (Rummery and Niranjan, 1994) and Q-learning (Watkins, 1989) algorithm, respectively. SARSA algorithm estimate the action-value function $Q(s,a)$ for the current behavior policy π, which is ε-greedy policy with respect to the current estimate of action-value function. The SARSA algorithm performs on every time step the following updates.

$$Q(s_t,a_t) := Q(s_t,a_t) + \alpha\left[r_t + \gamma Q(s_{t+1},a_{t+1}) - Q(s_t,a_t)\right] \tag{3.36}$$

Watkins (1989) develop an off-policy TD control algorithm known as Q-learning, which learns optimal action-value function, Q^* is directly approximated from learned action-value function, Q, regardless of the policy being followed. This algorithm has been proved early convergence (Sutton et al, 1998). The Q-learning algorithm performs the updates by following equation.

$$Q(s_t,a_t) := Q(s_t,a_t) + \alpha\left[r_t + \gamma \max_a Q(s_{t+1},a_{t+1}) - Q(s_t,a_t)\right] \tag{3.37}$$

The algorithms of TD, SARSA, and Q-learning algorithms are given in Figure 3.8, 3.9 and Figure 3.10.

Initialize $V(s)$ arbitrarily and $e(s) = 0$, for all $s \in S$
Loop (for each episode):
  Initialize s
  Loop (for each step of episode):
    $a :=$ action given by π(s)
    Take action a, observe r and s'
    $\delta := r + \gamma V(s') - V(s)$
    Loop (for $s \in S$)
      $V(s) := V(s) + \alpha \cdot \delta$
    Loop end
    $s := s'$
  Loop end
Loop end

Figure 3.8 Online TD(λ) algorithm

Initialize $V(s)$ arbitrarily and $e(s) = 0$, for all $s \in S$
Loop (for each episode):
  Initialize s
  Loop (for each step of episode):
    $a :=$ action given by π(s)
    Take action a, observe r and s'
    $\delta := r + \gamma V(s') - V(s)$
    Loop (for $s \in S$)
      $V(s) := V(s) + \alpha \cdot \delta$
    Loop end
    $s := s'$
  Loop end
Loop end

Figure 3.9 SARSA(λ) algorithm

Initialize $Q(s,a)$ arbitrarily for all s,a
Loop (for each episode):
  Initialize s,a
  Loop (for each step of episode):
    Take action a, observe r and s'
    Choose a' from s' using policy derived from Q (e.g. ε-greedy)
    $a^*: = \arg\max_b Q(s',b)$ (if a' ties for the max, then $a^*: = a'$)
    $\delta: = r + \gamma Q(s',a^*) - Q(s,a)$
    Loop (for $s \in S$)
      $Q(s,a) := Q(s,a) + \alpha \cdot \delta$
    Loop end
    $s := s'$; $a := a'$
  Loop end
Loop end

Figure 3.10 $Q(\lambda)$ algorithm

3.5. Generalization and function approximation

The reinforcement learning requires the storage of variety of mapping including $\pi(s): S \to A$ (policies), $V(s): S \to \Re$ (state value functions), $Q(s,a): S \times A \to \Re$ (state action value functions), $s'(s,a): S \times A \to S$ (deterministic transition), and $\pi(s,a): S \times A \to [0,1]$ (policy probabilities). In reinforcement learning algorithm the value functions play especially an important role in selecting an action, or evaluating the current state that can be used to select an action. The most common representation of state value functions for discrete state and action spaces is to employ a look-up table.

The mapping from the feature vector to look-up table can be uniform, non-uniform, or non-uniform with the detailed partitions in some parts of states. However, these methods require detailed prior knowledge of the environment for appropriate granularity or

placement of partitions. One solution for this problem can be to use adaptive resolution which construct an appropriate partition to the environment during the course of learning.

4. Stochastic Reservoir Systems Operations Models by Dynamic Programming

The goal of reservoir systems operations is to optimize the use of water for multiple purposes such as water supply and hydropower generation. Decision on how much water to release in any period and how much to retain in storage are difficult due to uncertainties in valuing future storage. Dynamic programming (DP) is one of the widely used optimization techniques for solving these kinds of sequential decision problems. Many stochastic reservoir system operations models have been developed and applied to reservoir systems operations for water resources management.

4.1. Implicit Stochastic Dynamic Programming

Young (1967) first introduced the implicit stochastic reservoir control problem, which requires synthetic streamflow generation, deterministic dynamic programming optimization, and regression analysis for inferring optimal operational policies. The one-dimensional (i.e., single reservoir) backwards DP recursion equation and the system state dynamics equation for the deterministic case is given in equations (4-1) and (4-2). Reservoir storage represents the state variable and the objective is to maximize the total discounted benefit (or rewards).

$$V_t(s_t) = \max_{a_t \text{ or } s_{t+1}} \{r_t + \gamma V_{t+1}(s_{t+1})\} \tag{4.1}$$

$$s_{t+1} = s_t + i_t - a_t \quad \text{or} \quad a_t = s_t + i_t - s_{t+1} \tag{4.2}$$

$$s_{t+1}^{\min} \le s_{t+1} \le s_{t+1}^{\max} \tag{4.3}$$

where, s_t is the reservoir storage at the beginning of period t, i_t is the reservoir inflow during that period, and a_t is the water released during period t. Notation $s_{t+1}^{\min}$ and $s_{t+1}^{\max}$ are dead storage and normal full storage, respectively, at the end of period t; r_t is the benefit (or reward) function for period t corresponding to s_t, s_{t+1}, i_t, and a_t; γ is the discount factor and $V_t(s_t)$ is the DP optimal return function (state value function) calculated recursively in equation (4.1). The recursion equations for the multidimensional case have basically the same structure as the one-dimensional case, except that all dealing with variables are represented as vectors. Evaporation and seepage loses are excluded in this formulation to simplify the presentation although they are easily included in the general formulation.

Multiple linear or nonlinear regression models can be applied to the deterministic optimization results for inferring feedback operation rules. In regression models, the reservoir releases or target storage can be inferred by the current reservoir storage and/or inflows. Artificial neural network (ANN) can be used to identify the nonlinear structure in operation rules.

4.2. Stochastic Dynamic Programming (SDP)

Deterministic dynamic programming requires the unrealistic assumption of perfect knowledge of inflows to the reservoir. This assumption can be modified by considering inflows to the reservoir as a random process described by probability distributions. If the inflow of the current period is considered known the SDP steady state policy is determined by the one-dimensional backward recursion equation with the state dynamics equation in the noninverted form.

$$V_t(s_t, I_t) = \max_{a_t}\{r_t + \gamma \sum_{I_{t+1}} p_{I_{t+1}} V_{t+1}(s_{t+1}, I_{t+1})\} \tag{4.4}$$

where I_t is the classed reservoir inflow during period t and $p_{I_{t+1}}$ is the marginal probability for the reservoir inflow state during period t+1. In addition, if the current inflow I_t is considered unknown the recursion equation is

$$V_t(s_t) = \max_{a_t}\{\sum_{I_t} p_{I_t}[r_t + \gamma V_{t+1}(s_{t+1})]\} \tag{4.5}$$

Little (1955) suggested a stochastic dynamic programming (SDP) approach using a periodic first-order Markov property of consecutive streamflows over time. This approach captures the conditional steamflow distribution. With this approach, inflows are defined probabilistically, conditioned on current period inflows. The recursion equation is as follows

$$V_t(s_t, I_t) = \max_{a_t}\{r_t + \gamma \sum_{I_{t+1}} p_{I_t I_{t+1}} V_{t+1}(s_{t+1}, I_{t+1})\} \quad (4.6)$$

where, $p_{I_t I_{t+1}}$ is the reservoir inflow transition probability from the inflow state during period t to the inflow state during period $t+1$.

An alternative recursion equation for the conditional case can be considered as a function of initial storage and the previous inflow when the current inflow is unknown, instead of the current inflow and forecasted inflow

$$V_t(s_t, I_{t-1}) = \max_{a_t}\{\sum_{I_t} p_{I_{t-1} I_t}[r_t + \gamma V_{t+1}(s_{t+1}, I_t)]\} \quad (4.7)$$

For the conditional case, equation (4.6) is preferred over equation (4.7) if it is easy to forecast the next period inflow due to the persistent nature of inflows from one period to the next. On the contrary, it is desirable to use the equation (4.5) instead of (4.4) for the unconditional case if forecasting inflows is difficult. Stedinger et al (1984) used streamflow forecasts directly as state variable but this is only valid when suitable long-term forecast models are available.

The result of SDP by conditional cases is a set of optimal release policies for each combination of discrete reservoir states and current inflows (or previous inflows). For the unconditional cases, a set of optimal release policies for discrete reservoir states is derived.

4.3. Sampling Stochastic Dynamic Programming (SSDP)

Kelman et al. (1990) proposed a variation of SDP called sampling stochastic dynamic programming (SSDP) which uses streamflow scenarios to represent the stochastic process. The functional equation for SSDP (Faber and Stedinger, 2001) is represented as

$$\max_{a_t}\{r_t + \gamma \sum_k p_{jk}^t [V_{t+1}(s_{t+1}, k)]\} \qquad \forall\, t \in \{1,...,T\} \tag{4.8}$$

$$V_t(s_t, j) = r_t + \gamma V_{t+1}(s_{t+1}, j) \qquad \forall\, t \in \{1,...,T\} \tag{4.9}$$

where scenario j represents flows of a particular historical, scenario k represents flows of another historical year that could; p_{jk}^t is the transition probability from scenario j to k; and t is time period, with T being the final period in the model.

Equation (4.8) requires transition probability p_{jk}^t and one simple choice for this is $p_{jk}^t = 1$ for $j = k$ and zero otherwise. This choice represents the deterministic case in which uncertainty among scenarios is not considered. Another simple choice is assuming an equal likelihood probability from a given scenario to all other scenarios in the next period by setting transition probability p_{jk}^t equal to 1/M, where M is the total number of scenarios. This assumption results in the following

$$\max_{a_t}\{r_t + \frac{\gamma}{M} \sum_{k=1}^{M} V_{t+1}(s_{t+1}, k)\} \tag{4.10}$$

SSDP method with more sophisticated transition probabilities using streamflow forecast information is described by Kelman et al.(1990) and Faber and Stedinger (2001). Although this method provided more efficient reservoir operations than use of historical streamflows, they will not be discussed here because it requires a long term forecast model.

4.4. Reinforcement Learning and Markov Decision Processes

All the previous stochastic dynamic programming approaches except SSDP with equation (4-10) require analysis of the streamflow processes represented by the historical streamflow records, either directly or indirectly. Implicit dynamic programming uses the indirect way of reproduction of the streamflow and conventional SDP need to find out the transition probabilities. A large water resources systems is complicated by the presence of multiple streamflows and demands. The model-free reinforcement learning approach is one way of handling such a complex system without the need for complex analysis of spatially and temporally correlated streamflow process.

The reservoir operation can be represented by the storage transition by Markov chain instead of analyzing the statistical structure of inflow when the current inflow is independent of the previous inflow. Let i_t has the following probability distribution

$$p_{i_t} = \Pr(i_t) \qquad \text{for } i_t = 0,1,... \tag{4.11}$$

The system state dynamics equation in (4.2) can be written as the following transition equation (Taylor and Karlin, 1984; Wang and Adams, 1986).

$$P_{s_t s_{t+1}}^{a_t} = \Pr(i_t = s_{t+1} - s_t - a_t) = p(s_{t+1} - s_t - a_t) \tag{4.12}$$

As described previously, system state value functions are not sufficient to determine a policy without an analysis of the streamflow processes as with other stochastic dynamic programming methods. If state-action values functions are used these kinds of complex analysis are not necessary. In this case, the $Q(\lambda)$ algorithm (refer to algorithm in Figure 3.9) can be used to evaluate the state-action values and the operation policy can be determined from these values.

Evaluating the state-action values requires state-action value function update whenever the agent visits s_t and takes the action a_t according to the ε-greedy policy scheme. The corresponding state-action value update equation for (4.5) is as follows

$$Q(s_t, a_t) := Q(s_t, a_t) + \alpha \underset{i_t}{E}\left[r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right] \tag{4.13}$$

where, $Q(s_t, a_t)$ is the state-action value function for the initial reservoir storage s_t at the beginning of period t and release a_t during period t. The expectation operator E is defined over individual streamflow i_t. Learning rate α controls how much weight is given to the reward just experienced. The equation inside the brackets indicates the amount of state-action value update. The steady state optimal policy a_t^* for the state s_t is derived by

$$a_t^*(s_t) := \arg\max_{a_t} Q(s_t, a_t) \tag{4.14}$$

The corresponding state-action value update equations for the SDP recursion equation of (4.6) is defined as

$$Q(s_t, I_t, a_t) := Q(s_t, I_t, a_t) + \alpha \underset{i_t(I_t)}{E}\left[r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, I_{t+1}, a_{t+1}) - Q(s_t, I_t, a_t) \right] \tag{4.15}$$

where $Q(s_t, I_t, a_t)$ is the state-action value function of the initial reservoir storage s_t and hydrologic state I_t at the beginning of period t. The expectation operator E is defined over individual streamflow i_t in the hydrologic state I_t. The steady state optimal policy a_t^* for the state (s_t, I_t) is derived by

$$a_t^*(s_t, I_t) := \arg\max_{a_t} Q(s_t, I_t, a_t) \tag{4.16}$$

The state-action value equation corresponding to the SSDP functional equation (4.10), along with optimal policy equation, is represented as:

$$Q(s_t, k, a_t) := Q(s_t, k, a_t) + \alpha \left[r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, k, a_{t+1}) - Q(s_t, k, a_t) \right] \tag{4.17}$$

$$a_t^* = \arg\max_{a_t} \underset{k}{E}\left[Q(s_t, k, a_t) \right] \tag{4.18}$$

where the expectation operator E is defined over individual scenario k.

Equations (4.13), (4.15) and (4.17) utilize discounted cumulative benefit (or reward) value function for which it is difficult to directly use this function for the boundary conditions in steady state periodic MDP model. Instead, the differences in value function between various storage states and a desirable state (e.g. normal full storage) can be used for the steady state boundary condition. However, the differences in value function are not consistent due to the partially updates of the values function in Q-Learning algorithm. The average reward model in equation (3.3) can be used to solve this problem. Weighting factors for the immediate reward and the discount factor are introduced. The value of value function converges to the magnitude of the average return R_t by these weighting factors. The corresponding equations for the equations (4.13), (4.15), and (4.17) using weighting factors are as follows

$$Q(s_t,a_t) := Q(s_t,a_t) + \alpha \underset{i_t}{E}\left[W_1 r_t + W_2 \max_{a_{t+1}} Q(s_{t+1},a_{t+1}) - Q(s_t,a_t) \right] \tag{4.19}$$

$$Q(s_t,I_t,a_t) := Q(s_t,I_t,a_t) + \alpha \underset{i_t(I_t)}{E}\left[W_1 r_t + W_2 \max_{a_{t+1}} Q(s_{t+1},I_{t+1},a_{t+1}) - Q(s_t,I_t,a_t) \right] \tag{4.20}$$

$$Q(s_t,k,a_t) := Q(s_t,k,a_t) + \alpha \left[W_1 r_t + W_2 \max_{a_{t+1}} Q(s_{t+1},k,a_{t+1}) - Q(s_t,k,a_t) \right] \tag{4.21}$$

$$W_1 + W_2 = 1 \tag{4.22}$$

$$W_2 = \gamma \tag{4.23}$$

where W_1 and W_2 are the weighting factors used for scaling and discounting of the action-value function. Larger values of W_1 represent higher values of current water use, whereas smaller values of W_1 represent higher future usage. This weighting factor enables use of the Monte Carlo simulation approach and SDP approaches. Relationships between the state value function and r_t are as follows.

$$\begin{aligned}
V^*(s_t) &= \max_{a_t} Q(s_t, a_t) \\
&= W_1 r_t^* + W_2 V^*(s_{t+1}) \\
&= W_1 r_t^* + W_2 (W_1 r_{t+1}^* + W_2 V^*(s_{t+2})) \\
&= W_1 r_t^* + W_2 W_1 r_{t+1}^* + W_2^2 (W_1 r_{t+2}^* + W_2 V^*(s_{t+3})) \\
&= W_1 r_t^* + W_2 W_1 r_{t+1}^* + W_2^2 W_1 r_{t+2}^* + W_2^3 (W_1 r_{t+3}^* + W_2 V^*(s_{t+4})) \\
&= \cdots
\end{aligned} \tag{4.24}$$

where V^* is the optimal state value function and r_t^* is optimal immediate reward for period *t*. The equation (4.24) can be expressed using equations (4.22) and (4.23).

$$V^*(s_t) = (1-\gamma)r_t^* + \gamma(1-\gamma)r_{t+1}^* + \gamma^2(1-\gamma)r_{t+2}^* + \gamma^3(1-\gamma)r_{t+3}^* + \gamma^4(1-\gamma)r_{t+4}^* + \cdots$$
$$= (1-\gamma)(r_t^* + \gamma r_{t+1}^* + \gamma^2 r_{t+2}^* + \gamma^3 r_{t+3}^* + \gamma^4 r_{t+4}^* + \cdots) \quad (4.25)$$

Assume optimal immediate reward r_t^* has the same magnitude of value for simplicity,

$$R = r_t^* \simeq r_{t+1}^* \simeq r_{t+2}^* \simeq r_{t+3}^* \simeq \cdots \quad (4.26)$$

Then equation (4.25) can be expressed as

$$V^*(s_t) = (1-\gamma)(R + \gamma R + \gamma^2 R + \gamma^3 R + \cdots + \gamma^n R + \cdots)$$
$$= (1-\gamma)R(1 + \gamma + \gamma^2 + \gamma^3 + \cdots + \gamma^n + \cdots)$$
$$= (1-\gamma)R\lim_{n\to\infty}\frac{(1-\gamma^n)}{(1-\gamma)} \quad (4.27)$$
$$= R\lim_{n\to\infty}(1-\gamma^n)$$

The optimal state value function in equation (4.27) has the average reward R with sufficiently large n. Larger value of discount factor γ require larger value of n to make the optimal state value function have the average reward. Therefore, large discount value of discount factor represents higher value of future water usage. The number of time step with respect to various discount factors required calculating the state value function is estimated by the following equation.

$$\gamma^n \approx 0 \tag{4.28}$$

This average state value function by weighting factors can directly applied to the boundary condition of steady state reservoir operation.

4.5. Transition probability in reinforcement learning

In order to apply conventional SDP, it is necessary to specify transition probabilities $p_{I_t I_{t+1}}$ from the current state I_t to the next state I_{t+1}. The transition probability matrices can be obtained from historical inflow data or synthetically generated inflow data. To define the inflow state, one can use percentile values directly from the historical data or from fitted theoretical probability distribution functions where parameters are estimated from the data. Not only are the serial correlations between inflows in consecutive time periods important, but also spatial cross correlations among multi-site inflows for watershed based operating system. However, cross correlations for multi-site inflows have been largely ignored when the mathematical manipulation is difficult for large number of reservoir cases. Direct percentile can be used in multi-site inflow analysis, as depicted in Figure 4.1.

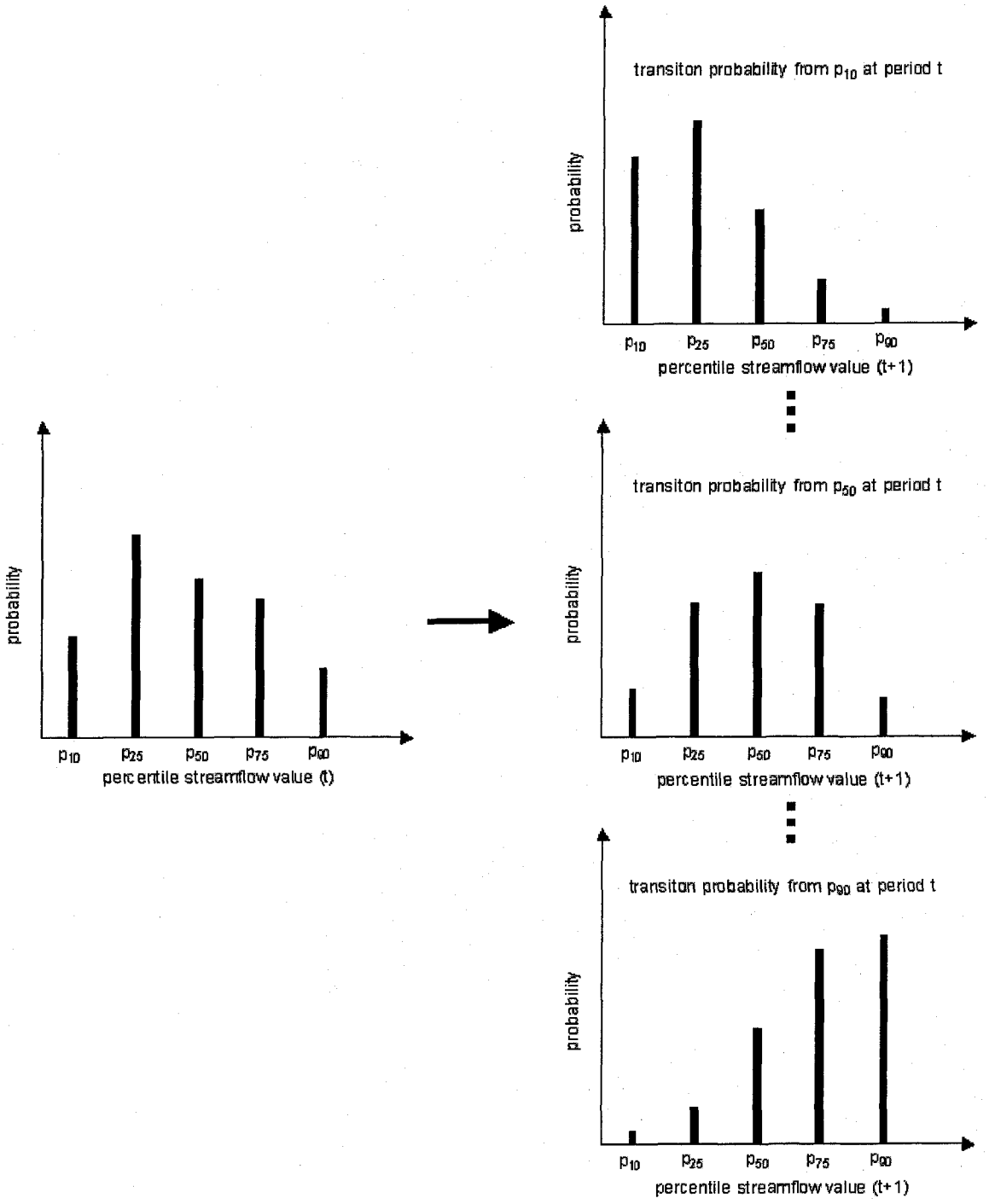


Figure 4. 1 A transition probability in SDP

Once the n finite states are defined, an (n x n) transition probability matrix P_n can be determined

$$P_n = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \cdots & \cdots & \cdots & \\ p_{n2} & p_{n2} & \cdots & p_{mn} \end{bmatrix} \tag{4.29}$$

where the p_{ij} have the following properties:

$$\sum_{j=1}^{n} p_{ij} = 1, \quad \text{for } i = 1,2,\cdots,n \tag{4.30}$$

and

$$p_{ij} \geq 0, \quad \text{for all } i \text{ and } j \tag{4.31}$$

Individual transition probabilities p_{ij} can be estimated by

$$p_{ij} = \frac{f_{ij}}{\sum_{j=1}^{n} f_{ij}}, \quad \text{for } i, j = 1,2,\cdots,n \tag{4.32}$$

The advantage of the Q-Learning approach is that it does not require an explicit transition structure as shown in Figure 4.1. However, Q-Learning poses discrete inflows defined for each state like SDP but their individual flow in the discrete state is used for simulation. Therefore, Q-Learning transition leads to the different state in the next time

period whereas SDP transition to the one state in the next time period when they are in the same state of reservoir and inflow. The relationship between the percentile values of streamflow at period t and reservoir storage volume at period t+1 using both SDP and Q-Learning when no release is applied is shown in Figure 4.2 for the unconditional case and Figure 4.3 for conditional case.

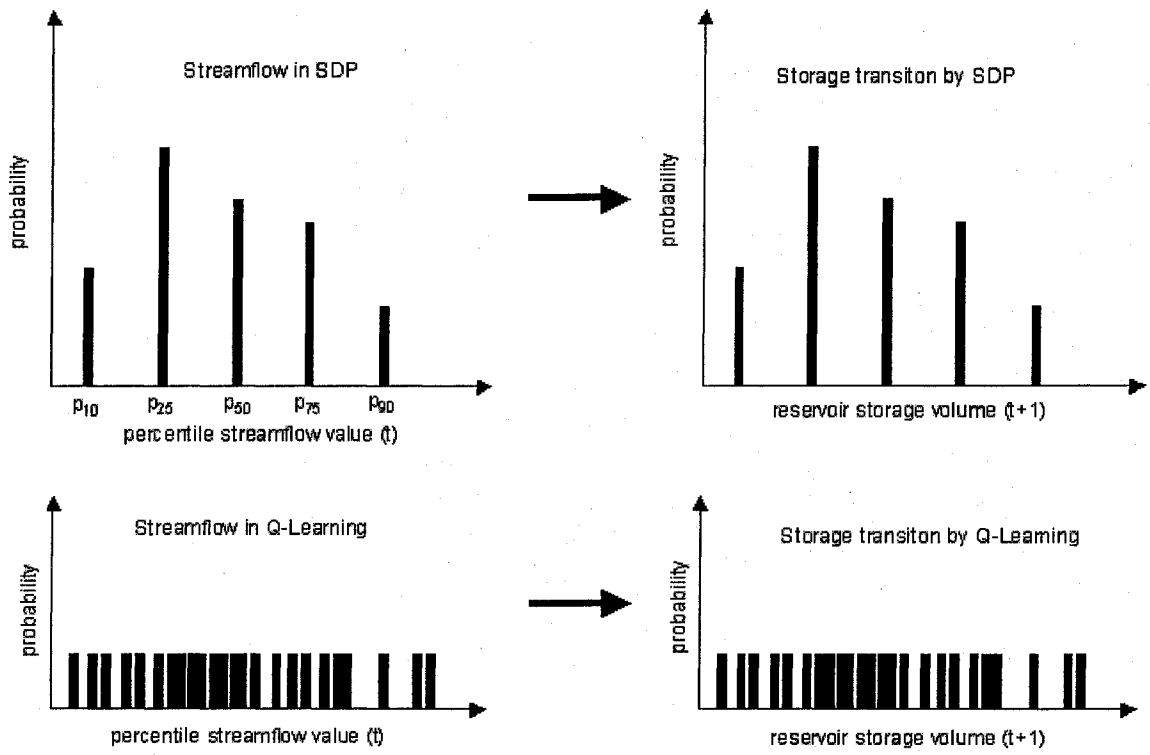


Figure 4.2 Reservoir storage transition by SDP and Q-Learning (unconditional case)

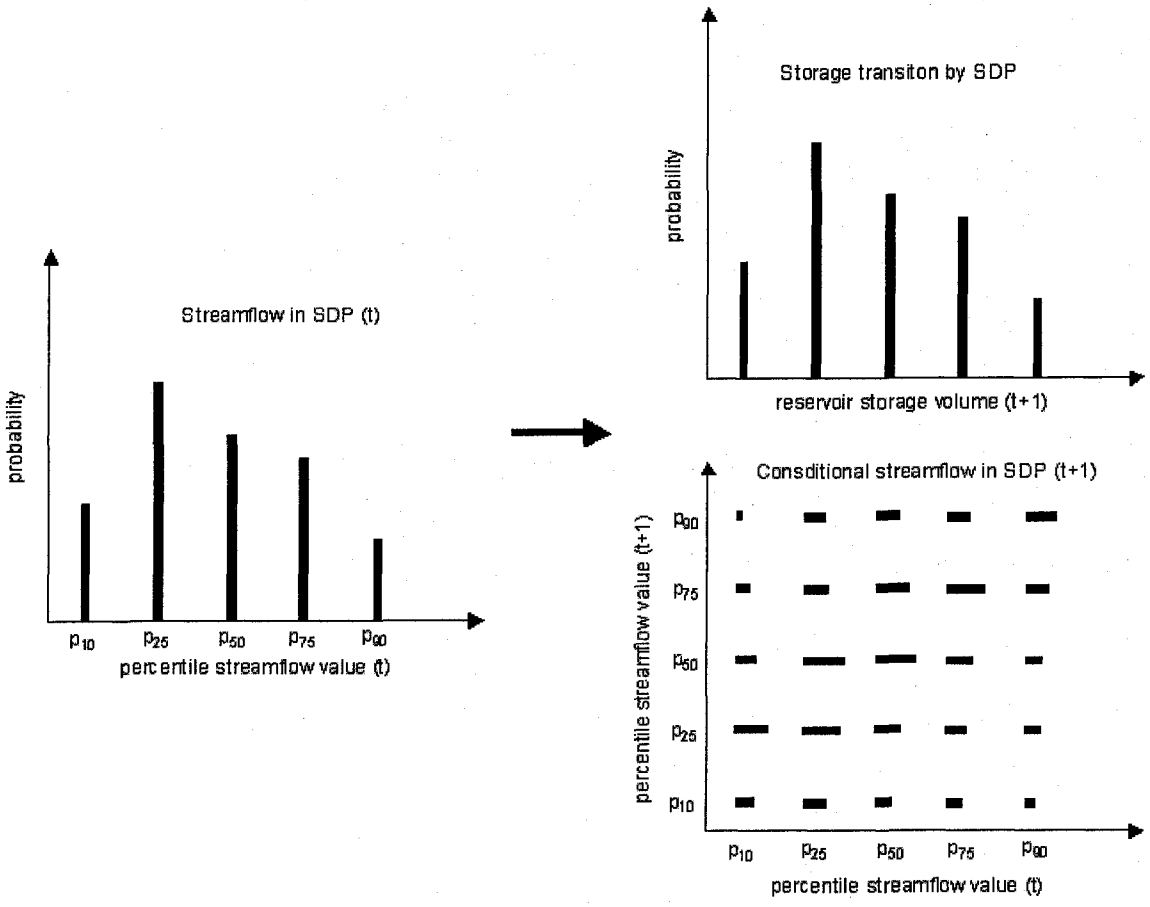


Figure 4. 3 Reservoir storage transition and Conditional streamflow by SDP (conditional case)

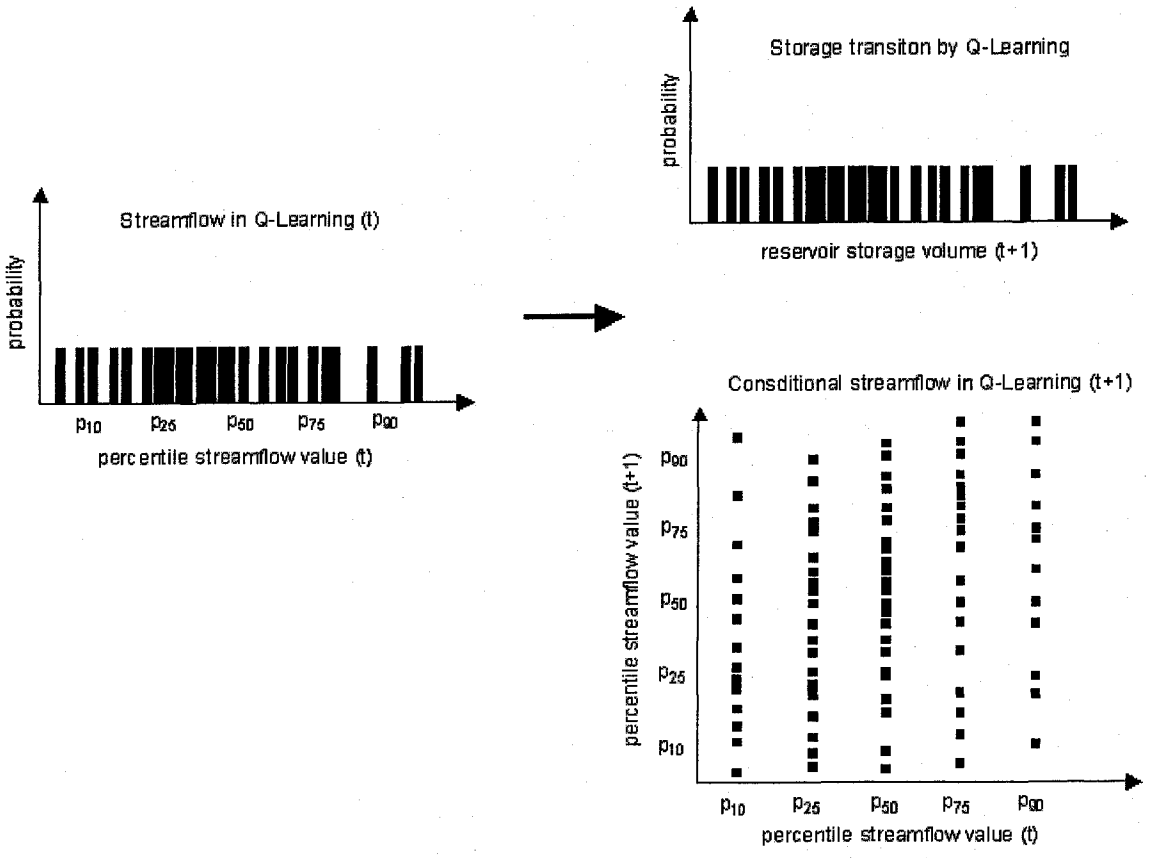


Figure 4. 4 Reservoir storage transition and Conditional streamflow by Q-Learning (conditional case)

4.6. Cluster Analysis for hydrologic state

In the previous section, the inflow state is defined by percentile values. Another approach is to apply to define the inflow state. By grouping a collection of data or observations into clusters, data within each cluster are more closely related to one another than data or observations assigned to different clusters.

