A REINFORCEMENT LEARNING ALGORITHM FOR OPERATIONS PLANNING OF A HYDROELECTRIC POWER MULTIRESERVOIR SYSTEM

by

ALAA EATZAZ ABDALLA

B.Sc. Ain Shams University, 1984
M.A.Sc. Katholieke Universiteit Leuven, 1990

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE STUDIES

(CIVIL ENGINEERING)

THE UNIVERSITY OF BRITISH COLUMBIA
April 2007

ABSTRACT

The main objective of reservoir operations planning is to determine the optimum operation policies that maximize the expected value of the system resources over the planning horizon. This control problem is challenged with different sources of uncertainty that a reservoir system planner has to deal with. In the reservoir operations planning problem, there is a trade-off between the marginal value of water in storage and the electricity market price. The marginal value of water is uncertain too and is largely dependent on storage in the reservoir and storage in other reservoirs as well. The challenge here is how to deal with this large scale multireservoir problem under the encountered uncertainties.

In this thesis, the use of a novel methodology to establish a good approximation of the optimal control of a large-scale hydroelectric power system applying Reinforcement Learning (RL) is presented. RL is an artificial intelligence method to machine learning that offers key advantages in handling problems that are too large to be solved by conventional dynamic programming methods. In this approach, a control agent progressively learns the optimal strategies that maximize rewards through interaction with a dynamic environment. This thesis introduces the main concepts and computational aspects of using RL for the multireservoir operations planning problem.

A scenario generation-moment matching technique was adopted to generate a set of scenarios for the natural river inflows, electricity load, and market prices random variables. In this way, the statistical properties of the original distributions are preserved.

The developed reinforcement learning reservoir optimization model (RLROM) was successfully applied to the BC Hydro main reservoirs on the Peace and Columbia Rivers. The model was used to: derive optimal control policies for this multireservoir system, to estimate the value of water in storage, and to establish the marginal value of water / energy. The RLROM outputs were compared to the classical method of optimizing reservoir operations, namely, stochastic dynamic programming (SDP), and the results for one and two reservoir systems were identical. The results suggests that the RL model is much more efficient at handling large scale reservoir operations problems and can give a very good approximate solution to this complex problem.

TABLE OF CONTENTS

LIST OF FIGURES

MDP	Markovian decision process
MM	Moment matching
MRE	Mean relative error
MW	Megawatt
MVW	Marginal value of water
NDP	Neuro dynamic programming
NFM	Network flow model
NLP	Non-linear programming
pdf	probability density function
PWL	Piecewise linear function
RL	Reinforcement learning
RLROM	Reinforcement learning reservoir optimization model
RP	Reliability programming
RTC	Real time control
SDP	Stochastic dynamic programming
SDDP	Stochastic dual dynamic programming
SLP	Stochastic linear programming
SLPR	Stochastic linear programming with recourse
STOM	Short term optimization model
TD	Temporal difference
TPM	Transition probability matrix
TRM	Transition reward matrix
WK	Weekday
WKPK	Weekday peak load hours
WKLO	Weekday light load hours
WKHI	Weekday high load hours
WE	Weekend

GLOSSARY

Agent	A learner or controller or decision maker
Age of the agent	The number of iterations performed by the agent
Capacity	The rated power output of a power plant, normally measured in MW
Demand	The rate at which electric energy is delivered to or by a system, generally expressed in MW, or averaged over a period of time (e.g., MWh)
Energy	The amount of electricity produced or used over a period of time usually MWh
Environment	The thing that the agent interacts with to provide a control signal. It could be a real system or a model of the real system.
Episode	Trial or iteration or cycle. Represents the planning period. Each episode ends at a terminal time period T
Greedy policy	A policy that follows always the best action
$\in -$ greedy policy	A policy that follows the best action with probability $\in$ while exploring new actions with probability $1-\in$
Planning horizon	The number of stages / time periods
Operating reserve	A specific level of reserve power should be available at all times to insure reliable electricity grid operation.
Step size	Learning rate parameter α
Sub-time step	The time step is divided to a number of sub-time steps to capture the variation in certain parameters or variables at a shorter time increment
Time step	The planning period T is subdivided to number of periods which are called the time steps or stages t
Target	A desirable direction to move in the next step
Terminal stage	The last period in an iteration (the end of the planning horizon)

ACKNOWLEDGEMENTS

I greatly acknowledge the support and invaluable insights of my thesis advisor Dr. Ziad Shawwash. Without his guidance this thesis would probably never have been completed. My understanding of topics relating to this research has been shaped by his guidance throughout the course of this research. I wish to extend my special thanks to Dr. S.O. Denis Russell for his wise advice and suggestions. I would like also to express my sincere thanks to Dr. A.D. Russell, Dr. J.A. Meech, Dr. R.G. Millar, and Dr. T.A. Sayed for their helpful comments and feedback throughout the course of this research.

Special appreciations are due to Dr. Thomas Siu for his continuous help and support. This thesis has benefited from his expertise, deep understanding, and explanations of operations planning of BC Hydro reservoir systems. I am also thankful to my colleagues in BC Hydro: D. Smith, D. Druce, Dr. G. Nash, H. Louie, W. Cheng, and Dr. G. Sreckovic for their valuable assistance and support. Interactions with UBC students have also been fruitful. I would like to acknowledge J. Li, Y. Tang, and R. Mazariegos for their help.

Special thanks are due to Dr. B. Abdulhai. My understanding of Reinforcement Learning benefited from the discussions I had with him and from his knowledge of artificial intelligence.

I am very grateful for my wife Dalal, for her love, patience and support during the entire period of my study. Thanks for my son Aly for bringing me great joy in stressful time. Lastly, and most importantly, I wish to thank my parents, Zeinab and Eatzaz, for their great care and encouragement during my studies and throughout my life.

1. INTRODUCTION

1.1. Background

In British Columbia, there are large number of dams distributed throughout the different basins and watersheds of the province. Most of the large dams in B.C. serve hydroelectric power purposes. The British Columbia Hydro and Power Authority (BC Hydro) operates a hydro dominated power generation system that includes 30 hydroelectric generating stations providing approximately 90 per cent of the system installed capacity (11,100 MW). These plants can be placed into four main groups: Peace system (2 plants) producing about 34% of the energy requirements, Columbia system (3 plants) producing 31%, Kootenay Canal and Seven Mile generating stations producing 13% and the remaining 23 plants that supply about 16% of the energy production. The balance of energy requirements is supplied from two thermal generating facilities and from energy purchases. The BC Hydro integrated system produced about 54,000 gigawatt-hours in 2004.

The main hydroelectric storage facilities in the BC Hydro system are: the Williston reservoir (39.4 billion m^3) on the Peace River and the Kinbasket reservoir (14.8 billion m^3) on the Columbia River. The two reservoirs provide multi-year storage and accordingly, they are used for the strategic and long-term operations planning of the BC Hydro system.

The reservoir systems in B.C. provide many important benefits in addition to hydroelectric power production. These include: domestic and industrial water supply, flood control, and recreation. During the operation of these reservoir systems, conflicts may arise between some of these uses, particularly in periods of sustained drought or storage deficits. Also, new pressures have been imposed on the reservoir system due to expanding energy needs, increasing water demands, and growing public awareness of environmental issues. In addition, publicly owned reservoir systems often have to deal with complex legal agreements (e.g. Columbia River Treaty), fisheries, and non-power release requirements, Federal/Provincial regulations (e.g. water licenses), and pressures from different interest groups. Recently increased attention has been given to improving the economic and operating efficiency of the existing reservoir systems.

The electrical transmission network in British Columbia is interconnected with Alberta and Western US systems. These interconnections allow for the purchase and sale of electricity in the wholesale electricity market, and therefore complicate the reservoir operations planning problem, since it must incorporate maximizing the profit from energy transactions in these two markets.

The complexities of the multipurpose, multiple reservoir systems generally require that release decisions be determined by an optimization and/or simulation model. Computer simulation models have been applied for many years, as powerful tools in reservoir systems management and operation. These models are descriptive of the behavior of the system (e.g. releases, storage), given a specified scenario (e.g. the sequence of flow data, storage, demands, priorities, and constraints); and they are able to illustrate the changes resulting from alternative scenarios. They are also useful in examining the long-term reliability of a proposed operating strategy when combined with Monte Carlo analysis. However, they are not well suited to develop the optimal strategy, particularly for large-scale systems.

Prescriptive optimization models on the other hand, offer the capability to systematically derive optimal solution given a specific objective and a set of constraints. The operating policies developed with optimization procedures often need to be checked with a simulation model in order to ensure their feasibility.

In BC Hydro, the operations planning process is carried out using several simulation and optimization models. This modeling system is divided in terms of time horizons. The cascaded operations planning modeling hierarchy in BC Hydro is categorized as follows:

- Long-term operations planning (strategic planning, 1-6 years in monthly time-steps).

- Medium-term operations planning (strategic/tactical planning, up to12 month period in hourly to weekly time-steps).
- Short-term operations planning (tactical planning, for one week in hourly to daily time-steps).
- Real-time operations planning (1 hour-1 week in hourly time-steps).

For long term planning, a marginal cost model (MCM) has been developed in-house at BC Hydro by Druce (1989, 1990), and kept up to date. The model applies Stochastic Dynamic Programming (SDP) to calculate the monthly marginal value of water stored in the Williston reservoir, the largest reservoir in the system, over a planning period of 4 to 6 years. The SDP model takes into account the uncertainties in inflows and market prices. The calculated marginal value of water stored in the Williston reservoir is then used as a proxy for the long-term system marginal cost. As well, the model develops a probabilistic forecast of BC Hydro system price signals, reservoir storage volumes and expected spills.

Shawwash et al. (2000) developed a short-term optimization model (STOM) that is used by BC Hydro operation engineers to determine the optimal hourly generation and trading schedules while meeting system demands. The model applies a linear programming technique in which the inflow, prices, and system loads are deterministic within the short-term period. The model has been modified and extended to the medium-term generalized optimization model (GOM) that is capable to handle longer planning periods ranging from hourly to monthly time-steps. The model also allows for variable time steps that can be specified on daily, weekly, and monthly intervals, and to include sub-time steps within a time step. Hence, GOM has the advantage of capturing variations in load, market prices, and generation schedules for shorter periods within the time step during weekdays or weekends.

Nash (2003) developed a stochastic optimization model for the Columbia and Peace reservoir systems (aggregated to two reservoirs). The model consists mainly of two sub-models. In the first model, which is the long-term model, the monthly storage value function and the marginal values of water derived by a DP-LP based model are passed to the second model. The second model is the shorter-term model that applies a stochastic linear programming with recourse algorithm (SLPR) in which the inflows, prices, and demands are defined by a scenario tree. The storage value curves generated by the long-term model are used as terminal value functions in the shorter-term model. The outputs of the shorter-term model for the two aggregated reservoir systems are the refined marginal values over the shorter periods and the operating decisions for each period. The present research work builds upon, extends, and enhances the above techniques and develops a new approach to solve the operations planning problem for multireservoir systems.

1.2. Problem Statement

The main objective of reservoir operations planning is to determine the release policies that maximize the expected value of the system resources over the planning period. This objective is generally achieved by maximizing the value of the energy produced while meeting firm domestic demands and taking into consideration the value of stored water at the end of the planning horizon. The reservoir operating strategy should provide answers to these questions at every decision point in time: when and how much water to store or release and when, where, and what quantity of energy to trade in the competitive wholesale electricity market while meeting the firm domestic demands and non-power requirements.

This control problem is challenged with three main sources of uncertainty with which a reservoir system operator has to deal with. The main sources of uncertainty are the inflows to the reservoir system. The system operation policy has to protect against situations of shortage where the system is unable to meet the demands in a dry year. Also, it has to be capable to store any sudden increase in the inflow while avoiding wasteful water spills. In addition, there is uncertainty in forecasting electricity demand that will affect the amount of energy generated since the firm demand must be met regardless of cost. There is also uncertainty in market prices that varies seasonally, daily, and between weekdays and weekends.

The marginal value of water represents the value of an extra increment of storage in a given reservoir ($/m^3). In the reservoir operations optimization problem, there is a trade-off between the marginal value of water and the present market price. If the market price is higher than the marginal value of water, then it is more profitable to sell energy. On the other hand, when the market price is lower than the marginal value of water, then it is more profitable to purchase energy from the market. The marginal value of water is equal to the derivative of the value of water function with respect to storage. The optimal use of the water in the reservoir corresponds to the point that minimizes the sum of immediate and future costs. This is also where the derivatives of the immediate cost function and future cost function with respect to storage become equal (Pereira et al. 1989).

To derive a set of realistic control policies and better estimate the value of system resources, there is a need to integrate BC Hydro's main reservoir systems into a single model that takes into consideration the uncertainties in the inflows, market prices, and the system load. This integration is essential to allow for the effect of interactions between the different reservoir systems on the amount of total energy produced by the system. The value of water in a given reservoir is a function of the storage level in that reservoir and those in other reservoirs. Thus, the value of water in storage in any reservoir cannot be established unless assumptions are made about the other storage variables in the system. In this research, the main hydroelectric power generation facilities of BC Hydro system are included in a long/medium term stochastic optimization model. These hydroelectric power generation facilities are located on several independent river systems, for example, Peace River (G.M. Shrum and Peace Canyon) and Columbia River (Mica, Revelstoke, and Keenleyside).

1.3. Goals and Objectives

The main goal of this research work is to develop and implement a long/medium term stochastic optimization model that serves as a decision support tool for the operations planning of the BC Hydro's reservoir systems. The model should be able to:

- Handle the dimensionality problem of this large-scale stochastic problem in an efficient manner,
- Provide forecasts of the expected value of revenues, energy generation, and expected market transactions (imports/exports),
- Model several river systems within the BC Hydro system,
- Address the main uncertainties in the operations planning problem (inflow, market price, and demand),
- Provide the marginal value of water for the major reservoirs, and
- Deal with variable time steps and be able to address different objective functions.

To achieve these goals, several objectives were identified:

1. Acquire an in-depth understanding and knowledge of the reservoir operations planning problem in general and of BC Hydro's reservoir systems in particular. This was achieved by thoroughly investigating the modeling environment at BC Hydro and other hydroelectric power generation entities. Special attention was given to integrating the generalized optimization model (GOM) with the stochastic optimization model.

2. Carry out an extensive review of the literature on reservoir optimization techniques with a particular emphasis on stochastic optimization techniques. This literature review was extended to include state of the art techniques developed and applied in the fields of machine learning and artificial intelligence environments. The aim of the literature review was to assess the merits and the drawbacks of the different optimization techniques and their potential to handle the complexity and the dimensionality problems of the large scale, stochastic, multiperiod, multireservoir operations planning problem.

3. Formulate the stochastic optimization model that addresses the uncertainties in inflow, system load, and market prices.

4. Investigate and develop the Reinforcement Learning (RL) technique with Function Approximation and Sampling techniques, an approach, that is becoming popular and seems to offer the promise of handling large scale stochastic problems

5. Test the performance of the RL model and develop an algorithm to implement it for the optimization of the operation of BC Hydro's main reservoir systems.

1.4. Organization of the thesis

This chapter presented the motivation, goals and the focus of the thesis. **Chapter 2** reviews the different techniques and approaches in handling the reservoir optimization problem cited in the literature. **Chapter 3** introduces the main concepts and computational aspects of using reinforcement learning (RL). **Chapter 4** describes the methodology and the mathematical formulation adopted in the development of the RLROM model. **Chapter 5** presents the results of applying the RL solution methodology for a single reservoir and testing the extended two reservoir problem. The model was then implemented on BC Hydro's main reservoir system on the Peace and the Columbia Rivers. **Chapter 6** provides conclusions and recommendations for future research work.

2. LITERATURE REVIEW

The literature review carried out in this research is presented herein from two perspectives: modeling approaches and optimization techniques. The modeling approaches commonly applied to the reservoir optimization problem can be grouped into two main categories: Implicit Stochastic optimization and Explicit Stochastic optimization. In terms of the optimization techniques, the reservoir optimization models can be classified as:

- Deterministic models including: Linear programming, Network Flow, Multiobjective, Non Linear programming, and Dynamic Optimzation Models,
- Stochastic optimization models including: Dynamic, Linear, Dual Dynamic Programming, Chance Constrained and Reliability Programming,
- Heuristic models including: Genetic Algorithms, Artificial Neural Networks, Fuzzy Programming, etc…

2.1. Modeling Approaches

Deterministic analysis of reservoir operational problems has several computational advantages over the stochastic analysis. Ignoring the stochasticity of the system simplifies the model resulting in more efficient performance; however this simplification introduces a bias in the results. Loucks et al. (1981) state that: "Deterministic models based on average or mean values of inputs, such as stream flows, are usually optimistic. System benefits are overestimated, and costs and losses are underestimated if they are based only on the expected values of each input variable instead of the probability distributions describing those variables".

Reservoir optimization models can be useful tools for identifying and evaluating the impacts of various alternative system operations. Yet, these models are not likely to be very useful unless they consider the uncertain conditions affecting the future performance of those systems. This includes the uncertain future demands imposed on those systems, the uncertain future costs of those systems and the uncertain quantities and qualities of the flow within those systems. Assumptions made regarding each of these sources of uncertainty can have a major impact on system planning and operation. These facts have served to motivate the development of stochastic models, models that take into consideration at least some of the important sources of uncertainty and their impacts on system design and operation.

Implicit stochastic reservoir optimization models optimize over series of random variables assuming perfect knowledge of the future. Multiple regression analysis of the results of the deterministic optimization model can be applied to generate operational rules. However, Labadie (1998) claims that regression analysis can result in poor correlations that may invalidate the operating rules.

Explicit stochastic optimization models deal with the probabilistic nature of random variables directly, rather than dealing with deterministic sequences. Accordingly, the optimization is performed without assuming perfect knowledge of future events. The benefit of this approach is to better quantify the impact of the uncertainty in the random variables and consequently come up with better reservoir management decisions. However, this approach is more computationally expensive than the implicit optimization approach.

Most of the explicit stochastic optimization models assume that unregulated inflows are the dominant source of uncertainty in the system and can be represented by appropriate probability distributions. These may be parametric or nonparametric based on frequency analysis. Other random variables that may be defined include: market prices and demands. Unregulated natural inflows may be highly correlated spatially and/or temporally. Explicit stochastic models use probability distributions of stream flow. This requires two main simplifications to keep the dimensionality of the problem manageable. First, discretization of the probability data, and second, relatively simple stochastic models are usually used (e.g., lag-1 Markov model). Most inflow sequences show serial correlation (Pereira et al., 1999), and are represented in modeling inflows by a lag-1 autoregressive or multivariate model.

2.2. Optimization Techniques

A broad array of mathematical models has been applied for the optimization of reservoir systems operations and management. The choice of the modeling technique relies largely on the characteristics of each application. The following sections briefly review optimization methods that are widely used to solve the reservoir system optimization problem, with a focus on the techniques that are applied in multireservoir systems. Yeh (1985) and Labadie (1998, 2004) presented a comprehensive in-depth state of the art review of the optimization techniques used in reservoir operation and management.

2.2.1. Deterministic Optimization Models

2.2.1.1. Linear Programming Models

One of the most favored optimization techniques in reservoir system models is linear programming (LP). LP requires that all the constraints and objective function be linear or be "linearizable" by applying one of the available linearization techniques, such as piecewise linearization or Taylor series expansion. LP models guarantee convergence to global optimal solutions. In addition, for large-scale reservoir optimization problems where the number of variables and constraints are large, decomposition techniques such as Dantzig-Wolf or Bender decomposition technique can be used to accelerate the solution process (Yeh, 1985). LP problem formulation is easy and LP problems can be readily solved by applying commercially available LP solvers.

Turgeon (1987) applied a monthly mixed integer LP model for site selection of hydropower plants to be built. Hiew et al. (1989) applied LP technique to an eight reservoir system in northern Colorado.

Shawwash et al. (2000) presented an LP short term optimization model (STOM), which has subsequently been developed to determine the optimal short term schedules that meet the hourly load and maximizes the return to BC Hydro resources from spot transactions in the Western U.S. and Alberta energy markets. Currently, the model is used in BC Hydro by the generation operations engineers to optimize the scheduling of the main reservoir systems. The authors state that using the model to develop the optimized schedule contributed between 0.25-1.0% in the form of additional revenue from sales and in the value of additional stored water. The authors indicate that one of the major benefits of using LP is the derived sensitivity analysis data that can be obtained from the simplex dual variables. As an example, the dual variable of load resource balance equation provides the system incremental cost at each time step. This information can be used in planning spot trading schedules and in real time operation of the system.

Other applications of the LP technique to the reservoir operations problem include: Martin (1986), Piekutowski et al. (1993), and Crawley and Dandy (1993).

2.2.1.2. Network Flow Models

Network flow models (NFM) have been applied in a broad range of water resource applications, as they are easy to formulate and efficient to solve. A reservoir system is represented as a network of nodes that are linked by arcs. Nodes could represent storage or non-storage points of confluence or diversion and arcs represent releases, channel flows, and carryover storage. This representation also has the advantage of easily defining piecewise linear functions through the specification of multiple links between nodes. Flow bounds and unit costs are defined by the flow limits and slopes of each linear piece (Labadie, 1997).

Lund and Ferreira (1996) applied a network flow algorithm to the Missouri River multireservoir system. The multireservoir system is optimized for a period of 90 years in monthly time steps. The authors concluded that system operation rules could be inferred from deterministic optimization applying a long hydrologic sequence.

Shawwash et al. (2000) observed that some of the methodology's limitations were encountered when using arcs to describe flow patterns in a complex system, such as BC

Hydro's, that includes a combination of very large and very small reservoir systems, were encountered.

2.2.1.3. Multiobjective Optimization Models

Labadie (1997) presented two approaches for dealing with multiobjective optimization problems. In the first approach, the primary objective is represented in the objective function while treating the other objectives are treated as constraints at desired target levels (epsilon method). The second approach assigns weights to each objective (weighting method).

Can and Houck (1984) applied a preemptive goal programming (PGP) approach to a four reservoir multipurpose system in the Green River Valley, Kentucky. In their comparative study with other LP models the authors concluded that PGP allows the flexible expression of policy constraints as objectives and it performed well compared with a more data intensive LP optimal operating model. The basic concept of PGP is to set aspiration levels (targets) for each objective and prioritize them. Attainment of the goals is sought sequentially. A significant advantage of PGP is that it does not require any penalty-benefit function, reducing the need for a detailed economic analysis. However, one drawback of PGP is that it does not allow trading a small degradation in a high priority objective for a large improvement in a lower priority objective (Loganathan and Bhattachatya, 1990). As goal programming (GP) relies on achieving predetermined target values, the global optima for objectives may not be explored.

2.2.1.4. Nonlinear Programming Models

Non-linear programming (NLP) is not as popular as LP and dynamic programming (DP) in solving reservoir systems optimization problems. The reason is basically because the optimization process is slow and can return inferior and non-optimal solutions. However, in cases where a problem cannot be realistically linearized, it may be solved as a NLP problem particularly with inclusion of hydropower generation in the objective function and/or the constraints. Labadie (1997) indicates that the most powerful and robust NLP algorithms available to solve reservoir system optimization problems include: Sequential Linear Programming (SLP), Sequential Quadratic Programming (SQP), Method of Multipliers (MOM), and the Generalized Reduced Gradient Method (GRG).

Recent applications of NLP to hydropower reservoir operations include: Tejada-Guibert et al. (1990), Arnold et al. (1994), and Barros et al. (2003). Barros et al. (2003) applied NLP model to a large scale multireservoir system in Brazil. This multiobjective optimization problem was solved applying LP and SLP using a Taylor series expansion. The authors concluded that the NLP model is the most accurate and suitable for real-time operations than the LP model.

2.2.1.5. Dynamic Programming Models

Dynamic programming (DP) is another powerful optimization technique that has been used extensively to solve reservoir system optimization problems. Unlike LP and NLP techniques that simultaneously solve the problem for all time periods, DP algorithms decompose the original problem into sub-problems that are solved sequentially over each stage (time period). DP formulation requires the definition of a set of state variables to describe the system state at the beginning of each stage and a set of decisions that transform the current stage state to the next one. DP has the advantage of handling nonlinear, nonconvex, and discontinuous objective and constraint functions. However, a major problem that limits the application of DP to large-scale multireservoir systems is the exponential growth in computation time as the number of discretized state variables increases. This is widely known as the curse of dimensionality (Bellman, 1957).

One of the earliest applications of deterministic DP to reservoir operation was by Young (1967), who studied a finite horizon, single reservoir operation problem.

Various extensions have been developed to overcome the curse of dimensionality in applying dynamic programming application to reservoir operation. Larson (1968) introduced incremental dynamic programming (IDP). The IDP procedure starts with a trial solution and the recursive DP equation examines adjacent states around the current solution. If better values are obtained, then the current solution is updated. Jacobson and Mayne (1970) developed a Differential Dynamic Programming (DDP) technique that uses analytical solution rather than discretization of the state space. Murray and Yakowitz (1979) extended this approach to constrained problems. Johnson et al. (1993) introduced the Coarse Grid Interpolation technique. This technique relies on using larger discretization intervals. Solution accuracy is retained by interpolating within a coarser grid structure.

2.2.2. Stochastic Optimization Models

2.2.2.1. Stochastic Dynamic Programming Models

Stochastic dynamic programming (SDP) is a powerful tool for studying multireservoir system operation because the stochastic nature of inflows and the nonlinear energy generation functions can be modeled explicitly. Interestingly, Yakowitz (1982) found that the first application of SDP preceded the application of deterministic DP by more than a decade. Lamond and Boukhtouta (1996, 2001) and Lamond (2003) presented a survey of stochastic reservoir optimization methods and models.

A multireservoir, multiperiod SDP model is formulated by considering the multiperiod optimization in stages. Each stage corresponds to one period. Release decisions are made to maximize current benefits plus the expected benefits from future operation, which are represented by a recursively calculated cost to go function. Solution of the SDP model for a multireservoir system yields the "cost-to-go" function and a release policy decision rule for each time period as a function of the system state variables.

Since optimization is performed conditionally on all discrete combinations of the state vector, the specter of the curse of dimensionality arises. For a multireservoir model with m discritization levels for n reservoirs, computational time and storage requirements are proportional to m^n.

The state variable typically includes the volume of water in reservoirs and sometimes a description of current, or forecasted hydrological conditions (Kelman et al., 1990). A periodic Markovian process typically describes reservoir inflows in SDP models. The choice of the inflow state variable in an SDP model depends on the system's characteristics as well as the information available for decision-making. In addition, computational considerations often influence how hydrologic information is represented in SDP. Huang et al. (1991) applied four types of representations of the inflow state variable to the Feitsui reservoir SDP optimization model in Taiwan. The authors found that using previous period inflows resulted in superior performance compared to the use of the present period inflows. Piccardi and R. Soncini (1991) found that policies derived from an SDP model without a hydrologic state variable resulted in simulated performance similar to that of policies derived using the previous period's inflow, although the SDP and simulation agreed more closely when the previous period's inflow were employed.

Tejada-Guibert et al. (1995) examined the value of hydrologic information in SDP multireservoir models by using different hydrological state variables for the Shasta-Trinity subsystem of the Central Valley project in California. Then, the SDP policies were compared using a simulation model assuming that the current inflows were known. The authors applied four types of models with different inflow state variables, and concluded that the value of using sophisticated inflow forecasting depends on several system characteristics, including the relative magnitude of water and power demands and the severity of the penalties on shortages. Turgeon (2005) applied a parametric approach to represent the inflows by a linear autoregression (AR) model, used to solve the SDP reservoir management problem. Instead of using the traditional lag-1 models, the author stated that there are many advantages in considering multilag autocorrelation of inflows. To avoid an increase in state space, the multilag autocorrelation of inflows was represented by the conditional mean of the daily inflow.

The use of SDP to optimize multireservoir systems is usually accompanied by the assumption that various natural inflows are not cross correlated. This results in solutions that provide a rough estimate of the optimal design or operation policy. To handle this problem, Yeh (1985) suggested the separation of the DP optimization and stream flow generation, or using an aggregation/decomposition methods similar to those proposed by Turgeon (1980).

a) Dynamic Programming with Successive Approximation

The Dynamic Programming with Successive Approximation (DPSA) method consists of breaking up the original multi-state variable problem into a series of one-state variable sub-problems in such a manner that the sequence of optimizations over the sub-problems converges to the solution of the original problem. Davis et al. (1972) used the DPSA to determine a local feedback policy for each reservoir for a network of reservoir-hydroplant complexes in parallel. Pronovost and Boulva (1978) have used Davis' method to obtain an open-loop policy, which gives near optimal results rather than local feedback to eliminate the drawback of this method. Turgeon (1980) concluded that to obtain an open-loop policy solution, the successive approximation method must be solved repetitively at the beginning of each period that may be computationally costly.

b) Aggregation and Decomposition SDP

Turgeon (1980) introduced the aggregation and decomposition method consisting of breaking-up the original n-state variable stochastic problem into n stochastic sub-problems of two-state variables that are solved by SDP. The final result of this method is a suboptimal global feedback operating policy for the system of n reservoirs. Furthermore, Turgeon (1980) assumed that the electrical energy generated by any plant is a constant times the turbine releases. Accordingly, instead of utilizing the reservoir storage as a state variable, a potential energy term is created for treating the nonlinearity of the power generation function.

Turgeon (1980) applied the DPSA and the aggregation/decomposition methods to a network of 6 reservoir hydroplant complexes. In his comparative study he concluded that the later gives a better operating policy with the same time and computer memory requirements. In addition, the computational effort of the aggregation/decomposition method increases linearly with the number of reservoirs since for each additional reservoir, only one additional DP two-state problem has to be solved.

Valdes et al. (1992) applied this technique to the 4 reservoir lower Caroni hydropower system in Venzuela. Disaggregation was performed both spatially and temporally, resulting in daily operational policies from the monthly equivalent reservoir policies. Saad et al. (1994) incorporated neural networks to improve the disaggregation process and to account for nonlinear dependencies between the system elements. The method was successfully applied to finding long-term operational policies for Hydro-Quebec's 5 reservoir hydropower system on the La Grande River. Labadie (1997) indicated that the main problem with the use of state aggregation/decomposition methods is the loss of information that occurs during the aggregation process.

c) The Principle of Progressive Optimality

Turgeon (1981) presented an algorithm based on the principle of progressive optimality of Howson and Sancho (1975), for which the state variables do not have to be discretized. He applied the technique to an example consisting of four hydropower plants in series to determine the optimal short time scheduling for multireservoir system consisting of 4 hydropower plants. The algorithm does not have the recursive equation in terms of the optimal value function and might be considered as a multidimensional continuous version of the IDP procedure.

This approach has the advantage of dealing with discontinuous return functions and with hydropower production functions that do not have to be linearized or approximated by convex linear functions. Also, there is no problem of dimensionality since only one trajectory of the reservoir storage must be stored in the computer memory. As this iterative procedure is a function of the selected initial solution, Turgeon (1981) proposed the use of DPSA with a very coarse grid of the state variables to obtain a good trial trajectory before using this approach, which can then be solved by a direct search method.

d) SDP with Function Approximation

The discretized “cost-to-go” function can be approximated by a continuous function. Since an approximate value for the cost-to-go function is only calculated at the discretized state values, the value of the function at other points can be estimated by interpolating nearby grid points. Several authors explored the reduction in computational effort possible when multivariate polynomials or piecewise polynomial functions are employed in SDP algorithms (Foufoula-Georgiou and Kitanidis (1988); Johnson et al. (1989); Johnson, et al 1993; Tejada-Guibert et al (1993), and Lamond (2003). Tejada-Guibert et al. (1993) concluded that computational savings are possible; mainly because: (1) the improved accuracy of higher order functions which results in good solutions even with a coarse state space discretization and (2) efficient gradient-based optimization algorithms can be used to compute better approximations to the optimal solutions.

Johnson et al. (1993) applied a high order spline function to approximate the cost-to go function so that a coarse discretization of the state space could be used. The spline is constructed of individual multivariate cubic polynomials, each defined over a sub-region of the state space domain. The spline coefficients were determined by requiring that the spline be able to interpolate the cost function values at each state space grid point. This approach proved to be successful in reducing the solution time for a system with two to five reservoirs. Tejada-Guibert et al. (1993 and 1995) applied these piecewise cubic functions to approximate the cost-to-go function for the five hydropower plants of the Shasta-Trinity system in North California. He also recommended the use of a sampling SDP algorithm suggested by Kelman et al. (1990) as an attractive approach to describing the distributions and temporal correlations of inflows.

Lamond (2003) applied a piecewise polynomial approximation of the future value function for a single hydroelectric reservoir model. The authors concluded that the adopted method is faster than both discrete DP value iteration and a continuous DP method using splines on a fixed grid. Also, they suggested that spline approximation is not well suited when the rewards are piecewise linear.

Lamond and Boukhtouta (2005) applied the neuro-dynamic programming approach (NDP) of Bertsekas and Tsitsiklis (1996) to approximate the cost-to-go function by a neural network. They applied the NDP to compute an approximate optimal policy for the control of a single hydroelectric reservoir with random inflows, concave, piecewise linear revenues from electricity sales taking into account the head variations and the turbine efficiency. Their NDP approach is based on a backward induction of a feed forward neural network with an input layer, hidden layer and a single output layer to approximate the future value function.

Lamond and Boukhtouta (2005) concluded that the NDP approximation architecture gives very smooth approximate functions, which allowed the use of a coarse discretization of the state and the inflow variables in the training step of the neural functions. Their findings reinforce and confirm Bertsekas (2001) claims that NDP can be impressively effective in problems where the traditional DP methods would be hardly applicable.

2.2.2.2. Stochastic Linear Programming

Stochastic linear programming (SLP) deals with uncertainty in the model parameters by considering a number of scenarios. Each scenario describes the values of the uncertain parameters and their probability of occurrence. The primary advantage of scenario-based stochastic models is the flexibility it offers in modeling the decision process and in defining scenarios, particularly when the state dimension is high. However, the difficulty with this modeling approach is that an ample number of scenarios result in a large scale linear programming problem, which in turn requires special solution algorithms that rely mainly on decomposition approaches.