The K-means algorithm is one of the most popular of the iterative descent clustering methods (Hastie et. al, 2001). The K-means clustering method finds clusters and cluster centers in a set of unlabeled data by selecting the desired number of cluster centers and iteratively moving the centers to minimize the total cluster variance.

A number of clusters K < N (when N is the total number of observations) is prespecified and each one is labled by an integer k $\in$ {1, ..., K}. Each observation is assigned to one and only one cluster. It is assumed that all variables are quantitative and the squared Euclidean distance is selected as the dissimilarity measure.

$$d(x_i, x_{i'}) = \sum_{j=1}^{p} (x_i - x_{i'})^2 = \| x_i - x_{i'} \|^2 \tag{4.33}$$

The within-point scatter can be represented by the following energy function

$$W(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(i')=k} \| x_i - x_{i'} \|^2 = \sum_{k=1}^{K} N_k \sum_{C(i)=k} \| x_i - \bar{x}_k \|^2 \tag{4.34}$$

where, $\bar{x}_k = (\bar{x}_{1k}, \cdots, \bar{x}_{pk})$ is the mean vector associated with the k[th] cluster, $C(i)$ is the cluster encoder assigning the i[th] observation to the k[th] cluster, and $N_k = 1$ (if $C(i) = k$ for i=1, 2, ⋯,N). The goal is to assign the N observations to the K clusters by minimizing the average dissimilarity of the observations from the cluster mean.

An iterative descent algorithm minimizes equation (4.35) can be obtained by noting that for any set of observations S.

$$C^* = \min_C \sum_{k=1}^{K} N_k \sum_{C(i)=k} \| x_i - \bar{x}_k \|^2 \tag{4.35}$$

$$\bar{x}_S = \arg\min_m \sum_{i \in S} \| x_i - m \|^2 \tag{4.36}$$

Hence we can cluster the given observation by solving the following optimization problem.

$$\min_{C, \{m_k\}_1^K} \sum_{k=1}^{K} N_k \sum_{C(i)=k} \| x_i - m_k \|^2 \tag{4.37}$$

The K-mean clustering algorithm is an optimization procedure that solves (4.37) in the following steps.

1. For a given cluster assignment C, the total cluster variance (4.37) is minimized with respect to $\{m_1, m_2 \cdots, m_k\}$ yielding the means of the currently assigned clusters (4.36).

2. Given a current set of means $\{m_1, m_1, \cdots, m_k\}$, (4.31) is minimized by assigning each observation to the closest (current) cluster means. That is,

$$C(i) = \arg\min_{1 \le k \le K} \| x_i - m_k \|^2 \tag{4.38}$$

3. Steps 1 and 2 are repeated sequentially until the assignments do not change

One example of the K-means clustering algorithm for the two-dimensional case is shown in Figure 4.5, where m_1, m_2, m_3 indicate the means of each cluster.

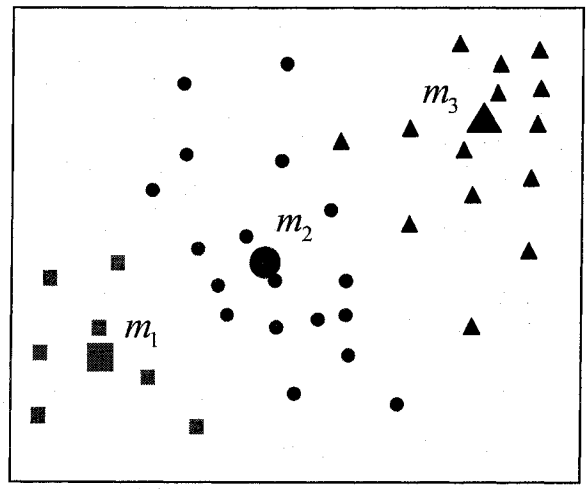


Figure 4. 5 Example of K-mean clustering

5. Case Study: Keum River Basin in Korea

5.1. Description of Keum River basin

The Keum River basin (Figure 5.1) in Korea was chosen as a case study to demonstrate the applicability of the reinforcement learning algorithm. It is one of the four major river basins in Korea and is located in the southwestern portion of the Korea Peninsula with basin area of 9,810 square kilometers and river main stem of 396 kilometers.

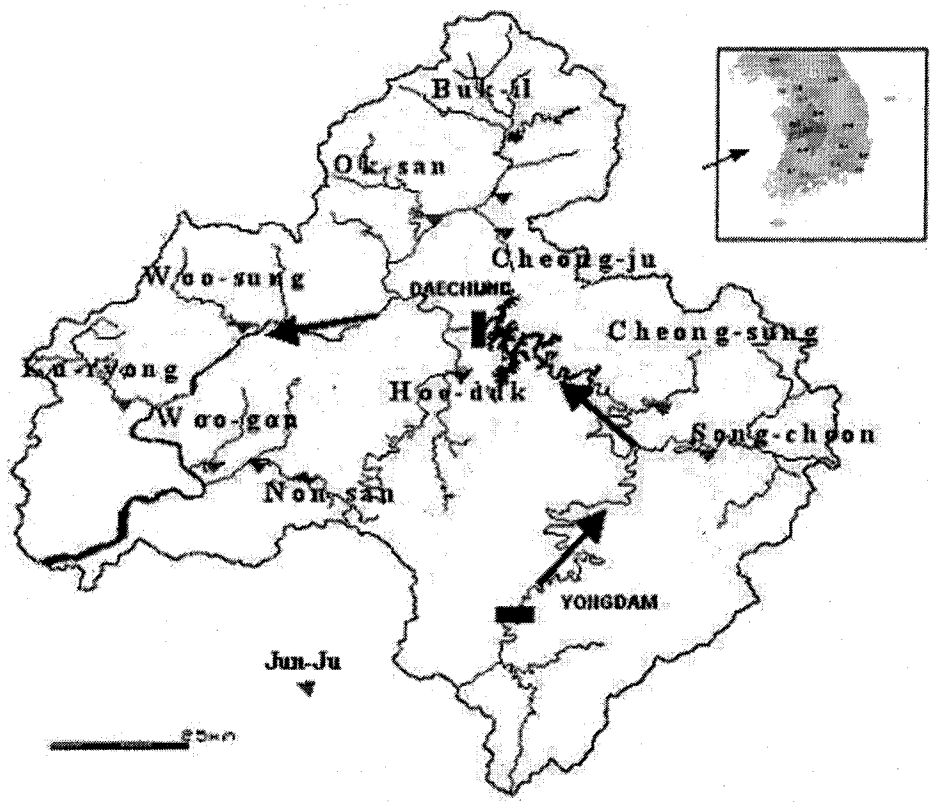


Figure5.1 Keum River basin in Korea

The Keum River basin consists of 12 sub basins and YongDam and DaeChung reservoirs, with operations in this river basin benefiting almost five million people. While satisfying municipal, industrial, and agricultural water demands are the primary purposes of the two reservoirs, they are also operated for flood control, hydropower generation, and maintaining instream flow requirements. The operation of this complex multipurpose, multireservoir system provides a challenging case study for demonstrating the reinforcement learning algorithm.

DaeChung reservoir is located in the central portion of the Keum River basin and has been operated more than 20 years. YoungDam reservoir has been recently constructed the upstream of DaeChung reservoir primary for providing transbasin water supply in JunJu region. Construction of YongDam reservoir has been built it has given rise to serious regional conflicts between the downstream region of Keum River basin and the Junju region receiving the transbasin water from YongDam reservoir. The main focus of the conflicts is that cities in the Keum River basin, which basically depend on DaeChunng reservoir do not want to share the water since they believe the water quality of DaeChung reservoir will deteriorate due to decreased flows in Keum River. Studies conducted on impacts of the diverted water on water quality in the Keum river basin conclude that the expected problems were not serious as long as certain conditions were placed on operation of YoungDam and waste water treatment plants were constructed. Consequently, the recent changes have required new coordinated water supply management in the basin. A brief schematic of the Keum River system is shown in Figure 5.2. Return flows are indicated by dashed line and transbasin diversions are specified by numbers 91, 92, and 93.

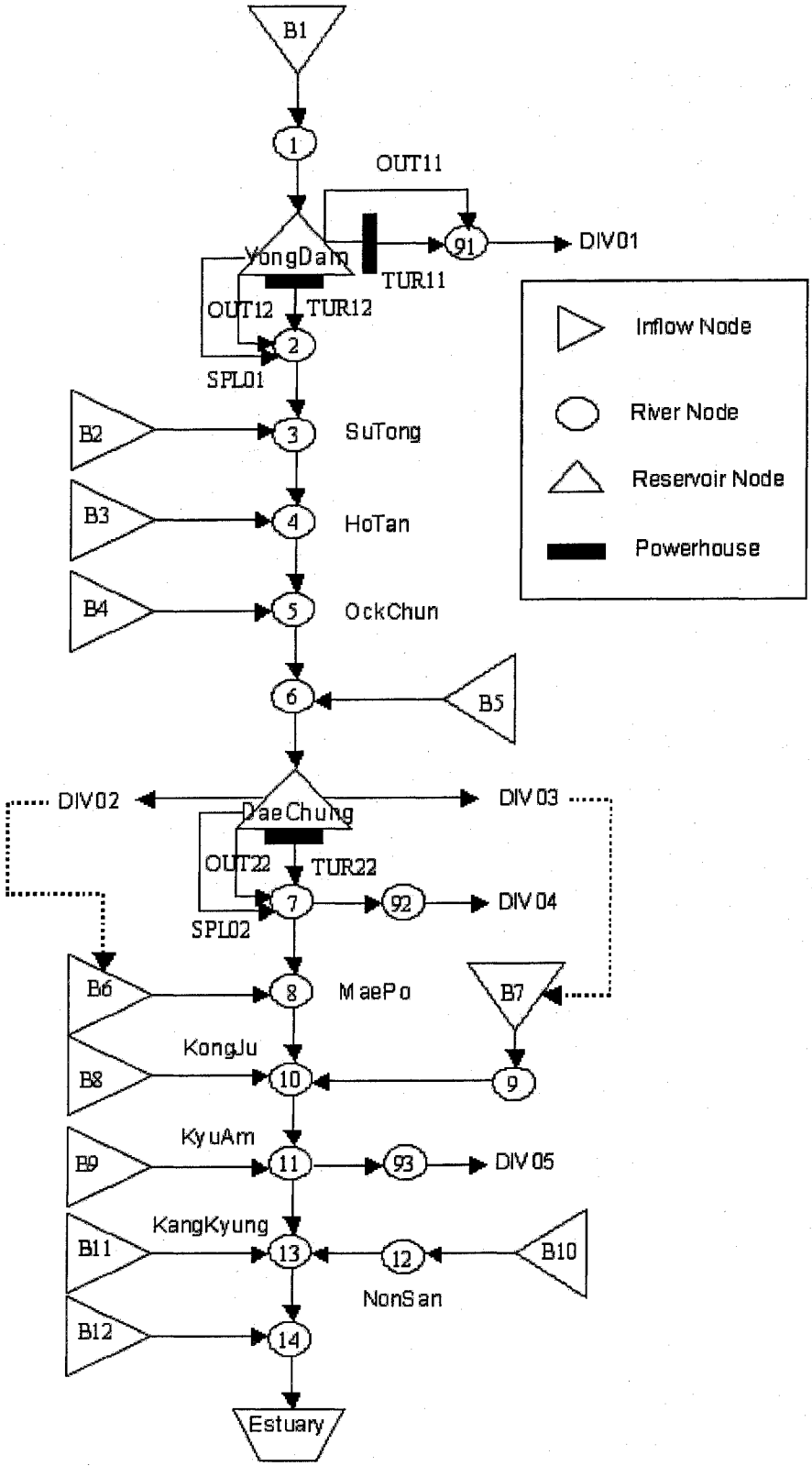


Figure5. 2 Brief schematic of Keum River basin

The bulk of inflow to the two reservoirs arrives during the flood season form June to September and is stored and used throughout the remainder of the year. The monthly inflow characteristics of YongDam and DaeChung reservoirs are calculated from 19 years of data and given in Tables 5.1 and 5.2. The unit used here is million cubic meters per month (MCM/mon). It can be seen that both YongDam and DaeChung reservoirs have a relatively high variation in flows during October to December and March to June.

Table 5.1 YongDam reservoir inflow characteristics

YongDam inflow	OCT	NOV	DEC	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP
Average (MCM/mon)	32.11	36.83	25.95	23.88	22.55	42.32	37.61	41.71	87.98	152.09	152.18	77.44
Minimum (MCM/mon)	1.82	0.69	0.27	1.42	1.28	0.31	0.26	0.27	0.17	1.65	1.68	2.94
Maximum (MCM/mon)	161.67	254.63	147.41	62.07	62.84	369.11	171.16	253.64	623.38	462.33	449.33	270.57
Standard Deviation	38.61	59.04	34.88	15.99	16.95	81.20	41.04	57.91	140.30	124.71	133.58	68.74
Coefficient of Variation	1.2	1.6	1.3	0.7	0.8	1.9	1.1	1.4	1.6	0.8	0.9	0.9

Table 5.2 DaeChun reservoir inflow characteristics

DaeChung inflow	OCT	NOV	DEC	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP
Average (MCM/mon)	94.8	58.9	50.2	66.5	70.2	83.6	87.5	88.3	189.8	488.0	358.4	277.1
Minimum (MCM/mon)	13.1	12.4	12.4	19.4	15.5	24.6	15.5	16.1	14.5	81.9	40.5	23.2
Maximum (MCM/mon)	330.0	121.0	167.8	180.2	282.4	209.4	309.5	367.6	834.3	1061.4	978.6	803.0
Standard Deviation	87.0	34.3	35.6	46.2	65.4	49.7	74.6	90.5	197.8	281.1	293.2	246.7
Coefficient of Variation	0.9	0.6	0.7	0.7	0.9	0.6	0.9	1.0	1.0	0.6	0.8	0.9

5.2. Operation Guidelines of Keum River basin

The priorities of water allocation in the sub basin are instream flows, domestic, industrial, agricultural water in the highest to the lowest. These water demands are first allocated before allocating any water demand in main stem of river basin to protect the preexisting water right in sub basin. Instream maintenance flows are provided by KOWACO (The Korea Water Resources Corporation), which defined as the 95% exceedence percentile of 19-year historical streamflows for each sub basin (KOWACO, 2003). Domestic and industrial water demands are estimated from records of previous water usage. The amounts of return flow from domestic and industrial water usage is assumed 65% of the diversion amount. The agricultural water is estimated from an agricultural consumptive use model and the amount of return flow are assumed 35% of the diversion. Lagging of these return flows is not considered so the return flows are returned to the river in the same time period as the diversion. These assumptions were made when KOWACO calibrate the rainfall runoff simulation by SSAR (KOWACO, 2003).

The highest priority for Yong-Dam reservoir operation is maintaining the instream flow requirement of five cubic meters per second (cms) in the downstream of YongDam reservoir and the second highest priority is the Jun-Ju transbasin demand. After the Jun-Ju demand is satisfied, up to 17.1cms of water is allowed to release to Keum River according to the reservoir level. YongDam reservoir operation guidelines were provided by KOWACO (2003).

DaeChung reservoir directly supplies water to Daejun (sub basin 6) and ChungJu (sub basin 7) for domestic and industrial use, with 65% of this water usage appearing are

return flow to sub basin 6 and 7, accordingly. Water allocation downstream of Dae-Chung reservoir is governed by a priority system that places domestic use first, followed by industrial, then agricultural water, and finally instream flow requirements. All the assumptions related to return flow in the main stem of Keum river are the same as the assumptions made for sub basins. The instream flow of 40cms at the Kyu-Am gauging station is recommended. These DaeChung reservoir operation guidelines were provided by KOWACO (2003).

The lowest priority is assigned to hydropower generation in the Keum River basin systems operations. Hyropower genetation at YongDam rservoir has 24 hours of hydropower generation with maximum power generation capacity of 59.43 MCM/mom (1^{st} power generator) and 16.6 MCM/mon (2^{nd} power generator). Five hours of daily peak time hydropower generation is recommended at DaeChung reservoir during the non-flood seasons (October through June), whereas the hydropower generation hours vary according to the inflow conditions during the flood seasons (July through September). The specific hydropower generation data was provided by KOWACO (2003).

Flood control at YongDam reservoir requires maintaining the reservoir level at or below 261.5 m (672.84 MCM) during flood season. However, the normal full storage level of 76.5 m (1242.7 MCM) at DaeChung reservoir is used as the flood control level since inflow to DaeChung reservoir is reduced after the construction of YongDam reservoir (KOWACO, 2003).

In addition to the monthly operational guidelines and constraints, an efficient water allocation procedure was requested by KOWACO to manage deep drought since Korea has recently experienced sever drought. The procedure was developed to allocate water

deficits in accordance with the type of demand and level of use. This procedure is based on a deficit sharing policy that is illustrated in Figure 5.3. This policy provides a dynamic approach to adjusting allocation of deficits considering the extent of demand satisfaction. According to the policy, all of the water demands are satisfied when there is sufficient water available. If there is a shortage in water supply, it will first satisfy the first municipal demand block. Water continues to apply to the municipal water demand blocks until it reaches 100 percent supply level. The allocation process continues in a similar manner for the industrial water demands and then the agricultural water demands. The numbers in each block indicates relative allocation priorities. Number in each indicates the relative priorities in Figure 5.3. This provides a realistic simulation of how water is actually allocated in the basin during drought. The relative priorities for Keum River basin operation guideline with deficit sharing policy are summarized in Table 5.3.

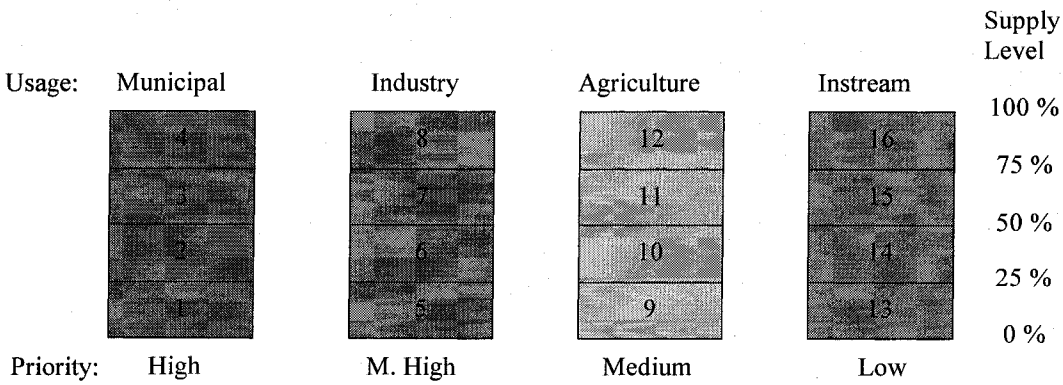


Figure 5. 3 Water supply deficit sharing policy

Table 5. 3 Relative water use priority system in Keum River basin

Priority	Usage	Remark
1	Instream Flow	Yong Dam Reservoir Down Stream
2	Jun Ju Diversion	Diversion from Yong Dam Reservoir
3	Instream Flow	12 Sub Basin
4	Domestic Water Supply Block (0-25%)	12 Sub Basin
5	Domestic Water Supply Block (25-50%)	12 Sub Basin
6	Domestic Water Supply Block (50-75%)	12 Sub Basin
7	Domestic Water Supply Block (75-100%)	12 Sub Basin
8	Industrial Water Supply Block (0-25%)	12 Sub Basin
9	Industrial Water Supply Block (25-50%)	12 Sub Basin
10	Industrial Water Supply Block (50-75%)	12 Sub Basin
11	Industrial Water Supply Block (75-100%)	12 Sub Basin
12	Agricultural Water Supply Block (0-25%)	12 Sub Basin
13	Agricultural Water Supply Block (25-50%)	12 Sub Basin
14	Agricultural Water Supply Block (50-75%)	12 Sub Basin
15	Agricultural Water Supply Block (75-100%)	12 Sub Basin
16	Domestic Water Supply Block (0-25%)	Diversion from Keum River and Diversion from Reservoirs
17	Domestic Water Supply Block (25-50%)	Diversion from Keum River and Diversion from Reservoirs
18	Domestic Water Supply Block (50-75%)	Diversion from Keum River and Diversion from Reservoirs
19	Domestic Water Supply Block (75-100%)	Diversion from Keum River and Diversion from Reservoirs
20	Industrial Water Supply Block (0-25%)	Diversion from Keum River and Diversion from Reservoirs
21	Industrial Water Supply Block (25-50%)	Diversion from Keum River and Diversion from Reservoirs
22	Industrial Water Supply Block (50-75%)	Diversion from Keum River and Diversion from Reservoirs
23	Industrial Water Supply Block (75-100%)	Diversion from Keum River and Diversion from Reservoirs
24	Agricultural Water Supply Block (0-25%)	Diversion from Keum River and Diversion from Reservoirs
25	Agricultural Water Supply Block (25-50%)	Diversion from Keum River and Diversion from Reservoirs
26	Agricultural Water Supply Block (50-75%)	Diversion from Keum River and Diversion from Reservoirs
27	Agricultural Water Supply Block (75-100%)	Diversion from Keum River and Diversion from Reservoirs
28	Agricultural Water Supply Block (75-100%)	Diversion from Keum River and Diversion from Reservoirs

5.3. Development of model

In the study, the objectives are to minimize the water demand deficits and reservoir spills, and to maximize the hydropower generation. These multiple objectives are combined into a single objective function for the dynamic programming optimization using the weighting method. Equation (5.1) shows the immediate reward (return) during every time with the weighting method.

$$\sum_{i=1}^{N_{Generator}} w_1 P_i - \sum_{j=1}^{N_{Diversion}} \sum_{k=1}^{N_{Share}} w_{2jk} \left(\frac{100 \times (D_{jk} - U_{jk})}{D_{jk}} \right)^2 - \sum_{l=1}^{N_{Reservoir}} w_3 SP_l \qquad (5.1)$$

where, w_1, w_{2jk}, w_3 are priority weighting factors for hydropower generation, diversions at the j^{th} diversion point and the k^{th} deficit sharing block, and reservoir spill, respectively; $N_{Generator}(=3)$ is the number of hydropower generators, $N_{Diverstion}(=25)$ is the number of diversion points, $N_{Share}(=4)$ is number of deficit sharing blocks, and $N_{Reservoir}(=2)$ is number of reservoirs; P_i is the hydropower generation at generator i and SP_l is the spilled water from reservoir l; D_{jk} is demand at diversion point j and deficit sharing block k and U_{jk} is the actual diversion water at diversion point j and deficit sharing block k.

Hydropower generation is generally calculated by the following equation

$$P_t = 9.8/3{,}600 \eta Q_t \overline{H}_t \qquad (5.2)$$

where P_t : hydropower production (GWh); η : turbine efficiency; $\overline{H}_t$: net head (m); Q_t : turbine Discharge (MCM). However, regression equations developed by historical hydropower generation data were used for calculating hydropower generation instead of using equation (5.2) since the turbine efficiencies are not available or are not accurate enough. Although the regression equations are linear, the non-linearity of the power equation is still preserved due to the non-linear relationship between storage and reservoir elevation. The regression equations for reservoir storage vs. elevation and hydropower generation appear in Tables 5.4 and 5.5. Other reservoir constraints are given in Table 5.6. The municipal and industrial demands used in the optimization are derived by averaging the demands in water years 2001 and 2002 to reflect the current situations of Keum River basin and they are given in Tables 5.7 and 5.8. The agricultural demands are the average of 19 years data from 1983 to 2002 and are given in Tables 5.9. The instreamflow for sub basin are the 95% exceedence percentile of 19-year historical flows for each sub basin and all other insreamflow requirements are provided by KOWACO.

Table 5.4 Regression equations for storage vs. elevation

	Equation	Remark	R^2 Value
YongDam	$H=aV^b$	a= 13.5987816, b=0.2425, H: elevation (m), V: reservoir volume(MCM)	0.9899 (19 data points)
DaeChung	$H=aV^b$	a= 174.2207985, b=0.0617, H: elevation (m), V: reservoir volume(MCM)	0.9983 (15 data points)

Table 5.5 Regression equations for hydropower generation

	Equation	Remark	R^2 Value
Yong Dam hydropower (JunJu)	z=a+bx+cy	a= -12977.807, b=70.905005, c=399.44509, x: elevation (m), y: turbine discharge (MCM/mon), z: power generation (Kwh)	0.9979 (15 data points)
Yong Dam hydropower (Keum)	z=a+bx+cy	a= -1303.9479, b= 29.878616, c= 90.919088, x: elevation (m), y: turbine discharge (MCM/mon), z: power generation(Kwh)	0.6243 (15 data points)
DaeChung hydropower	z=a+bx+cy	a= -23023.348, b= 324.95317, c= 101.71787, x: elevation (m), y: turbine discharge (MCM/mon), z: power generation(Kwh)	0.9897 (228 data points)
Dae Chung hydropower generation hours	$y=a+bx^{0.5}$	a= -77.224857, b=24.14873, x: total discharge (MCM/mon), y: hydropower generation hours for flood season only (July, August, September)	0.9323 (57 data points)
DaeChung turbine discharge	y=a+bx	a=-15.40454, b=0.81944428, x: hydropower generation hours	0.9863 (57 data points)

Table 5.6 Reservoir physical constraints

	Yong Dam	Dae Chung
Maximum turbine release	Jun Ju : 66 MCM/mon (24hr generation) (Turbine Release = 16.3538 + 0.18994 * Elevation) Keum : 60 MCM/mon (24hr generation) (Turbine Release = -135.5424 + 0.7774 * Elevation)	707 MCM (Variable peak time generation)
Minimum storage	68.75 MCM (228.5 m)	451.7 MCM (60.0 m)
Normal full storage	742.5 MCM (263.5 m)	1241.7 MCM (76.5 m)

Table 5.7 Municipal water demands in Keum River basin

Municipal (MCM/mon)	OCT	NOV	DEC	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP
B1	0.27	0.26	0.27	0.27	0.24	0.27	0.26	0.27	0.26	0.27	0.27	0.26
B2	0.27	0.26	0.27	0.27	0.24	0.27	0.26	0.27	0.26	0.27	0.27	0.26
B3	0.27	0.26	0.27	0.27	0.24	0.27	0.26	0.27	0.26	0.27	0.27	0.26
B4	1.07	1.04	1.07	1.07	0.97	1.07	1.04	1.07	1.04	1.07	1.07	1.04
B5	3.21	3.11	3.75	3.62	3.02	3.62	3.24	3.62	3.50	4.04	3.88	3.76
B6	19.82	18.27	18.88	17.81	16.93	18.75	18.14	19.82	19.18	20.89	21.83	20.22
B7	9.91	9.07	9.37	8.70	8.23	9.11	8.81	9.64	9.33	10.18	10.71	9.85
B8	1.34	1.30	1.34	1.34	1.21	1.34	1.30	1.34	1.30	1.34	1.61	1.30
B9	1.61	1.56	1.61	1.34	1.45	1.61	1.56	1.61	1.56	1.61	1.87	1.56
B10	1.07	1.04	1.07	1.07	0.97	1.07	1.04	1.07	1.04	1.07	1.07	1.04
B11	0.54	0.52	0.54	0.54	0.48	0.54	0.52	0.54	0.52	0.54	0.54	0.52
B12	0.54	0.52	0.54	0.54	0.48	0.54	0.52	0.54	0.52	0.54	0.54	0.52
Div01	21.92	25.69	13.79	23.80	21.66	23.91	17.06	11.93	21.84	10.23	13.14	13.02
Div02	18.60	18.38	16.94	20.10	16.46	17.85	19.56	19.22	17.22	20.25	19.31	18.28
Div03	2.97	3.08	2.71	3.17	2.74	2.93	3.05	2.94	2.63	2.94	2.84	2.80
Div04	8.23	8.04	7.99	7.81	7.12	8.13	8.13	8.95	9.22	9.69	9.69	9.15
KJ_mun	0.13	0.13	0.13	0.15	0.18	0.17	0.18	0.16	0.17	0.19	0.18	0.20
Div05	5.52	5.09	5.27	5.33	4.32	4.74	4.59	4.62	4.65	4.91	4.96	4.72

Table 5.8 Industrial water demands in Keum River basin

Industrial (MCM/mon)	OCT	NOV	DEC	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP
B1	0.54	0.52	0.54	0.54	0.48	0.54	0.52	0.54	0.52	0.54	0.54	0.52
B2	0.80	0.78	0.80	0.80	0.73	0.80	0.78	0.80	0.78	0.80	0.80	0.78
B3	0.54	0.52	0.54	0.54	0.48	0.54	0.52	0.54	0.52	0.54	0.54	0.52
B4	5.09	4.92	5.90	5.62	4.60	5.62	5.18	5.62	5.44	6.16	5.89	5.70
B5	7.37	7.13	8.45	8.04	6.53	8.04	7.52	8.04	7.78	8.84	8.57	8.29
B6	1.47	1.43	1.48	1.34	1.45	1.61	1.56	1.61	1.56	1.61	1.87	1.56
B7	2.28	2.20	2.28	2.14	2.18	2.41	2.33	2.41	2.33	2.41	2.68	2.33
B8	0.27	0.26	0.27	0.27	0.24	0.27	0.26	0.27	0.26	0.27	0.27	0.26
B9	0.27	0.26	0.27	0.27	0.24	0.27	0.26	0.27	0.26	0.27	0.27	0.26
B10	0.27	0.26	0.27	0.27	0.24	0.27	0.26	0.27	0.26	0.27	0.27	0.26
Div01	9.39	11.01	5.91	10.20	9.28	10.25	7.31	5.11	9.36	4.38	5.63	5.58
Div02	8.41	7.81	8.40	8.48	7.79	8.72	9.39	11.26	9.81	10.58	10.15	10.03
Div03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ShinTanJin	0.86	0.89	0.78	0.89	0.76	0.79	0.83	0.87	0.80	0.85	0.87	0.80
BooYong	0.25	0.24	0.22	0.25	0.22	0.23	0.26	0.26	0.26	0.30	0.28	0.27
Div05	2.36	2.18	2.26	2.28	1.85	2.03	1.97	1.98	1.99	2.10	2.13	2.02

Table 5.9 Agricultural water demands in Keum River basin

Agricultural (MCM/mon)	OCT	NOV	DEC	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP
B1	0.00	0.00	0.00	0.00	0.00	0.00	1.17	13.11	19.25	10.93	20.23	7.29
B2	0.00	0.00	0.00	0.00	0.00	0.00	0.77	8.92	13.25	7.84	13.59	4.92
B3	0.00	0.00	0.00	0.00	0.00	0.00	0.47	5.47	7.87	4.7	7.95	2.86
B4	0.00	0.00	0.00	0.00	0.00	0.00	1.01	11.67	17.09	10.91	18.52	6.03
B5	0.00	0.00	0.00	0.00	0.00	0.00	1.81	21.08	31.27	19.59	32.28	11.99
B6	0.00	0.00	0.00	0.00	0.00	0.00	2.3	26.25	38.8	26.54	44.02	16.38
B7	0.00	0.00	0.00	0.00	0.00	0.00	3.84	42.47	64.74	36.53	70.45	27.34
B8	0.00	0.00	0.00	0.00	0.00	0.00	1.54	17.24	26.08	16.43	28.74	11.12
B9	0.00	0.00	0.00	0.00	0.00	0.00	3.35	37.86	57.3	34.98	61.99	24.35
B10	0.00	0.00	0.00	0.00	0.00	0.00	1.35	15.34	22.36	14.27	26.02	9.56
B11	0.00	0.00	0.00	0.00	0.00	0.00	2.19	24.77	36.81	24.16	42.5	16.19
B12	0.00	0.00	0.00	0.00	0.00	0.00	2.8	31.9	47.78	33.5	55.59	20.8
KJ_agr	0.00	0.00	0.00	0.00	0.00	0.00	0.88	3.96	37.99	25.07	21.85	18.04
KA_agr	0.00	0.00	0.00	0.00	0.00	0.00	1.29	5.80	55.70	36.76	32.05	26.45
KK_agr	0.00	0.00	0.00	0.00	0.00	0.00	0.69	3.09	29.69	19.59	17.08	14.10

Table 5.10 Instream water demands in Keum River basin

Instream Flow (MCM/mon)	OCT	NOV	DEC	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP
B1	3.11	3.11	3.11	3.11	3.11	3.11	3.11	3.11	3.11	3.11	3.11	3.11
B2	2.07	2.07	2.07	2.07	2.07	2.07	2.07	2.07	2.07	2.07	2.07	2.07
B3	1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30
B4	3.63	3.63	3.63	3.63	3.63	3.63	3.63	3.63	3.63	3.63	3.63	3.63
B5	4.15	4.15	4.15	4.15	4.15	4.15	4.15	4.15	4.15	4.15	4.15	4.15
B6	2.60	2.60	2.60	2.60	2.60	2.60	2.60	2.60	2.60	2.60	2.60	2.60
B7	6.22	6.22	6.22	6.22	6.22	6.22	6.22	6.22	6.22	6.22	6.22	6.22
B8	2.07	2.07	2.07	2.07	2.07	2.07	2.07	2.07	2.07	2.07	2.07	2.07
B9	3.11	3.11	3.11	3.11	3.11	3.11	3.11	3.11	3.11	3.11	3.11	3.11
B10	1.56	1.56	1.56	1.56	1.56	1.56	1.56	1.56	1.56	1.56	1.56	1.56
B11	1.81	1.81	1.81	1.81	1.81	1.81	1.81	1.81	1.81	1.81	1.81	1.81
B12	1.81	1.81	1.81	1.81	1.81	1.81	1.81	1.81	1.81	1.81	1.81	1.81
Yong Dam	12.96	12.96	12.96	12.96	12.96	12.96	12.96	12.96	12.96	12.96	12.96	12.96
Kyu Am	103.68	103.68	103.68	103.68	103.68	103.68	103.68	103.68	103.68	103.68	103.68	103.68

Unlike other reservoir optimization model there is no simplification of the simulation procedure representing the basin as precise as possible. It includes hydropower generation routine, reservoir water allocation routine, and sub basin water allocation routine. DaeChung reservoir water allocation routine and sub basin water allocation routine require the optimization routine to represent the deficit sharing policy and priority system in the basin. The deficit sharing policy allocation is based on the equation (5.3).

$$\arg\min_{U_{jk}} \sum_{j=1}^{N_{Diversion}} \sum_{k=1}^{N_{Share}} w_{2k} \left(\frac{100 \times (D_{jk} - U_{jk})}{D_{jk}} \right)^2 \tag{5.3}$$

$$\sum_{k=1}^{N_{Share}} U_{jk} \le \sum_{k=1}^{N_{Share}} D_{jk} \qquad \text{for } j = 1, 2, \cdots, N_{Diversion} \tag{5.4}$$

$$U_{jk+1} = 0 \text{ if } U_{jk} < D_{jk} \quad \text{for } j = 1, 2, \cdots, N_{Diversion};\ k = 1, 2, \cdots, N_{Diversion} - 1 \tag{5.5}$$

where $N_{Diverstion}(=25)$: the number of diversion points; $N_{Share}(=4)$: the number of deficit sharing blocks; D_{jk} : the demand at diversion point j and deficit sharing block k; U_{jk} : the actual diversion at each diversion point and each deficit sharing block. Water balance equation for each control points such as diversion point and inflow point is in equation (5.6). Any loses such as channel loss is ignored in the study.

$$q_{down} = q_{up} + q_{in} + q_{return} - \sum_{k=1}^{N_{share}} U_k \tag{5.6}$$

$$\sum_{k=1}^{N_{share}} U_k < q_{up} + q_{in} + q_{return} \tag{5.7}$$

where q_{down} and q_{up}: downstream and upstream control point flows; q_{in} and q_{return}: local downstream inflow and return flow from upstream water uses; U_k : the diversion at downstream control point for the deficit sharing block k.

5.4. Development of Optimal Operational Policies

Various stochastic dynamic programming approaches were applied to developing optimal coordinated operating rules for the two reservoir system of the Keum River basin. Three stochastic dynamic programming approaches were considered including implicit stochastic dynamic programming, SSDP, and reinforcement learning. The conventional SDP is excluded due to the complexity of deriving transition probability of 12 sub basin inflows. Several options for defining the streamflow transition and discount factors in SSDP and reinforcement learning are tested.

5.4.1. Implicit Stochastic dynamic programming

5.4.1.1. Streamflow generation by SAMS

A number of stochastic streamflow generation models have been suggested, such as HEC-4 (U.S. Army Corps of Engineers, 1971), LAST (Lane and Frevert, 1990), SPIGOT (Grygier and Stedinger, 1990) and SAMS (Salas et al, 2000) are available. Large and

complex water resource systems such as the Keum River basin that include multiple reservoirs, diversions, and inflows require generation of multivariate periodic hydrologic time series that considers both spatial and temporal correlation structures.

The most widely used multivariate periodic time series models are multivariate PAR (periodic autoregressive) and PARMA (periodic autoregressive moving average) and Contemporaneous PAR and PARMA. Although models can reproduce the periodic means, standard deviations, skewness, and periodic correlations, these models may not be able to reproduce the annual characteristics (Salas, 1980). When it is necessary to preserve the statistical characteristics of both annual and seasonal time scales, then disaggregation modeling approaches are recommended (Salas, 1980). The dissagregation approach was originally presented by Valencia and Schaake (1973), which was then modified by Mejia and Rousselle (1976) who modified the basic Valencia and Schaake model to preserve the covariances of the first season of a year and any preceding season (Salas, 1993). Lane (1981) proposed a disaggregation approach in the temporal domain.

In this study, disaggregation approach (Spatial: Mejia and Rousselle, Temporal: Lane) in SAMS (Salas et. al, 2000) and 19 years of historical data provided by KOWACO (Korea Water Resources Corporation) were used to generate the ten sets of 50-year monthly streamflow data. Some the basic statistics of the generated data are given on Figure 5.4. The compared statistics include monthly mean, standard deviation, coefficient of variation, skewness, minimum, maximum, annual lag-1 auto correlation function (ACF) for each sub basin inflows, and season-to-season correlation. Most of the statistics are reasonably well reproduced by the disaggregation model. The generated minimum values are lower than those of the historical data and the generated maximum values are

higher. The annual lag-1 auto correlation function for each sub basin inflows shows they are not statistically significant.

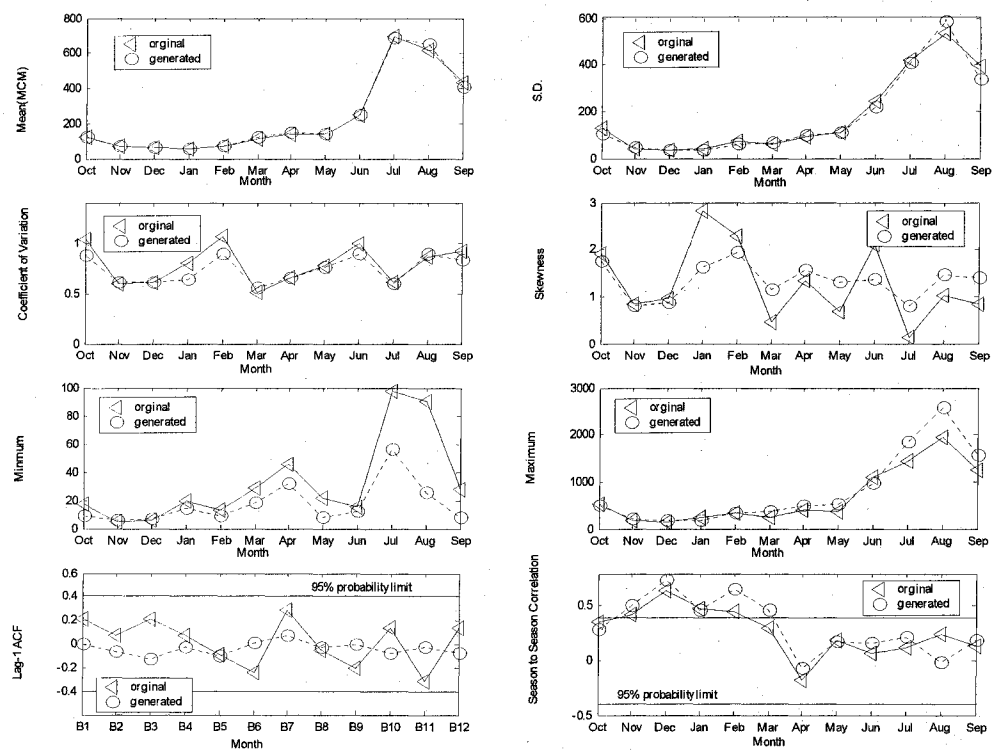


Figure 5.4 Basic statistics of generated data

Annual statistics are compared in the following tables include mean, standard deviation, coefficient of variation, skewness, minimum, maximum. Most of the statistics are reasonably well reproduced by the disaggregation model. The minimum and maximum values are not comparable since the length of the historical data is reasonably short. However, the generated minimum values are lower than those of historical and the generated maximum values are higher than those of historical.

Table 5.11 Annual statistics comparison of sub basin inflow

	Mean (MCM/mon)		Standard Deviation (MCM/mon)		Coefficient of Variation		Skewness	
	Historical	Generated	Historical	Generated	Historical	Generated	Historical	Generated
B1	732.66	769.74	535.31	605.45	0.73	0.78	1.91	1.47
B2	342.50	363.94	112.55	186.37	0.32	0.51	0.01	0.96
B3	256.46	272.00	91.314	128.59	0.35	0.47	0.41	0.75
B4	704.77	783.42	240.94	359.15	0.34	0.45	0.09	0.61
B5	806.06	859.03	273.73	392.25	0.33	0.45	-0.08	0.66
B6	519.72	505.65	221.66	236.28	0.42	0.46	0.65	0.61
B7	1078.1	1020.90	360.58	438.43	0.33	0.42	0.39	0.56
B8	535.45	527.84	233.40	238.34	0.43	0.45	0.30	0.66
B9	642.60	587.26	243.38	273.17	0.37	0.46	0.88	0.72
B10	317.68	313.67	136.05	167.37	0.42	0.53	0.54	0.88
B11	358.58	329.96	151.40	156.28	0.42	0.47	1.06	0.80
B12	336.29	317.87	164.79	167.51	0.49	0.52	0.89	1.05
DC	2773.10	2780.00	1114.6	1093.90	0.40	0.39	0.08	0.33

	Minimum (MCM/mon)		Maximum (MCM/mon)	
	Historical	Generated	Historical	Generated
B1	152.84	6.16	2465.00	4044.80
B2	163.33	31.26	539.59	1099.50
B3	114.65	22.55	444.21	776.95
B4	316.96	71.16	1111.00	2126.90
B5	333.05	107.69	1198.00	2494.10
B6	202.11	47.52	964.20	1349.80
B7	481.32	129.23	1862.70	2453.90
B8	200.95	58.00	990.40	1586.40
B9	284.77	54.76	1283.90	1681.50
B10	114.45	16.96	604.24	1010.00
B11	153.29	35.20	758.17	1079.90
B12	117.72	24.39	681.45	1103.70
DC	968.02	299.84	4946.90	5882.60

5.4.1.2. Deterministic Dynamic Programming and Regression Analysis

Development of the optimal rules for integrated operation of the two reservoirs was performed using the generalized dynamic programming software package CSUDP (Labadie, 2003). Stochastically generated inflow sequences are treated deterministically in the dynamic programming optimization, with optimal operations found for each sequence. Statistical analysis is then performed on the optimal results to derive monthly operational rules. Subroutines for CSUDP were developed to represent the two-reservoir system, the multiple demand points, and the differing priorities of water usage. Backwards DP recursion equation and the system state dynamics equation for noniverted form used in CSUDP is as follows.

$$V_t(\vec{s}_t) = \max_{\vec{s}_{t+1}}\{\vec{r}_t + \gamma V_{t+1}(\vec{s}_{t+1})\} \tag{5.8}$$

$$\vec{a}_t = \vec{s}_t + \vec{i}_t - \vec{s}_{t+1} \tag{5.9}$$

$$\vec{s}_{t+1}^{\min} \le \vec{s}_{t+1} \le \vec{s}_{t+1}^{\max} \tag{5.10}$$

The CSUDP model was run with ten sets of 50-year monthly data. The first and last 5 years in each set were eliminated to minimize impacts of boundary conditions. The optimal releases and storage results were organized by months and analyzed using multiple linear and non-linear regressions to develop monthly operational rules. Tables 5.12 and 5.13 show the regression equation for Yongdam and DaeChung. Most of these

equations have high R^2 Values which indicates the regression model provide good prediction for data points. Examples of the results are shown in Figures 5.5 and 5.6.

Table 5.12 YongDam CSUDP Rule Equations by Regression

(x: YD Inflow, y: YD Beginning Storage, z: YD Ending Storage, q: DC release)

Month	Equation	Remark	R^2 Value
October	z=a+bx+cy	a=6.0404324, b=0.5547587, c=0.93146005	0.987
November	z=a+bx+cy	a=6.6116753, b=0.56476797, c=0.92188506	0.985
December	z=a+bx+cy	a=6.2166042, b=0.58636115, c=0.92415844	0.986
January	z=a+bx+cy	a=6.7187947, b=0.61352824, c=0.91218659	0.984
February	z=a+bx+cy	a=5.2201193, b=0.81094379, c=0.90415502	0.986
March	z=a+bx+cy	a=6.3464651, b=0.66014198, c=0.90950869	0.982
April	z=a+bx+cy	a=7.7658821, b=0.62957685, c=0.89240777	0.978
May	z=a+bx+cy	a=7.5550822, b=0.62037185, c=0.89668031	0.972
June	q=a+bx+cy	a=-395.15696, b=0.6968931, c=0.85517116 (Above 400 MCM Inflow)	0.946
	z=a+bx+cy	a=6.316589, b=0.7380634,7 c=0.89894787 (Below 400 MCM Inflow)	0.978
July	q=a+bx+cy	a=-682.81763, b=0.96913508, c=0.95640806 (Above 500 MCM Inflow)	0.989
	z=a+bx+cy	a=17.246056, b=0.74662692, c=0.89691337 (Below 500 MCM Inflow)	0.932
August	q=a+bx+cy	a=-661.07909, b=0.97428142, c=0.89072911 (Above 400 MCM Inflow)	0.993
	z=a+bx+cy	a=5.9283342, b=0.71996397, c=0.91325633 (Below 400 MCM Inflow)	0.917
September	q=a+bx+cy	a=-709.90178, b=0.97008485, c=0.97535359 (Above 300 MCM Inflow)	0.996
	z=a+bx+cy	a=4.8738803, b=0.6568893, c=0.93536306 (Below 300 MCM Inflow)	0.973

Table 5.13 DaeChung CSUDP Rule Equations by Regression

(x: DC Inflow, y: DC Beginning Storage, z: DC Ending Storage q: DC release)

Month	Equation	Remark	R^2 Value
October	$z=a+bx+cy$	a=85.760531, b=0.45927192, c=0.82831452	0.945
November	$z=a+bx+cy$	a=93.801851, b=0.40780308, c=0.81305869	0.958
December	$z=a+bx+cy$	a=104.81938, b=0.2592738, c=0.7966079	0.955
January	$z=a+bx+cy$	a=113.6369, b=0.42440024, c=0.76782586	0.954
February	$z=a+bx+cy^{2.5}$	a=391.90274, b=0.61055865, c=1.4745765e-05	0.925
March	$z=a+bx^{1.5}+cy^{2}$	a=356.70833, b=0.026892917, c=0.0005433317	0.943
April	$z=a+bx^{2}+cy^{2}$	a=365.46579, b=0.0011661811, c=0.00051173733	0.951
May	$z=a+bx^{1.5}+cy^{2.5}$	a=403.82406, b=0.011877325, c=1.5011611e-05	0.929
June	$z=a+bx+cy$	a=-217.50498, b=1.046812, c=1.0700347 (Above 600 MCM Beginning Storage)	0.948
		a=-59.253794, b=0.55337194, c=0.8928085 (Below 600 MCM Beginning Storage)	0.746
July	$q=a+bx^{2}+cx^{0.5}$	a=0.25508825, b=0.00020297886, c=8.205721	0.701
August	$q=a+bx^{1.5}+cy$	a=-97.186854, b=0.01373869, c=0.21572517	0.936
September	$q=a+bx+cy+dx^{2}+ey^{2}+f xy$	a=-83.329333, b=-0.31041706, c=0.21579538, d=0.00027483596, e=-6.8172136e-05, f=0.00064852052	0.943

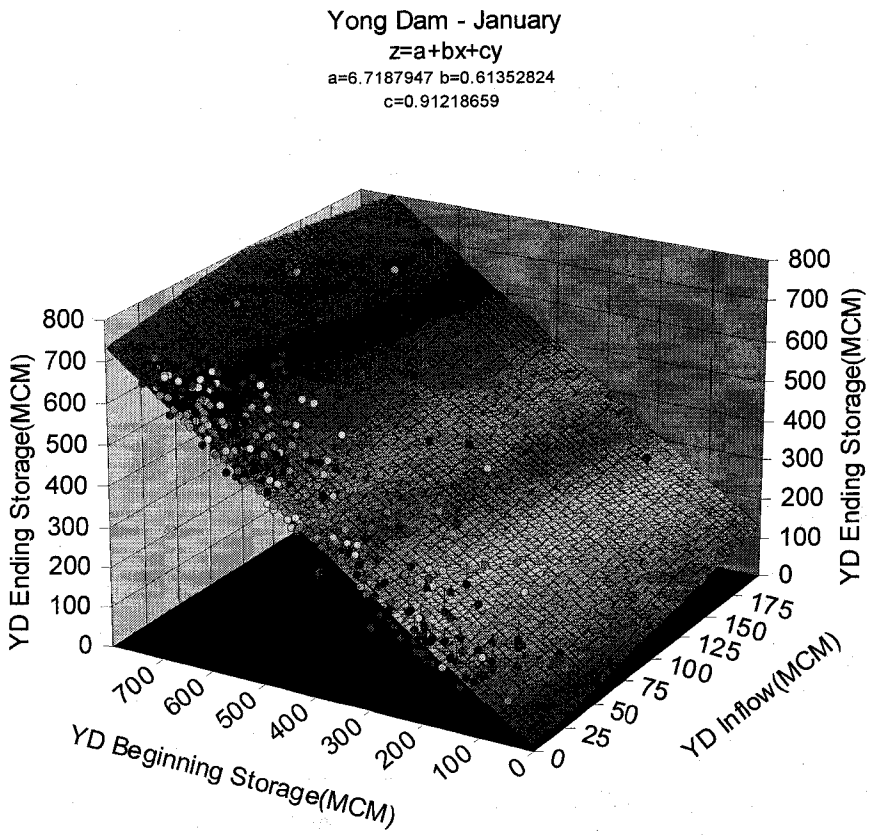


Figure 5.5 Fitting the regression rules for the integrated operation at Yong Dam reservoir (January)

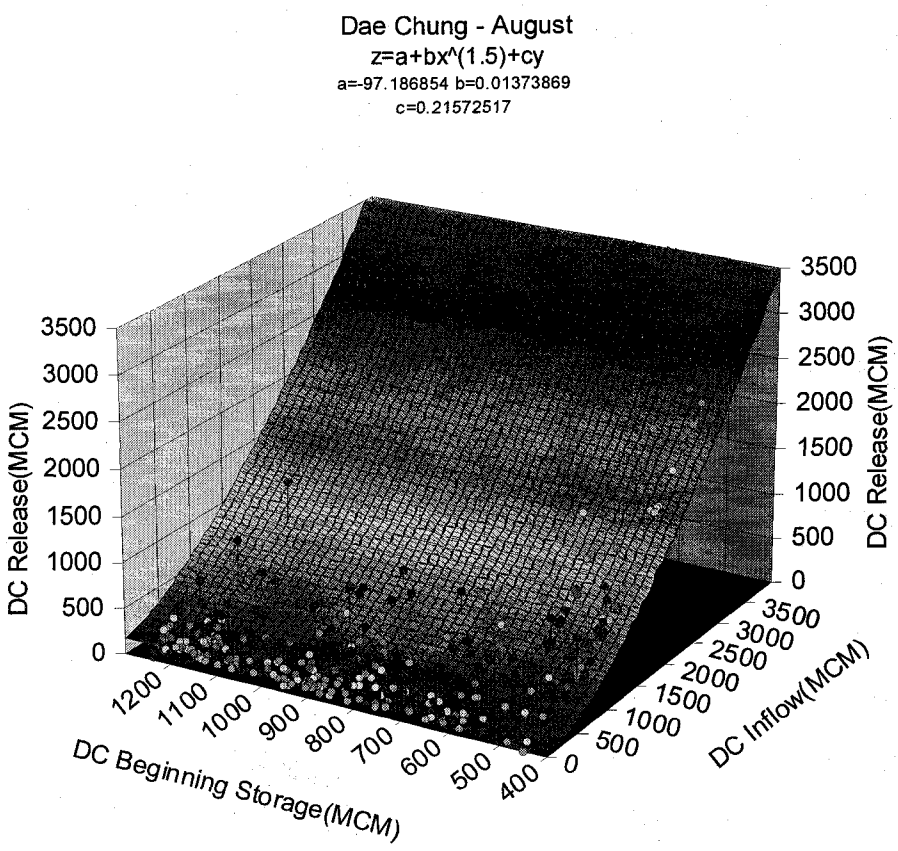


Figure 5.6 Fitting the regression rules for the integrated operation at Dae Chung reservoir (August)

5.4.2. Model Description for Explicit Stochastic Dynamic Programming

Although the accuracy of dynamic programming optimization depends on the size of the discretization intervals and discretization scheme, a small number of state discretization is required to avoid excessive computational time. Klemes (1977) derived some absolute constraints the minimum number of discretization in order to assure a satisfactory result. The actual number of discrete storage states N used in DP optimization is required to be larger than the following minimum values suggested by Klemes constrains.

$$N \geq \left[\frac{C}{2(D - x_{\min})} + 2 \right] \tag{5.11}$$

$$N \geq \left[\frac{C}{2(x_{\max} - D)} + 2 \right] \tag{5.12}$$

where C: active storage capacity; D: reservoir draft (release); and $x_{\min}$ and $x_{\max}$: minimum and maximum reservoir inflows. The active storage capacities of DaeChung and YoungDam reservoirs are 790 and 684 million cubic meters (MCM), respectively. The reservoir drafts are determined by Tables 5.7 to 5.10, and they are 40 MCM for YoungDam reservoir and 38 MCM for DaeChung reservoir. The minimum and maximum reservoir inflows are (0.17 MCM, 623.38 MCM) for YoungDam reservoir and (12.4 MCM, 1061.4 MCM) for DaeChung reservoir. The minimum number of discretization for YongDam is 11 and DaeChung requires 17 discretizations.

In the Q-Learning and SSDP models, the reservoir storage states are discretized into 21 grid points for YongDam reservoir and 24 grid points for DaeChung reservoir representing 34 MCM. YoungDam release is evenly discretized into grid value by 10 MCM from 3 to 133 MCM (14 points). DaeChung release is unevenly discretized into 27 points from 17 to 707 million cubic meters, with 15 MCM is used for grid value up to 212 MCM and then 40 MCM is used after 212 MCM. Spill of YongDam reservoir is allowed to Keum River basin at the normal full storage level of 263.5 m (742.57 MCM) from September to May and flood control level of 261.5 m (672.84 MCM) from June to August. For DaeChung reservoir, the normal full storage level of 76.5 m (1241.7 MCM) is set for the entire year. The flood control level of DaeChung reservoir originally was set during the flood season but it is now set to the normal full storage level after the construction of YongDam reservoir.

Although the available data covers a 19-year period, only 15 years of data from October 1983 to September 1998 is used for generating operation policies in the Q-Learning and SSDP models and 4 years of data from October 1999 to October 2002 is left for policy performance evaluation. Historical estimated agricultural demands provided by KOWACO are used for the optimization instead of using averaged agricultural demand.

The primary difficulty in applying dynamic programming to multireservoir systems is the curse of dimensionality. In addition, water allocation procedure for the deficit sharing policy and priority system in the basin requires considerable computational time. Since it is computationally intractable to update all the state value function values with such a procedure, one novel idea is to use the pre-calculated immediate benefit (reward).

Fortunately, this is possible since Q-Learning and SSDP models utilize the finite discretized grid points of state and decision spaces. The computational time used to calculate all possible immediate reward values for all the possible grid points, including reservoir storages, releases, months, and years, is almost four hours of computer time on a 1 GHz desktop workstation. If this routine is directly imbedded in the Q-Learning and SSDP models, this computational time should be multiplied by the number of iteration required for convergence. This pre-calculation of immediate reward values can separate the feasible decisions from all possible decisions.

The application of Q-Learning and SSDP requires the interpolation between the value function points of interests. Various forms of function approximation techniques have been developed such as polynomial and spline approximation (Foufoula-Georgiou, 1991; Johnson, 1993). A linear interpolation can be appropriate for approximately functions but the computation burden of dynamic programming is increased in the case of multiple states even with this simple interpolation approach. Alternatively, the discrete values of interest can be estimated by the nearest discrete point. In this study, the linear interpolation method and the nearest discrete point method are used to save computational time.