Stochastic linear programming with recourse (SLPR) utilizes scenarios to represent the uncertainty in model parameters in the form of stages. The SLPR in its simple form subdivides the problem into two stages. The first stage decisions are proactive or planning decisions, which are made with certainty, while the second stage decisions are reactive or operating decisions. Accordingly, SLPR models support the "here and now decision", while providing a number of "wait and see" strategies depending on which scenario unfolds. These models are non-anticipative: in each stage, decisions must be made without knowledge of the realization of random variables in future stages.

When each inflow scenario is treated deterministically, the deterministic variables represent the set of first stage decisions and the stochastic variables represent future release decisions corresponding to a specific scenario. It should be noted that only the first stage decisions are actually implemented, since the future decisions are not known with certainty. Following the implementation of the first stage decisions, the problem is then reformulated starting with the next period decisions, and solved over the planning horizon.

The first applications of two-stage and multi-stage SLP to reservoir management (Pereira and Pinto, 1985, 1991) used the Benders Decomposition Method (Benders, 1962: Van Slyke and Wets, 1969). This method is powerful because it allows a large-scale problem to be solved iteratively. Moreover, using this technique in a nested form allows multi-stage problems to be decomposed by both scenario and decision period (Birge, 1985).

Jacobs et al. (1995) applied SLP using Benders decomposition to a three reservoir hydropower system in Northern California. Decomposition of the linear programming problem into smaller network flow optimization problems resulted in significant computational savings over attempts at direct solution.

Dantzig and Infanger (1997) combined Benders decomposition (Benders, 1962) with the importance sampling technique to reduce the variance of the sample estimates. The dual of the multistage formulation measures the impact of future responses, which is fed back to the model's present time in the form of cuts. These cuts are sequentially added at different stages of the multi-stage dynamic system. Dantzig and Infanger (1997) indicated that these cuts constitute a set of generated rules that guide the control problem to balance between present and future costs and drive the system away from future infeasibilities and towards optimality. Kracman et al. (2006) developed a multistage multiobjective SLP reservoir planning model for the Highland Lakes system in Texas applying generated scenarios using a quantile sampling technique. The authors state that this scenario generation technique, which was adopted, preserves the spatial correlation of the random inflows.

2.2.2.3. Stochastic Dual Dynamic Programming

Stochastic Dual Dynamic Programming (SDDP) developed by Pereira (1989) represents an interesting mix of stochastic linear and dynamic programming optimization techniques. SDDP solves a multidimensional stochastic dynamic programming problem and it approximates the future cost function (FCF) as a piecewise linear function. Unlike conventional SDP, which discretizes the state space and solves the FCF for all points, SDDP samples the state space and solves the DP problem iteratively. The SDDP approach, as presented by Pereira et al. (1999), is described in the following paragraphs.

The first phase starts with a backward recursion calculation of the FCF. The slope of the FCF around a given state is calculated by solving a series of one stage LPs for each inflow scenario. The slopes of the FCF at the different states are estimated from the dual variable of the mass balance constraint, as these multipliers represent the change in the objective function value with respect to storage $(\partial f / \partial S)$. The resulting cost-to-go, which is based on the highest value in each state (convex hull), represent a lower bound for the actual FCF. In the second phase a Monte-Carlo simulation is performed in a forward pass which simulates the system operation using the release policy calculated so far. Similar to the backward recursion calculations, a set of one stage LP problems has to be solved for each inflow scenario. The upper bound of the FCF is estimated as the mean of the Monte Carlo simulation results. To address the uncertainty around the true expected value of the cost function, Pereira et al. (1999), used the 95% confidence intervals to estimate the confidence limits around the true values.

The Optimal solution is obtained if the lower bound of the FCF lies inside the confidence limits. If not, a new iteration with backward and forward calculations has to be performed adding additional sets of states (additional cuts or plans to the FCF). The states that the simulation passes through are used in the new backward recursion.

It should be noted that the planes obtained in each iteration are retained and the new planes are added to the set generated so far. This is in contrast to the conventional SDP where a new cost-to-go table has to be developed in each stage.

Pereira (1989), and Pereira and Pinto (1991) applied the SDDP to a hydrothermal scheduling problem in Brazil. Rotting and Gjelsvik (1992) applied the SDDP to seasonal planning for system of 35 reservoirs in a 28 river systems, which represents a part of the Norwegian hydropower system. The system is operated to minimize the thermal operating costs while considering the terminal value of water storage. They concluded that the SDDP procedure is successful and convergence of the algorithm is obtained with a saving in run time over the basic SDP approach by a factor of 16. Halliburton (1997) however, states that: "convergence is questionable for both the U.S. Bonneville Power Administration (BPA) and New Zealand systems". He summarized the difficulties with SDDP as: non convergence, long CPU time, difficulties in setting the large number of interacting penalties, and the inability to handle certain type of constraints (non convex, applying across a number of time periods, etc...).

2.2.2.4. Chance-Constrained Programming and Reliability Programming

Chance-Constrained Programming (CCP) considers the probability conditions on constraints. Typically, the probability of satisfying a constraint is required to be greater than a threshold value. These constraints have the impact of tightening the restrictions on reservoir releases at the desired risk levels, thereby encouraging more conservative operational strategies. CCP converts a stochastic type problem to a deterministic-type one, and then solves the deterministic equivalent.

Loucks and Dorfman (1975) concluded that the use of chance constraints leads to overly conservative rules for reservoir operation. Takeuchi (1986) invoked a CCP model to solve a real-time reservoir operation problem. The chance-constraints were set on the probability of the reservoir becoming empty. Changchit et al. (1989) combined CCP with goal programming to operate a multiple reservoir system.

Yeh (1985) concluded that CCP formulations neither explicitly penalize the constraint violation nor provide recourse action to correct constraint violations as a penalty. Hogan et al. (1981) warned that the practical usefulness of CCP is seriously limited and it should not be regarded as substitution for stochastic programming. Labadie (1997) indicated that the CCP does not represent the true risk that must be estimated by performing Monte Carlo analyses on the proposed operational policies.

Colorni and Fronza (1976) initiated the application of reliability programming (RP) to the reservoir management problem that was regarded as an extension to the CCP. In their model, risk was accounted for by choosing different probability values that constrain the degree of satisfying the contracted release. Reznicek and Cheng (1991) expressed the probability of the constraints as decision variables and were therefore incorporated in the objective function.

2.2.3. Heuristic Models

Heuristics methods are criteria, methods, or principles for deciding that among several alternative courses of action are the most effective in achieving certain goals (Pearl, 1984). Heuristic algorithms cannot guarantee global optimum solutions, however they are well-suited to problems that are difficult to formulate and solve by applying algorithmic methods (e.g. non-linear-nonconvex functions). Genetic algorithms (GA) and artificial neural networks are the most commonly used heuristic methods for the reservoir operations planning problem.

Recently, Ant Colony Optimization (ACO) algorithms, which are evolutionary methods based on the foraging behavior of ants, have been successfully applied to a number of benchmark combinatorial optimization problems (Dorigo et al., 2000). ACO was inspired by the behavior of ants in finding the shortest route between their nest and a food source. Jalai et al. (2006) applied ant ACO to the Dez reservoir in Iran for a finite horizon and a single time series of inflows. The authors concluded that the ACO algorithm provided improved release policies as compared with another GA model. The same conclusion of ACO outperforming the GA algorithm was found by Kumar and Reddy (2006) who applied ACO to a multipurpose reservoir in India.

2.2.3.1. Genetic Algorithm

Genetic algorithm (GA) is a powerful population oriented search method based on the principle of Darwinian natural selection and survival of the fittest. GA performs optimization through a process analogous to "the mechanics of natural selection and natural genetics" (Goldberg, 1989).

Genetic algorithms deal with a population of individual candidate solutions (strings/chromosomes), which undergo changes by means of genetic operations of reproduction through selection, crossover, and mutation operations. These solutions are ranked according to their fitness with respect to the objective function. Based on their fitness values, individuals (parents) are selected for reproduction of the next generation by exchanging genetic information to form children (crossover). The parents are removed and replaced in the new population by the children to keep a stable population size. The result is a new population (offspring) with normally better fitness. After a number of generations, the population is expected to evolve artificially, and the (near) optimal solution will be reached. The global optimum solution however cannot be guaranteed since the convexity of the objective function can't be proven. The GA adjusts populations of release rule structures based on values of the fitness (objective) function values according to the hydrologic simulation model results.

Wardlaw and Sherif (1999) successfully applied GA to a four-reservoir system in which a global optimum was achieved. The authors concluded in their evaluative study that GA provides robust and acceptable solutions and could be satisfactorily used in real-time operations with stochastically generated inflows. Haung et al (2002) applied a genetic algorithm based-stochastic dynamic programming to cope with the dimensionality problem in a parallel multireservoir system in northern Taiwan to derive a joint long-term operation policy. Haung et al (2002) concluded that although the use of GA-based SDP may be time consuming as it proceeds from generation to generation, the model could overcome the "dimensionality curse" in searching solutions. Reis et al. (2005) proposed a hybrid genetic algorithm and linear programming approach for multireservoir operation planning. Their model handled the stochastic future inflows by a three stage tree of synthetically generated inflows. They applied their approach to a hypothetical hydrothermal four reservoir system and compared the results with a SDDP model. The authors concluded that the hybrid scheme offers some computational advantages over the SDDP model. However, it is computationally more time consuming.

2.2.3.2. Artificial Neural Networks

An artificial neural network (ANN) is a model inspired by the structure of the brain that is well suited to complicated tasks such as pattern recognition, data compression and optimization. In neural network terminology, a formal neuron simulates the behavior of the biological neuron whose dendrites collect the energy from its input signals and whose axon transmits a signal to other neurons. In the formal neuron, the energy from the dendrites is presented by a weighted sum of the input variables, and the axon transmission is represented by applying a transfer function to the weighted sum of inputs. The training of the ANN is usually performed using a gradient-type back propagation procedure, which determines the values of the weights on all interconnections that best explain the input-output relationship.

ANN has been used within SDP models to approximate the "cost-to-go" function with fewer sampling points. Saad et al. (1994) applied an ANN to the long-term stochastic operation of the hydroelectric multireservoir system of Quebec's La Grande River. The neural network was trained to disaggregate the storage level of each reservoir of the system for an aggregated storage levels for the system. The inputs to the network are the aggregated storage levels determined by SDP for the aggregated reservoirs. The neural network is trained by applying a large number of equally likely stream flow sequences. Saad et al. (1994) concluded that in comparison with the principal components approach, ANN is more efficient.

Raman and Chandramouli (1996) used ANN to obtain optimal release rules based on initial storage, inflows, and demands for Aliyar reservoir in Tamil Nadu, India. The ANN was trained by applying the results of a deterministic DP model. Raman and Chandramouli (1996) concluded that simulation of operation with rules obtained by the trained ANN outperformed those produced by linear regression analysis, as well as optimal feedback rules obtained from explicit stochastic optimization using SDP.

2.2.3.3. Fuzzy Programming

Several researchers have used fuzzy set theory and fuzzy logic to deal with uncertainties associated with the reservoir operation problem. Fuzzy set theory is generally used to describe imprecision and vagueness. In fuzzy logic, variables are partly represented by several categories and the degree of belongingness to a set or category can be described numerically by a membership number between 0 and 1.0. Russell and Campbell (1996) used fuzzy logic to derive operating rules for a hydroelectric plant, where the inflow and price of energy can vary. Tilmant et al. (2001) developed a fuzzy SDP approach with fuzzy objectives and fuzzy intersections between immediate and future release decisions consequences. Mousavi et al (2004) developed a technique called fuzzy-state stochastic dynamic programming (FSSDP) for reservoir operation that considers the uncertainties in the hydrologic variables and the imprecision due to variable discretization as fuzzy variables. The transition probabilities are considered by defining a fuzzy Markov chains.

2.3. Sampling Techniques

In the applications of stochastic programming models for the reservoir optimization problem we are usually faced with the problem of how to represent the random variables (inflow, demand, prices). The problem becomes rather complex with multivariate random vectors, particularly if these vectors are correlated. Generation of data trajectories or scenarios represents the most challenging and time consuming part of the solution of stochastic optimization problems. The objective is to generate trajectories or scenarios that best approximate the given distribution of the random variables in a computationally manageable way in the optimization model. Different sampling-based approaches have been proposed to handle the problem of generating scenarios. A number of these methods have been presented by Kaut and Wallace (2003). The following is a brief overview of the generation of data trajectories and sampling methods.

2.3.1. Time Series Models

Time series models are intended to replicate the spatial and temporal structure of the random variables. Examples of time series models include: Autoregressive models, Moving Average Models, and Bayesian Vector Autoregression model (VAR). Many of the reported applications of SDP in reservoir management models use lag-1 autoregressive or multivariate model. The use of time series to generate data trajectories involves selecting a model and estimating its parameters. These two steps add to the uncertainty of the analysis. Vogel and Stedinger (1988) have documented that errors arising from parameter estimation often overwhelm issues of model choice.

2.3.2. Conditional Sampling

Because of its simplicity, conditional sampling is the most popular method for generating scenario trees in stochastic programming. It is based on approximating probability measures by empirical ones generated by random samples. Because of computational restrictions, these samples cannot be very large, so the empirical measures can be poor approximations of the original ones. Pennanen and Koivu (2002) show that modern integration quadratures provide a simple and attractive alternative to random sampling. These quadratures are designed to give good approximations of given probability measures by a small number of quadrature points. Loretan (1997) applied principal component analysis to reduce the dimensionality of the scenario tree. Sampling from principal components, allows correlated random vectors to be obtained.

2.3.3. Sampling from Specified Marginals and Correlations

Many techniques are available to generate random samples from univariate distributions (Devroye, 1986). These techniques are not applicable in sampling multivariate vectors particularly if they are correlated. In the case of multivariate distributions, some algorithms were developed assuming the correlation matrix and marginal distributions (beta, lognormal, Pearson, etc...) are fully specified (e.g. Cario and Nelson, 1997). Other algorithms sample correlated random variables applying partially specified multivariate distributions (e.g. Luri and Goldberg 1998). However the user specifies the marginal moments. The various algorithms also differ in the degree to which dependencies among variables are specified. Most algorithms require only the correlation matrix, but a few require higher order product moments. Parish (1990) presented a method for generating random variables from multivariate Pearson distribution, with the knowledge of all product moments to the fourth order.

2.3.4. Moment Matching

This method relies on describing the marginal distributions by their moments (mean, variance, skewness, kurtosis, etc.) as well as a correlation matrix, and possibly other statistical properties. Hoyland and Wallace (2001) developed a scenario generation algorithm, which constructs multi-dimensional scenario trees with specified moments and correlations, by solving a single, very large, least squares problem. To improve the speed of the solution procedure, Hoyland et al. (2003) introduced a new algorithm that speeds up the procedure by decomposing the least squares problem into *n* univariate random variables, each satisfying a specification for the first four moments. Then, the different marginal distributions were combined so that the joint distribution satisfies the specified correlations and moments by applying a Cholesky decomposition and a cubic transformation in an iterative procedure. Lurie and Goldberg (1998) applied a similar multivariate decomposition approach but starting with parametric marginal distributions instead of the marginal moments.

Although Hoyland et al. (2003) could not guarantee convergence to their proposed procedure, they concluded that their experience shows that it would converge if the moment's specifications were possible and there were enough scenarios. They also stated that a potential divergence or convergence to the wrong solution is easy to detect. Accordingly, there is no risk of ending up using the incorrect tree in the optimization procedure. In terms of computer time, they found trees with 1000 scenarios representing 20 random variables took less than one minute.

2.3.5. Path Based Methods

These methods start by generating several data paths (or fans), which can be done through the use of parametric or nonparametric methods as suggested by Dupacova et al. (2000). In many application areas there exist advanced continuous and discrete time stochastic models and historical time series that serve to calibrate these models. A global scenario generation can be achieved with the calibrated model, by simulating many sample paths. These models employ a specified type of probability distributions. Nonparametric methods can be applied to large families of probability distributions, which cannot be indexed by a finite dimensional parameter (distribution free methods). The next step is to delineate the initial structure of the scenario tree, i.e. the number of stages and the branching scheme. The additional step to build the scenario tree includes applying ad hoc methods, by cutting and pasting the data paths in an intuitive way. The other possibility, as proposed by Birge and Mulvey (1996), is to apply cluster analysis in a multi-level clustering or bucketing scheme that exploits the whole sequences of observed/simulated data.

2.3.6. Optimal Discretization

Pflug (2001) developed a method for constructing a scenario tree with optimal discretization on the basis of a simulation model of the underlying stochastic process by using a stochastic approximation technique. This method is different from other methods described earlier in that it constructs the whole scenario tree at one time. However, the method deals only with univariate processes.

2.3.7. Scenario Reduction

This method involves developing a much smaller number of scenarios, and it determines a scenario subset of prescribed cardinality or accuracy and a probability measure based on this set that is the closest to the initial distribution in terms of a natural probability metric. All deleted scenarios have probability zero. Romisch and Heitsch (2003) presented two new algorithms for computing optimally reduced probability measures approximately. One advantage of the reduction concept is its generality. No requirements are imposed on the stochastic data processes (e.g. the dependency or correlation structure of the scenarios, the scenario probabilities or the dimension of the process).