5.4.3. Sampling Stochastic Dynamic Programming

The sampling stochastic dynamic programming model presented in equation (4.10) is used for this study. Although SSDP is mostly used for on-line operation with forecasted streamflows, it can also be applied to off-line policy generation with historical streamflow sequences. The most difficult aspect of applying SSDP is defining future

value of reservoir storage. However, this problem can be solved by discount weighting factors. Formulation of the SSDP model for this study is

$$\max_{a_t}\{W_1 r_t + \frac{W_2}{M}\sum_{m=1}^{M} V_{t+1}(\vec{s}_{t+1}, k)\} \tag{5.13}$$

$$V_t(\vec{s}_t, j) = W_1 r_t + W_2 V_{t+1}(\vec{s}_{t+1}, j) \tag{5.14}$$

$$\vec{s}_{t+1} = \vec{s}_t + \vec{i}_t - \vec{a}_t \tag{5.15}$$

$$\vec{s}_{t+1}^{\min} \le \vec{s}_{t+1} \le \vec{s}_{t+1}^{\max} \tag{5.16}$$

$$\vec{a}_{t+1}^{\min} \le \vec{a}_{t+1} \le \vec{a}_{t+1}^{\max} \tag{5.17}$$

$$W_1 + W_2 = 1 \tag{5.18}$$

where, t : time period ; $\vec{s}_t$: reservoir storage vector for period t; $\vec{a}_t$: reservoir release vector for period t; $\vec{i}_t$: reservoir inflow vector for period t; j: streamflows of a particular historical year up to period t; ; k: streamflows of another historical year that could occur after period t; W_1 and W_2 : weighting factors for immediate reward and future value function; and γ : discount factor

Examples of operation rules for January and August are shown in Figures 5.7 and 5.8, which show that YongDam reservoir operation is readably independent of DaeChung reservoir storage due to the structural restriction on the YongDam reservoir release as well as the large amount of transbasin water release to Jun-Ju region. However, increased YongDam reservoir releases are made in January (one of dry month) when DaeChung reservoir has low storage level. When the discount factor of 0.95 is used W_1 and W_2 are set to 0.05 and 0.95 respectively.

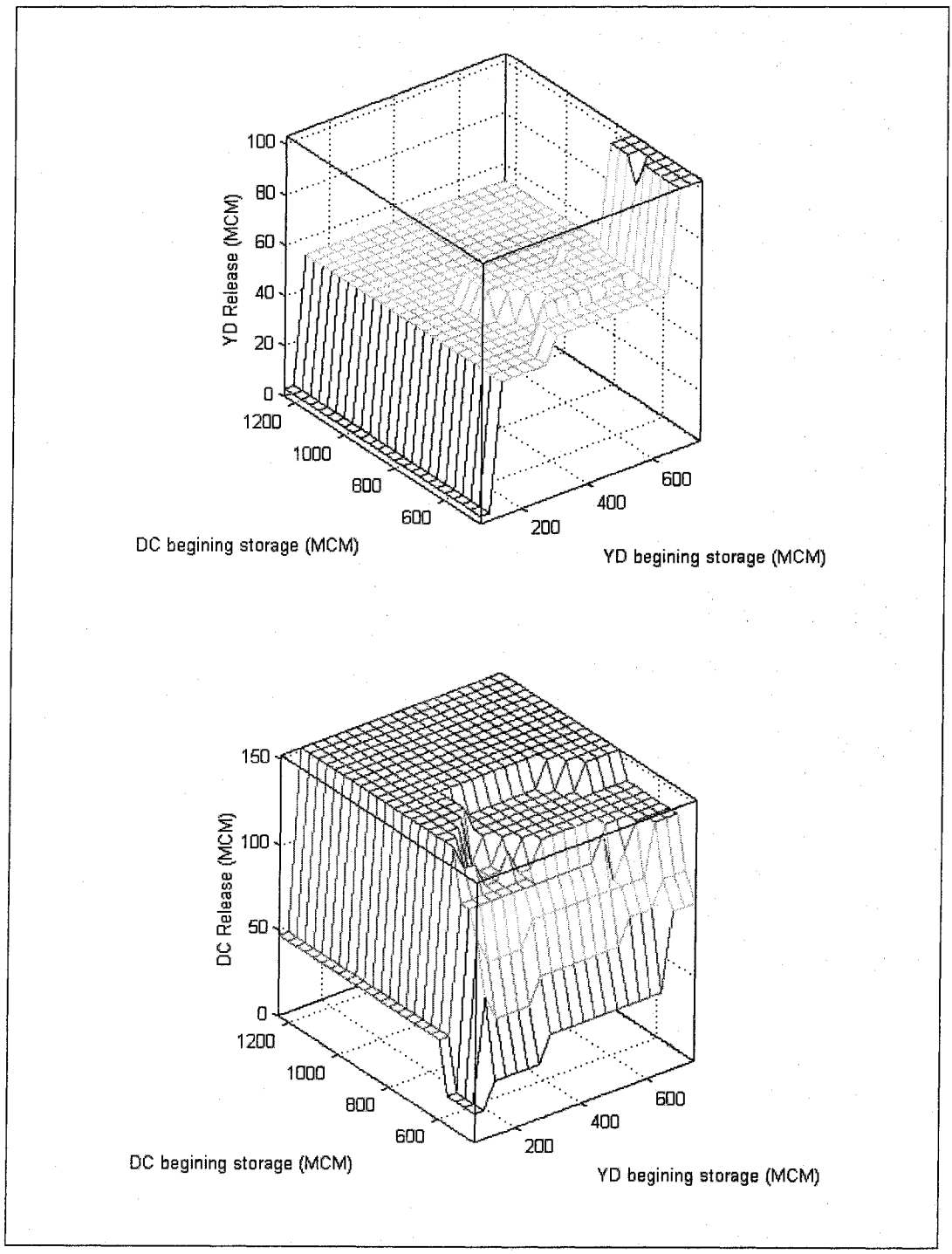


Figure 5.7 SSDP operation rules by discount factor of 0.95 (January)

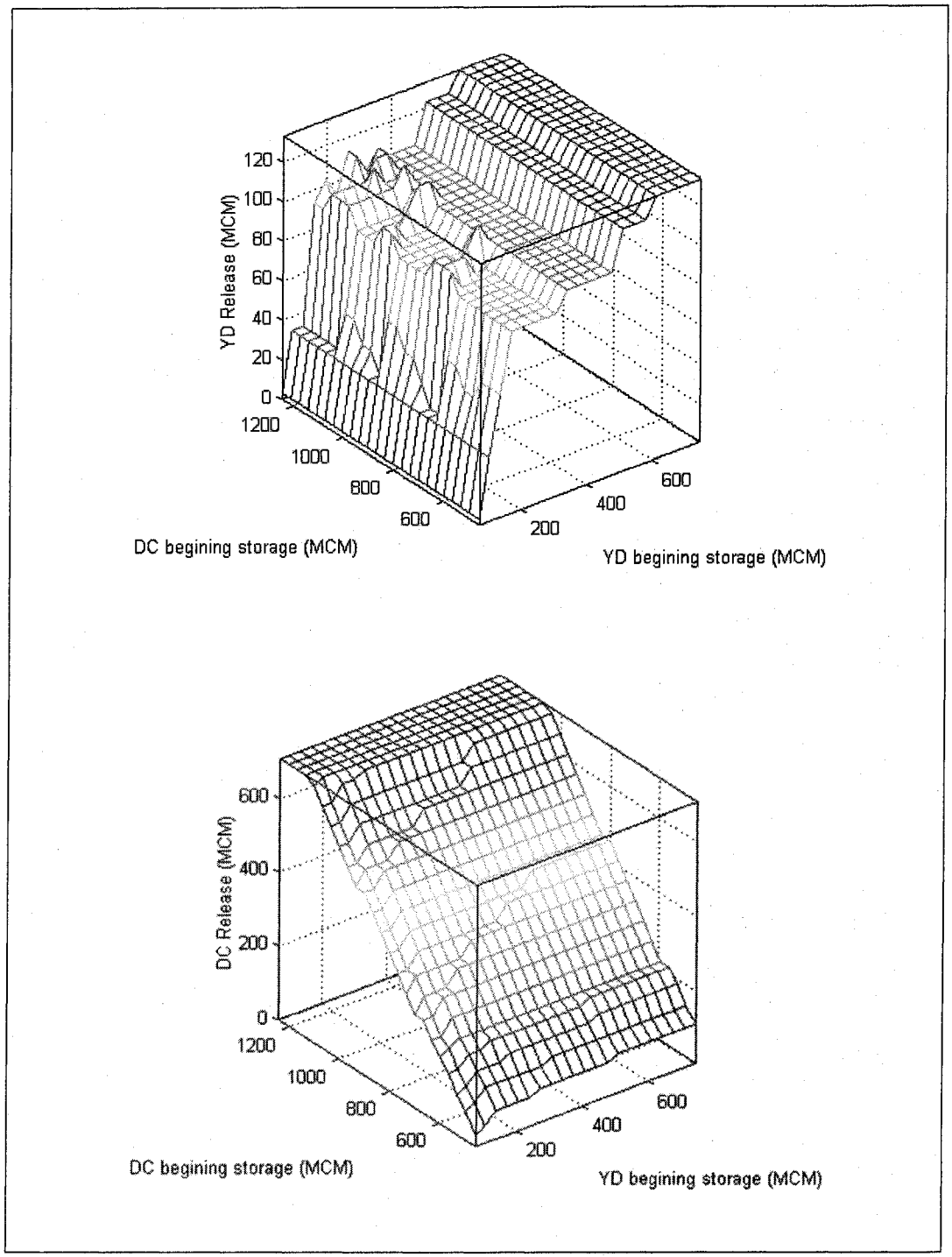


Figure 5.8 SSDP operation rules by discount factor of 0.8 (August)

5.4.4. Unconditional Reinforcement Learning (Q-Learning) Model

Formulation of unconditional Q-Learning with weighting factors is given in equations (5.19) through (5.21). Equations (5.13) through (5.18) are not repeated here.

$$Q(\vec{s}_t, \vec{a}_t) = \underset{\vec{I}_t}{E}\left[W_1 r_t + W_2 \max_{\vec{a}_{t+1}} Q(\vec{s}_{t+1}, \vec{a}_{t+1}) \right] \tag{5.19}$$

$$\vec{a}_t^* = \arg\max_{\vec{a}_t} Q(\vec{s}_t, \vec{a}_t) \tag{5.20}$$

$$W_1 + W_2 = 1 \tag{5.21}$$

where $Q(\vec{s}_t, \vec{a}_t)$ is action value function for unconditional inflow state for period t.

One example of operation rules for January and August with discount factor of 0.95 are shown on Figure 5.9 and 5.10. Other operation rules with different discount factors are similar patterns. The YongDam reservoir release rules are independent of DaeChung reservoir storage due to the structural restriction on the YongDam reservoir release as well as the large amount of transbasin water release to Jun-Ju region. However, The DaeChung reservoir release rules are influenced by YongDam reservoir storage since the more release from YongDam reservoir is made by policy rules. Less release is determined when the reservoir level is high comparing to SSDP operation rules.

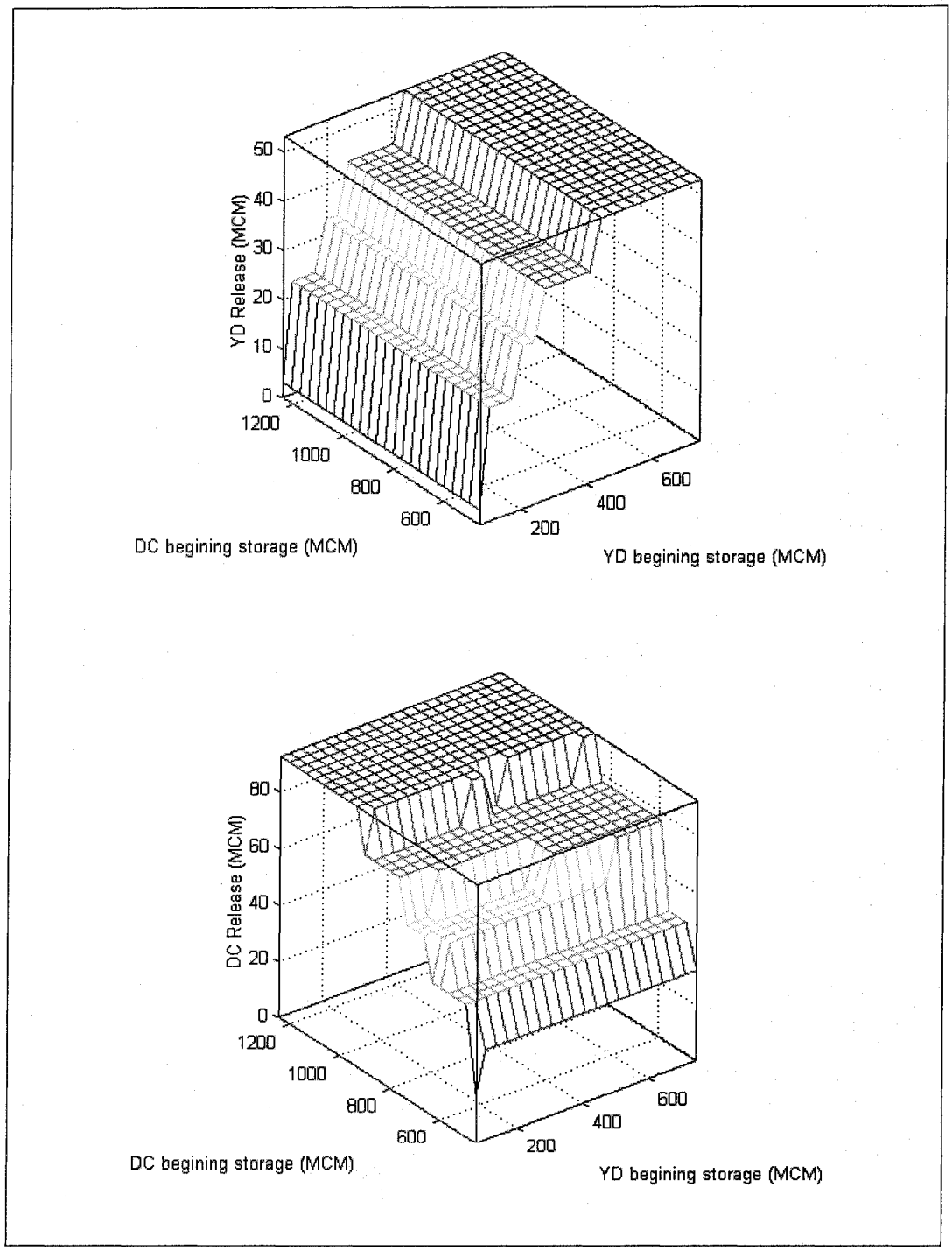


Figure 5.9 Q-Learning operation rules (January, discount=0.95)

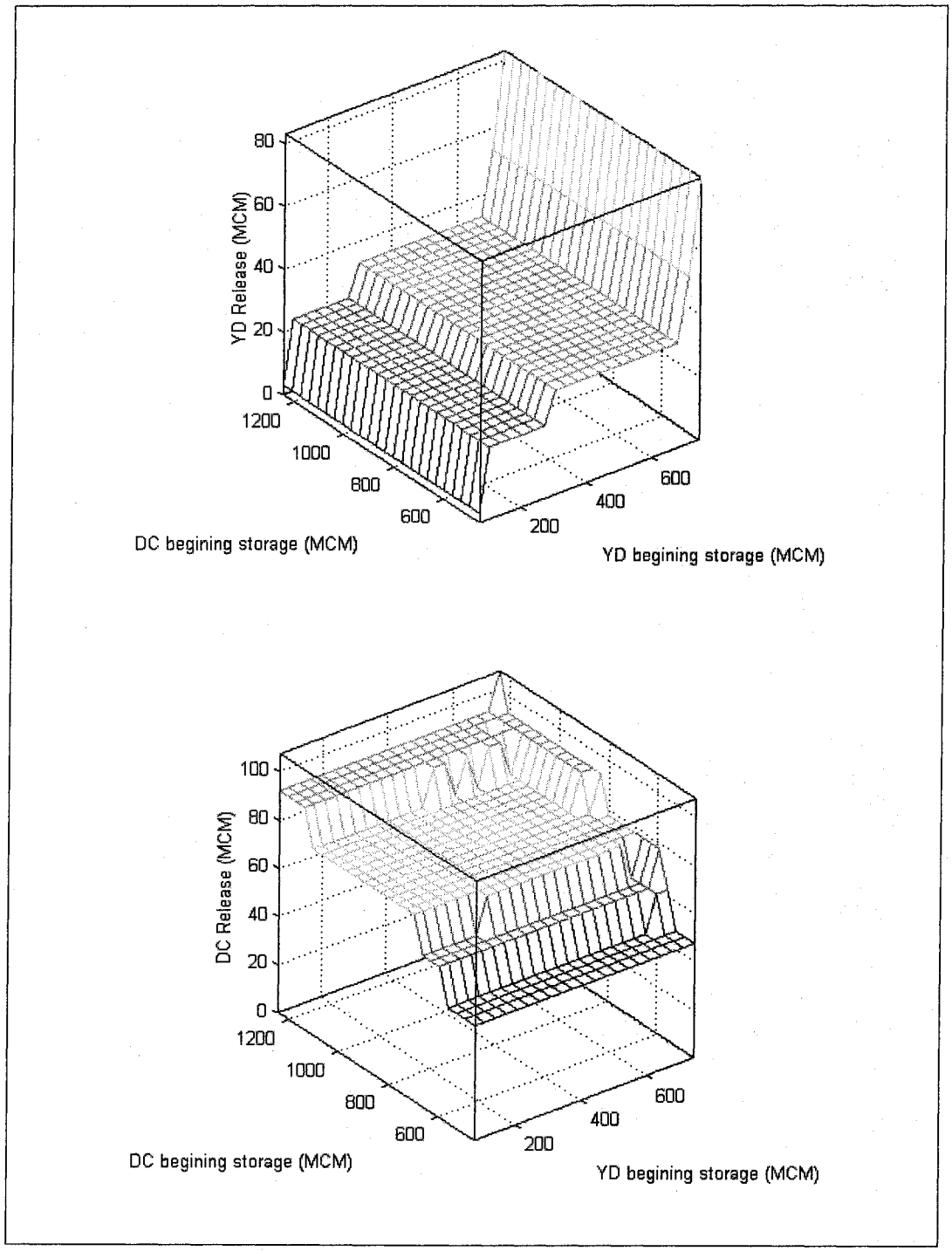


Figure 5.10 Q-Learning operation rules (August, discount=0.95)

5.4.5. Conditional Reinforcement Learning (Q-Learning) Model

Formulation of conditional Q-Learning is given in the equations (5.22) and (5.23). Equations (5.13) through (5.18) are not repeated here.

$$Q(\vec{s}_t, \vec{I}_t, \vec{a}_t) = \underset{\vec{I}_t(\vec{I}_t)}{E}\left[W_1 r_t + W_2 \max_{\vec{a}_{t+1}} Q(\vec{s}_{t+1}, \vec{I}_{t+1}, \vec{a}_{t+1}) \right] \tag{5.22}$$

$$\vec{a}_t^* = \arg\max_{\vec{a}_t} Q(\vec{s}_t, \vec{I}_t, \vec{a}_t) \tag{5.23}$$

where $Q(\vec{s}_t, \vec{I}_t, \vec{a}_t)$ is action value function for conditional inflow state for period t and $\vec{I}_t$ is hydrologic state vector at period t.

K-mean clustering and percentile value approaches are used to define hydrologic states in the system. Instead of considering all the individual streamflows from each sub basin, three sets of aggregated flows are used for defining the hydrologic state for simplicity. The first aggregated flow is YongDam reservoir natural inflow, designated as sub basin 1. The second is the total inflow to the DaeChung reservoir, except for the YongDam release. This includes natural inflows from sub basin 2 to 5 located between YonDam reservoir and DaeChung reservoir. The third is the DaeChung downstream flows, involving the natural flows of sub basin 6 to 12.

The percentile approach uses the 25^{th} and 75^{th} percentile values of each of the aggregated flows to define three hydrologic states as dry, average, and wet. Only three hydrologic states are defined here due to the small number of data available. Weighting factors are used in both approaches to determine the hydrologic state of inflows. Higher

weighting factors are assigned to the DaeChung and YongDam reservoir inflows than for DaeChung downstream flow in order to reflect higher valuation of reservoir inflows. The hydrologic states for August during the flood season for the two approaches are shown in the Figures 5.11 and 5.12. The non-flood season example is shown in the Figures 5.13 and 5.14. Both approaches to hydrologic state definition are used for developing the conditional operational rules in Q-Learning.

The reservoir operation rules for various hydrologic conditions and discount factors are shown on Figures 5.15 though 5.19. The influence of YongDam reservoir storage can be seen in the release rules for DaeChung reservoir, and low discount factors result in increased releases due to the lower valuation of future reward.

The relationship between number of updates of the state-action value function and mean absolute error is shown in Figure 5.20. The optimal mean absolute error is calculated by comparing the current state-action value function and the optimal state-action value function. The results show that high discount factor requires more iteration since the current action-value functions considers more future action-value functions.

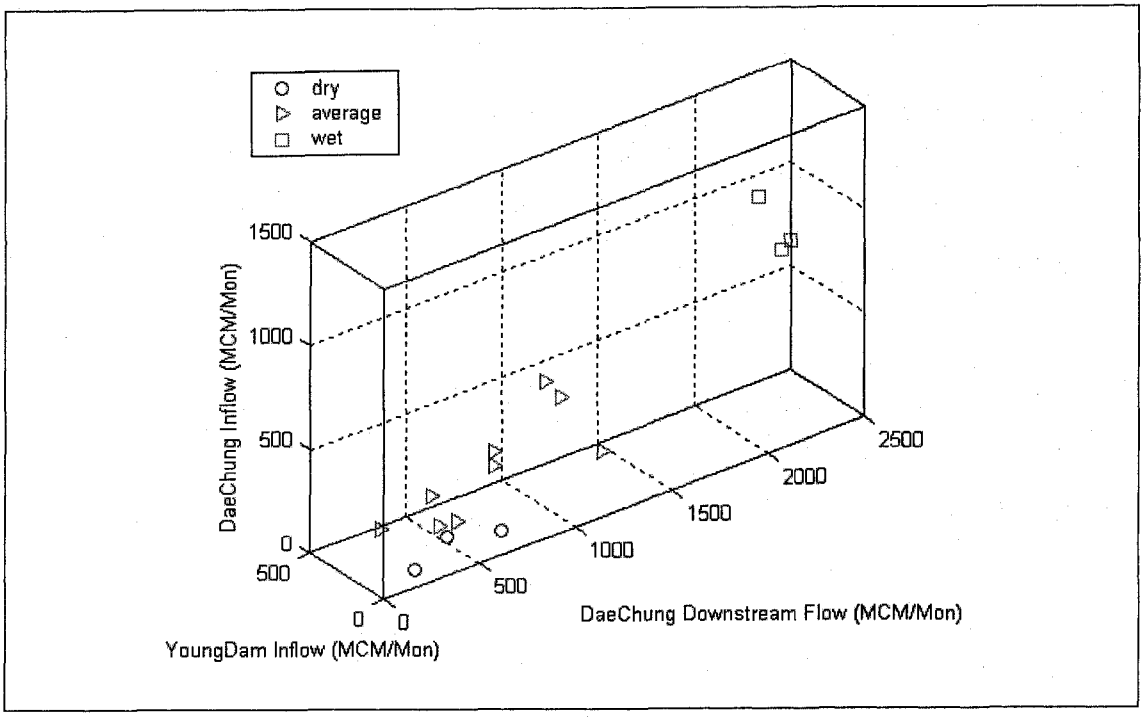


Figure 5.11 Hydrologic state by percentile approach (August)

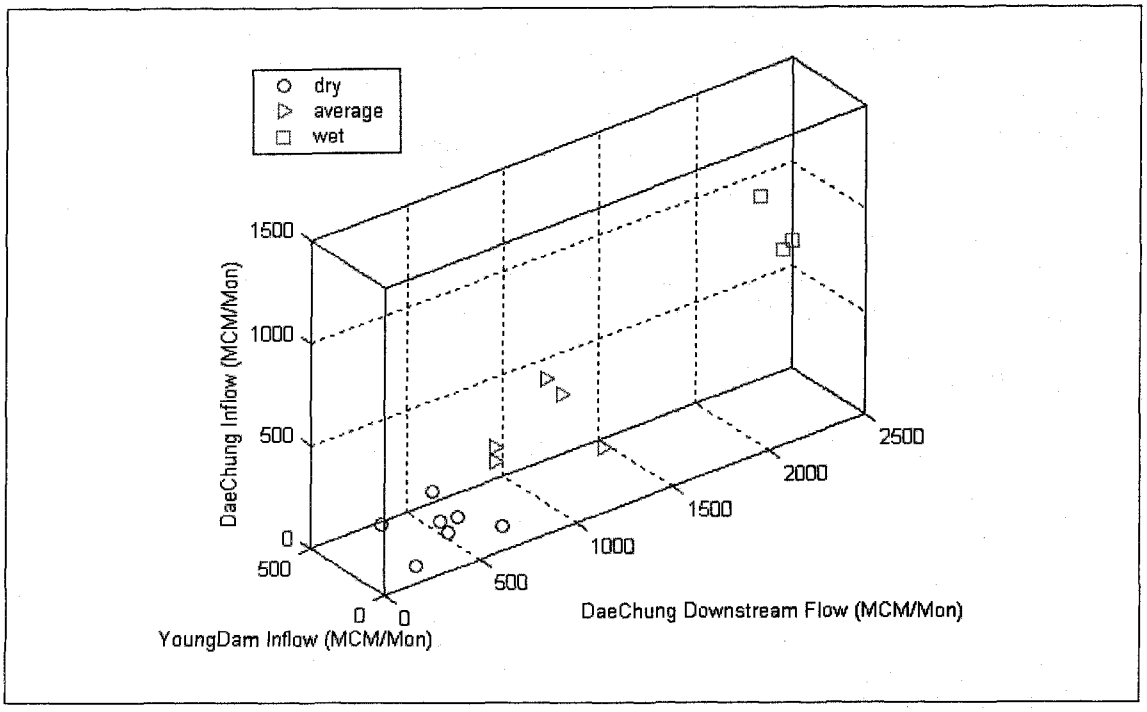


Figure 5.12 Hydrologic state by percentile approach (August)

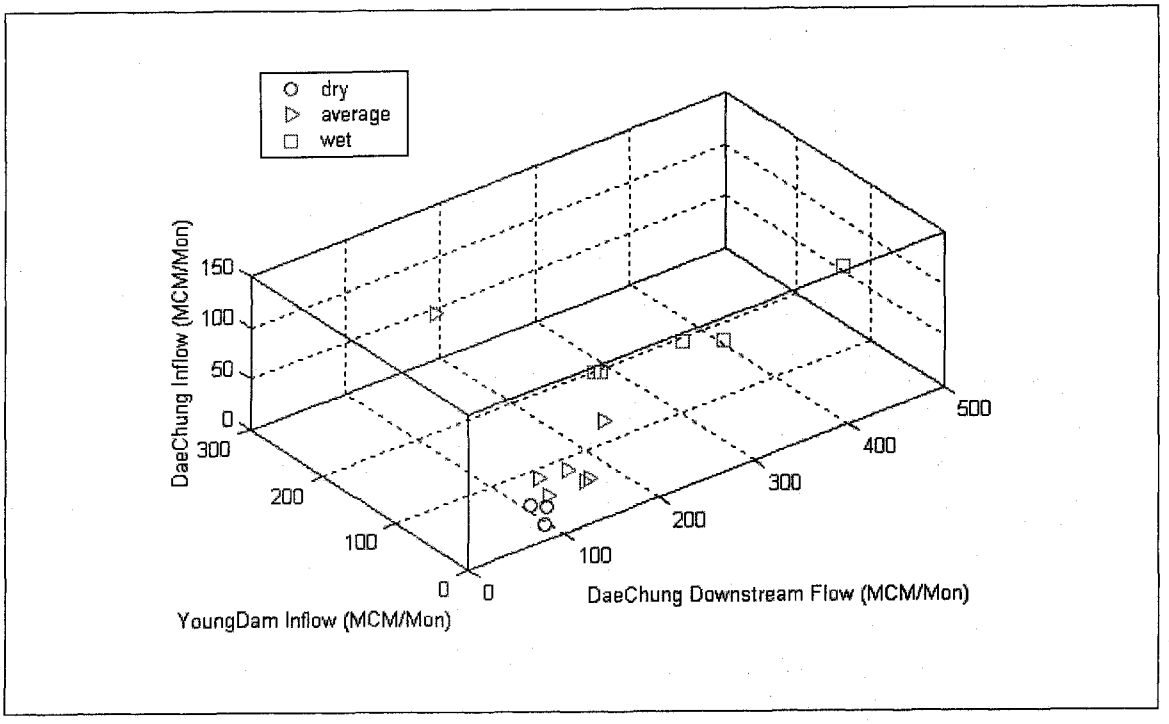


Figure 5.13 Hydrologic state by percentile approach (November)

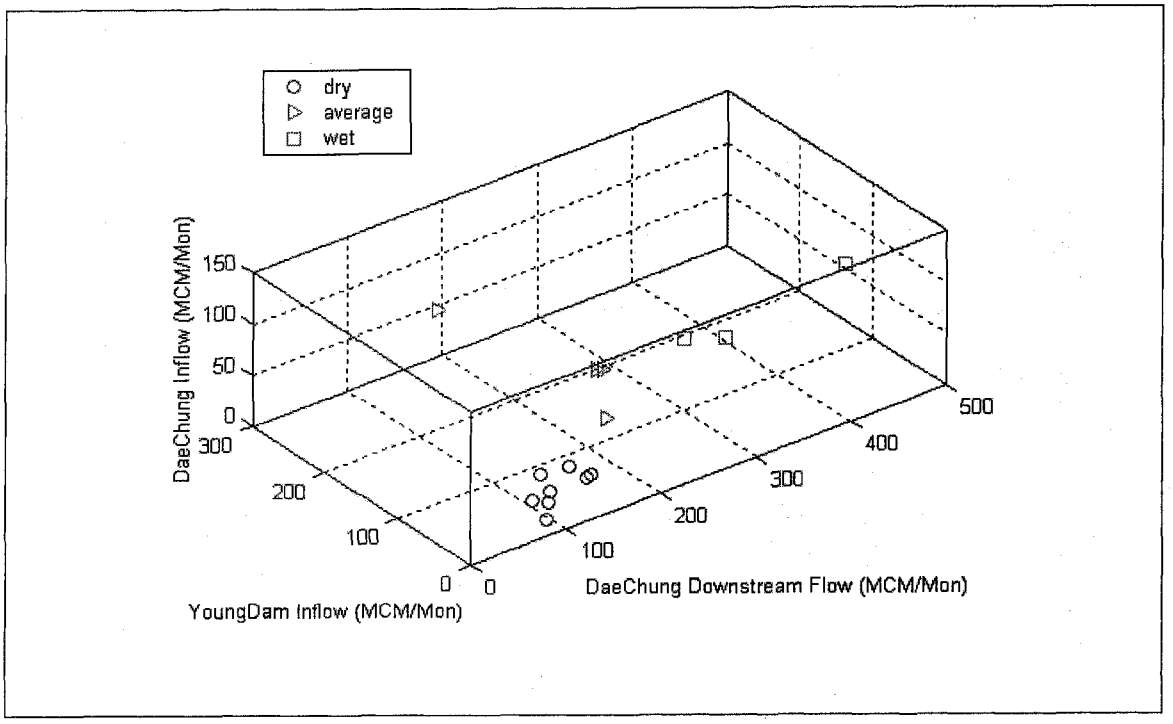


Figure 5.14 Hydrologic state by K-mean clustering approach (November)