2.3.8. Interior Sampling Methods

Interior sampling is an another class of sampling methods in which several samples are used at different steps of a particular optimization procedure, for example to estimate function values, gradients, optimality cuts, or bounds, corresponding to the second-stage expected value function. Higle and Sen (1991) suggested stochastic decomposition methods. Infanger (1994) applied importance sampling that generates samples within the L-shaped algorithm for stochastic linear programming. Importance sampling is typically presented as a method for reducing the variance of the expected estimate of a stochastic variable by carefully choosing a sampling distribution.

2.4. RL Approach

Conventional optimal control methods, dynamic programming for instance, suffer from the 'curse of dimensionality', wherein the large dimensionality of the system at hand and the exponential growth of its possible states prohibit the attainment of an optimal solution even using the fastest computers available today, and most likely in the future. The literature survey conducted has revealed that this area of research is still very active, as new solution techniques are being investigated and developed.

One possible angle from which the problem can be tackled is through the use of machine learning techniques from the field of artificial intelligence (AI), particularly Reinforcement Learning (RL). RL has two key advantages over conventional control methods: the potential for learning how to control a larger system in a shorter time, and the ability to do so with or without a formal model of the system. Reinforcement learning (RL) has adapted key ideas from various disciplines namely: machine learning, operations research, control theory, psychology, and neuroscience to produce some very successful engineering applications (Sutton and Barto 1998).

RL overcomes the curse of dimensionality through the use of function approximation, which allows RL to use much larger state spaces than classical sequential optimization techniques such as dynamic programming. In addition, using sampling, RL can be applied to large-scale problems where it is too complicated to explicitly evaluate and enumerate all the state transition probabilities. Modern reinforcement learning could be applied to both trial and error learning without a formal model of the environment, and to planning activities with a formal model of the environment, where an estimate of the state-transition probabilities and immediate expected rewards could easily be evaluated.

Sutton and Barto (1998), Bertsekas and Tsitsiklis (1996) state that: "RL has become popular as an approach to artificial intelligence because of its simple algorithms and mathematical foundations and also because of a series of successful applications". Sutton (1999) concluded that this approach has already proved to be very effective in many applications as it has produced the best of all known methods for playing backgammon (Tesauro, 1995), dispatching elevators (Crites at al. 1996), job-shop scheduling (Zhang W. and Dietterich 1996), and assigning cellular-radio channels (Singh and Bertsekas 1996).

Ernst et al. (2003) applied RL for power systems stability control. Abdulhai et al. (2003) applied RL for true adaptive traffic signal control. In the water resources sector the application of this approach has been very limited. Wilson (1996) applied the RL technique in the real-time optimal control of hydraulic networks. Bhattacharya et al. (2003) successfully applied the RL technique in real time control (RTC) to Delfland water system in the Netherlands, which includes Delft, Hague, and part of Rotterdam covering an area of about 367 km^2 and consisting of about 60 polders with 12 pumping stations. Bhattacharya et al. (2003) concluded that in all applications involving some sort of control functions (urban drainage systems, polder water level maintenance, and reservoir operation), RL has substantial potential.

RL is a machine learning approach that can be used to derive an optimal control strategy. RL concerns the problem of a learning agent interacting with its environment to achieve a goal (Sutton, 1999). The agent continuously maps situations to actions so as to maximize a reward signal. The learner is not told what to do, as in most forms of machine learning techniques, but instead must discover which actions yield the most rewards by trying them (Sutton, 1999). These two characteristics, trial and error search and delayed reward, are the two most important distinguishing features of reinforcement learning.

2.5. Conclusions

The literature review carried out shows that this area of research is still very active and that different optimization approaches and modeling techniques are being tried for dealing with the reservoir systems optimization problem. The review shows that employing an explicit stochastic optimization approach would be the most advantageous since it provides the best representation of this complex problem. The main obstacle that needs to be addressed and resolved, however, is the high dimensionality of the problem.

From this literature review, it can be concluded that DP algorithms remains a very powerful technique for handling the nonlinear, stochastic large-scale reservoir optimization problem. Among the numerous efforts attempted to alleviate the curse of dimensionality problem, which is aggravating the large-scale SDP method, function approximation techniques and/or sampling techniques resulted in some successful applications in the multireservoir hydropower generation operations planning problem. One promising approach addressing the possibility of combining these two techniques (function approximation and sampling techniques) within an SDP formulation is the Reinforcement Learning (RL) technique. The following chapter presents the main concepts and computational aspects of RL techniques.

3. THE REINFORCEMENT LEARNING APPROACH

3.1. Introduction

Reinforcement learning (RL) is a computational approach for learning from interactions with an environment and from the consequences of actions to derive optimal control strategies. RL has adapted key ideas from various disciplines namely: machine learning, operations research, control theory, psychology, and neuroscience (Sutton and Barto, 1998). RL has become popular as an approach to artificial intelligence because of its simple algorithms and mathematical foundations (Bertsekas and Tsitsiklis, 1996) and because of a number of successful applications in different domains, e.g. control problems, robot navigation, economics and management, networking, games, etc... (Sutton, 1999).

The successful applications of RL surveyed and the key advantages that RL offers in handling large-scale problems provided the motivation to research the possibility of applying this approach to solve the large-scale problem of operation planning of multireservoir systems.

The following sections of this chapter introduce the main concepts and computational aspects of the RL methods and presents the distinguishing features and elements of the RL approach including the trial and error learning of policies, the concept of delayed rewards, and the exploration and exploitation of policies. The chapter also focuses on the three main classes of methods to solve the RL problem, namely, (1) dynamic programming algorithms and its relation with Markovian decision process (MDP) and the Bellman principle of optimality, (2) Monte Carlo methods, and (3) the Temporal-Difference learning methods. More advanced RL methods that unify the basic ideas of the above three methods are also described; these include the eligibility traces and function approximation, and generalization. More comprehensive reviews of RL can be found in Sutton and Barto (1998), Kaelbling et al. (1996), and Bertsekas and Tsitsiklis (1996).

3.2. Reinforcement Learning Approach versus Dynamic Programming

Dynamic Programming (DP) is a very powerful technique for handling sequential, nonlinear, and stochastic optimization problems. DP guarantees the attainment of optimal solutions to MDPs. However, DP requires that values of the transition probabilities and the transition rewards of the system be calculated. The transition probabilities and the expected immediate rewards are often known as the theoretical model of the system. For large scale systems that involve several stochastic variables, constructing the model of the environment is quit a difficult task. Gosavi (2003) states that: "Evaluating the transition probabilities often involves evaluating multiple integrals that contain the probability density functions (pdfs) of random variables. It is for this reason that DP is said to be plagued by the curse of modeling".

Compared with DP methods, linear programming methods (LP) can also be used to solve MDPs. Sutton and Barto (1998) indicated that LP becomes impractical at a much smaller state space (by a factor of about 100) and concluded that for the largest problems, DP methods are the only feasible and practical solution methods.

For DP problems, assuming a system with *m* state discretization and *n* reservoirs, the computational time and storage requirement is proportional to m^n. Consider the case of a system with hundred state discretization for each of two reservoirs, the number of possible state combinations in one period is $100^2=10^4$. This exponential increase in the state space is often known as the curse of dimensionality (Bellman, 1957). Assuming that there are five possible actions for each state, the transition probability matrix of each action would consist of $5 \times 10^4 \times 10^4 = 5 \times 10^8$ elements. As this simple example shows, it is obvious that for such problems with large state space, storage of the transition matrices will be a difficult task indeed.

Obviously, the two main problems limiting the capabilities of DP are the excessive memory needed to store large tables and the very long computation time required to fill those tables. One possibility to tackle these problems is through the use of machine learning techniques from the field of artificial intelligence (AI), particularly Reinforcement Learning (RL). RL offers two key advantages in handling problems that are too large to be solved by conventional control methods:

1. The ability to solve MDPs with or without the construction of a formal model of the system. By using ***sampling*** (simulation), RL can be applied to large-scale problems that are too complicated to explicitly evaluate and enumerate all the transition probabilities and the expected immediate rewards of the system. This way RL provides a way to avoid the curse of modeling.

2. The potential for learning how to control a larger system. RL can overcome the curse of dimensionality through the use of ***function approximation*** methods. For small scale problems, RL stores the elements of the value function in lookup tables called *Q*-Tables (tabular methods). However, as the state space increases, RL can use function approximation methods, which require the use of a limited number of parameters to approximate the value function of a large number of states. The following figure highlights the difference in the methodology between the DP and RL.

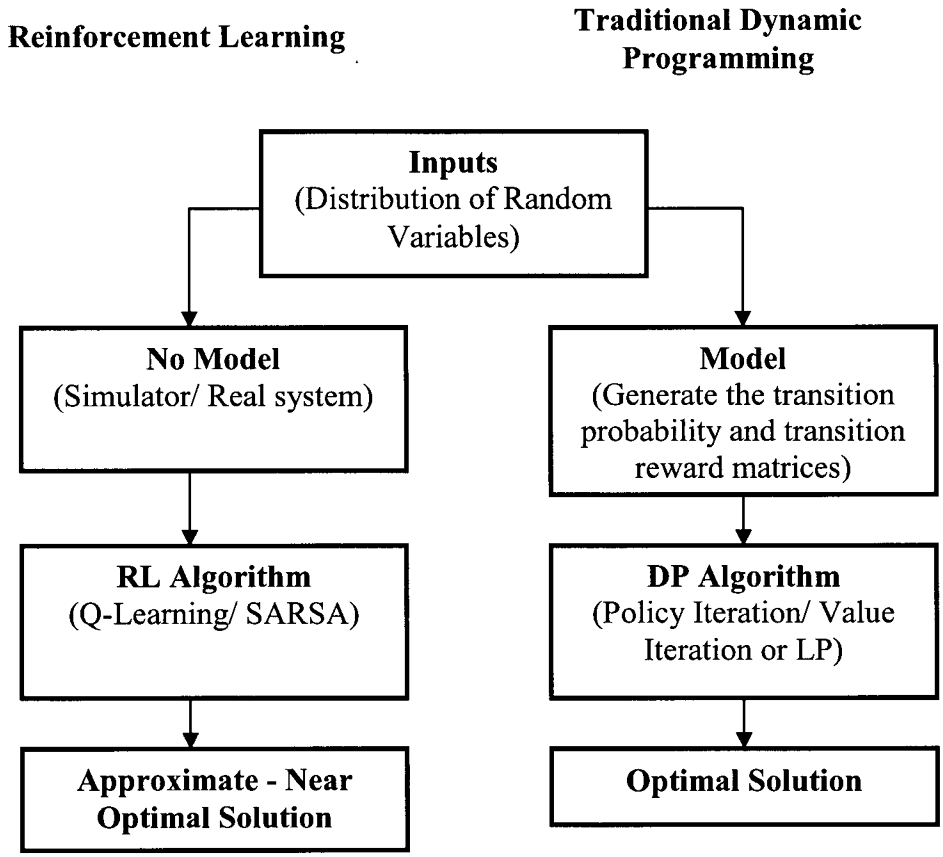

Reinforcement Learning
Traditional Dynamic Programming
Inputs
(Distribution of Random Variables)
No Model
(Simulator/ Real system)
Model
(Generate the transition probability and transition reward matrices)
RL Algorithm
(Q-Learning/ SARSA)
DP Algorithm
(Policy Iteration/ Value Iteration or LP)
Approximate - Near Optimal Solution
Optimal Solution

Figure 3.1. Reinforcement Learning and Dynamic Programming comparison (adopted from Gosavi, 2003)

3.3. Reinforcement Learning Problem Formulation

RL concerns the problem of a learning agent that relies on experience gained from interacting with its environment to improve performance of a system over time. The learner (or the decision maker) is called the agent and the object it interacts with is called the environment. The environment could be a simulator of the system or the real system. Both the agent and the environment constitute a dynamic system.

Unlike supervised learning techniques, which require examples of input-output pairs of the desired response to be provided explicitly by the teacher, the learning agent is not told what to do. Instead, the agent must continuously learn from the consequences of its actions. The agent perceives how well its actions perform by receiving a reward signal. This reward signal indicates that the agent should do or how to modify its behavior without specifying how to do it. The agent uses this signal to determine a policy that leads to achieving a long term objective. This trial and error interaction with the environment and the delayed rewards are the two main distinguishing features of the reinforcement learning method.

3.3.1. RL Basic Elements

The main components of a RL algorithm are: an agent, an environment and a reward function. The interaction between the agent and its environment can be modeled within the framework of Markov Decision processes. The agent and the environment interact in a sequence of discrete time steps, $t = 0, 1, 2, 3, \ldots$. At each step the agent receives some indication of the current state of the environment, $s_t \in S$, where S is the set of all states and then it selects an action $a_t \epsilon A$, where A is a finite set containing all possible actions. The agent interacts with the environment and receives feedback in the form of a stochastic transition to a new state s_{t+1} and receives a numerical reward $r(s_t, a_t)$ as defined by the reward function. Through this delayed reward and guided search process, the agent learns to take appropriate actions that maximize the cumulative rewards over time (Sutton, 1999). A schematic representation of the agent-environment interaction is presented in Figure 3.2.

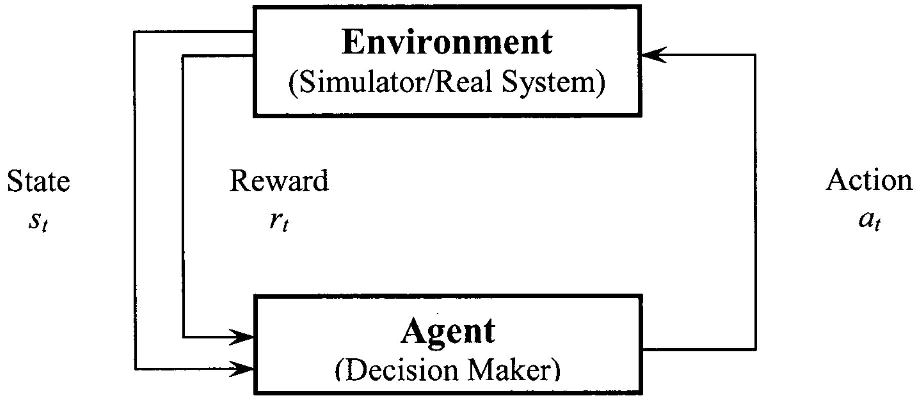


Figure 3.2. Agent-Environment interaction in RL

3.3.2. Exploration/ Exploitation

One of the key aspects of reinforcement learning is that the learner needs to explicitly explore its environment in its search for good rewards. The feedback that the agent receives from its environment indicates how good the action was, but it does not indicate whether it was the best or the worst action possible (Sutton and Barto, 1998). Accordingly, two conflicting objectives arise during the action selection process. One objective is to achieve high-valued short-term rewards by selecting actions that are already known to be good (exploit). On the other hand, the agent has to explore new actions for better action selections in the future.

Two popular methods for balancing the exploration and exploitation in RL are the ε-greedy and softmax action selection rules. In the ε-greedy method, the learner behaves in a greedy way most of the time (by selecting the highest estimated action value), while a non-greedy exploratory action will be taken every now and then with a small probability ε (by selecting a random action uniformly regardless of its value). The advantage of the ε-greedy method is that, in the limit and as the number of trials increases, every action is sampled an infinite number of times. The probability of selecting the optimal action converges to greater than 1-ε (Sutton and Barto, 1998). The disadvantage of the ε-greedy method, however, is that in the exploration process it chooses equally among all actions, regardless of the estimated value of the chosen action. The result is that the learner could choose equally between the worst action and the second best action in the exploration process.

In the softmax action selection method, the action selection probabilities are ranked according to their estimated values. The greedy action is then selected as the one with the highest action selection probability, and all other actions are then weighted proportionally to their estimated action values. Frequently, the softmax method uses a Gibb or Boltzmann distribution to choose the probability of an action *a, P*(*a*):

$$p(a) = \frac{e^{Q_t(a)/\tau}}{\sum_{b=1}^{n} e^{Q_t(b)/\tau}} \tag{3.1}$$

where $Q(a)$ and $Q(b)$ are the estimated action values and τ is a temperature parameter that controls the probability distribution. Higher temperatures result in actions with equal probability of selection (exploration). On the contrary, low temperatures cause a greater difference in the probability of selecting actions according to their estimated values (exploitation). Gradually, the temperature τ decreases over time to limit the exploration process.

3.3.3. Return Functions

One can distinguish two main types of RL tasks: episodic and continuous tasks. In episodic tasks, the horizon represents a finite number of steps in the future. There is a terminal state where the episode ends. On the other hand, in continuous tasks, the horizon represents an infinite sequence of interactions between the environment and the agent. The goal of the agent is to maximize the accumulated future rewards, and the return function R_t is a long term measure of such rewards. In the case of finite horizon tasks, the return is the sum of the rewards from the beginning to the end of the episode:

$$R_t = r_{t+1} + r_{t+2} + r_{t+3} + \dots + r_T \tag{3.2}$$

where T is the number of stages in the episode, R_t is the reward received after t time steps. For continuous tasks, the infinite horizon discounted model takes the long-term rewards into account by discounting the rewards received in the future by discount factor γ, where $0 \leq \gamma \leq 1$. The return function then becomes:

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{3.3}$$

where k is the number of time steps in the future. The discount rate determines the present value of future rewards. If $\gamma = 0$, the agent is said to be myopic and only considers immediate rewards. As γ increases, the returns increase and the agent gives more consideration to future rewards. In the mathematical formulation of the reservoir operation planning model presented in chapters 4 and 5, it is assumed that the current period rewards is realized at the end of each time period, accordingly the discount factor $\gamma$is applied to both of the present period and the future rewards. In the following sections the focus will be on discounted rewards as this approach is more appropriate to reservoir operation problems.

3.3.4. Markovian Decision Process (MDP)

RL relies on the assumption that the system dynamics can be modeled as a Markovian decision process (MDP). An environment is said to satisfy the Markovian property if the signal from its state completely captures and summarizes the past outcomes in such a way that all relevant information are retained. In general, the response of the environment at time $t+1$ to an action taken at time t depends on past actions. The system dynamics can be defined by specifying the complete probability distribution:

$$p_{ss'r}^{a} = p\{s_{t+1} = s', r_{t+1} = r \mid s_t, a_t, r_t, s_{t-1}, a_{t-1}, \ldots, r_1, s_0, a_0\} \qquad (3.4)$$

for all s', r and all possible values of past events: s_t, a_t, r_t, ..., r_1,s_0, a_0.

The MDP framework has the following elements: state of the system, actions, transition probabilities, transition rewards, a policy, and a performance metric (return function). MDP involves a sequences of decisions in which each decision affects what opportunities are available later. The Markov property means that the outcome of taking an action to a state depends only on the current state. If the state and action space in a MDP are finite then it is called a finite Markov decision process.

If the state signal has the Markov property, then the environment's response at *t+1* depends only on the state and the action representations at *t*. The environment's dynamics of the MDP can be defined by the ***transition probability***:

$$p^{a}_{ss'} = p\{s_{t+1} = s' | s_t = s, a_t = a\} \tag{3.5}$$

where $p^{a}_{ss'}$, is the probability of moving to state s′ for a given state *s* and action *a*. and the expected value of the immediate reward:

$$R^{a}_{ss'} = E\{r_{t+1} | s_t = s, a_t = a, s_{t+1} = s'\} \tag{3.6}$$

These two quantities, $P^{a}_{ss'}$ and $R^{a}_{ss'}$ completely specify the most important aspects of the dynamics of an MDP process.

3.4. Reinforcement Learning Algorithms

The goal of RL algorithms is either to evaluate the performance of a given policy (***prediction problems***) or to find an optimal policy (***control problems***). In prediction problems, the value function (***state-value function***) for a given policy π is estimated as follows:

$$V^{\pi}(s) = E_{\pi}\{R_t | s_t = s\} = E_{\pi}\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s\} \tag{3.7}$$

where policy π is a mapping from states $s \in S$ to the probability of selecting each possible action. This mapping is called the agent's policy π, where $\pi(s,a)$ is the probability of taking an action *a* when the agent is in state *s*. In control problems, the value function (***action-value function***) for policy π is defined as follows:

$$Q^{\pi}(s,a) = E_{\pi}\{R_t | s_t = s, a_t = a\} = E_{\pi}\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a\} \tag{3.8}$$

where $Q^{\pi}(s,a)$ is defined as the expected return starting from state s, taking action a, and thereafter following policy π. The recursive relationship property of value functions between the value of state s and the value of its possible successor states is:

$$V^{\pi}(s) = E_{\pi}\{R_t \middle| s_t = s\} = \sum_{a} \pi(s,a) \sum_{s'} p^{a}_{ss'} \left[R^{a}_{ss'} + \gamma V^{\pi}(s') \right], \forall s \in S \tag{3.9}$$

Equation (3.9) is the Bellman equation, which states that the value of state s must be equal to the discounted value of the expected next state plus the expected reward along the way. The value function $V^{\pi}(s)$ is the unique solution to the Bellman equation. The policy π is better than the policy π' if $V^{\pi}(s) \geq V^{\pi'}(s)$ for all $s \in S$. The optimal policy which has a better value function than other policies is defined as:

$$V^{*}(s) = \max_{\pi} V^{\pi}(s), \qquad \forall s \in S \tag{3.10}$$

The optimal action-value function $Q^*(s,a)$ in terms of $V^*(s)$ is:

$$Q^*(s,a) = E_{\pi}\{r_{t+1} + \gamma V^*(s') | s_t = s, a_t = a\} = \sum_{s'} p^{a}_{ss'} \left[R^{a}_{ss'} + \gamma V^{*}(s') \right] \tag{3.11}$$

Once we have the optimal value function V^* for each state, then the actions that appear best after a one-step search will be optimal actions. Hence, a one-step-ahead search yields the long-term optimal actions. With Q^*, the agent does not have to do a one-step-ahead search: for any state s, it can simply find any action that maximizes $Q^*(s,a)$. The action-value function effectively memorizes or stores the results of all one-step-ahead searches. It provides the optimal expected long-term return as a value that is locally and immediately available for each state-action pair (Sutton and Barto, 1998).

The following sections describe the fundamental three classes of methods for solving the RL problem. These methods are: dynamic programming (DP), Monte Carlo techniques (MC) and the temporal difference learning methods (TD). Dynamic programming is a planning method, which requires a model of the environment (***Model-based***), whereas the MC and TD methods are learning methods, which can learn solely from basic experience without using a model of the environment (***Model-free***).

3.4.1. Dynamic Programming Methods (DP)

Dynamic programming provides the theoretical foundation of RL algorithms. DP methods are used to solve MDPs, assuming a perfect knowledge of the model of the environment. The key idea of DP methods is the use of value functions and the Bellman equation of optimality recursively to guide the search for optimal policies. DP methods are ***bootstrapping*** methods, as they update one estimate of the value function based on the estimate of a successor states. The two most widely used DP methods for calculating the optimal policies and the value functions are the ***policy iteration*** and the ***value iteration***. The following is a brief overview of these two methods.

The ***policy evaluation*** method refers to the iterative computation of the value functions V^{π} for a given policy π. Initial values of all states are assumed V_0^{π} and successive approximation of the value function is obtained by applying the Bellman equation as a recursive update rule:

$$V_{k+1}^{\pi}(s) \leftarrow \sum_{a} \pi(s,a) \sum_{s'} P_{ss'}^{a} [R_{ss'}^{a} + \gamma V_k^{\pi}(s')], \forall s \in S, a = \pi(s), k = 0,1,2,... \tag{3.12}$$

In practice, and for practical considerations, a stopping criteria for the iterative process is commonly used when the term $\max_{s \in S} |V_{k+1}(s) - V_k(s)|$ is sufficiently small (Sutton and Barto 1998).

The estimated action-values are used as a basis to find a better policy. If the action-value $Q^{\pi}(s,a) > V^{\pi}(s)$ for some $a \neq \pi(s)$, then action a is taken and the policy is changed to π' where $\pi'(s) = a$. This process of taking greedy actions with respect to the current policy is called ***policy improvement***. The new greedy policy π' is given by:

$$\pi'(s) \leftarrow \arg\max_{a} \sum_{s'} P^{a}_{ss'}[R^{a}_{ss'} + \gamma V^{\pi}(s')] \tag{3.13}$$

where $\arg\max_{a}$ denotes the action a at which the value function is maximized. Combining the policy evaluation step with the policy improvement step yields the ***policy iteration*** algorithm. Thus we can obtain a sequence of improved policies and value functions: $\pi_0 \xrightarrow{E} V^{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} V^{\pi_1} \xrightarrow{I} ....\pi^* \xrightarrow{E} V^*$. Where E denotes policy evaluation and I denotes policy improvement.

Another way of solving MDPs is the ***value iteration*** algorithm. Similar to the policy iteration method, the value iteration also combines the policy improvement step and the policy evaluation step. However, in the value iteration algorithm the policy evaluation step is truncated after one sweep over the state space and is followed by a policy improvement step. The value iteration estimates the optimal policy directly as the maximum to be taken over all actions:

$$V^{\pi}_{k+1}(s) \leftarrow \max_{a} \sum_{s'} P^{a}_{ss'}[R^{a}_{ss'} + \gamma V_k(s')], \qquad \forall s \in S \tag{3.14}$$

The policy iteration and value iteration methods converge in the limit to the optimal value function $V^*(s)$ due to the contraction property of the operator (3.14) (Bertsekas and Tsitsiklis, 1996).

3.4.2. Monte Carlo Methods (MC)

As stated earlier, DP methods require that a model of the environment be available, including transition probabilities and the expected immediate rewards. However, in many cases the exact model of the system is not known and in other cases, such as large scale systems, constructing the model could be a difficult task indeed. In such cases, learning the value function and the optimal policies directly from experience could be more efficient. MC methods estimate the value function from the experience of the agent. This experience is gained by sampling sequences of states, actions, and rewards from on-line or simulated interaction with an environment.

MC methods use the mean return of many random samples to estimate the expected value function V^{π} and the action-value function Q^{π}. As more samples are observed, their average return converges to the expected value of the value function. A sample return is the sum of the rewards received starting from state s or a state-action pair (s,a), and that follows policy π until the end of a learning episode. As complete returns can only be obtained at the end of such episodes, MC methods can only be defined for finite horizon tasks.

One can design MC control methods by alternating between policy evaluation and policy improvement for the complete steps of the episode. Observed returns at the end of the episode are used for policy evaluation and then for improving the policy of all visited states in the episode. There is one complication that arises in this method however; the experience gained by interaction with the environment contains samples only for the actions that were only generated by policy π but the values of all other possible actions are not included in the estimate. Those values are needed for comparing alternatives in the policy improvement step. Therefore, maintaining sufficient exploration is a key issue in MC control methods. There are two approaches to assure that the agent is selecting all actions often, namely ***on-policy*** and ***off-policy*** control methods.

In ***on-policy*** control method, the agent uses a soft stochastic policy meaning that $\pi(s,a) > 0$ for all $s \in S$ and all $a \in A$ to evaluate and improve the performance of the same policy. The agent commits to continuous exploration and tries to find the best policy in the process.

The other approach is the ***off-policy*** method: the agent uses one policy to interact with the environment and generates a *behavior policy*. Another policy which is unrelated to the behavior policy is evaluated and improved, and is called the *estimation policy*. An advantage of this approach is that the agent learns a deterministic (e.g., greedy) optimal policy (*estimation policy*) while following an arbitrary stochastic policy (*behavior policy*) thereby, ensuring sufficient exploration in the process.

MC methods differ from DP methods in two main ways (Sutton and Barto 1998): First, they operate on sample experience. Therefore, they can be used for direct learning from interaction with the environment without a model. Second, they do not bootstrap; i.e. they do not build their value estimates for one state on the basis of the estimates of successor states.

3.4.3. Temporal Difference (TD)

Temporal difference (TD) learning methods represent a central and key idea to RL. TD methods are considered as a class of incremental learning procedures specialized where a credit is assigned to the difference between temporally successive predictions (Sutton 1988).

The TD methods combine the ideas of DP and MC methods. Similar to MC methods, the TD approach can learn directly from real or simulated experience without a model of the environment. TD methods share with DP the bootstrapping feature in estimating the value function (Sutton and Barto 1998).

However, TD methods have some advantages over the DP and MC methods. TD methods use sample updates instead of full updates as in DP methods. The agent observes only one successor state while interacting with the environment rather than using values of all possible states and weighing them according to their probability distributions. Accordingly, learning the optimal policy from experience does not require constructing a model of the environment's dynamics.

In addition, unlike MC methods, TD methods do not need to wait until the end of the episode to update their estimate of the value function. The simplest versions of such algorithms are usually referred to as one-step TD or TD(0). TD(0) wait only for the next time step to update their estimates of the value function based on the value function of the observed state transition $V(s_t)$ and the immediate rewards received from the environment r_{t+1}. The following update is performed on each time step:

$$V(s_t) \leftarrow V(s_t) + \alpha_t[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)] \tag{3.15}$$

where α_t is a step-size parameter, $0 \leq \alpha_t \leq 1$ which represents the learning rate. The update rule presented above computes a ***stochastic approximation*** of the V^{π}, which states that:

$$\textit{New Estimate} \leftarrow \textit{Old estimate} + \textit{Step\_size} \,.\, [\textit{Target} - \textit{Old estimate}] \tag{3.16}$$

The target for the TD update is $r_{t+1} + \gamma V(s_{t+1})$. The term [*Target - Old estimate*] represents the error in the estimate or the temporal difference between two successive evaluations of the value function. This error is reduced by taking a step toward the target. The step-size can be defined as *1/n* where *n* is the number of samples generated. This stochastic approximation algorithm which produces a sequence of estimates of $V(s_t)$ such that the error → 0, is based on an old algorithm (Robbins and Monro, 1951).

For any fixed policy π, the TD algorithm is proven to converge to V^{π} in the limit with probability 1 if the step-size decreases according to the ***stochastic approximation*** conditions: $\sum_{t=0}^{\infty} \alpha_t = \infty$ *and* $\sum_{t=0}^{\infty} \alpha_t^2 < \infty$. The first condition guarantees that the steps are large enough to overcome any initial conditions or random fluctuations. The second condition guarantees that eventually the steps become small enough to assure convergence. (Sutton and Barto 1998).

In the case of the control problem, i.e. estimation of the action values $Q(s,a)$, TD methods are used for the evaluation or the prediction part of the policy iteration algorithm. As with the MC methods, sufficient exploration is required to assure convergence which again can be achieved applying either the ***on-policy*** or the ***off-policy*** approaches.

3.4.3.1. SARSA

The name SARSA is attributed to the update rule that is applied in this algorithm which follows the sequence of events: current state (s_t), current action (a_t), resulting rewards (r_{t+1}), next state (s_{t+1}), and next action (a_{t+1}). SARSA is an on-policy TD control algorithm that learns and improves Q^{π} for policy π, which selects and follows its actions. π is ε-greedy regarding the estimated Q so far. The update rule that is performed at each time step is:

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1},a_{t+1}) - Q(s_t,a_t)] \qquad (3.17)$$

This update is done after every transition from a non-terminal state s_t. If s_{t+1} is terminal, then $Q(s_{t+1}, a_{t-1})$ is defined as zero. Similar to TD(0), SARSA converges with probability 1 to an optimal policy and action-value function as long as all state-action pairs are visited for an infinite number of times and the policy converges in the limit to the greedy policy.

3.4.3.2. Q-Learning

Q-learning, first introduced by Watkins (1989), is regarded as one of the breakthroughs in RL. The simplest form of the Q-learning algorithm, which is the one step tabular Q-learning, is based on the temporal difference method TD(0). In this method, the elements of the estimated action-value function are stored in a so called Q-table. The agent uses the experience from each state transition to update one element of the table (Sutton 1999).

The Q-table has an entry, $Q(s,a)$ for each state s and action a. After taking action a_t, the system is allowed to transition from state s_t to the next state s_{t+1}. The immediate reward received as a feedback from the environment is used to update the Q-table for the selected action. The next time step action value estimate $Q(s_{t+1},a_{t+1})$ used in the update is selected according to the ε-greedy policy. This is achieved by selecting the next state-action as the one with the maximum estimated value most of the time and, with a small probability $(1\text{-}\varepsilon)$, a random exploration action is selected.

The following is a procedural list of the *Q*-Learning algorithm:

```
Initialize $Q(s,a)$ arbitrarily (to any feasible values)
Repeat for each episode:
  Initialize $s$
  Repeat for each step in the episode:
      Choose $a$ from s using policy derived from $Q$ (e.g. $\varepsilon$-greedy)
      Take action $a$, observe r, $s'$
  $Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$
  $s \leftarrow s'$
until $s$ is terminal
```

If every state-action pair is visited infinitely often and the learning rate decreased over time, the *Q*-values converges to Q^* with probability 1 (Sutton and Barto, 1998). *Q*-Learning is an off-policy algorithm in the sense that the agent tries to learn the value of the optimal policy while following an arbitrary stochastic policy which is independent of the policy followed by the agent. An example including sample numerical calculations using the *Q*-learning method is presented in Appendix A.