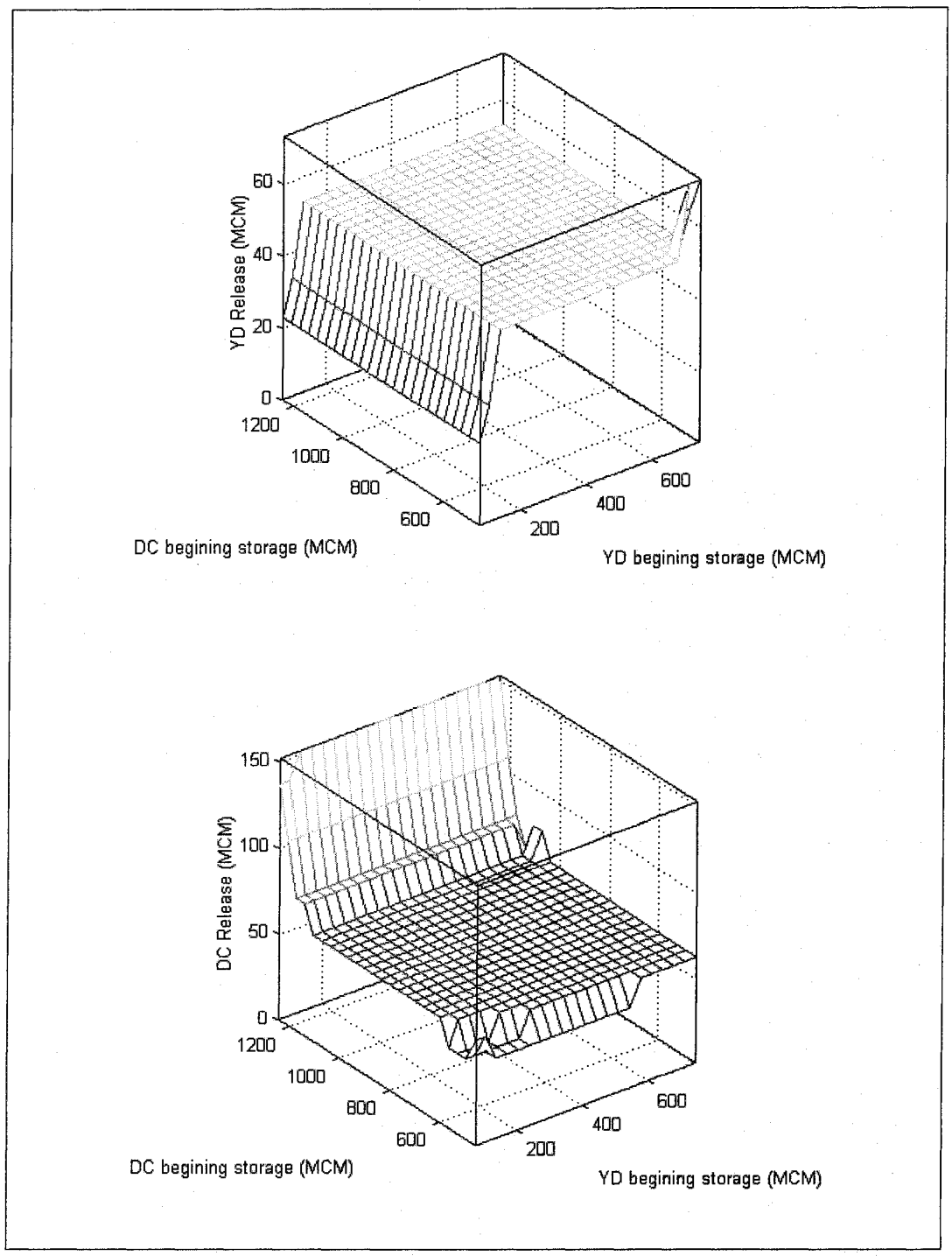


Figure 5.15 Q-Learning operation rules (November, Wet, discount =0.5)

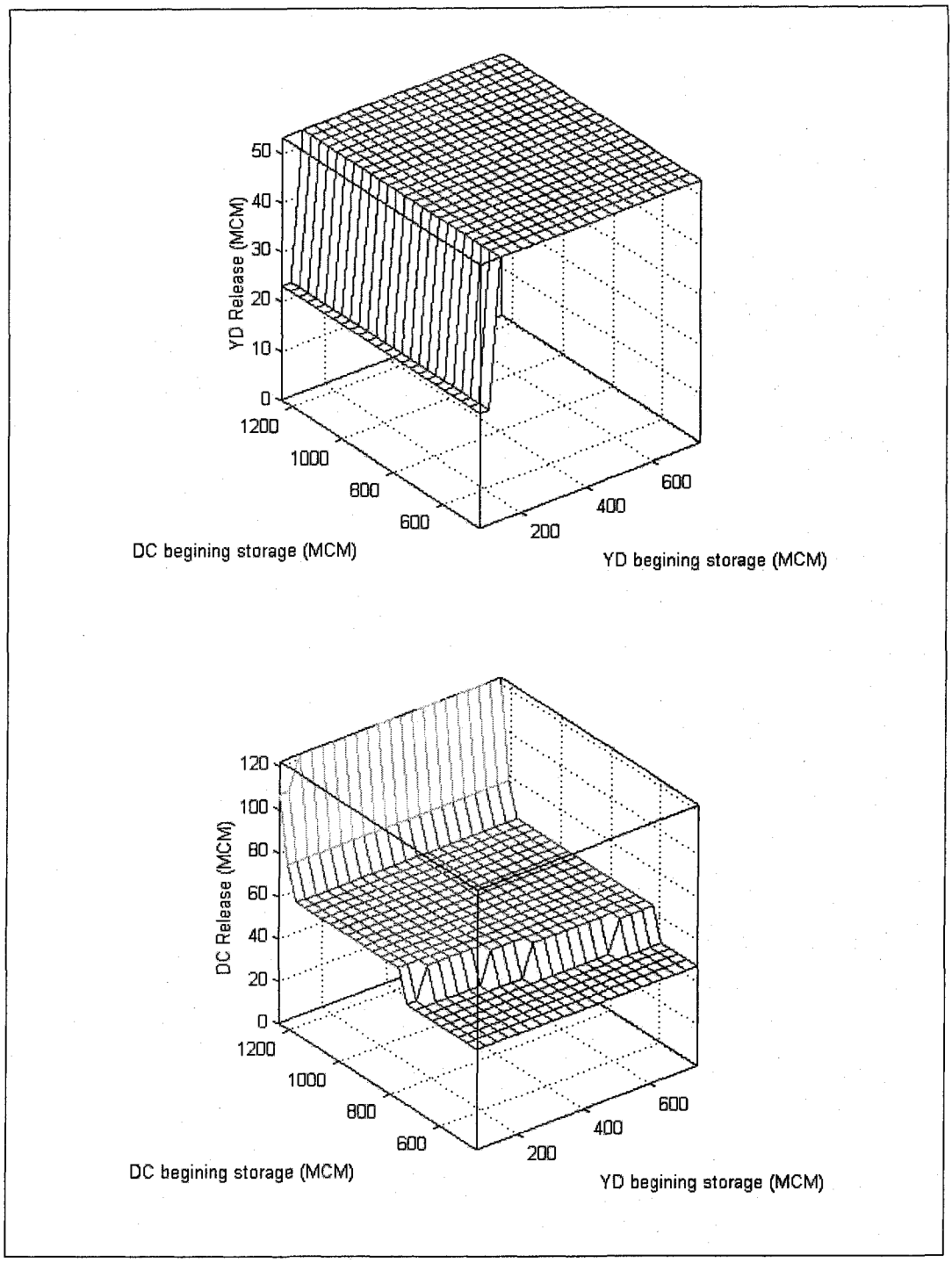


Figure 5.16 Q-Learning operation rules (November, Wet, discount =0.8)

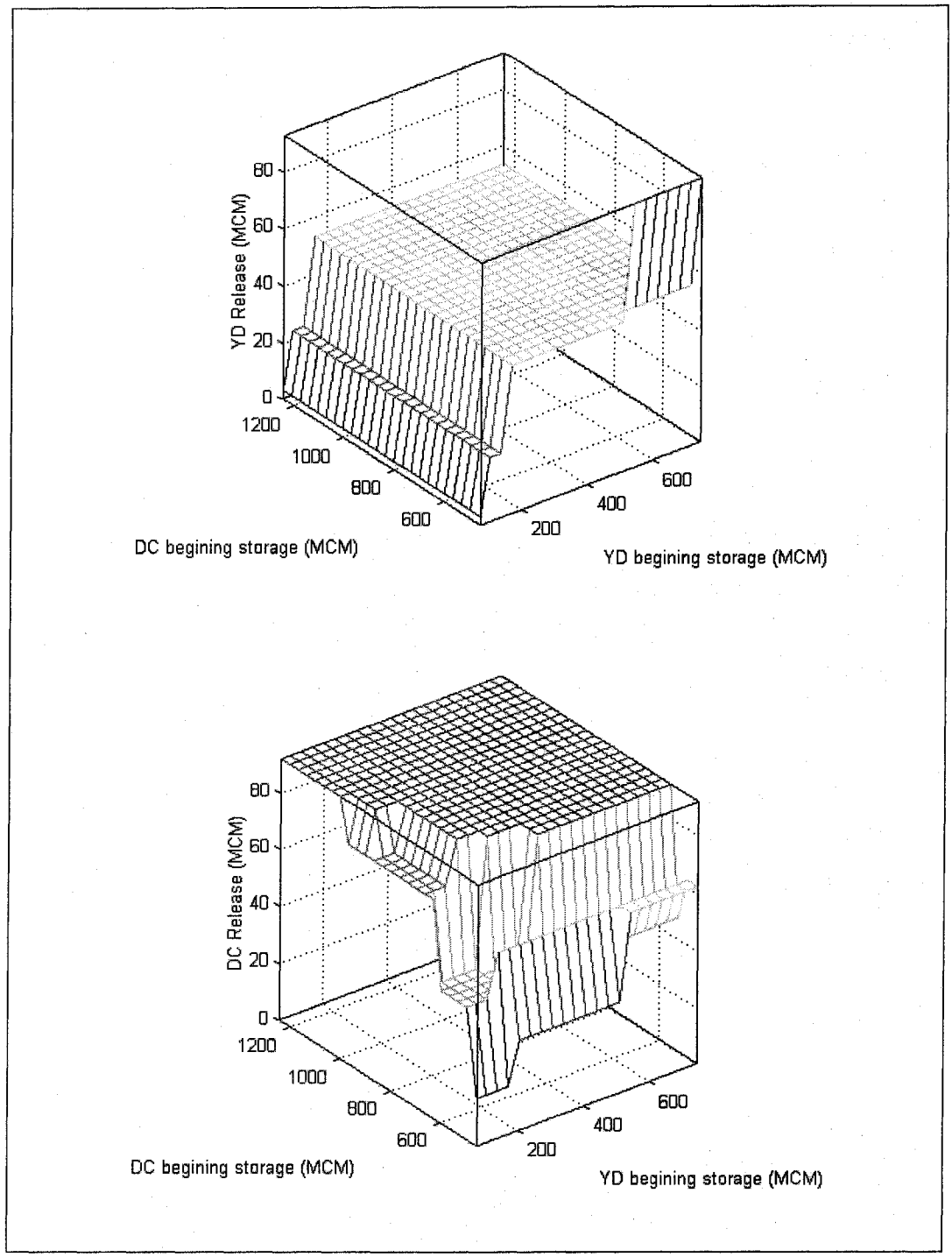


Figure 5.17 Q-Learning operation rules (January, Dry, discount =0.8)

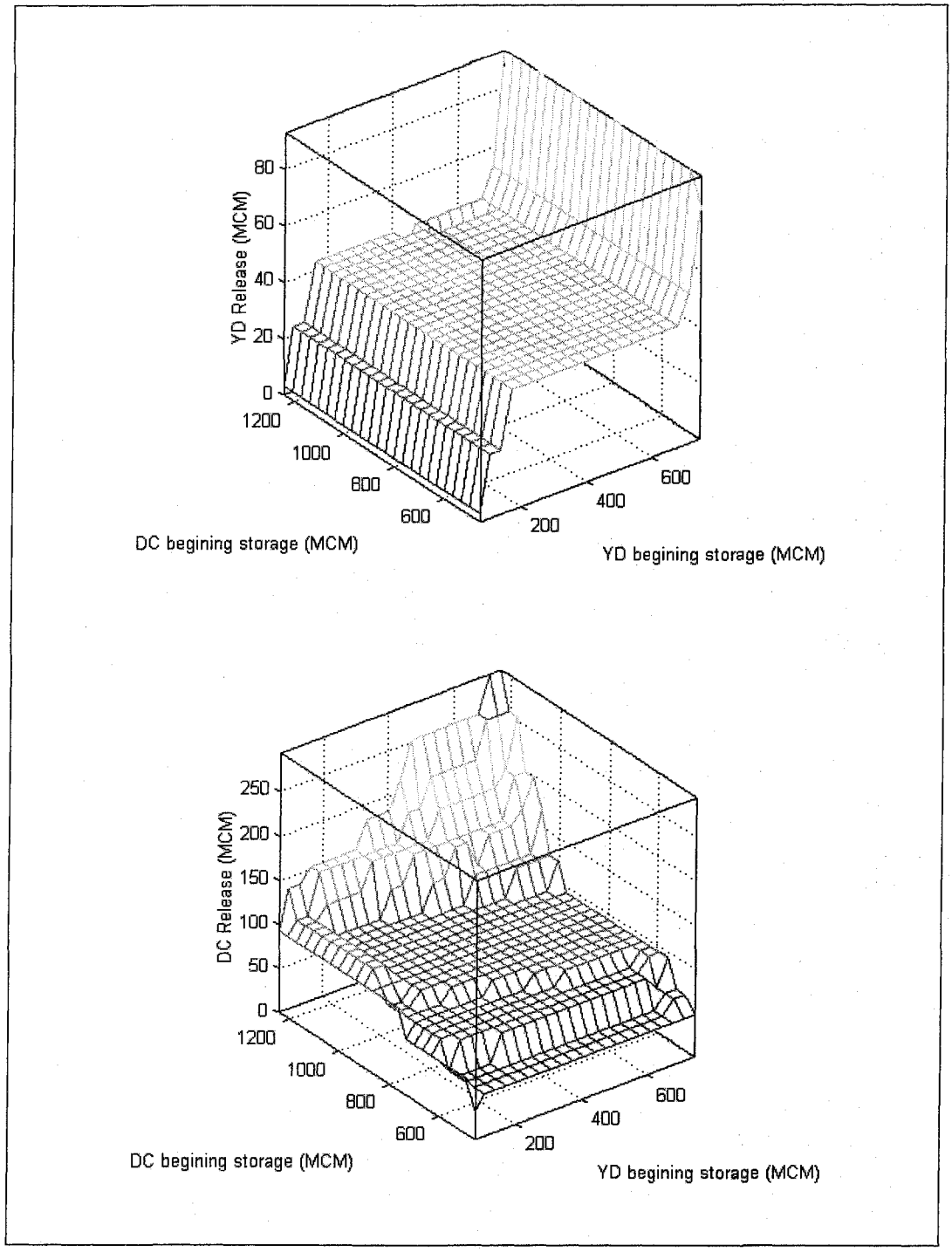


Figure 5.18 Q-Learning operation rules (August, Average, discount =0.8)

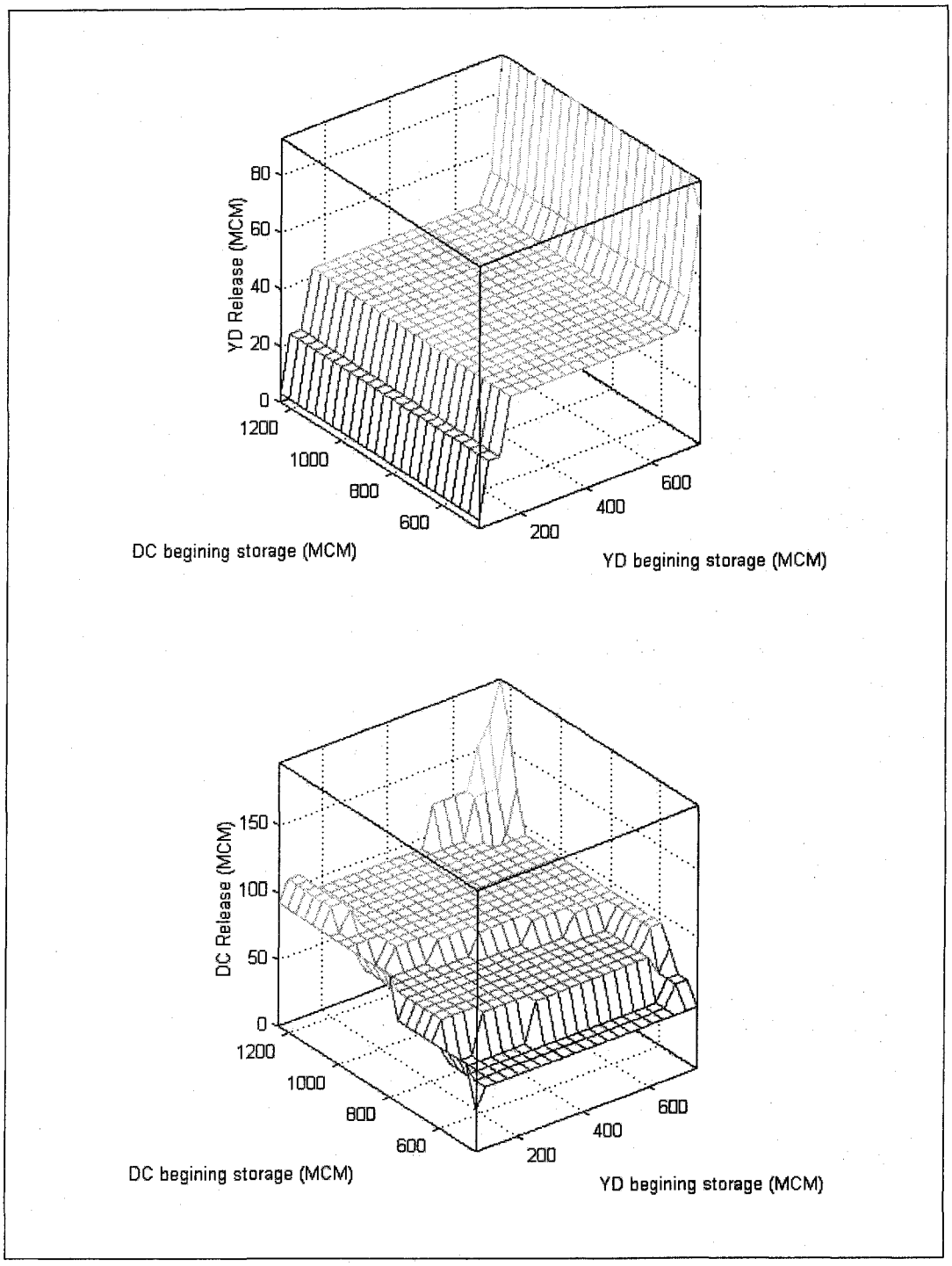


Figure 5.19 Q- Learning operation rules (August, Average, discount =0.9)

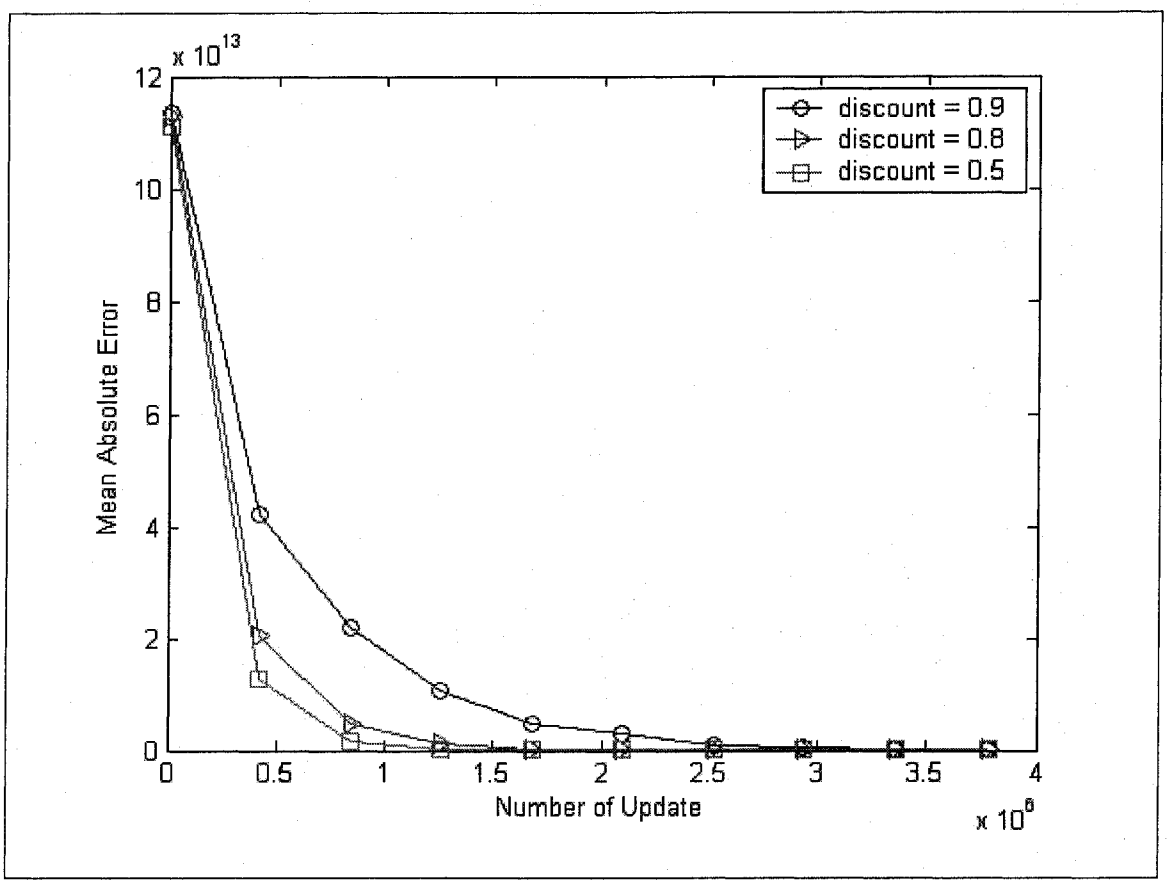


Figure 5.20 Number of update of Q value function and mean absolute error

5.5. Evaluation of the Optimal Operating Rules

The simulation analysis is used to evaluate the derived optimal operating policies in previous sections. The historical data for the period October 1998 to September 2002 are used for simulation analysis. Figure 5.21 shows the time series plot of YongDam reservoir inflow for simulation period and the boxplot of 15-year data from October 1983 to September 2002. Figure 5.22 shows the time series plot of DaeChung reservoir inflow for simulation period and the boxplot of 15-year data from October 1983 to September 2002. Low flow sequences are found in both reservoirs during 1999 and 2000

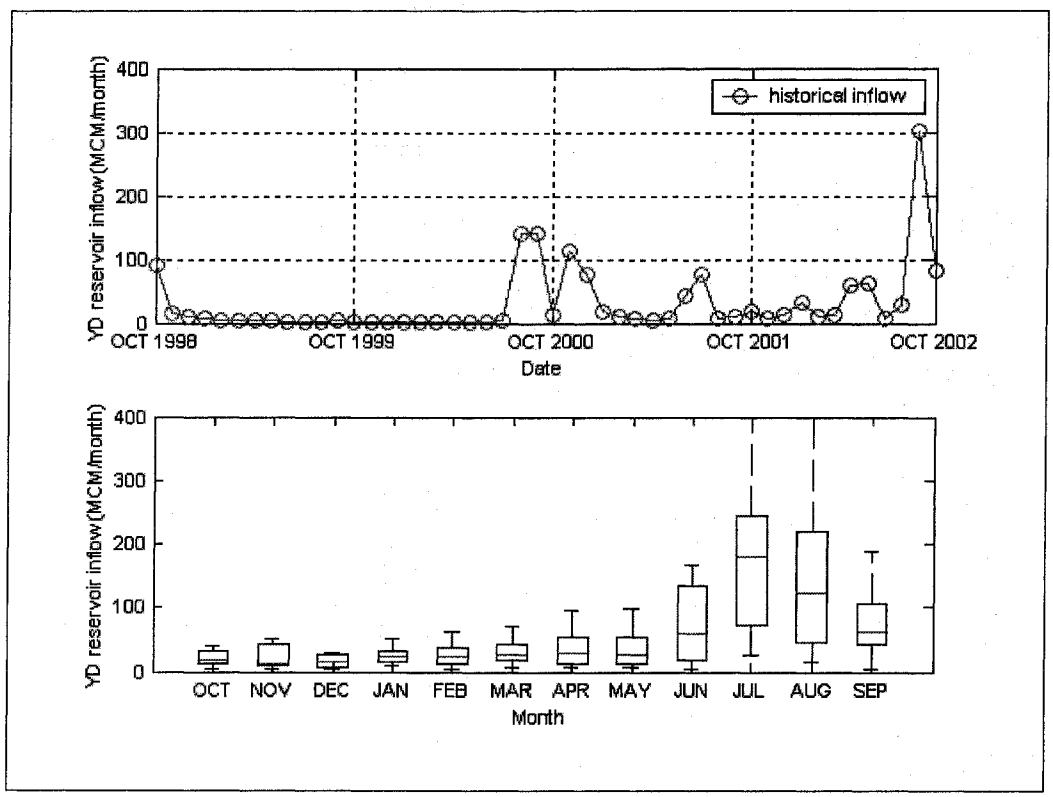


Figure 5.21 YongDam reservoir historical inflow

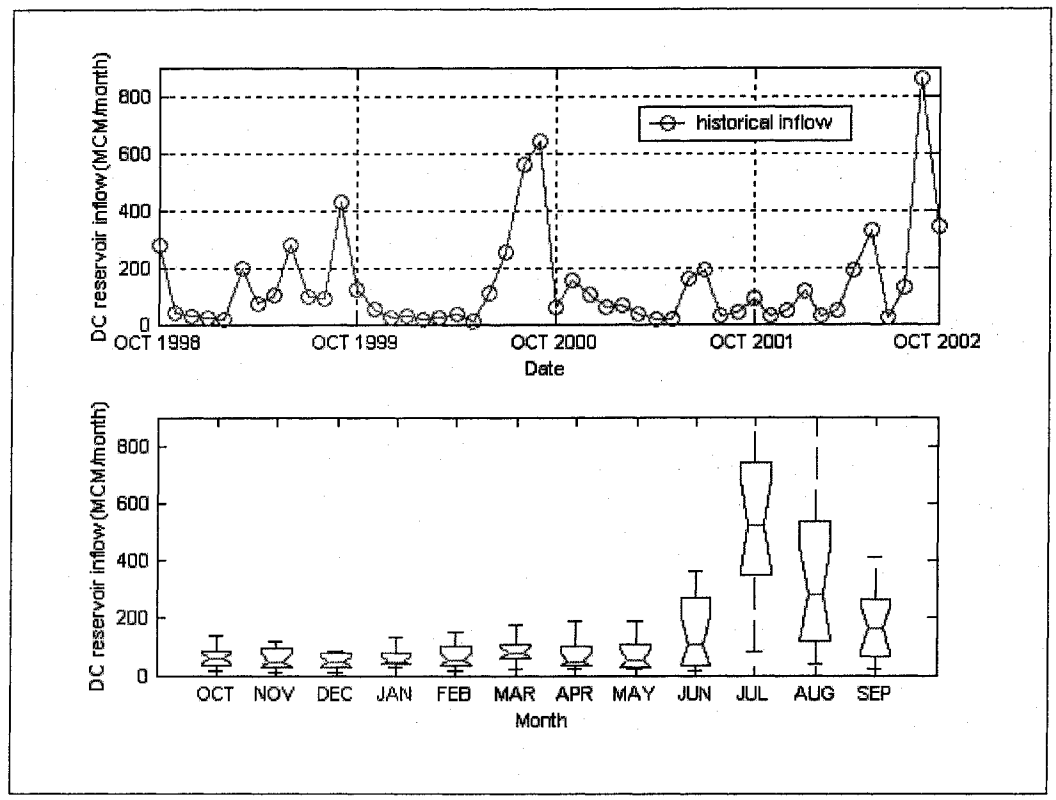


Figure 5.22 Time series plot for DaeChung reservoir inflow

5.5.1. Simulation Analysis

Performances of release policies developed previously are evaluated using simulatioin analysis. The same performance measure used in the optimization models is also applied in the simulation analysis. In addition, ending reservoir storage levels and reservoir releases for each month are compared, as well as monthly and four year total performance measures. Table 5.14 specifies all the models evaluated in the study. The first column indicates the model option, such as CSUDP, SSDP, Q-Learning, and optimal policy. Hydrologic state in the second column specifies whether the model includes the basin-wide flow condition as a system state, whereas K-mean clustering and percentile approach are used to classify the hydrologic state. The options are only available for the

Q-Learning model. Discount factors are applied to the SSDP and Q-Learning models, with value ranging from 0.7 to 0.95. From the equation (4.28), the discount factors of 0.7 and 0.95 require 12 and 84 months to represent 98% of values functions. The scenario labels are summarized in the last column and are used for further analysis. The labeling indicates for the model used, the hydrologic state classification method, and the discount factor. The scenarios in the study are categorized into two groups according to the provided operational rules. The first group includes CSUDP and Q-Learning with K-mean clustering or percentile approaches. This group provides the conditional operating rules for the reservoirs according to the hydrologic state defined in the specified manner, SSDP can also employ the hydrologic states with forecasted streamflows, but this hydrologic state is not used here due to the absence of forecasting model. The second group includes SSDP and Q-Learning without the hydrologic state, which provides the unconditional operating rules for the reservoirs.

Table 5.14 Model descriptions

Model	Hydrologic State	Discount Factor	Model Label
CSUDP	Inflow to the reservoir	NA	CSUDP
SSDP	NA	0.95	SSDP095
		0.80	SSDP08
Q-Learning	NA	0.95	Q095
		0.70	Q07
Q-Learning	Percentile	0.90	QP09
		0.80	QP08
Q-Learning	K-mean clustering	0.95	QK095
		0.90	QK09
Deterministic DP	NA	NA	OPT

5.5.2. Comparison of unconditional operation policies

Unconditional policies generated by the SSDP and Q-Learning models are compared to evaluate model performance. The performance of Q095, Q07, SSDP095, and SSDP08 are compared to the best attainable performance measure by deterministic DP (OPT) with the perfect knowledge of future inflow. YongDam reservoir operations for the various models are compared in Figure 5.23. It can be seen that the relative performance of Q095 is close to the OPT and reservoir storage is not draw down to the minimum storage level. SSDP with a discount factor of 0.8 displays the poorest performance in the operation, showing a particular disadvantage in dealing with the low flows occurring during 1999 - 2000. The low discount factor operates the reservoir more myopically than the high discount factor due to the low valuation of future reward. Even though the flow scenarios are uncertain, the sequences themselves are known by assumption in SSDP model. It seems like that the sequence information is restricted due to the absence of forecasting model. Relative releases from YongDam reservoir are included in the lower portion of the figure, showing reduced release during the low flow season. The OPT model is the only method can handle this challenge.

Figure 5.24 provides the DaeChung reservoir operation results for the models, SSDP models also display poor performance. It can be seen that all the models attempt to refill the reservoir by the end of flood season (September). This reflects the realistic need in the reservoir operations to prepare for the dry season prior to June. The ending reservoir storage for May by the OPT model is close to minimum storage for DaeChung reservoir, which is only possible if perfect foresight of future streamflows.

Monthly performance measures and total performance measure are compared in Figure 5.25, with the OPT and Q095 models both maintaining the consistency in the performance measures over the simulation period. OPT provides less performance measures in the beginning of the operation to prepare for the low streamflow conditions in the future while Q095 provides less performance measures during the low streamflow conditions. The total performance measure indicate that the unconditional Q-Learning models perform better with high discount factors, whereas the SSDP models do not provide desirable operation rules. High discount factors generate more risk adverse operating rules since they assign the higher future value of water usage. It can be found that SSDP method does not produce useful operating rules without forecasting model.

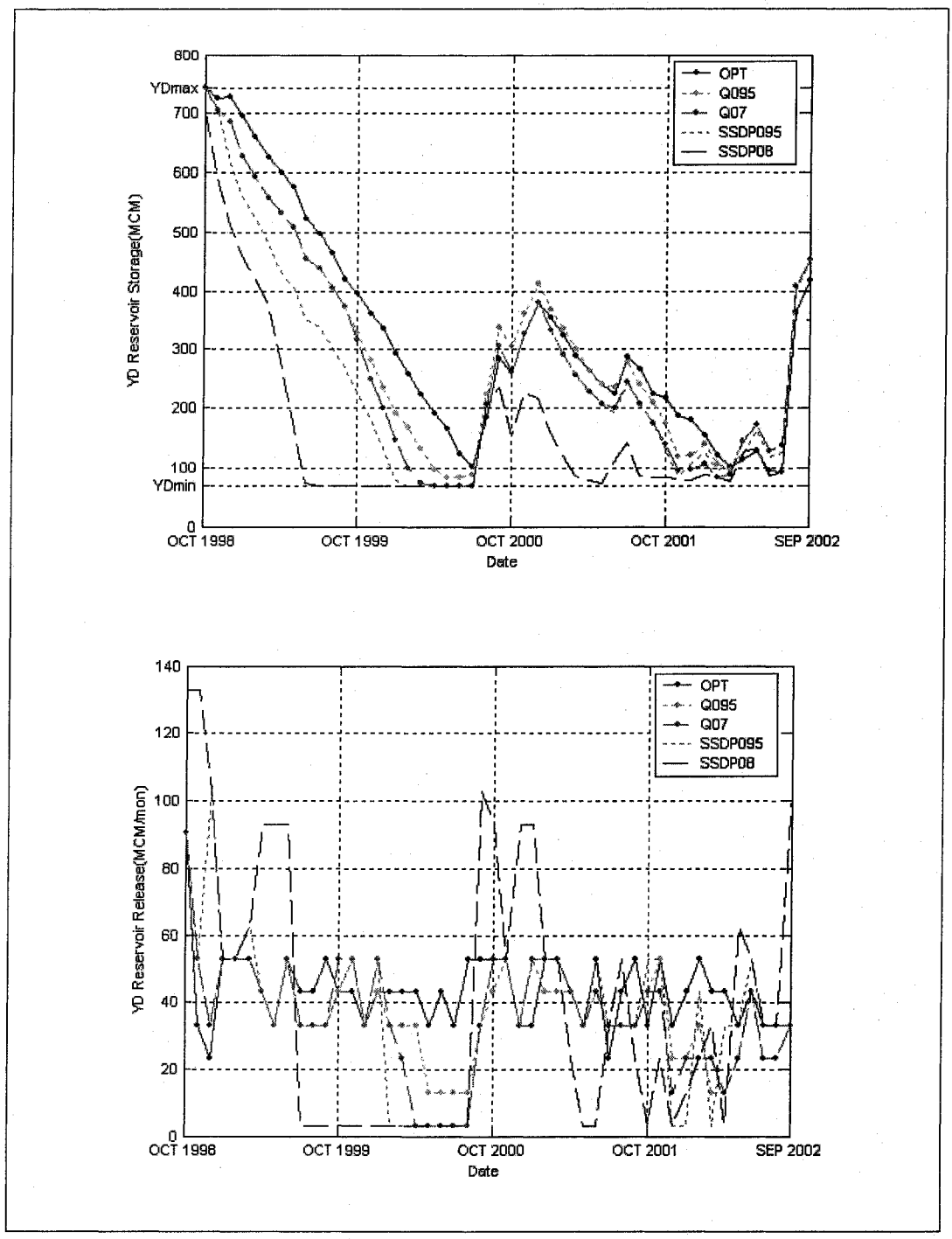


Figure 5. 23 Unconditional Operation Policy Comparisons

(YongDam Reservoir Beginning Storage and Release)

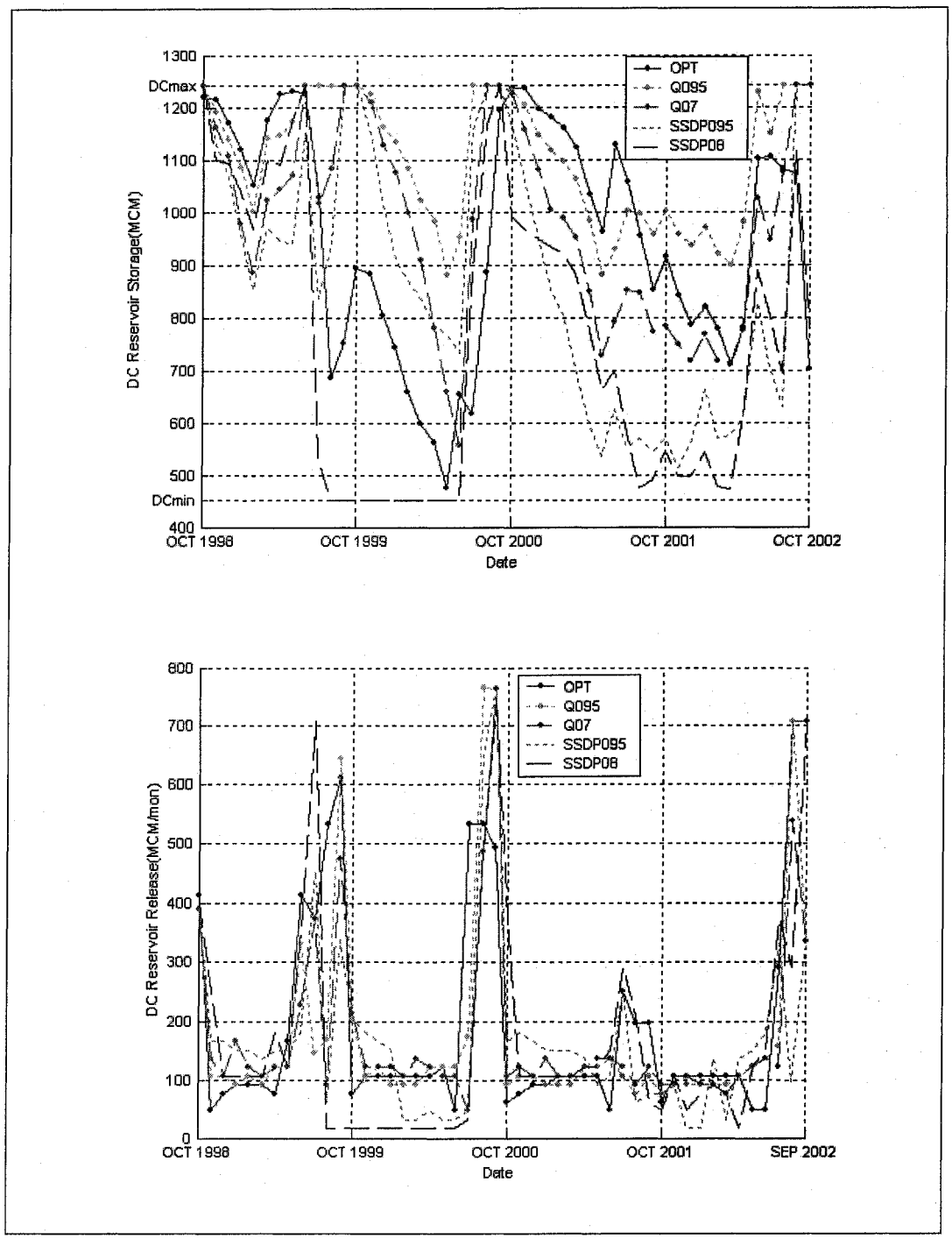


Figure 5.24 Unconditional Operation Policy Comparisons

(DaeChung Reservoir Beginning Storage and Release)

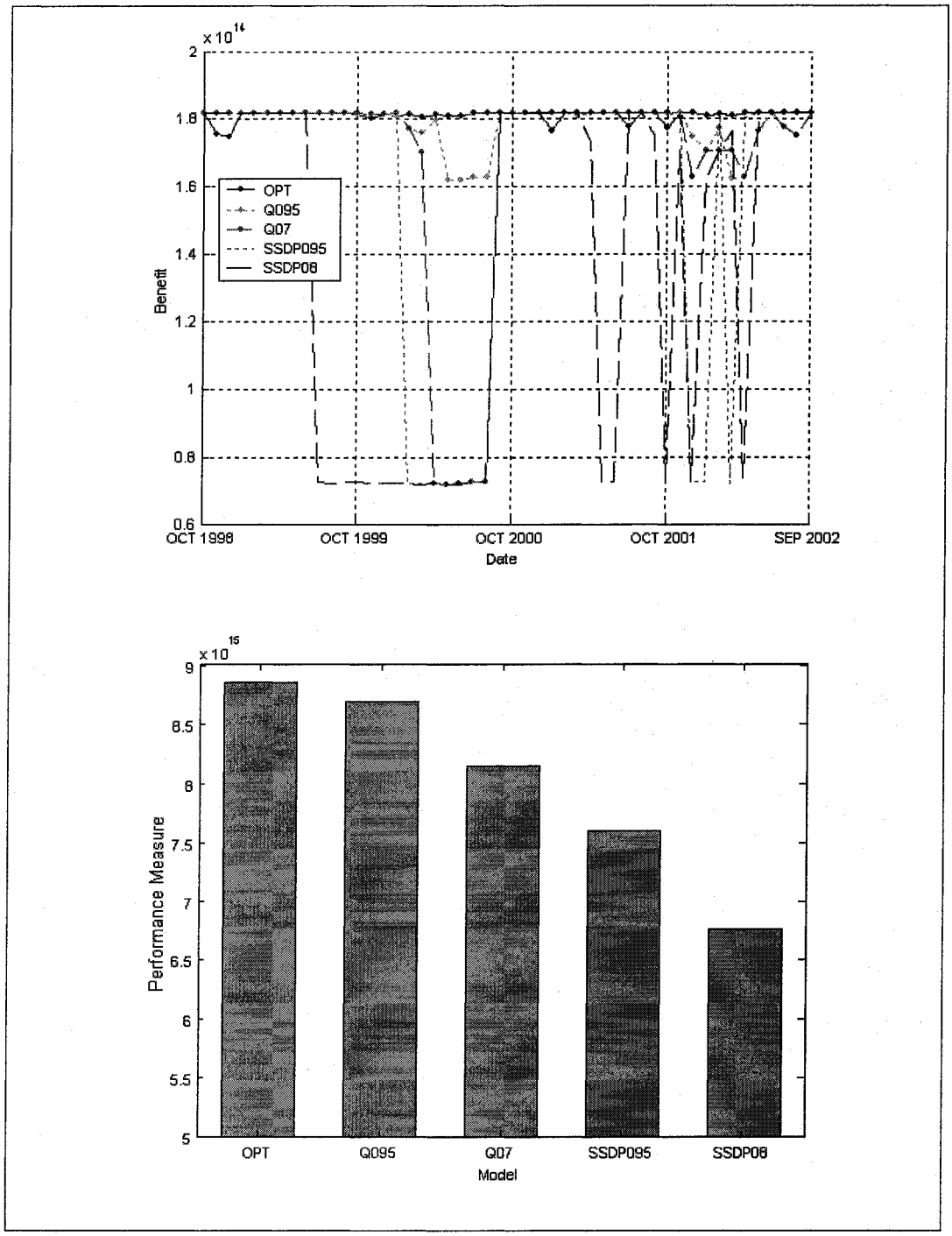


Figure 5.25 Unconditional Operation Policy Comparisons

(Monthly and Total Performance Measures)

5.5.3. Comparison of conditional operation policies

Conditional policies by Q-Learning models are compared in this section, with QK095, QK09, QP09, and QP08 chosen for hydrologic state definition comparison. QK models use K-mean clustering approach to define hydrologic state whereas QP models use percentile values to define hydrologic state. Reservoir operations from the various models are compared in Figures 5.26 and 5.27. Through all the models behave similarly, QK models perform slightly well. However, QK models perform poorly during the low flow during 1999 - 2000, even with high discount factors and hydrologic states unable to overcome this problem.

Figure 5.28 shows the monthly and total performance measures and indicates that QK095 and QK09 provide similar total performance measures. All the models provide less performance measures during the low flow sequences. The total performance measure indicates that hydrologic state information has more influence on model performance than the discount factor. Way of defining hydrologic state appears to make a significant performance difference.

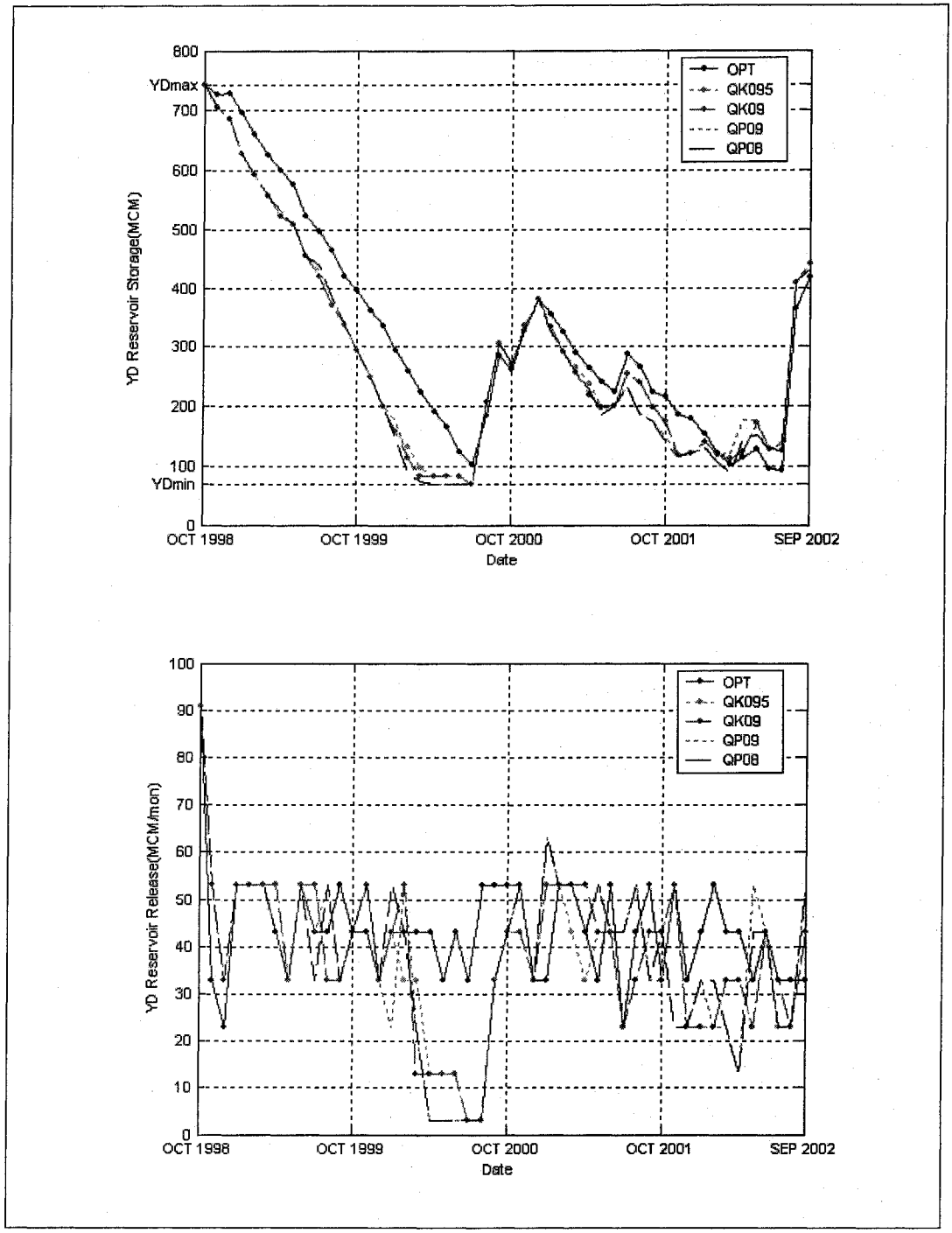


Figure 5.26 Conditional Operation Policy Comparisons

(YongDam Reservoir Beginning Storage and Release)

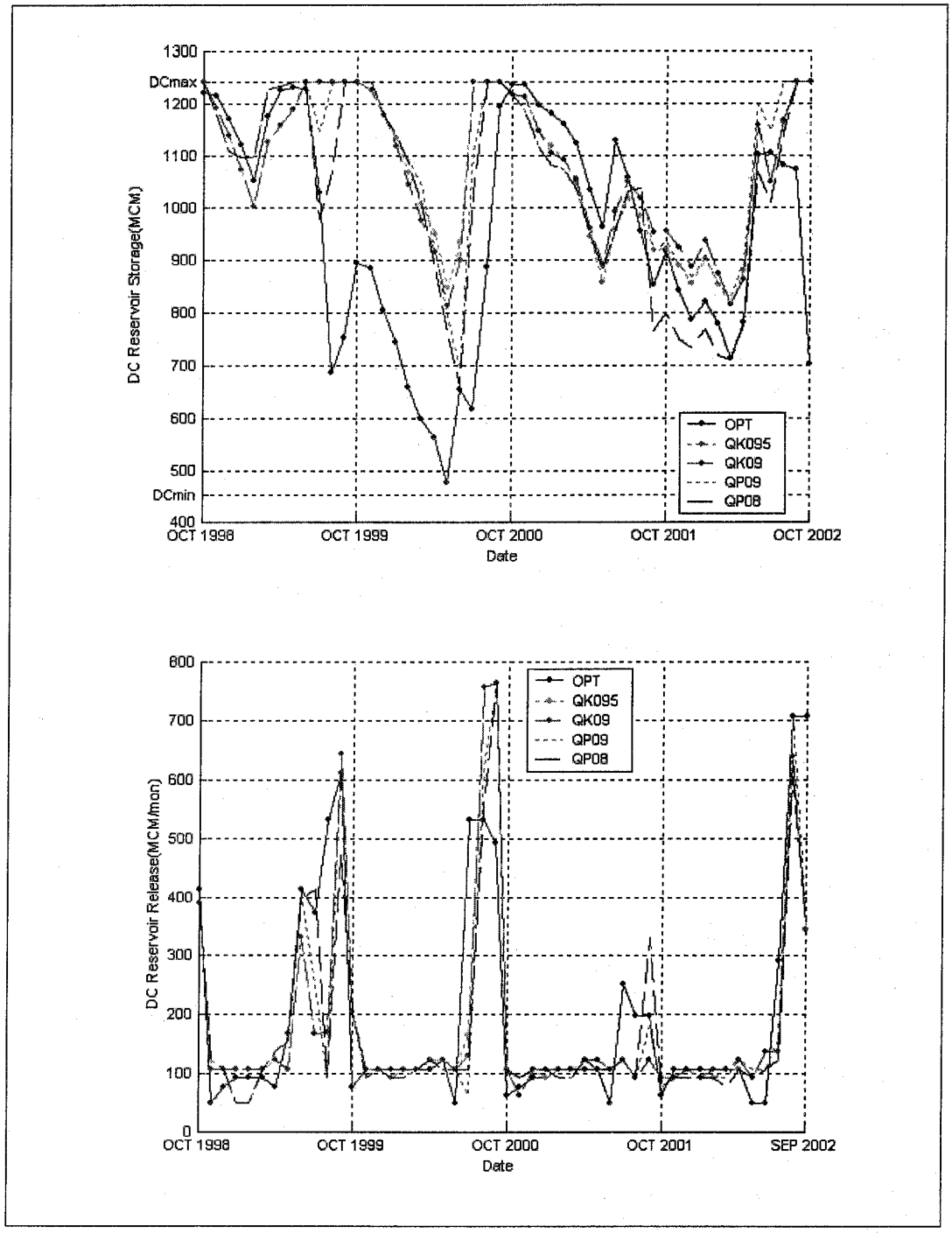


Figure 5.27 Conditional Operation Policy Comparisons

(DaeChung Reservoir Beginning Storage and Release)

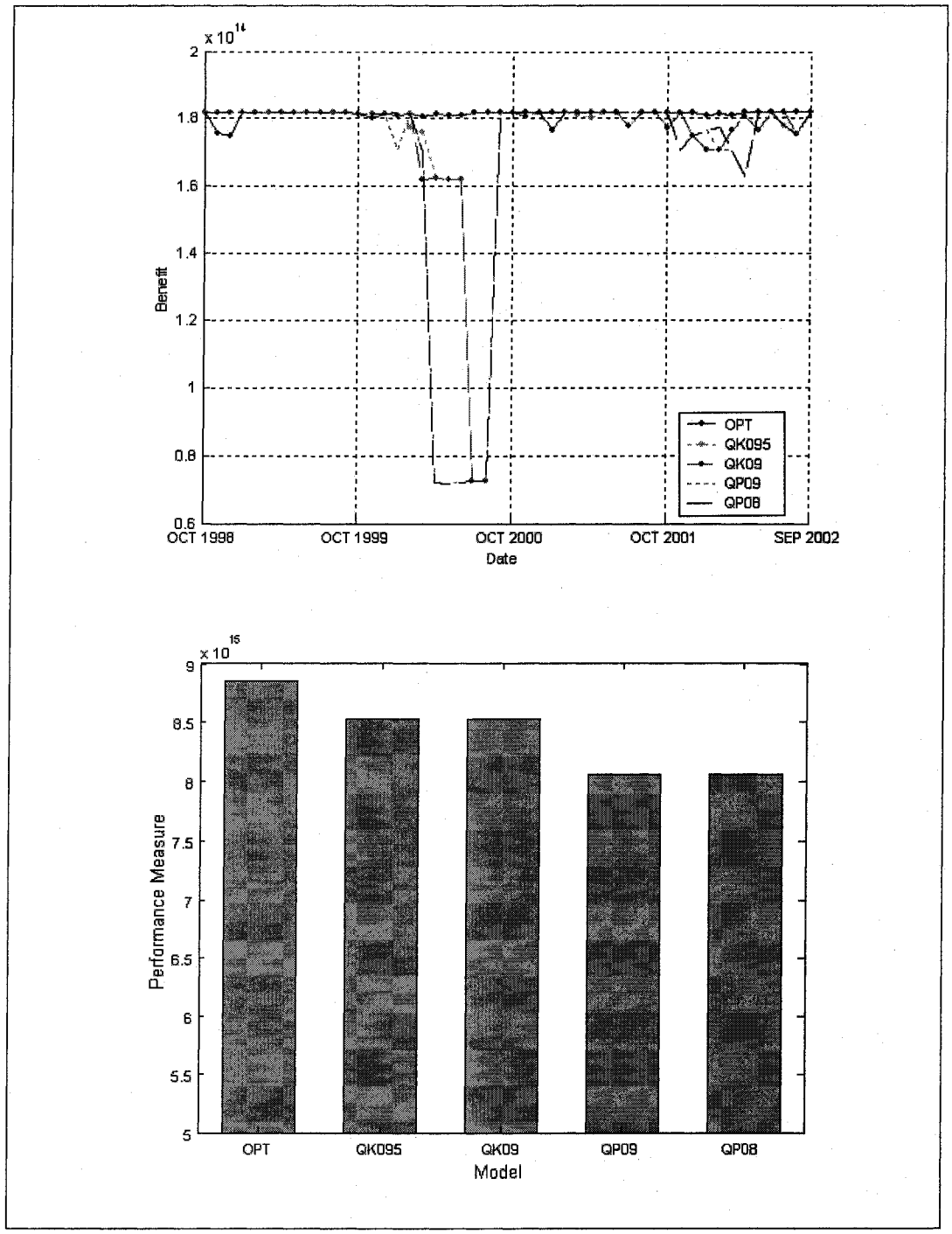


Figure 5.28 Conditional Operation Policy Comparisons

(Monthly and Total Performance Measures)

5.5.4. Implicit vs. Explicit Stochastic Optimization

Several explicit stochastic optimization models including Q095, QK095, and SSDP095 are compared with implicit stochastic optimization model (CSUDP) to investigate model performances. Operation of YongDam reservoir is compared for the various models in Figure 5.29. CSUDP results are considerably different from the other models, with latter showing similar behavior. All the models including OPT draw down reservoir storage to the minimum in May 2000. However, the reservoir storage operated by CSUDP is not drawn to the minimum reservoir storage since the reduced release are determined due to the extremely low flow condition. The CSUDP uses reservoir inflow as well as beginning of reservoir storage to determine operating rule, whereas other explicit stochastic optimization models use only beginning of reservoir storage. More stable releases are found in operations with OPT, Q095, and QK095 during dry seasons.

As shown in Figure 5.30, the DaeChung reservoir operations indicating that CSUDP results are quite similar to those of SSDP095. The CSUDP operation reduces DaeChung reservoir release to recover the normal full reservoir storage at the end of the water year. This results in more releases during the dry season and less releases during the flood season. This reduced release may not impact on the performance measures significantly due to the abundant flow conditions during flood season. More stable releases represented in operations with OPT, Q095, and QK095 during dry seasons.

Figure 5.31 compares the monthly and total performance measures, showing that Q095 and OPT provide consistency in monthly performance measures, which the other models provide less performance measures during the low flow sequences. CSUDP has better performance in terms of large performance measure reduction than QK095 and

SSDP095. The total performance measure indicates that the conditional or unconditional Q-Learning approaches outperform both CSUDP and SSDP. CSUDP and QK095 produced almost the same amount of total performance measures although their monthly performance measures are different.