3.4.3.3. Eligibility Traces

Monte Carlo methods perform updates based on the entire sequence of observed rewards until the end of the episode. On the other hand, the one-step TD methods use the immediate reward and the sample next state estimate to perform the update. In between the one step and full episode backup, there are n-step possible backups, based on n-steps of discounted truncated returns R_t^n and the discounted estimated value of the nth next state $\gamma^n V_t(s_{t+n})$. The n-step return is defined as:

$$R_t^n = r_{t+1} + \gamma r_{t+2} + \ldots\ldots + \gamma^{n-1} r_{t+n} + \gamma^{n-1}(V_{t+n}) \tag{3.18}$$

The n-step backups are still TD methods as they still change an estimate based on an earlier estimate:

$$V_{t+1}(s_t) \leftarrow V_t(s_t) + \alpha[R_t^n - V_t(s_t)] \tag{3.19}$$

One step further is to compute the updates of the estimated value function based on several n-step returns. This type of learning is denoted by TD(λ) algorithm, where λ is an eligibility trace parameter (trace-decay parameter), where $0 \leq \lambda \leq 1$. The TD(λ) algorithm averages the n-step backups each weighted proportionally to λ^{n-1}. The backup of this λ-return is defined as:

$$R_t^\lambda = (1-\lambda)\sum_{n=1}^{\infty} \lambda^{n-1} R_t^{(n)} \tag{3.20}$$

It is obvious from the above equation that by setting the $\lambda = 1$, we get the MC updates, whereas by setting $\lambda = 0$ we get the one-step TD(0) updates.

$$V_{t+1}(s_t) \leftarrow V_t(s_t) + \alpha[R_t^\lambda - V_t(s_t)] \tag{3.21}$$

A simpler and more efficient way of implementing the TD(λ) method is the ***backward view TD(λ) learning algorithm***. This algorithm introduces an additional memory variable associated with each state at each time step called the ***eligibility trace*** (e_t). An eligibility trace is a temporary record of the occurrence of an event, such as visiting a state or taking an action. This variable specifies the eligibility of a particular event in updating the value function. At each time step, the eligibility trace for all states decay by $\gamma\lambda$ except for the visited states in that time step which is incremented by 1 as follows:

$$e_t(s) = \begin{cases} \gamma\lambda e_{t-1}(s) & if \quad s \neq s_t \\ \gamma\lambda e_{t-1}(s) + 1 & if \quad s = s_t \end{cases} \tag{3.22}$$

In other words, eligibility traces can be thought of as weighted adjustments to predictions occurring n-steps in the past; more recent predictions make greater weight changes. The TD(0) error or temporal difference at time step t is denoted as δ_t is calculated as follows:

$$\delta_t = r_{t+1} + \gamma V_t(s_{t+1}) - V_t(s_t) \tag{3.23}$$

On every time step, all the visited states are updated according to their eligibility trace:

$$V_{t+1}(s_t) \leftarrow V_t(s_t) + \alpha\delta_t e_t(s) \tag{3.24}$$

Although eligibility traces require more computation than TD(0) methods, they offer significantly faster learning, particularly when rewards are delayed by many steps. TD(λ) is proven to converge under stochastic approximation conditions with probability 1 (Tsitsiklis, 1994). Figure 3.3 presents a schematic comparison of the DP, MC, and TD ebackup methods and the calculation of the value function.

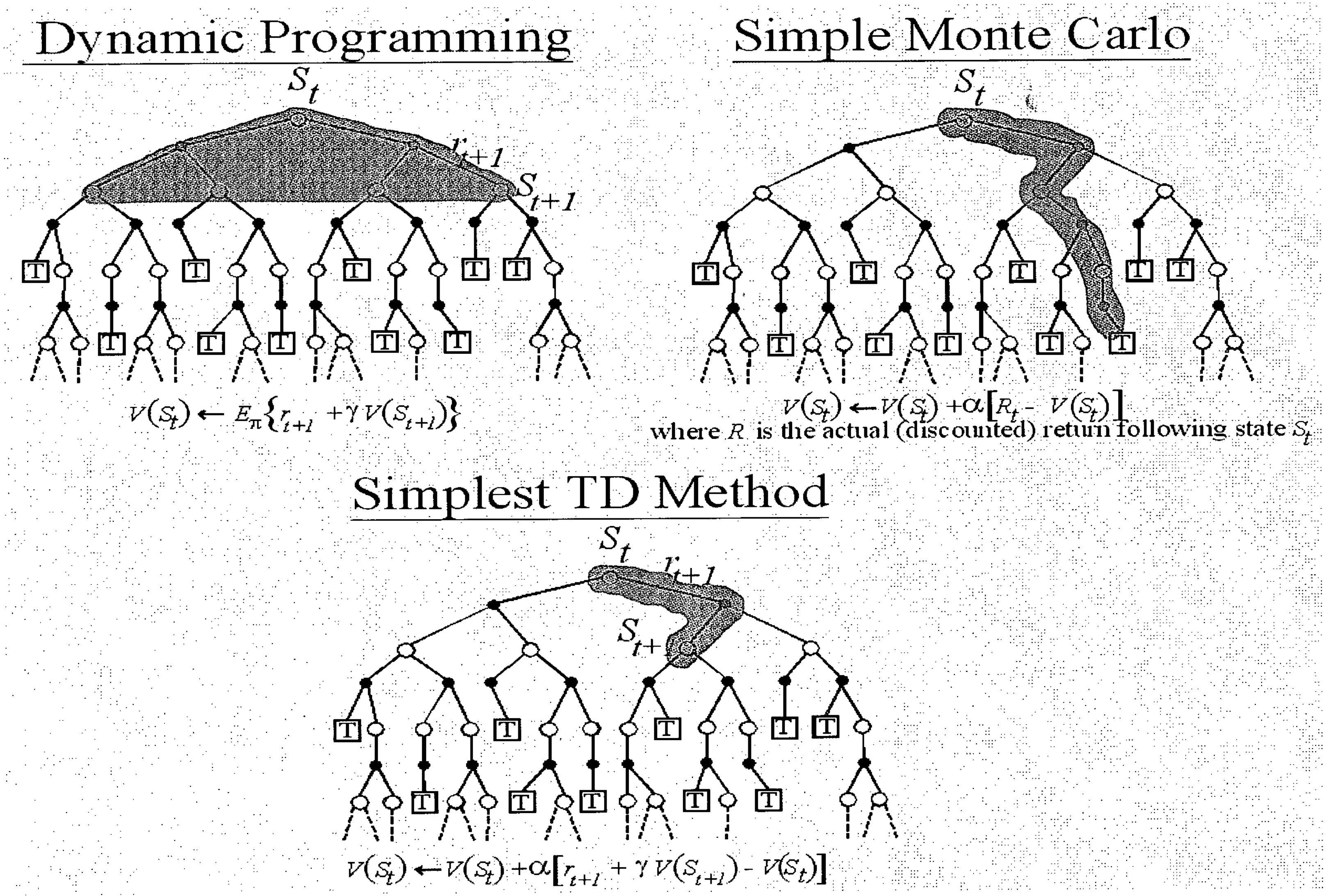

Dynamic Programming
S_t
r_{t+1}
S_{t+1}
T
$V(S_t) \leftarrow E_\pi\{r_{t+1} + \gamma V(S_{t+1})\}$
Simple Monte Carlo
S_t
T
$V(S_t) \leftarrow V(S_t) + \alpha[R_t - V(S_t)]$
where R is the actual (discounted) return following state S_t
Simplest TD Method
S_t
r_{t+1}
S_{t+}
T
$V(S_t) \leftarrow V(S_t) + \alpha[r_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$

Figure 3.3. Comparison of Different RL Approaches (Adopted from Sutton and Barto 1998).

3.5. Function Approximation

When the state space is finite, the most straightforward DP approach is to use a lookup table to store the value function for each state or state-action value combination. In reality, the state space could be quite large or even infinite and could include continuous variables. By using model free RL methods one can avoid the need to construct the transition probability matrix. However, this does not solve the problem completely as the large memory and long time needed to fill in the elements of the lookup-tables still represents a problem. In such cases using look-up tables does not yield practical results.

RL overcomes this problem by applying generalization and function approximation techniques. Here, estimating the Q-values for unvisited state-action pairs require generalization from those states that have already been visited. Function approximation can be done in a number of ways, such as: (1) function fitting (neural networks and regression), (2) function interpolation (K-nearest-neighbors and Kernel methods), and (3) state aggregation (Gosavi 2003). Watkins (1989) used the Cerebeller Model Articulation Controller (CMAC) and Tesauro (1995) used back propagation for learning the value function in backgammon.

As an example, consider applying function fitting techniques for a MDP with A actions in each state, for state $s \in S$:

$$Q(s,a) = f_a(s) \tag{3.25}$$

The idea is to store the Q-factors for a given action as a function of the state index. Assumming s is a scalar; the function $f_a(s)$ can be approximated by:

$$f_a(s) = A + Bs + C.s^2 \tag{3.26}$$

Thus instead of storing each Q-value for action a, we only need to store the values for the set of parameters: A, B, C. The Q-value for state s and action a is represented by the value $f_a(s)$. Obviously, less storage space is needed and a large state space can thus be handled.

3.6. On-Line and Off-Line

RL methods can be implemented in two modes: on-line and off-line. The on-line mode consists of using an RL driven agent directly on the real system. This mode is particularly interesting when it is difficult to model the system or when some phenomena are difficult to reproduce in a simulation environment. Moreover, and as the agent is learning continuously, it can adapt quickly to changing operating conditions. The main drawback of the on-line mode is that the agent may jeopardize system stability because at the beginning of the interaction no experience is available to the RL agent to adequately control the system.

One solution to this problem is to first let the agent interact with a simulation environment (off-line mode). The RL agent then can be implemented on the real system where it would benefit from the experience it has acquired in the simulation environment and will be able to improve its behavior from interaction with the real system. Alternatively, one may extract off-line learned policies and implement them in the real system without any further learning.

On-line and off-line implementation of RL should be differentiated from the on-line and off-line RL algorithms. In on-Line RL algorithms, similar to SARSA, the agent is learning and improving the same policy that it is following in selecting the actions. On the other hand, with an off-policy algorithm such as Q-learning, the agent is gaining useful experience even while exploring actions that may later turn out to be non-optimal.

3.7. Summary

This chapter provided an overview of RL and its main algorithms. The classification of the methods presented is intended to give the reader an idea about the different dimensions of RL. By using sampling and function approximation techniques, RL has the potential to be applied to larger systems than any other classical optimization technique. Modern reinforcement learning methods could be applied to both trial and error learning without a formal model of the environment, and to planning activities with a formal model of the environment, where an estimate of the state-transition probabilities and immediate expected rewards can be easily evaluated.

The introduction of RL into the water resources systems domain is relatively new. However, the advantages that RL offers in dealing with large-scale problems, makes it a promising area of research in that field. A RL based approach is adopted in this research work to develop a stochastic optimization model for the solution of the multireservoir operation planning problem as described in the following chapter.

4. REINFORCEMENT LEARNING MULTIRESERVOIR OPTIMIZATION MODEL (RLROM) DEVELOPMENT

4.1. Introduction

In general, the reservoir operation planning problem for a hydro dominated power system, such as the BC Hydro system, is to find the optimal operation strategy for each time period during the planning horizon that maximizes the long term value of resources while serving the domestic load in British Columbia and thereby to minimize the cost of electricity to the ratepayers in the Province. This objective can be accomplished by coordinating and optimizing the use of all available generation resources while taking advantage of market opportunities.

Releasing more water now could result in high immediate benefits, but with less water left in storage for the future there would be less benefit in the future. On the contrary, releasing less water now could result in gaining more benefits in the future. Accordingly, the decisions taken in any given planning period will affect both the present return and the opportunities that could be available in the future. The challenge then is to link the present decisions with their future consequences. The approach followed in this research work for solving this problem relies on the concept of the marginal value of water (MVW). By definition, the MVW represents the incremental value of water in storage expressed as dollar per cubic meter second-day ($/cms-day).

Optimal dispatch from hydro plants is established when the trade-offs between the present benefits, expressed as revenues from market transactions, and the potential expected long-term value of resources, expressed as the marginal value of water stored in the reservoirs, are equal. Accordingly, as long as the value of releasing water is higher than the value of storing water in the reservoir then the operator's optimal planning decision is to continue generation.

Another implication of applying the water value concept is that hydro plants are dispatched based on their expected water value in storage. Within the BC Hydro system, the multiyear storage capability of the GM Shrum Dam on the Peace River and Mica Dam on the Columbia River projects and their large production capabilities dictate the need for a much higher level of coordination in planning the operation of these two particular basins, than with smaller projects. Ignoring inter-basin and system dependencies on the Peace and Columbia basins could lead to unrealistic operations modeling for the entire system. For this reason, it is important to be able to model the Peace and the Columbia basins and their interaction with the electricity markets in detail to truly reflect an optimal integrated system operation. The two reservoirs operation cannot be optimized separately as the benefits obtained from the operation of one reservoir cannot be directly specified as a function of the storage level in that reservoir alone. Rather, it is a function of both plants (GM Shrum and Mica). The challenge then is to model this large-scale multireservoir system in an integrated manner while addressing the different sources of uncertainty in a way that a large-scale stochastic program can handle.

To solve this large-scale problem, a stochastic optimization model for the operations planning of a multireservoir hydroelectric power generation system is proposed. In this research work, the methodology adopted follows a Reinforcement learning (RL) artificial intelligence approach. The following sections describe the methodology followed in the development of the proposed Reinforcement Learning Reservoir Optimization Model (RLROM). The details of the mathematical formulation and the solution algorithm of the reservoir optimization problem are presented in the following sections.

4.2. Mathematical Formulation

The primary objective of the proposed RLROM model is to develop a set of system operation planning strategies that maximize the overall value of BC Hydro resources at every time period over a planning time horizon given the uncertainty in the forecasted domestic load, the forecasted market prices, and the random natural inflows. The main operation planning decisions consist of: the timing and quantities of import and export, in addition to the timing, location and quantity of water to store or draft from the reservoirs. Another significant outcome of the operation planning model is to establish the MVW in storage for the main multiyear storage reservoirs in the BC Hydro system. The established MVW obtained from the RLROM model can potentially be used for estimating target storage values in the medium term optimization models. Moreover, the calculated marginal values can be used in making tradeoff decisions using the clearing prices for short term wholesale energy market transactions. Marginal values of water are determined from the derivative of the value function with respect to storage ($/cms-d).

The following notation is used in the mathematical formulation of the RL model:

J Number of reservoirs included in the RL Model, where $\{j \in 1,...,J\}$,

Ω Number of scenarios of random variables, where $\{\omega \in 1,...,\Omega\}$,

T final time period in a cycle (iteration or episode),

N Iteration number or the age of the agent,

N_{max} Max number of iterations,

$\{S_j\}$ A set of discretized storage volume for reservoir j, where $S_j = \{s_j^1, s_j^2, ..., s_j^{n_j}\}$ in cms-d,

n_j Number of state discretization for reservoir j,

d Number of actions,

$\{A\}$ Set of actions, where $A_t = \{a_t^1, a_t^2, ..., a_t^d\}$,

t Current time period (stage) in a cycle (iteration or episode), where $t \in \{1,\ldots, T\}$,

$s_{j,t}$ Storage volume in reservoir j at beginning of the period t in cms-d,

$s_{j,t+1}$ Storage volume in reservoir j at end of the period t in cms-d,

s' State vector representing a possible combination of different storage increments of the different reservoirs within the state space where, $s' = (s_1^{i_1}, s_2^{i_2},..,s_J^{i_J})$, with $i_j \in \{1,...,n_j\}$ *for j=1*, ..., J in cms-d,

a_t Action (decision variable): forward sales (pre-sales) at time period t, where a $\in \{ a_1,a_2,\ldots, a_d\}$ in MWh,

$I_{j,t}^{\omega}$ Local natural inflow to reservoir j in period t for scenario ω, where $\omega \in \{1,.., \Omega\}$ in cms,

L_t^{ω} System load at time period t for scenario ω in MWh,

P_t^{ω} Market price at time period t for scenario ω in \$-Canadian/MWh,

$Q_{T_{j,t}}$ Turbine releases from reservoir j during period t in cms,

$Q_{S_{j,t}}$ Spill flow from reservoir j during period t in cms,

$r_t(s',a)$ Rewards of taking action a_t and transition to state s'_{t+1} at end of period t in \$-Canadian,

$Q_t(s',a)$ Q-value function when the system state is s'_t and action a_t is selected at the beginning of time period t in cms,

$f_t(s')$ Expected value function when the system state is s'_t in the beginning of time period (stage) t in \$-Canadian,

γ Monthly discount factor of future rewards,

α_n Learning rate (step size) parameter in iteration n, where $n \in \{1,\ldots, N\}$,

$\tilde{e}$ Vector of random events that influence the operation of the multireservoir system, where $\tilde{e} = (I^{\omega}, L^{\omega}, P^{\omega})$,

$\in_n$ Exploitation rate in iteration n (probability of random action in $\in$-greedy policy),

π Policy, decision making rule,

$\pi(s')$ action taken in state s' under deterministic policy π, and

$\pi(s',a)$ probability of taking action a in state s' under stochastic policy π.

4.2.1. SDP Formulation

One possible way to solve this reservoir optimization Markovian Decision Problem (MDP) is to apply one of the traditional SDP techniques, such as the value iteration technique. This formulation involves a sequence of decisions in which each decision affects what opportunities are available in the future. For any given time period and at any starting state, the outcome from applying an action to a state depends only on the current state. Also, the effects of the outcomes are not deterministic.

The basic structure of the SDP algorithm can be characterized by:

t Discrete time,

s_t State variable,

a_t Control; decision to be selected from a given set,

r_t Reward signal,

$\widetilde{e}$ Set of random variables, and

T Horizon.