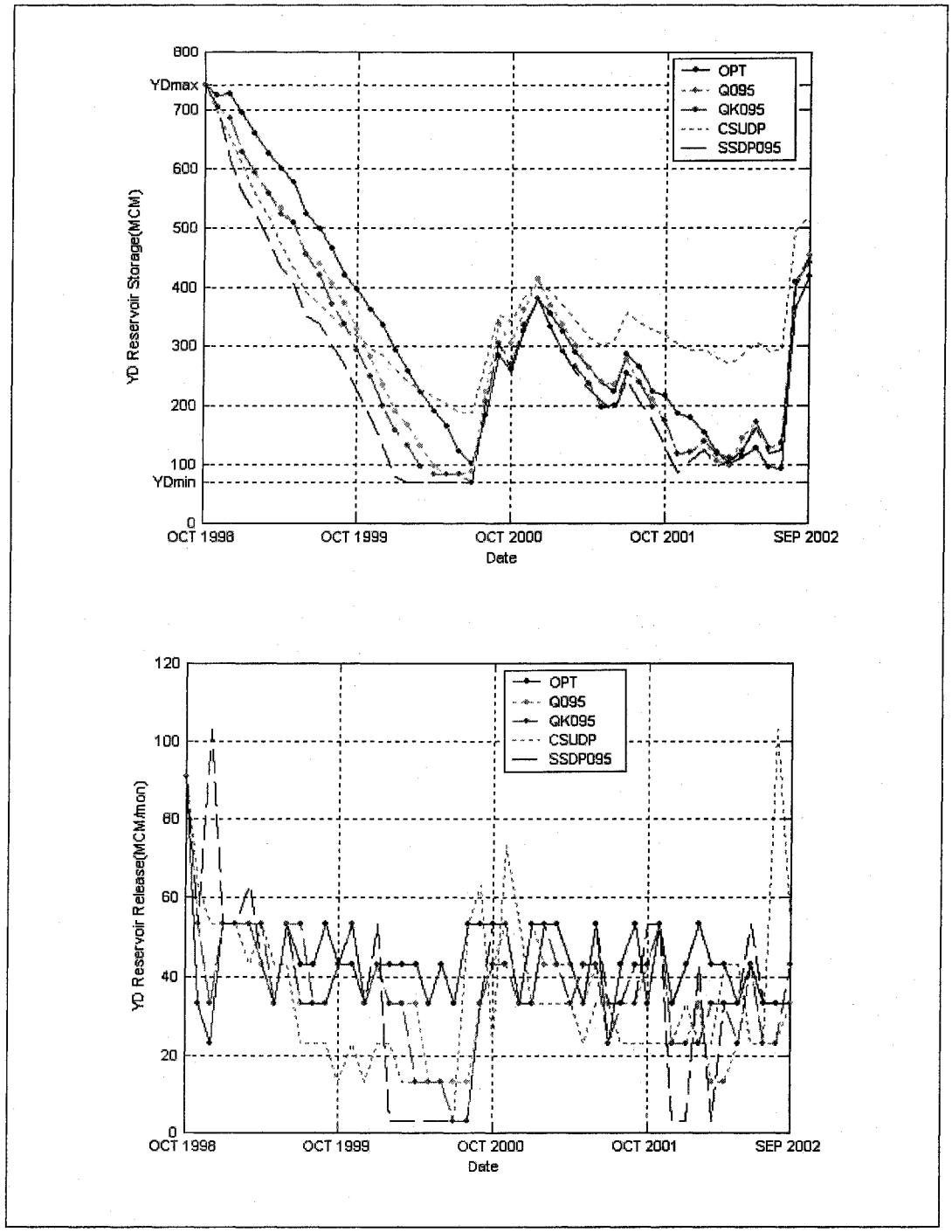


Figure 5.29 Operation Policy Comparisons – Implicit vs. Explicit Stochastic Optimization (YongDam Reservoir Beginning Storage and Release)

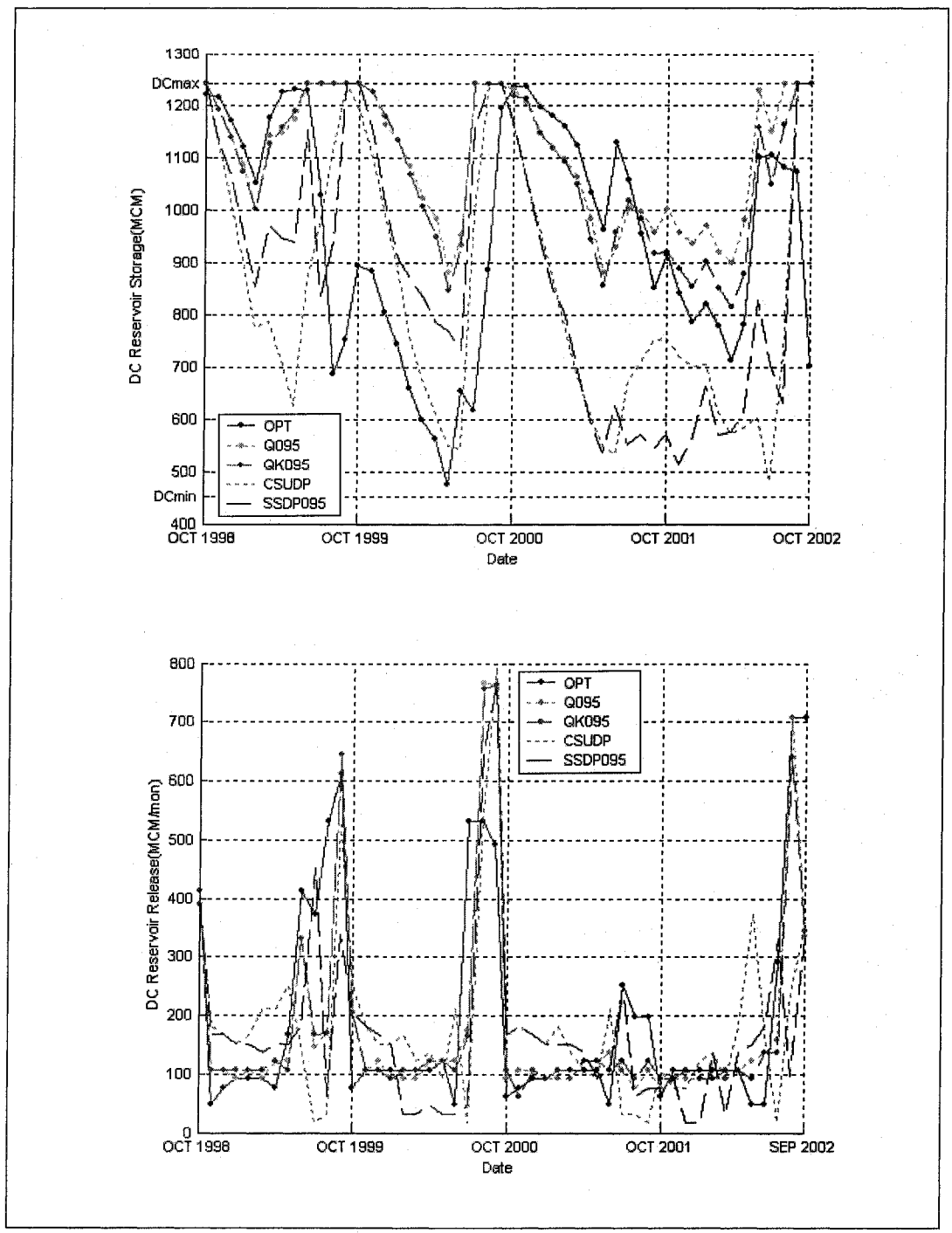


Figure 5.30 Operation Policy Comparisons – Implicit vs. Explicit Stochastic Optimization (DaeChung Reservoir Beginning Storage and Release)

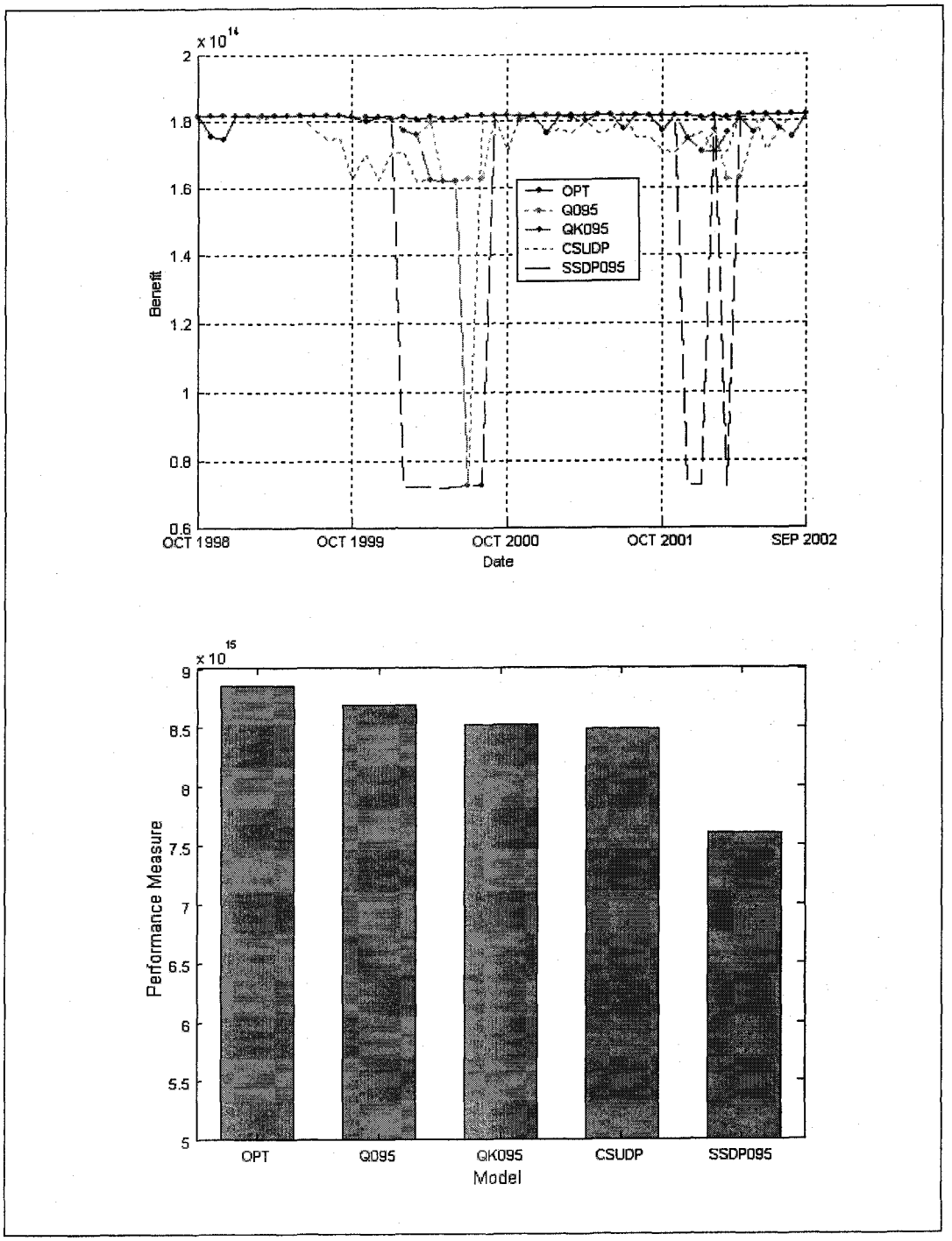


Figure 5.31 Operation Policy Comparisons – Implicit vs. Explicit Stochastic Optimization (Monthly and Total Performance Measures)

5.5.5. Initial condition of reservoir storage

In the previous section, the initial conditions for reservoir storage are assigned to as normal full storage (YD = 742.75MCM and DC = 1241.7 MCM). To represent a more realistic case, the actual end of month reservoir storages of September 1998 are used to evaluate the model performance. The initial conditions for reservoir storage are also assigned to as 80% of normal full storage (YD = 606.75 MCM and DC = 1074.8 MCM).

The Q095, QK095, SSDP095, CSUDP, OPT models are used to investigate the performance with the new initial storage conditions. Figure 5.32 shows YongDam and DaeChung reservoir operations by comparing the reservoir storage levels. The subindex "f" in Q095f and CSUDPf indicates the normal full storage option. The storage volumes for YongDam reservoir are identical after two years of operation. The same thing occurs for DaeChung reservoir after one year of operation. Operation of YongDam and DaeChung reservoirs is compared in Figures 5.33 to 5.35, showing no impact on model performance under these changes.

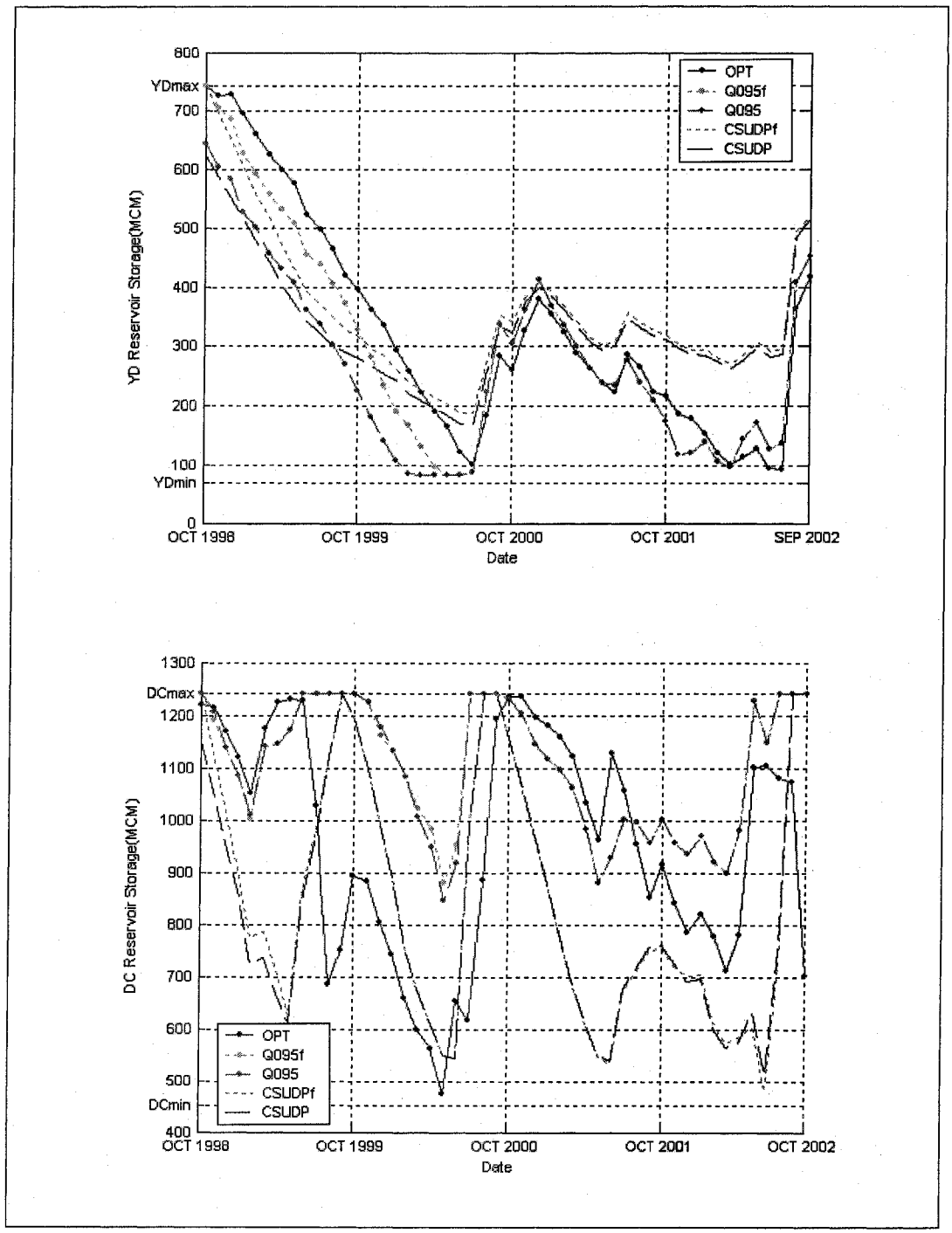


Figure 5.32 YD and DC Reservoir Beginning Storage Comparisons

(Normal Full Storage vs. 80% Normal Full Storage)

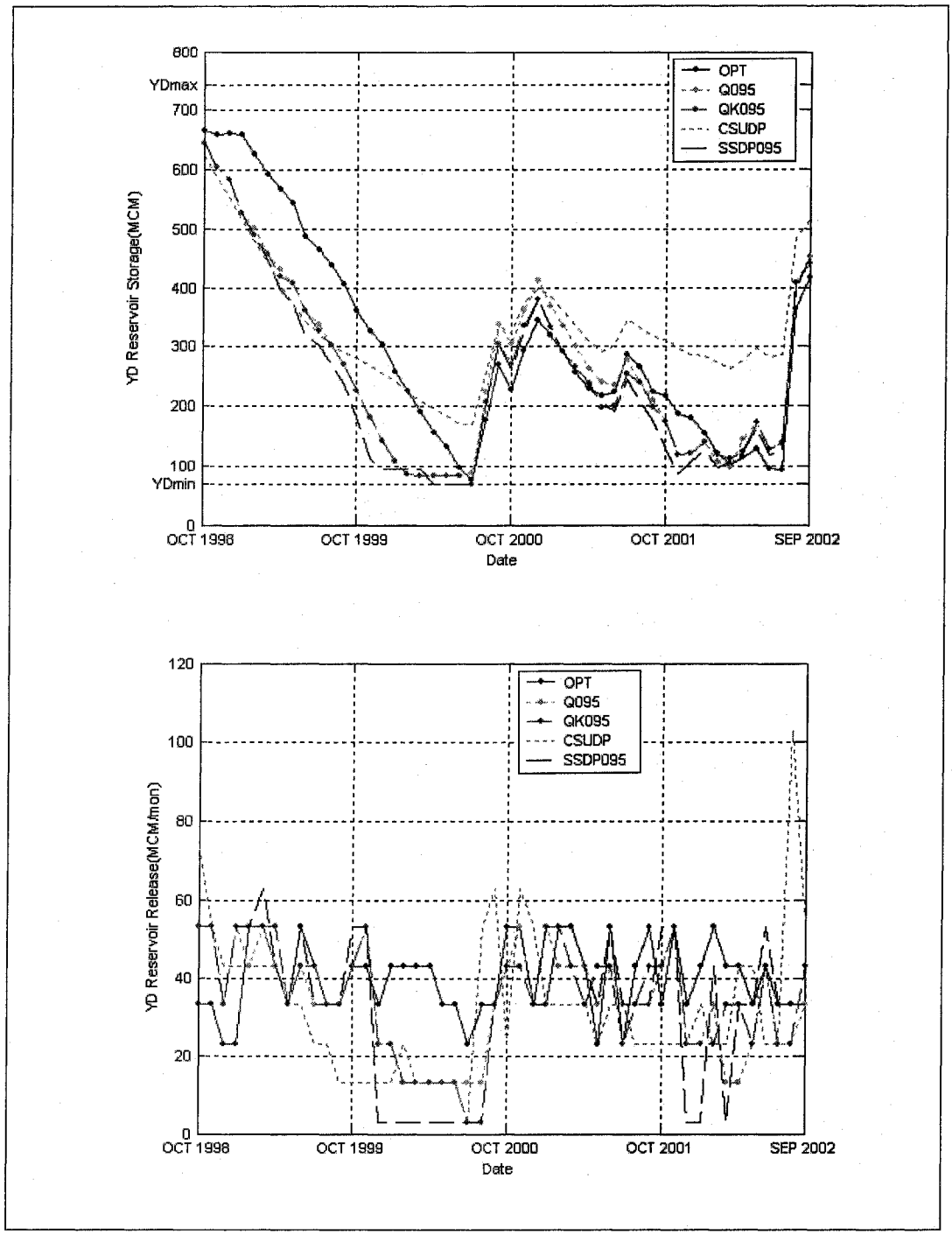


Figure 5.33 YD Reservoir Beginning Storage and Release

(80% Normal Full Storage)

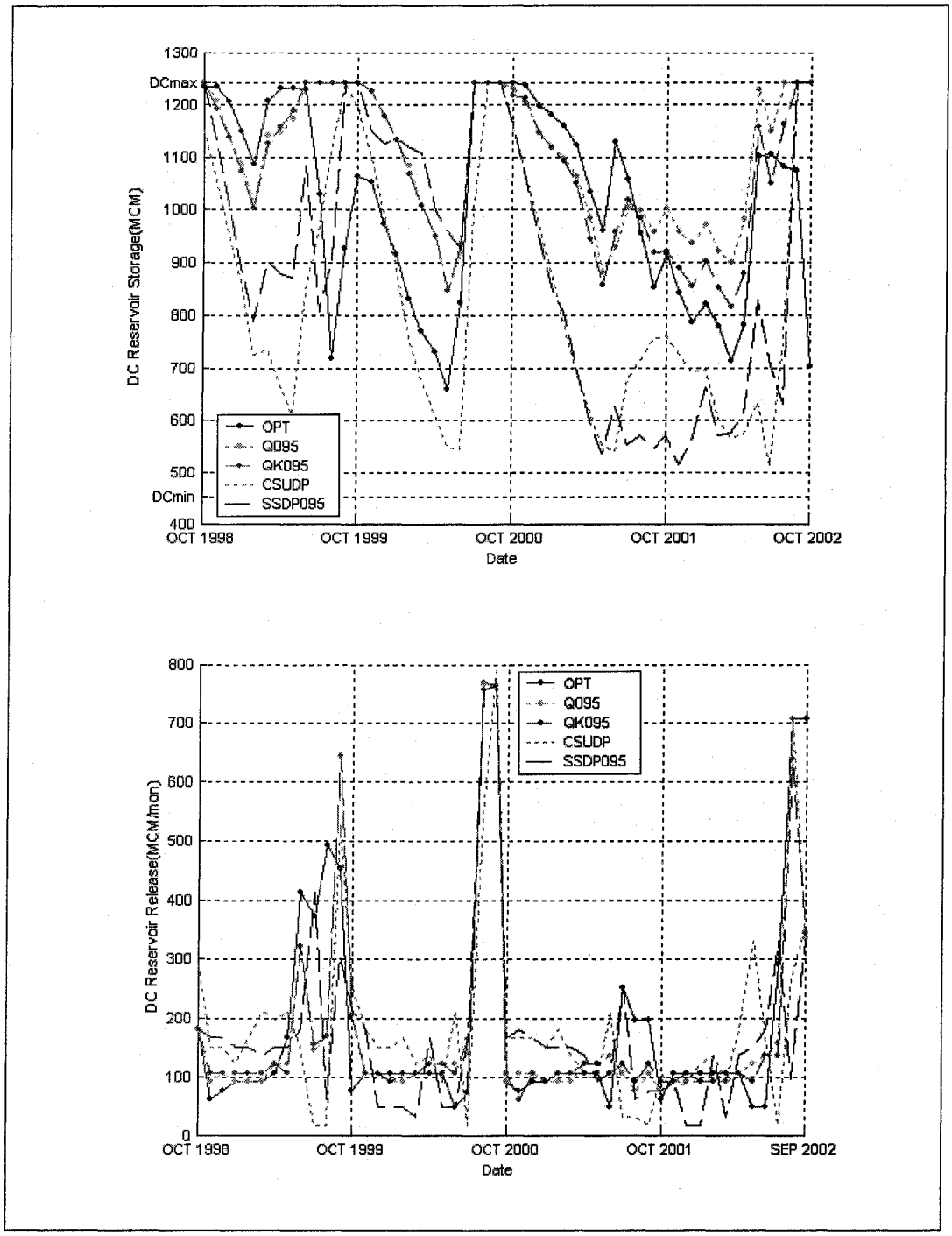


Figure 5.34 DC Reservoir Beginning Storage and Release

(80% Normal Full Storage)

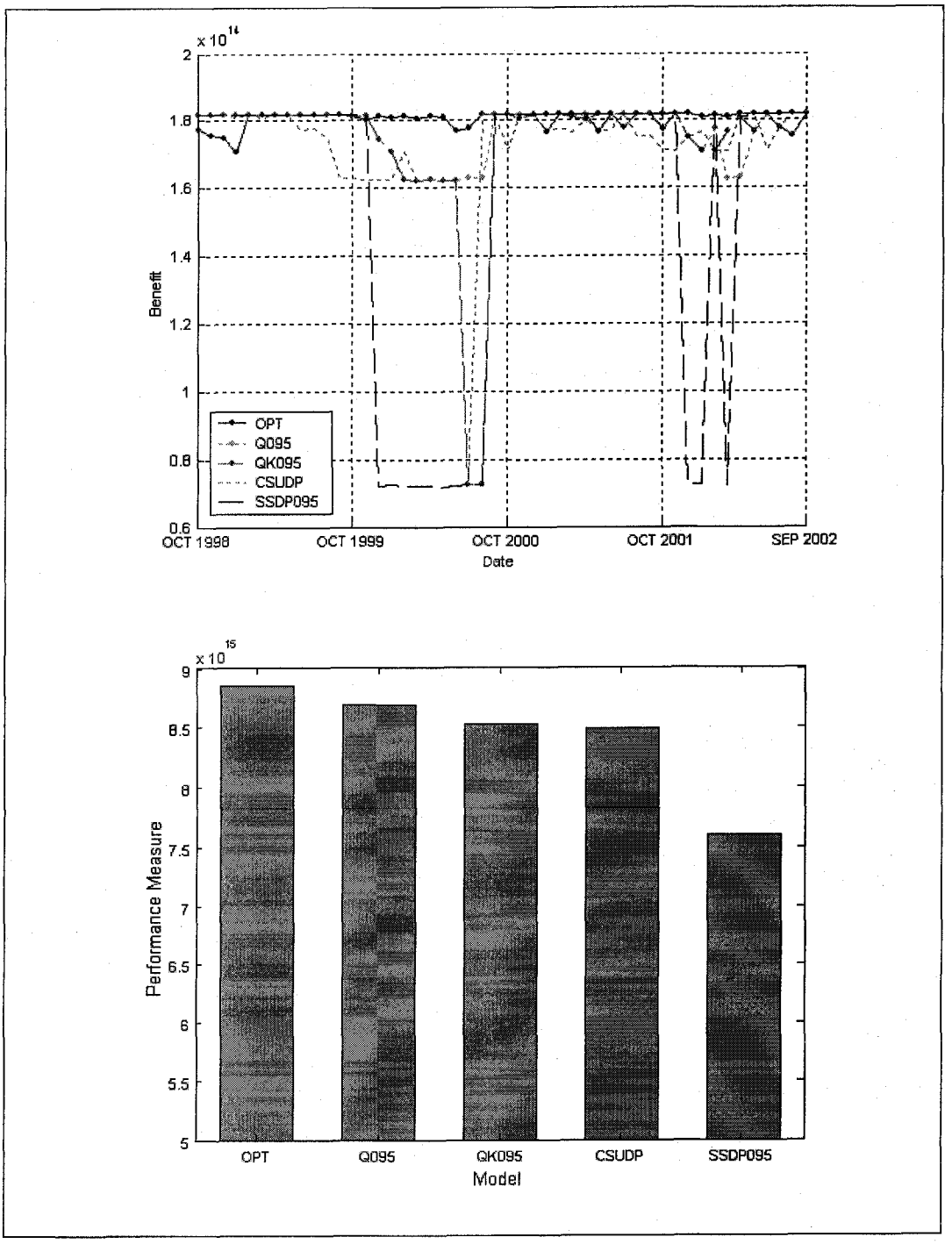


Figure 5.35 Operation Policy Comparisons - 80% Normal Full Storage

(Monthly and Total Performance Measures)

6. Summary and Conclusions

6.1. Summary

The analysis of large-scale water resources systems is often complicated by the presence of multiple reservoirs and diversions, the uncertainty of unregulated inflows and demands, and conflicting objectives. Reinforcement learning is presented herein as a new approach to solving the challenging problem of stochastic optimization of multi-reservoir systems. The conventional stochastic DP models have applied to the limited system requiring simplification and approximations that operators are unwilling to accept (2004, Labadie). The purpose of this study is to establish an optimization framework with reinforcement learning for realistic and reliable operation of multi-reservoir system so that it can reduce a gap between the theoretical investigation and the practical implementation.

Reinforcement learning is a computational framework for agents to operate and learn a dynamic environment system to achieve their long-term performance goals without any external supervisor. Reinforcement learning can be categorized as model-based approach and model-free approach. The model-based approach requires learning a model of the environment's dynamics such as state transition probability, whereas the model-free approach does not require learning a model. In this sense, the model-based approach is similar to conventional stochastic dynamic programming (SDP), except the solution technique involves a variation of policy iteration. The model-free approach uses historical data without requiring knowledge of the state transition probabilities in the system. The optimal value function and control policy are learned by continual interaction with the

environment. In addition, reinforcement learning does not require synthetic streamflow generation and inference methods required in implicit optimization method.

The Keum River basin in Korea was chosen as a case study to demonstrate the applicability of reinforcement learning basin wide reservoir operation. The Keum River basin consists of 12 sub basins and two major reservoirs, and the operation of this river basin includes water supply, flood control, hydropower generation, and instream flow requirements. These multiple objectives are combined into a single objective function for the dynamic programming optimization using the weighting method, assuming a unique performance measure for multi-objective exists. A detailed simulation procedure for Keum River basin is developed accurately to reflect the basin characteristics and consider every component in the basin. The simulation procedure additionally includes deficit sharing policy, was developed to allocate water deficits in accordance with the type of demand and level of use. This provides a dynamic approach to adjusting allocation of deficits considering the extent of demand satisfaction.

One of the model-free approaches called Q-Learning is used for generating integrated monthly operation rules for the Keum River basin. The Q-Learning model is evaluated by comparing with the implicit stochastic dynamic programming and sampling stochastic dynamic programming (SSDP) approaches. Implicit stochastic dynamic programming used SAMS for multivariate synthetic streamflow generation and CSDUP for development of integrated operation policies. The SSDP was used for long-term operation policy uses historical streamflow sequences although it was originally designed to use in realtime operation with forecasting model. The boundary condition for SSDP was also handled by the average return model with discount and scale weighting factors

that were used in Q-Learning approach. Although the available data covers a 19-year period, 15 years of data from October 1983 to September 1998 is used for generating operation policies in the Q-Learning and SSDP models and 4 years of data from October 1999 to October 2002 is used for policy performance evaluation and verification. The CSUDP model was run with ten sets of 50-year monthly data which is synthetically generated by analyzing 19-year historical data.

Evaluation of the stochastic basin-wide operational models considered several options relating to the choice of hydrologic state and discount factors as well as various stochastic dynamic programming models. The Q-Learning model performance with the hydrologic state definition by K-mean clustering was compared to the traditional percentile approach. The impact of discount factor on the operation policy was also investigated. Several stochastic optimization explicit models and implicit model are compared to identify the performance differences. Historical reservoir storage volume was used as an initial condition of simulation to represent the more realistic case for model verification purpose.

6.2. Conclusion

The overall conclusions drawn by this study can be briefly summarized in the followings.

1) Large scale basin-wide reservoir operation rules were derived from Q-Learning models as well as various explicit stochastic dynamic models. The serial and cross correlations of streamflow were preserved by using historical steamflow data.

2) The primary advantage of the Q-Learning model is that predetermined transition probabilities of inflow and post inference procedure to derive the operational rules are not required.

3) The operating rules by Q-Learning models outperformed the other rules derived by SSDP and implicit stochastic optimization models.

4) Unconditional operation rules from the Q-Learning model outperformed the conditional operation rules by 2%. This is due to the streamflow characteristics of Keum River basin or the serial correlation of Keum River basin is not strong enough to be explained by the conditional model or the coarse representation of hydrologic state definition.

5) Hydrologic state definition by k-mean clustering approach outperformed the simple percentile approach by 5%. It can be a good alternative to define hydrologic state in complex water resource system like Keum River basin.

6) A multiple linear or nonlinear regression model by implicit stochastic dynamic programming is well performed in basin-wide reservoir operation. However, the releases from the reservoirs are fluctuated according to the reservoir inflow condition.

7) Since SSDP was originally designed to use for the real time operation with a precise forecasting model it is not suitable for deriving long-term optimal operation rules.

8) Simulation by operation rules with the lower discount factor resulted in poor reservoir operation due to the lower valuation of future flow.

9) Optimal operation rules by Q-Learning using hydrologic state were not affected by discount factors since hydrologic state itself considers the uncertainty of future inflow.

10) Boundary condition for steady state model is easily handled average reward model with discount and scale weighting factors.

6.3. Recommendations for Future Work

The ultimate goal of developing steady state operational rules is to improve system performance in real-time operation. The stochastic optimization model usually results in the improvement in system performance. However, applying boundary condition for the real time operation model is difficult due to the difficulty of evaluating the future value of stored water in reservoir. The average reward Q-Learning approaches with model can directly be applied to real-time reservoir operation as boundary condition. Based on this modeling scheme, optimal immediate decision can be easily obtained by real-time operation model and Q-Learning model. In addition, the operation result and newly available data can be used for updating steady state optimal operation rules applying Q-Learning algorithm.