The mathematical formulation of the reservoir optimization problem can be expressed as follows:

Objective function:

$$f_t(s'_t) = \underset{a_t}{Max}\, E\{\gamma[r_t(s'_t, a_t, \widetilde{e}_t) + f_{t+1}(s'_{t+1})]\} \tag{4.1}$$

Constraints:

$$s_{j,t+1} = s_{j,t} - \sum_{j'=1}^{J}(C^T_{j'j}Q_{T_{j',t}} + C^S_{j'j}Q_{S_{j',t}}) + I_{j,t} \quad \forall j, j' \in J, \quad \forall s \in S \tag{4.2}$$

$$s^l_{j,t} - \Delta s_{j,t} \le s_{j,t} \le s^u_{j,t} \qquad \forall j \in J, \quad \forall s \in S \tag{4.3}$$

$$a^l_t \le a_t \le a^u_t \qquad \forall a \in A \tag{4.4}$$

where $C^{T}_{j'j}$ and $C^{S}_{j'j}$ are the elements of the matrices representing the hydraulic connection between the reservoirs in terms of the turbine outflow and spill outflow respectively from reservoir j' to reservoir j. $C^{T}_{j'j}$ and $C^{S}_{j'j}=0$ if there is no physical hydraulic connection between the reservoirs, $C^{T}_{j'j}$ and $C^{S}_{j'j}=1$ if $j=j'$, and $C^{T}_{j'j}$ and $C^{S}_{j'j}=-1$ for $\forall j \neq j'$ and reservoir j is physically connected to reservoir j'.

For any given system state, the objective is then to maximize the expected long term value of resources resulting from the reservoir operating decisions taken in the current period and for decisions that could be made out in the future.

Equation (4.2) represents the set of constraints describing the continuity of flow for each reservoir (j) considering the physical setting of any reservoir (j') within the river systems. In case the storage exceeds the maximum limit the reservoir spills. The minimum storage is treated as a soft constraint and storage below the minimum storage constraint is penalized.

Equation (4.3) represents the upper and lower limits on the storage constraint. Equation (4.4) represents the constraints on the upper and lower limits on the decision variable.

4.2.2. RL Formulation

Similar to the conventional SDP methods, the mathematical formulation of the RL model is cast as a Markovian Decision Problem. The theoretical basis of the RL approach, as presented in the previous chapter, formed the foundation for the methodology adopted in the development of the reinforcement learning multireservoir optimization model (RLROM). In the proposed algorithm, the agent (controller) first interacts with a simulation environment that models the real reservoir system, to learn the optimal control policies (Figure 4.1). Once the agent learns the optimal value function, it can use its knowledge at any time period and in any given system state to control the system or to estimate the MVW. Alternatively, the agent can keep on learning continuously as it controls the system, and it can adapt quickly to changing operating conditions and keep on updating and enhancing its knowledge about different operating conditions of the system.

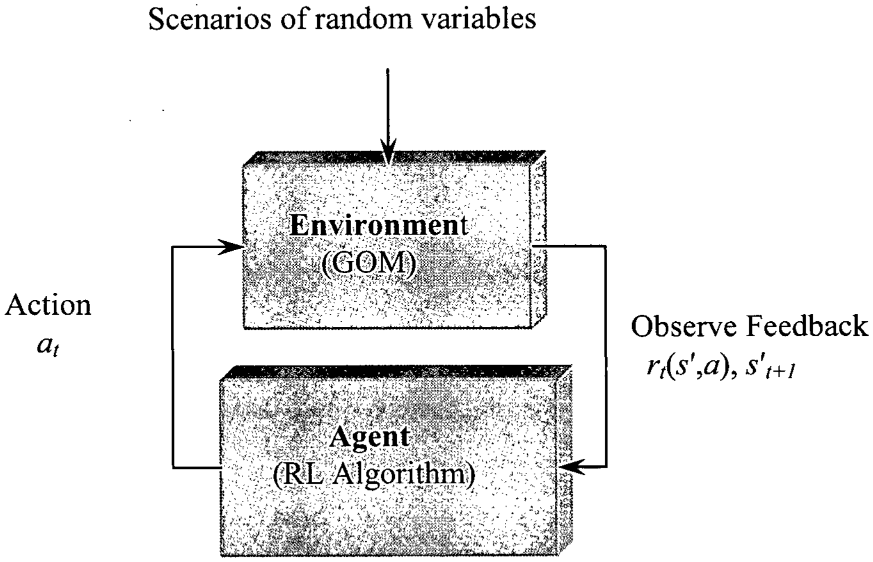


Figure 4.1. RL agent-environment interface in learning mode

In this research, a reinforcement learning (Q-Learning) algorithm which relies on the stochastic approximation of the value iteration is adopted. The main idea of the proposed RL formulation is to apply a sampling technique while performing the iterations rather than computing the expected rewards and the transition probabilities. The transition state s'_{t+1} and rewards $r_t(s,a)$ are generated from the state-action pair (s_t,a_t) by simulation. This RL formulation can be regarded as a combination of value iteration and simulation. Accordingly, rather than estimating the value function for each state (Equation 4.1), one can compute a value for each state-action pair which is known as the Q-value or Q-factor applying the following Q-Learning formula:

$$Q_t^N(s',a) = Q_t^{N-1}(s',a) + \alpha_t\left\{\gamma[r_t\ (s',a) + \max_{a_{t+1}} Q_{t+1}^{N-1}(s',a)] - Q_t^{N-1}(s',a)\right\} \tag{4.5}$$

where N is the iteration number and t is the time period (stage). It should be noted that the discounting is applied to both of the current and future rewards as it is assumed in the model formulation that the rewards of each time period are realized at the end of each period. The details of the RLROM model components and the proposed solution algorithm are described in detail in the following sections.

The proposed RLROM model is linked to a Generalized Optimization Model (GOM) for the hydroelectric reservoir optimization model as described in section (4.2.10). At each time period, the agent sends a signal to GOM describing the state of the system s_t' and the action it has taken a_t. GOM returns an estimate of the rewards $r_t(s',a)$ that informs the agent how well it did in selecting the action, and it generates a set of transition states s_{t+1}'. The RL agent controls the forward sales (presales) decisions, and based on these decisions, GOM determines the optimal system operation in terms of the release and generation strategies. This agent-environment interaction process is performed at each time period over the planning horizon and for a set of generated scenarios that models the uncertainty in electricity load, market prices, and natural inflows. The stochastic modeling of these random variables is described in detail in section (4.2.8).

While the main concern is to establish the optimal control strategies and the MVW of water in the two main multiyear storage reservoirs, linking the RL model with the GOM model allowed the inclusion of more reservoirs in the optimization process. The model optimizes the operation of the five main plants within BC Hydro system, namely: GM Shrum, Peace Canyon, Mica, Revelstoke, and Keenleyside. Therefore, the operation planning decisions developed by the RL algorithm encompasses the major plants within the BC Hydro system.

4.2.3. State Space

At any time period, the state of the system is typically expressed as the amount of available storage in each reservoir at the beginning of that time period. The reservoir storage variable used to represent the system state is defined by a *J* dimensional space. The continuous storage of reservoir *j* is discretized to n_j discrete storage values. The state space can be represented as a cartesian product of all state combinations as follows:

$$\{s_1^1,\ldots,s_1^{n_1}\}\otimes\ldots\otimes\{s_j^1,\ldots,s_j^{n_j}\}$$

It is assumed that the number of discretized states for reservoir *j* is constant during the *T* periods of the planning study.

4.2.4. Action (Decision Variable)

The decision variable considered in developing the RL model is the forward sales (pre-sales) to/ from the US and Alberta markets. The decision variable (a_t) is discretized at each time period to d_t decisions. For each time period, the forward sales are subdivided into three categories namely: peak, high, and low to capture the effects of price variations during the heavy load hours (HLH) and light load hours (LLH).

Traditionally, the turbine release from each reservoir is the common choice of the decision variable in stochastic reservoir optimization models. However, in this form of the simulation based optimization of the reinforcement learning model, the turbine releases are calculated during the interaction process between the agent and the environment (GOM). In the proposed RL model formulation, the turbine releases from the different reservoirs are based on the current state of the agent, the presale decision and the adopted scenario of the random variables.

One of the main advantages to the choice of presales as the decision variable is that it reduces the dimensionality of the problem. This is mainly due to the fact that presale is a system variable which is independent of the number of the reservoirs involved in the RL model. In comparison, if the turbine releases were used as decision variables in the RL framework that increase the dimensionality of the decision space by many orders of magnitude.

4.2.5. Exploration/Exploitation Rate

One possibility for the agent is to pick the action with the highest Q-value for the current state- this can be defined as **exploitation**. As the agent can learn only from the actions it tries, it should try different actions ignoring what it thinks is best, some of the time- and this can be defined as **exploration**. At the initial stages of the learning process, exploration makes more sense, as the agent does not know much. The adopted equation for estimating the probability that the learning agent should select the best action in the algorithm developed herein is proposed by Michael Gasser, (2004). The equation states that:

$$\varepsilon = 1 - e^{-\zeta N} \tag{4.6}$$

where:

ε = exploitation rate,

N = the number of iterations or the age of the agent, and

ζ = exploitation parameter.

It is clear that as the number of iterations increases, the exploitation rate increases. Eventually, when the number of iterations gets larger, the agent will tend to select the greedy actions, and these actions represent the policy learned by the agent. Figure (4.2) presents the exploitation rate as a function of the agent's age.

4.2.6. Learning Rate

The learning rate α controls the step size of the learning process. In fact, it controls how fast we modify our estimates of the action-value function. Usually, the iterations start with a high learning rate that allows fast changes in the Q-values and then the rate gradually decreases as time progresses - as shown in Figure (4.2). This is a basic condition to assure the convergence of the Q-Learning approach as identified by Sutton and Barto (1998). The adopted formulation in the developed model is a polynomial learning rate as given by Even-Dar and Mansour, (2003) as follows:

$$\alpha=1/N^{\psi} \qquad (4.7)$$

where:

N = Number of iterations (the age of the agent),

ψ = Parameter, where $\psi \in (0.50,1.0)$.

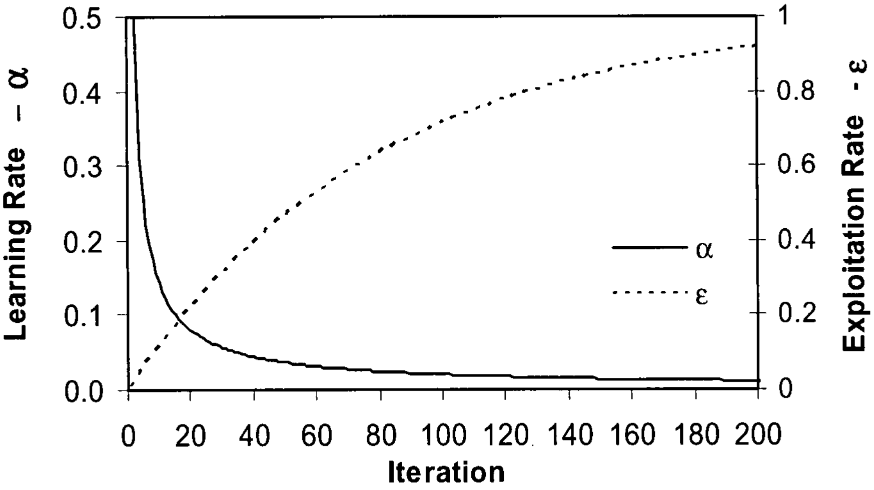


Figure 4.2. Learning rate and exploitation rate

4.2.7. Rewards

The reward function $r_t(s',a)$ is very important in the communication process between the environment and the agent. It helps the agent to learn how to perform actions that will achieve its goals. Depending on the reward signal received from the environment, the agent perceives how well it did in selecting the action. Generally, the objective is to maximize the sum of the reinforcements received from the environment over time. In the multireservoir problem we are dealing with is an infinite time horizon problem, therefore a discount factor, γ, is introduced. In the multireservoir model, the agent should learn the operation policy $\pi^*(s')$ that maximizes the benefits from taking action a_t when the agent is in state s'.

$$r_t(s',a)=\sum_{N=0}^{\infty}\gamma^N r_{t+N+1} \tag{4.8}$$

At each time period, and based on the current state of the system and the agent's decision regarding forward sales, the optimization model (GOM) determines the optimal control strategy that maximizes the benefits. The reward function, which is selected to maximize the value of resources, is calculated at the end of each time period. The details of the GOM optimization model are presented in section 4.2.10.

4.2.8. Stochastic Modeling of Random Variables

In modeling multireservoir hydro-electric power generation system, we are interested in establishing the optimal operation planning strategies and the expected value/marginal value of the resources given the uncertainty in natural inflows, electricity demand, and market prices. This problem is complicated because of the multidimensional probability distribution of the random variables and their inter dependencies. One approach to fit the marginal distribution of the random variables is to use time series autoregressive models. In these models, the autocorrelation of the random variables are modeled by a continuous Markov decision process.

The methodology adopted in this research relies on approximating the continuous distributions and stochastic processes by discretization to a number of scenarios rather than the Markov description of the stochastic variables.

To represent the inflow, electricity load, and market price random variables, a moment matching (MM) scenario generation approach developed by Hoyland et al.

(2003) has been adopted. In this approach, the marginal distributions are described by their first four moments (mean, variance, skewness, and kurtosis). In addition, the correlation matrix is also specified. The method generates a discrete joint distribution that is consistent with the specified values of the marginal distributions and the correlation matrix.

The moment matching algorithm first decomposes the multivariate problem to univariate random variables that satisfy the first four moments for each random variable. Then, it applies an iterative procedure that involves simulation, decomposition and various transformations to preserve the original marginal moments and the correlation matrix. The details of the algorithm are given by Hoyland et al. (2003).

Historical stream flow records for the Peace and the Columbia rivers are used to estimate the distribution properties of the inflow random variable. Considering the case of the Peace and the Columbia River systems and assuming a monthly time step in the RLROM model, the number of random inflow variables for one year will be twenty four, as inflow in each month is considered as a random variable. First, the first four moments and the correlation matrix for the twenty four variables are calculated. Then, this information is fed as an input to the MM model. The generated inflow scenarios, I_t^{ω}, represents the cross and serial correlation between the inflows at the different time periods for the two river systems for one year. The outcome of this process is a reduced (manageable) number of inflow scenarios that preserve the properties of the original distributions.

In the Pacific Northwest, electricity price variations are correlated to runoff volumes in the northwest. In dry years, power production falls and prices increase accordingly, while in wet years, there is more power available and prices decrease proportionally. Also, prices vary to a large degree across the day. Therefore, the average forecasted market prices are adjusted to reflect the relationship between runoff volumes and market prices. This relationship is represented by a regression relationship between the Dalles monthly runoff volume in the US and the Mid-Columbia market price. The following approach was adopted to represent the price variability in the RLROM model. First, a stochastic variable for the annual Dalles runoff volume was introduced in the moment matching scenario generation model in addition to the twenty four inflow variables of the Peace and Columbia inflows. Then, the scenarios generated for the Dalles runoff were correlated to the heavy load hour (HLH) and light load hour (LLH) price factors (multipliers) using polynomial regression relationships. Finally, the average price forecast for the Mid-Columbia was multiplied by the HLH and LLH price multipliers to generate the HLH and LLH prices for the scenarios.

BC Hydro's system load forecast was estimated using a Monte Carlo energy model. The load forecast is mainly impacted by: (1) the economic growth measured by the gross domestic product (GDP), (2) electricity prices billed to BC Hydro's customers, (3) elasticity of the load with respect to electricity prices and economic growth, and (4) energy reduction due to demand side management (DSM). The continuous distribution of the BC Hydro system load forecast is represented by the values of the median (P_{50}) and the two quantiles (P_5 and P_{95}). Starting from these three points, the expected value and the variance of the forecasted load are then estimated using the method of Pearson and Tukey (1965). Then, a Monte Carlo simulation was carried out, using a Log-Normal probability distribution, to generate several thousand scenarios. The generated scenarios were then aggregated in a histogram with a specified number of intervals that represent the required probably distribution. Based on the generated data, the first four moments and the correlation of the electricity load with the inflow and price variables were estimated and used in the moment matching algorithm. The details of the data used, the results, and an analysis of the scenario generation process is presented in chapter 5.

4.2.9. Lookup Tables and Function Approximation

During the experimental and testing phase of the RLROM model development, the state space was discretized into a small number of points. The state space in this case is the reservoir storage levels for the different reservoirs. For this limited number of state space variables, the learning process for the Q-values was implemented using lookup tables. Lookup tables are used to store and update the function representing the value of water in storage at each time period. At each time period, and for each discrete point on the state space, there exist a row of entries in the lookup table. The elements of each row are: reservoirs storage s', action a_t, reward value $r_t(s', a)$ and Q-value $Q_t(s', a)$.

In real applications of multireservoir optimization, the number of grid points grows exponentially as the number of reservoirs increases. This results in a much larger state space and the use of lookup tables becomes impractical. Accordingly, some sort of function approximation is required, which will enable the calculation of the future value function $f_{t+1}(s')$ at any point within the state space without the need to store every value. Two properties of this function need to be considered in the development process. First, the storage value function is typically a concave nonlinear function. Second, the target storage (end of period storage) of each reservoir is a function of the storage levels in other reservoirs.