In this study, the Q-Learning approach was applied to the complex two-reservoir system with the complex representation of basin wide modeling. However, extending this Q-Learning approach to systems consisting of four or more reservoir system causes the dimensionality problem. The excessive computational burden by additional state variables limits the application of Q-Learning approach to more complex reservoir system. Future research on the stochastic optimization of multi-reservoir operations may employ information extraction techniques such as K-mean clustering and the localized policy improvement technique with the aggregated information.

The operation of this river basin includes multiple objectives such as water supply, flood control, hydropower generation. These multiple objectives were combined into a single objective function for the dynamic programming optimization using the weighting method in the study, assuming a unique performance measure for multi-objective exists. The more realistic approach requires applying economic and social cost coefficients instead of priority weighting factors.

A detailed simulation procedure for Keum River basin is developed accurately to reflect the basin characteristics and consider every component in the basin. However, the use of aquifers for storing water and the detailed modeling of groundwater is not considered in the study. This work can improve water supply reliability and efficiency by controlling the total water resources system.

References

Archibald T. W., C.S. Buchanan, K.I.M. McKinnon, L.C. Thomas. (1999). "Nested Benders decomposition and dynamic programming for reservoir optimization." Journal of the Operational Research Society, 50(5), 468-479.

Archibald T. W., K. I. M. McKinnon and L. C. Thomas. (1997). "An aggregate stochastic dynamic programming model of multireservoir systems." Water Resources Research, 33(2), 333-340.

Arunkumar S. and K. Chon. (1978). "On optimal regulation policies for certain multi-reservoir system. Operations Research, 26, 551-562.

Askew, A. (1974a). "Optimum reservoir operating policies and the imposition of a reliability constraint." Water Resources Research, 10(1), 51-56

Askew, A. (1974b). "Chance-constrained dynamic programming and the optimization of water resources systems." Water Resources Research, 10(6), 1099-1106

Askew, A. (1975). "Use of risk premiums in chance-constrained dynamic programming." Water Resources Research, 11(6), 862-866

Askew, A. (1978). "Comments on 'Reliability-Constrained Dynamic Programming and Randomized Release Rules in Reservoir Management' by Lewis A. Rossman." Water Resources Research, 14(1), 159-160

Barto, A. G., S. J. Bradtke and S. P. Singh. (1995). "Learning to act using real-time dynamic programming." Artificial Intelligence, Special Volume, Computational Research on Interaction and Agency, 72(1), 81-138

Bellman, R. (1957). Dynamic programming, Princeton University Press, Princeton, NJ.

Bellman, R. (1961). Adaptive control process: a guided tour, Princeton University Press, Princeton, NJ.

Bellman, R. and S. Dreyfus. (1962). Applied Dynamic Programming. Princeton University Press, Princeton, NJ

Benders, J. F. (1962). "Partitioning procedures for solving mixed-variables program problems," Numerische Mathematik, Vol. 4, 238-252.

Bertsekas, D. P. (1995). Dynamic Programming and Optimal Control. Athena, Belmont, MA.

Bertsekas, D. P. and J. N. Tsitsiklis. (1996). "Neuro-dynamic programming." Athena Scientific, Belmont, MA

Bhaskar, N. R. and E. E. Whitlatch. (1980). "Derivation of monthly reservoir release policies." Water Resources Research, 16(6), 987-993.

Bhattacharya, B., A. H. Lobbrecht, D. P. Solomatine. (2002). "Control of water levels of regional water systems using reinforcement learning." Proc. 5th Int. Conference on Hydroinformatics. Cardiff, UK. 1-6.

Bhattacharya, B., A. H. Lobbrecht, D. P. Solomatine. (2003). "Neural networks and reinforcement learning I control of water systems" Journal of Water Resources Planning and Management-ASCE, 129(6), 458–465.

Braga Jr., B.P.F., Yeh W. W-G., Becker L., and Barros M.T.L. (1991). "Stochastic optimization of multiple-reservoir-system operation." Journal of Water Resources Planning and Management-ASCE, 117(4), 471–481.

Buras, N. 1965. "A three dimensional optimization problem in water-resources engineering." Operational Research Quarterly, 16, 419-428.

Butcher, W. (1971). "Stochastic dynamic programming for optimum reservoir operation." Water Resources Bulletin, 7(1), 115-123

Chapman D., and L. P. Kaelbling. (1991). "Input generalization in delayed reinforcement learning: An algorithm and performance comparisons," in Proceedings of the International Joint Conference on Artificial Intelligence Sydney, Australia.

Chandramouli V., H. Raman. (2001). ''Multireservoir modeling with dynamic programming and neural networks.'' Journal of Water Resources Planning and Management-ASCE, 127(2), 89–98.

Chen, V.C.P., D. Ruppert, and C.A. Shoemaker. (1999). "Applying experimental design and regression splines to high-dimensional continuous-state stochastic dynamic programming." Operations Research, 47(1), 38-53.

Collins, M.A. (1977). "Implementation of an optimization model for operation of metropolitan reservoir system." Water Resources Bulletin, 13(1), 57-70.

Damas, M., M.Salmeron, A.Díaz, J.Ortega, A.Prieto, G.Olivares. (2000). "Genetic Algorithms and Neuro-dynamic Programming: Application to Water Supply Networks." Proc. of the 2000 Congress on Evolutionary Computation (CEC2000). La Jolla, CA, USA.7-14.

Damas, M., M. Salmeron, J. Ortega, G. Olivares. (2001). "Hybrid Framework for Neuro-dynamic Programming: Application to Water-Supply Networks." Sixth International Work-Conference on Artificial and Natural Neural Networks (IWANN 2001). Granada, Spain, 13-15 Junio, 2001. 719-727.

Doran, D. G. (1975). "Efficient transition definition for discrete state reservoir analysis." Water Resources Research, 13(2), 483-485

Faber B.A , J.R. Stedinger. (2001). "Reservoir optimization using sampling SDP with ensemble streamflow prediction (ESP) forecasts" Journal of Hydrology. 249 (1-4), 113-133

Feiring, B. R., T. Sastri, and L. S. M., Sim. (1998). "A stochastic programming model for water resource planning." Mathematical and Computer Modeling. 27(3), 1-7.

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

Foufoula-Georgiou, E. (1991). "Convex interpolation for gradient dynamic-programming." Water Resources Research. 27(1), 31-36.

Gal, S. (1979). "Optimal management of a multi-reservoir water-supply system." Water Resources Research, 15(4), 737-749.

Gessford, J. and S. Karlin (1958). "Optimal policy for hydroelectric operations", in Studies in the mathematical theory of inventory and production, edited by K. J. Arrow, S. Karlin, H. Scarf, 179-200, Stanford University Press, Stanford, CA.

Giles, J. and W. Wunderlich. (1981). "Weekly multipurpose planning model for TVA reservoir system," Journal of Water Resources Planning and Management-ASCE, 107(2), 1-20.

Gjelsvik A. (1982). "Stochastic seasonal planning of multi-reservoir hydro-electric power systems by differential dynamic programming." Modeling Identification And Control, 3(3), 131-149

Grygier, J.C. and J.R. Stedinger. (1985). "Algorithms for optimizing hydropower systems." Water Resources Research, 21(1), 1-10.

Grygier, J. C., and J. R. Stedinger, 1990, SPIGOT, A Synthetic Streamflow Generation Software Package, technical description, version 2.5, School of Civil and Environmental Engineering, Cornell University, Ithaca, N.Y.

Howard, R. (1960). Dynamic programming and Markov Processes. Mass. Institute of Technology Press, Cambridge, MA.

Jacobsen, D. and D. Mayne. (1970). Differential dynamic programming, Elsevier, New York, NY.

Johnson, S.A. , J.R. Stedinger, C.A. Shoemaker, Y. Li, and J.A. Tejadaguibert. (1993). "Numerical-solution of continuous-state dynamic programs using linear and spline interpolation." Operations Research. 41(3), 484-500.

Kaelbling, L. P., M. L. Littman, and A. W. Moore. (1996). Reinforcement learning: A survey. "Journal of Artificial Intelligence Research", 4, 237-285.

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

Karamouz, M., M.H. Houck, and J.W. Delleur. (1992). ''Optimization and simulation of multiple-reservoir systems.'' Journal of Water Resources Planning and Management-ASCE, 118(1), 71–80.

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

Kim Y.O., R.N. Palmer. (1997). "Value of seasonal flow forecasts in Bayesian stochastic programming," Journal of Water Resources Planning and Management-ASCE, 123(6), 327-335.

Klemes V. (1977). "Discrete Representation of Storage for Stochastic Reservoir Optimization." Water Resources Research, 13(1), 149-158.

Koutsoyiannis D., A. Efstratiadis, and G. Karavokiros (2002). "A Decision Support Tool for the Management of Multi-Reservoir Systems" Journal of the American Water Resources Association, 38(4), 945-958.

KOWACO (2003). Keum River basin operation guideline for MODSIM.

Kuman, P. R. and P. Varaiya. (1986). Stochastic Systems: Estimation, Identification, and Adaptive Control. Prentice-Hall, Englewood Cliffs, NJ.

Labadie, J.W. (2004). "Reservoir system optimization models." Journal of Water Resources Planning and Management-ASCE, 130(2) 93-111.

Labadie, J.W. (2003). Generalized Dynamic Programming Package, CSUDP, Documntation and User Guide Version 3.2a, Water Resources Planning and Management Division, Dept. of Civil Engineering, Colorado State University, Fort Collins, CO.

Lane, W. L., 1981, Corrected Parameter Estimates for Disaggregation Schemes, Inter. Symp. On Rainfall Runoff Modeling, Mississippi State University.

Lane, W. L., and D. K. Frevert, 1990, Applied Stochastic Techniques, personal computer version 5.2, users manual, Bureau of Reclamation, U.S. Dep. of Interior, Denver, Colorado.

Larson, R. E. (1968). State increment dynamic programming, American Elsevier, New York, NY.

Larson R.E., A.J. Korsak. (1970). "Dynamic programming successive approximations technique with convergence proofs." Automatica, 6(2), 245-252.

Little, J. D. C. (1955). "The use of storage water in a hydroelectric system," Journal of the Operations Research Society of America, 3, 187-197.

Liu, B. (2001). "Uncertain programming, a unifying optimization theory in various uncertain environments." Applied Mathematics and Computation. 120 (1-3). 227-234.

Lund, J.R., I. Ferreira. (1996). ''Operating rule optimization for Missouri River reservoir system.'' Journal of Water Resources Planning and Management-ASCE, 122 (4) 287-295.

Lund, J. R. and J. Guzman. (1999). ''Derived operating rules for reservoirs in series or in parallel.'' Journal of Water Resources Planning and Management-ASCE, 125(3) 143-153.

Mawer, P., and D. Thorn. (1974). "Improved dynamic programming procedures and their practical application." Water Resources Research. 10(2),183-190.

Mejia, J. M., and J. Rousselle, 1976. "Disaggregation Models in Hydrology Revisited." Water Resources Research, vol. 12(2), 185-186.

Mizyed N. R., J. C. Loftis, and D.G. Fontane. (1992). ''Operation of large multireservoir systems using optimal-control theory.'' Journal of Water Resources Planning and Management-ASCE, 118(14) 371-387.

Murray D. and S. Yakowitz. (1979). "Constrained differential dynamic programming and its application to multi-reservoir control." Water Resources Research, 15(5), 1017-1027.

Nopmongcol, P. and A. Askew. (1976). "Multi-level incremental dynamic programming." Water Resources Research, 12(6), 1291-1297.

Oliveira, R. and D.P. Loucks. (1997). "Operating rules for multireservoir systems." Water Resources Research, 33(4), 839-852.

Peng, C. and N. Buras. (2000). "Dynamic operation of a surface water resources system." Water Resources Research, 39(9), 2701-2709.

Pereira M.V.F., L.M.V.G. Pinto. (1985). "Stochastic optimization of a multireservoir hydroelectric system - a decomposition approach," Water Resources Research, 21(6), 779-792

Pereira M.V.F., L.M.V.G. Pinto. (1991). "Multistage stochastic optimization applied to energy planning." Mathematica Programming, 52 (2), 359-375

Philbrick, C.R. and P.K. Kitanidis. (1999). "Limitations of deterministic optimization applied to reservoir operations" Journal of Water Resources Planning and Management-ASCE, 125(3) 135-142.

Philbrick, C.R. and P.K. Kitanidis. (2001). "Improved dynamic programming methods for optimal control of lumped-parameter stochastic systems." Operations Research, 49 (3), 398-412.

Raman, H. and V. Chandramouli. (1996). "Deriving a general operating policy for reservoirs using neural network." Journal of Water Resources Planning and Management-ASCE, 122(5), 342-347.

Reznicek, K. and T. C. E. Cheng. (1991). "Stochastic modeling of reservoir operations." European Journal of Operational Research, 50, 235-248.

Roefs, T. G. and L. D. Bodin. (1970). "Multireservoir operation studies," Water Resources Research, 6(2), 410-420.

Rummery, G. A. and M. Niranjan. (1994). On-line Q-learning using connectionist systems. Technical Report CUED/FINFENG/TR 166. Engineering Department, Cambridge University.

Russell, C. (1972). "An optimal policy for operating a multi-purpose reservoir." Operations Research, 20(6), 1181-1189.

Saad M, A. Turgeon. (1988). "Application of principal component analysis to long-term reservoir management," Water Resources Research, 24 (7), 907-912

Saad M, A. Turgeon A., and J.R. Stedinger. (1992). "Censored-data correlation and principal component dynamic-programming," Water Resources Research, 28 (8), 2135-2140

Saad, M., A. Turgeon, P. Bigras and R. Duquette. (1994). "Learning disaggregation technique for the operation of long-term hydroelectric power systems," Water Resources Research, 30(11), 3195-3202.

Salas, J. D., N. Saada, C.H. Chung, W.L. Lane, and D.K. Frevert, 2000, Stochastic Analysis, Modeling, and Simulation (SAMS) Version 2000, users manual, Water Resources, Hydrologic and Environmental Sciences Engineering Research Center, Fort Collins, Colorado.

Salas, J. D., 1993, Analysis and Modeling of Hydrologic Time Series, Handbook of Hydrology, Chap. 19, pp.19.1-19.72, edited by D. R. Maidment, McGraw-Hill, Inc., New York.

Salas, J. D., J. W. Delleur, V. Yevjevich, and W. L. Lane, 1980, Applied Modeling of Hydrologic Time Series, WWP, Littleton, Colorado.

Samuel, A. L. (1959). "Some studies in machine learning using the game of checkers." IBM Journal on Research and Development, 210-229.

Schweig, Z., and J. A. Cole (1968). "Optimum control of linked reservoirs." Water Resources Research, 4(3), 479–497.

Sen, S. (2001). Stochastic Programming, Computational Issues and Challenges. In Encyclopedia of Operations Research and Management Science, 2nd edition, ed. S. Gass and C. Harris, 748-751. Boston, Kluwer Academic Publishers.

Sniedovich, M. and D. R. Davis. (1975). "Comment on 'Chance-Constrained Dynamic Programming and Optimization of Water Resource Sytems' by Arthur J. Askew." Water Resources Research, 11(6), 1037-1038.

Stedinger J.R., B.F. Sule, and D.P. Loucks. (1984). "Stochastic dynamic-programming models for reservoir operation optimization." Water Resources Research. 20(11), 1499-1505.

Sutton, R. S. (1990). "Integrated architectures for learning, planning, and reacting based on approximating dynamic programming," in Proceedings of the Seventh International Conference on Machine Learning, 216--224, San Mateo, CA. Morgan Kaufmann.

Sutton, R. S. (1991). "Planning by incremental dynamic programming," in Birnbaum, L. A. and Collins, G. C., editors, Proceedings of the Eighth International Workshop on Machine Learning, 353--357, San Mateo, CA. Morgan Kaufmann.

Sutton, R.S. and A.G. Barto. (1998). Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA.

Tai, F-K. and C. Goulter. (1987). "A stochastic dynamic programming based approach to the operation of a multi-reservoir system." Water Resources Bulletin, 23(3), 371-377.

Taylor, Howard M. and S. Karlin. (1984). An introduction to stochastic modeling. Academic Press, Orlando.

Tejada-Guilber J., S. Johnson, and J. Stedinger. (1995). "The value of hydrologic information in stochastic dynamic programming models of a multireservoir system." Water Resources Research, 3(10), 2571-2579.

Trezos T. and W. Yeh. (1989). Stochastic dynamic programming applied to multi-reservoir systems, in Computerized Decision Support Systems for Water Managers. J. W. Labadie et al., editors, American Society of Civil Engineers, New York

Trott, W. and W. Yeh. (1971). Multi-Level Optimization of a Reservoir System. Presented at the Annual National Environmental Engineering Meeting, ASCE, St. Louis, MO, October, 1971

Turgeon, A. (1980). "Optimal short-term hydro scheduling from the principle of progressive optimality." Water Resources Research, 16(2), 275-283.

Turgeon, A. (1981). "Optimal operation of multireservoir power systems with stochastic inflows." Water Resources Research, 17(3), 481-486.

Turgeon, A. (1985). The Weekly Scheduling of Hydroelectric Power Plants subject to Reliability Constraints. Hydro-Quebec, Canada

Turgeon, A. and R. Charbonneau. (1998). "An aggregation-disaggregation approach to long-term reservoir management." Water Resources Research, 34(12), 3585-3594.

U. S. Army Corps of Engineers, 1971, HEC-4 Monthly Streamflow Simulation, Hydrologic Engineering Center, Davis, California.

Valdes, J.B., J. Montbrun-DiFilippo, K.M. Straepeck, and P.J. Restrepo. (1992). "Aggregation-disaggregation approach to multireservoir operation," Journal of Water Resources Planning and Management-ASCE, 118(4), 423-444.

Valencia, D., and J. C. Schaake, Jr., 1973. "Disaggregation Processes in Stochastic Hydrology." Water Resources Research, 9(3), 580-585.

Wang, D. and Adams, B. (1986). "Optimization of Real-Time Reservoir Operations with Markove Decision Processes'' Water Resources Research, 22(3), 345-352.

Watkins, D. W. and D. C. McKinney. (1997). "Finding robust solutions to water resources problems," Journal of Water Resources Planning and Management-ASCE, 123(1), 49-58.

White, D.J. (1963). "Dynamic programming, Markov chains, and the method of successive approximations." Journal of Mathematical Analysis and Applications, 6, 373-376.

Willis R., B.A. Finney B.A, and W.S. CHU. (1984). "Monte-carlo optimization for reservoir operation.'' Water Resources Research, 20(9), 1177-1182.

Wurbs, R. A. (1993). ''Reservoir-system simulation and optimization models.'' Journal of Water Resources Planning and Management-ASCE, 119(4), 455–472.

Wurbs, R. A. (1996). Modeling and Analysis of Reservoir System Operations. Prentice Hall PTR, Upper Saddle River, NJ.

Yakowitz, S. (1982). ''Dynamic programming review.'' Water Resources Research, 18(4), 673–696.

Yeh, W. (1985). ''Reservoir management and optimization models: A state-of-the-art review.'' Water Resources Research, 21(12), 1797–1818.

Yeh, W. and W. Trott. (1972). Optimization of Water Resources Development, Optimization of Capacity Specification for Components of Regional, complex, Integrated, Multi-Purpose Water Resources Systems. UCLA, Eng. Rep. 7245, Univ. of California, Los Angeles, CA

Yevjevich, V. (1992). "Water and civilization." Water International, 17, 163-171.

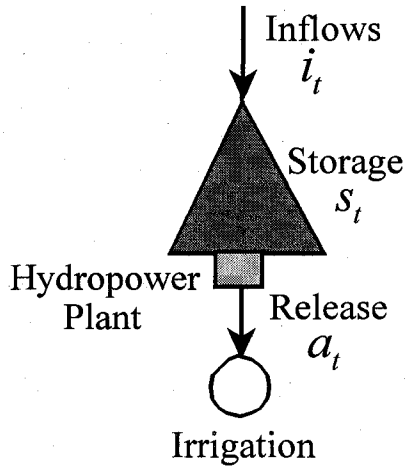
Yi, J. E., (1996). Decision support system for optimal basin-wide scheduling of hydropower units, Colorado State University, Fort Collins, CO

Young, G. K., Jr. (1967). ''Finding reservoir operating rules.'' Journal of Hydraulic Division-ASCE, 93(6), 297–321.

APPENDIX: STOCHASTIC DYNAMIC PROGRAMMING–EXAMPLE PROBLEM

A reservoir is to be operated over 2 time periods. The discrete probabilities for the inflows are given as follows:

Stage t	Inflows (i_{tk})		Probabilities (p_{tk})	
	k=1	k=2	k=1	k=2
1	1	2	0.2	0.8
2	3	4	0.3	0.7



The following constraints apply to storage volumes s_t and releases a_t:

$3 \le s_t \le 5$; $\Delta s = 1$

$0 \le a_t \le 4$; $\Delta a = 1$

The ideal target storage level is 4, because the reservoir is used for hydropower. However, the ideal discharge is 2 to meet downstream irrigation needs. Therefore, the objective function is:

$$\min \sum_{t=1}^{2} \left[(s_{t+1} - 4)^2 + (a_t - 2)^2 \right]$$

(a) Conventional SDP Algorithm (Noninverted form)

stage 2:

$$V_2(s_2) = \min_{a_2} \sum_{k=1}^{2} p_{2k}\left[(s_{3k}-4)^2 + (a_2-2)^2\right]$$

subject to:

$s_{3k} = s_2 - a_2 + i_{2k}$

s_2	a_2	i_{2k}	s_{3k}	p_{2k}	$(s_{3k}-4)^2$ $+(a_2-2)^2$	$V_3(s_{3k})$		$V_2(s_2)$
3	0	3	6(infeas)					
		4	7(infeas)					
	1	3	5					
		4	6(infeas)					
	2	**3**	**4**	**0.3**	**0**	0	**0.7**	**0.7**
		4	**5**	**0.7**	**1**	0		
	3	3	3	0.3	2	0	1.3	
		4	4	0.7	1	0		
	4	3	2(infeas)					
		4	3					
4	1	3	6(ineas)					
		4	7(infeas)					
	2	3	5					
		4	6(infeas)					
	3	**3**	**4**	**0.3**	**1**	0	**1.7**	**1.7**
		4	**5**	**0.7**	**2**	0		
	4	3	3	0.3	5	0	4.3	
		4	4	0.7	4	0		
5	1	3	7(infeas)					
		4	8(infeas)					
	2	3	6(infeas)					
		4	7(infeas)					
	3	3	5					
		4	6(infeas)					
	4	**3**	**4**	**0.3**	**4**	0	**4.7**	**4.7**
		4	**5**	**0.7**	**5**	0		

stage 1:

$$V_1(V_1) = \min_{a_1} \sum_{k=1}^{2} p_{1k}\left[(s_{2k}-4)^2 + (a_1-2)^2 + V_2(s_{2k})\right]$$

subject to:

$$s_{2k} = s_1 - a_1 + i_{1k}$$

s_1	a_1	i_{1k}	s_{2k}	p_{1k}	$(s_{2k}-4)^2$ $+(a_1-2)^2$	$V_2(s_{2k})$		$V_1(s_1)$
3	0	1	4	0.2	4	1.7	8.9	
		2	5	0.8	5	4.7		
	1	**1**	**3**	**0.2**	**2**	**0.7**	**2.7**	**2.7**
		2	**4**	**0.8**	**1**	**1.7**		
	2	1	2(infeas)	0.2				
		2	3	0.8				
4	1	1	4	0.2	1	1.7	5.9	
		2	5	0.8	2	4.7		
	2	**1**	**3**	**0.2**	**1**	**0.7**	**1.7**	**1.7**
		2	**4**	**0.8**	**0**	**1.7**		
	3	1	2(infeas)					
		2	3					
5	1	1	5					
		2	6(infeas)					
	2	1	4	0.2	0	1.7	4.9	
		2	5	0.8	1	4.7		
	3	**1**	**3**	**0.2**	**2**	**0.7**	**2.7**	**2.7**
		2	**4**	**0.8**	**1**	**1.7**		
	4	1	2(infeas)					
		2	3					

Final Results for noninverted form:

s_t	$V_1(s_1)$	$a_1^*(s_1)$	$a_2^*(s_2)$
3	2.7	1	2
4	1.7	2	3
5	2.7	3	4

(b) Q-Learning Algorithm (Noninverted form)

Set $V_1(s_1)$=10 (maximum total cost at stage 1), $V_2(s_2)$=5(maximum total cost at stage 2), $V_3(s_3)$=0

iteration 1:

stage 2:

$$V_2(s_2) = \min_{a_2} \sum_{k=1}^{2} p_{2k}\left[(s_{3k}-4)^2 + (a_2-2)^2\right]$$

subject to:

$$s_{3k} = s_2 - a_2 + i_{2k}$$

s_2	a_2	i_{2k}	s_{3k}	p_{2k}	$(s_{3k}-4)^2$ $+(a_2-2)^2$	$V_3(s_{3k})$		$V_2(s_2)$
3	2	3 4	4 5	0.3 0.7	0 1	0	0.7	**0.7**
4	4	3 4	3 4	0.3 0.7	5 4	0	4.3	**4.3**
5	4	3 4	4 5	0.3 0.7	4 5	0	4.7	**4.7**

stage 1:

$$V_1(s_1) = \min_{a_1} \sum_{k=1}^{2} p_{1k}\left[(s_{2k}-4)^2 + (a_1-2)^2 + V_2(s_{2k})\right]$$

subject to:

$$s_{2k} = s_1 - a_1 + i_{1k}$$

s_1	a_1	i_{1k}	s_{2k}	p_{1k}	$(s_{2k}-4)^2$ $+(a_1-2)^2$	$V_2(s_{2k})$		$V_1(s_1)$
3	0	1 2	4 5	0.2 0.8	4 5	4.3 4.7	9.42	**9.42**
4	2	1 2	3 4	0.2 0.8	1 0	0.7 4.3	3.78	**3.78**
5	2	1 2	4 5	0.2 0.8	0 1	4.3 4.7	5.42	**5.42**

iteration 2

stage 2:

$$V_2(s_2) = \min_{a_2} \sum_{k=1}^{2} p_{2k}\left[(s_{3k}-4)^2 + (a_2-2)^2\right]$$

subject to:

$s_{3k} = s_2 - a_2 + i_{2k}$

s_2	a_2	i_{2k}	s_{3k}	p_{2k}	$(s_{3k}\text{-}4)^2$ $+(a_2\text{-}2)^2$	$V_3(s_{3k})$		$V_2(s_2)$
3	3	3 4	3 4	0.3 0.7	2 1	0	1.3	0.7
4	3	3 4	4 5	0.3 0.7	1 2	0	1.7	**1.7**
5	4	3 4	4 5	0.3 0.7	4 5	0	4.7	4.7

stage 1:

$$V_1(s_1) = \min_{a_1} \sum_{k=1}^{2} p_{1k}\left[(s_{2k}-4)^2 + (a_1-2)^2 + V_2(s_{2k})\right]$$

subject to:

$s_{2k} = s_1 - a_1 + i_{1k}$

s_1	a_1	i_{1k}	s_{2k}	p_{1k}	$(s_{2k}\text{-}4)^2$ $+(a_1\text{-}2)^2$	$V_2(s_{2k})$		$V_1(s_1)$
3	1	1 2	3 4	0.2 0.8	2 1	0.7 1.7	2.7	**2.7**
4	2	1 2	3 4	0.2 0.8	1 0	0.7 1.7	1.7	**1.7**
5	2	1 2	4 5	0.2 0.8	0 1	1.7 4.7	4.9	**4.9**

iteration 3

stage 2:

$$V_2(s_2) = \min_{a_2} \sum_{k=1}^{2} p_{2k}\left[(s_{3k}-4)^2 + (a_2-2)^2\right]$$

subject to:

$s_{3k} = s_2 - a_2 + i_{2k}$

s_2	a_2	i_{2k}	s_{3k}	p_{2k}	$(s_{3k}-4)^2 +(a_2-2)^2$	$V_3(s_{3k})$		$V_2(s_2)$
3	2	3 4	4 5	0.3 0.7	0 1	0	0.7	0.7
4	3	3 4	4 5	0.3 0.7	1 2	0	1.7	1.7
5	4	3 4	4 5	0.3 0.7	4 5	0	4.7	4.7

stage 1:

$$V_1(s_1) = \min_{a_1} \sum_{k=1}^{2} p_{1k}\left[(s_{2k}-4)^2 + (a_1-2)^2 + V_2(s_{2k})\right]$$

subject to:

$s_{2k} = s_1 - a_1 + i_{1k}$

s_1	a_1	i_{1k}	s_{2k}	p_{1k}	$(s_{2k}-4)^2 +(a_1-2)^2$	$V_2(s_{2k})$		$V_1(s_1)$
3	1	1 2	3 4	0.2 0.8	2 1	0.7 1.7	2.7	2.7
4	2	1 2	3 4	0.2 0.8	1 0	0.7 1.7	1.7	1.7
5	2	1 2	4 5	0.2 0.8	0 1	1.7 4.7	4.9	4.9

iteration 4:

stage 2:

$$V_2(s_2) = \min_{a_2} \sum_{k=1}^{2} p_{2k}\left[(s_{3k}-4)^2 + (a_2-2)^2\right]$$

subject to:

$s_{3k} = s_2 - a_2 + i_{2k}$

s_2	a_2	i_{2k}	s_{3k}	p_{2k}	$(s_{3k}\text{-}4)^2 + (a_2\text{-}2)^2$	$V_3(s_{3k})$		$V_2(s_2)$
3	2	3 4	4 5	0.3 0.7	0 1	0	0.7	0.7
4	3	3 4	4 5	0.3 0.7	1 2	0	1.7	1.7
5	4	3 4	4 5	0.3 0.7	4 5	0	4.7	4.7

stage 1:

$$V_1(s_1) = \min_{a_1} \sum_{k=1}^{2} p_{1k}\left[(s_{2k}-4)^2 + (a_1-2)^2 + V_2(s_{2k})\right]$$

subject to:

$s_{2k} = s_1 - a_1 + i_{1k}$

s_1	a_1	i_{1k}	s_{2k}	p_{1k}	$(s_{2k}\text{-}4)^2 + (a_1\text{-}2)^2$	$V_2(s_{2k})$		$V_1(s_1)$
3	1	1 2	3 4	0.2 0.8	2 1	0.7 1.7	2.7	2.7
4	2	1 2	3 4	0.2 0.8	1 0	0.7 1.7	1.7	1.7
5	3	1 2	3 4	0.2 0.8	2 1	0.7 1.7	2.7	**2.7**

Final Results for noninverted form:

s_t	$V_1(s_1)$	$a_1^*(s_1)$	$a_2^*(s_2)$
3	2.7	1	2
4	1.7	2	3
5	2.7	3	4

Number of iteration required for Q-Learning convergence (50 trials):