In this research work, two alternative techniques for function approximation were investigated. The first approach is a function fitting approach using polynomial regression. An alternative function fitting approach was tested using a linear interpolation technique. However, instead of using one function to approximate the entire state space, it was divided into a finite number of segments or pieces using a piecewise linear (PWL) interpolation technique. The following figure illustrates an example of approximating the concave nonlinear value of water in storage function with four linear pieces.

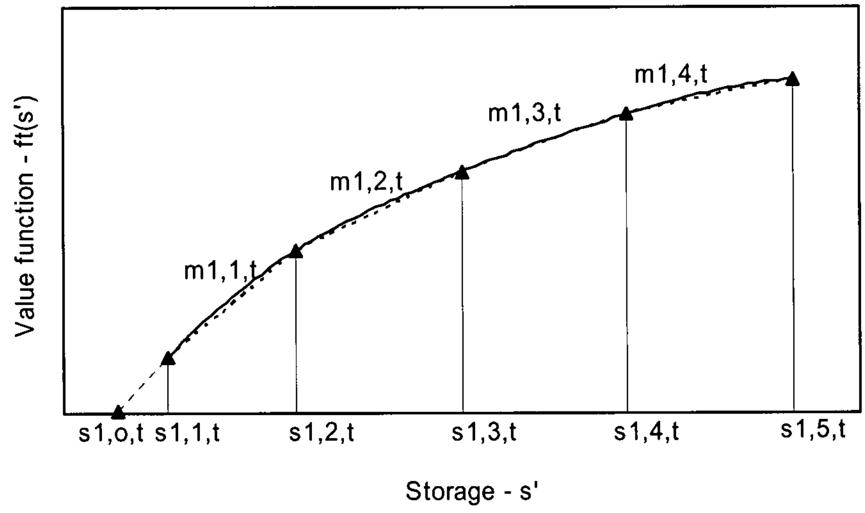


Figure 4.3. PWL approximation of storage value function

As an example, $s_{1,3,t}$ represents the third break point of reservoir 1 storage at time t for a given storage level of the other reservoirs. Similarly, $m_{1,3,t}$ is the slope of the third segment of the PWL function of reservoir 1 at time t, and $s_{0,3,t}$ represents the intercept of the curve with the storage axis. During the testing and development phase, the advantages of using PWL formulation over nonlinear formulation were noticeable in terms of more stability in the results and also faster implementation in the RLROM model. The faster implementation is mainly attributed to the way that AMPL mathematical programming language handles PWL functions. In AMPL, PWL functions are defined by: a set of breakpoints (grid points of the state space), the slope of the different segments, and the intercept. In the course of the iterations, for any target storage point on the state space, the PWL function approximation is implemented to calculate the storage value function for the multidimensional state space as presented in Figure 4.4 in a 3D view. The PWL function deals with one state variable at a time. As an example, consider the case of the value function being a function of the storage value in two reservoirs (for example: GMS and Mica). First, PWL functions are constructed for the different storage grid points of Mica as a function of different storage levels in GMS as shown in Figure 4.5. At the target storage of Mica, a PWL curve is constructed as a function of the different GMS storage grid points. This PWL function was then used to evaluate the value of water in storage as shown in Figure 4.6.

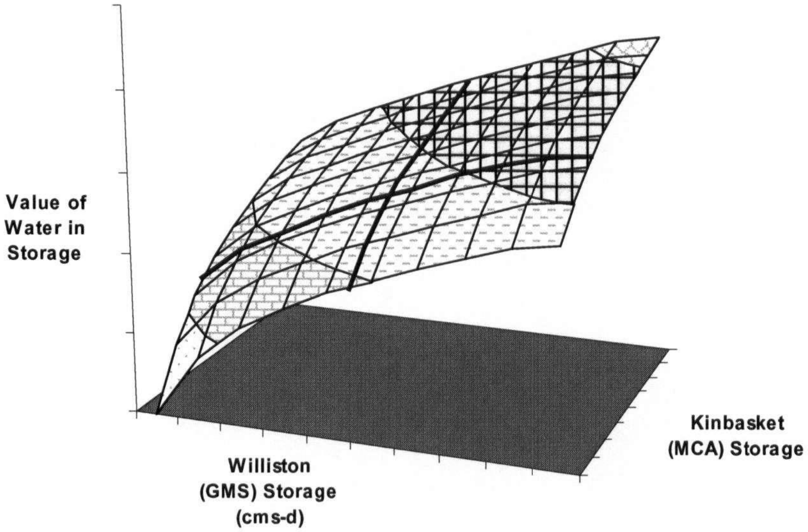


Figure 4.4. 3-D view of the PWL value of storage function as a function of GMS and Mica storage

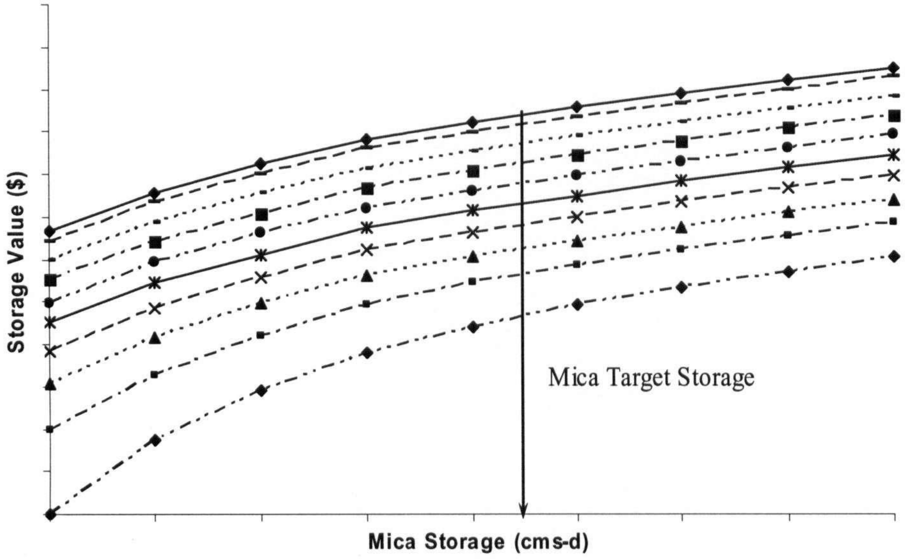


Figure 4.5. 2-D view of the PWL value of storage function as a function of GMS and Mica storage

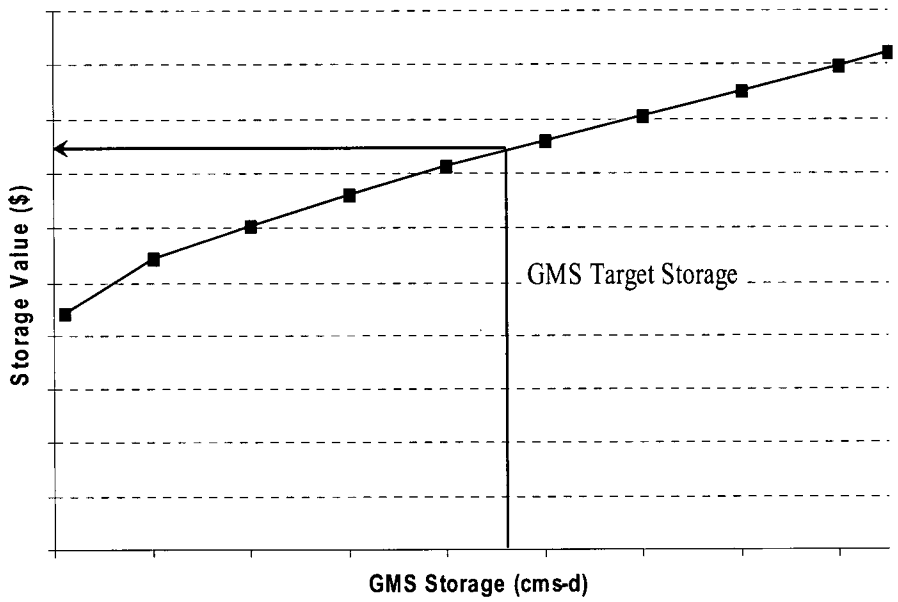


Figure 4.6. PWL value of storage function as a function of GMS storage at Mica target storage

4.2.10. Model of the Environment

The RL agent needs to interact with an environment while learning to control the system. This environment could either be the real system or a model of the system. In this research work, the BC Hydro generalized optimization model, GOM, was used as the simulation environment of the real system. The GOM model was adapted from the short term optimization model (STOM) developed by Shawwash et al. (2000). GOM, which incorporates the basic optimization formulation of STOM, was developed to give its user the flexibility for a more generalized form so that it can be used over longer time horizons and at various time resolutions. The model is currently used at BC Hydro as an analytical tool that assists the operations planning engineers to simulate and optimize the operation planning of the integrated BC Hydro system. The primary objective of the model was to develop optimal system operation schedule given a deterministic forecast of system inflows, domestic load and market prices while maximizing the value of BC Hydro resources.

GOM is a variable time step model with the capability to model sub-time steps in detail. The time steps may be hourly, daily, weekly, sub-monthly and/or monthly. The GOM system includes a detailed hydraulic simulation that calculates the hydraulic balance and calculates generation and turbine limits for each time step. Sub-time steps may be used to further divide the time step into shorter time periods to reflect different load conditions within a time step, as derived from the load-duration curves. For example, for a time step that is greater than or equal to a week, the load-duration curves are used to represent both weekday and weekend load shapes. The sub-time step thus provides a more detailed view of the load and resource balance, and the market trade-offs under different load conditions (i.e. super peak load, peak load, heavy load, shoulder load and light load hours). The load-resource balance and trade-off optimization is performed for each sub-time step.

The non-linear power generation function is represented in GOM as a piecewise linear surface function where the generation is calculated as a function of the forebay level $FB_{j,t}$, turbine discharge $Q_{Tj,t}$, and unit availability $U_{j,}$, $G_{j,t} = f(FB_{j,t}, Q_{Tj,t}, U_{j,t})$. This function was developed with an optimal unit commitment and loading assumption. Accordingly, each point on the piecewise linear surface function represents the maximum generation attainable given the set of turbine discharge, forebay, and the number of units available for commitment. The procedure, which was followed to prepare these plant production functions for the BC Hydro plants is described in detail in Shawwash, (2000). The following figure illustrates an example of the PWL function for a typical hydroelectric power generating plant.

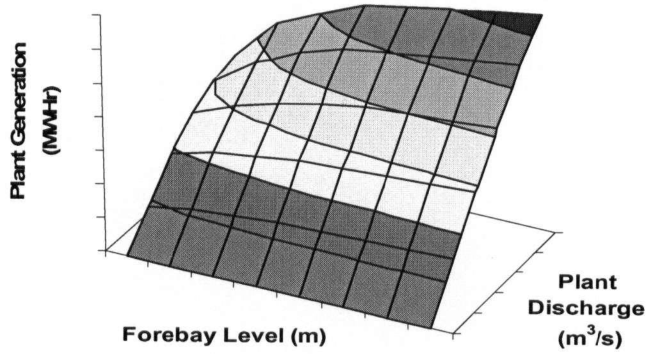


Figure 4.7. Piecewise linear hydropower production function

The formulation of the GOM model was modified to run interactively with the RL model as detailed below. At each time step GOM receives the information from the RL model on the initial state of the system, it then updates the system constraints and solves the one stage reservoir optimization problem and then it passes the optimized results back to the RL model. During the learning phase of the RL agent, the information passed to the RLROM model constitutes the transition state and the reward signal. In the final iteration, the optimal operation polices are derived, including the plant generation, market transactions (import/export), and turbine releases.

Hence, linking the RLROM with the GOM has the advantage of capturing the diurnal variation in load, market prices, and generation schedules for the shorter periods within the time step, either during weekdays/ weekends or Heavy load hours (HLH)/ Light load hours (LLH). The following is a description of the GOM model formulation including the decision variables, constraints, and the objective function:

Decision Variables

$Q_{T_{k,t,h}}$ Turbine release from reservoir k at time period t and sub-time step h, in cms,

$Q_{S_{k,t,h}}$ Spill (non-power) release from reservoir k at time period t and sub-time step h, in cms,

$G_{k,t,h}$ Generation from plant k at time period t and sub-time step h, in MWh,

$Spot_{US_{t,h}}$ Spot transaction (import/export) to US market at time period t and sub-time step h, in MWh, and

$Spot_{AB_{t,h}}$ Spot transaction (import/export) to Alberta market at time period t and sub-time step h, in MWh.

Constraints

Hydraulic continuity equation

$$S_{j,t+1} = S_{j,t} - \left[\sum_{k=1}^{K}\sum_{h=1}^{h_n}\left(Q_{T_{k,t,h}} * C_{kj}^{T} + Q_{S_{k,t,h}} * C_{kj}^{S}\right) * H_{t,h} + I_{k,t} * H_{t}\right]/24 \qquad \forall k,t \qquad (4.9)$$

and at t=T: $S_{j,t+1} = S_{j,1}$ (4.10)

where:

C_{kj}^{T} and C_{kj}^{S} are the elements of the matrices representing the hydraulic connection between the reservoirs in terms of the turbine outflow and spill outflow respectively from reservoir k to reservoir j. C_{kj}^{T} and C_{kj}^{S}=0 if there is no physical hydraulic connection between the reservoirs, C_{kj}^{T} and C_{kj}^{S}=1 if j=k, and C_{kj}^{T} and C_{kj}^{S}=-1 for $\forall j \neq k$ and reservoir j is physically connected to reservoir k,

$H_{t,h}$= number of hours in sub-time step h at time step t and $h \in (1,2,\ldots,h_n)$,

H_t = number of hours in time step t.

Storage bounds constraint

$$S_{k,t}^{Min} - \Delta S_{k,t} \le S_{k,t} \le S_{k,t}^{Max} \qquad \forall k,t \tag{4.11}$$

where the storage is expressed as a PWL function of the reservoir forebay Fb_k, $S_{k,t} = f(Fb_{k,t})$. This function, which is not part of the optimization model, is used to relate the storage volume to the reservoir elevation (Forebay) within the GOM model. $\Delta S_{k,t}$ is a variable representing the deviation from the minimum storage limit which is penalized in the objective function. In case the storage exceeds the maximum limit the reservoir spills. The minimum storage is treated as a soft constraint and storage below the minimum storage constraint $\Delta S_{k,t}$ is penalized in the objective function.

Power generation constraint

$$G_{k,t,h} = f(FB_{k,t,h}, Q_{T_{k,t,h}}, U_{k,t,h}) \qquad \forall k,t,h \tag{4.12}$$

Total plant generation constraint

$$G_{Tk,t,h} = G_{k,t,h} + G_{k,t,h} * OR_k \qquad \forall k,t,h \tag{4.13}$$

where:

OR_k = the percentage of operating reserve from plant k. The operating reserve is a specific level of reserve power should be available at all times to insure reliable electricity grid operation.

$G_{T_{k,t,h}}$ = the total of plant generation and operating reserve from plant k at time period t, in MWh, and h sub-time step.

Load resources balance (LRB) constraint

$$\sum_{k=1}^{K} G_{k,t,h} + G'_{t,h} - G_{f_{t,h}} - Spot_{US_{t,h}} - Spot_{AB_{t,h}} = L_{t,h} \qquad \forall t,h \tag{4.14}$$

where:

$G'_{t,h}$ = the fixed and shaped generation from other small hydro and thermal plants, not included as a decision variables in the optimization problem.

$G_{f_{t,h}}$ = the forward sales; this information is passed at each time period from the RL model.

$L_{t,h}$ = the load at time period *t*, and at subtime step *h* in MWh.

Spot US Transactions constraint

$$T^{Max}_{US_{t,h}} \geq Spot_{US_{t,h}} \geq T^{Min}_{US_{t,h}} \qquad \forall t,h \tag{4.15}$$

where T^{Max}_{US} = the inter-tie transmission limit from BC to the US and T^{Min}_{US} is the inter-tie transmission limit from the US to BC.

Spot Alberta Transactions constraint

$$T^{Max}_{AB_{t,h}} \geq Spot_{AB_{t,h}} \geq T^{Min}_{AB_{t,h}} \qquad \forall t,h \tag{4.16}$$

where T^{Max}_{AB} = the inter-tie transmission limit from BC to Alberta and T^{Min}_{AB} is the inter-tie transmission limit from Alberta to BC.

Generation limits constraint

$$G_{k,t}^{Min} \leq G_{T_{k,t,h}} \leq G_{k,t}^{Max} \qquad \forall k,t,h \tag{4.17}$$

Objective Function (*MaxRev*):

$$\begin{aligned} &Maximize: \quad \sum\nolimits_{h=1}^{h_n} \left\{ (Spot_{US_{t,h}} + Spot_{AB_{t,h}}) * H_{t,h} * P_{t,h} + G_{f_{t,h}} * H_{t,h} * P_{f_{t,h}} \right\} \\ &+ \sum\nolimits_{k=1}^{K} \left\{ (s_{k,t+1}^{N} - s_{k,t+1}^{N-1}) * MVW_{k,t}^{N-1} \right\} + \sum\nolimits_{h=1}^{h_n} \left\{ \Delta S_{k,t} * c_k \right\} \qquad \forall t \end{aligned} \tag{4.18}$$

where $P_{f_{t,h}}$ is the forward market price. The ***MaxRev*** objective function maximizes the value of power generation at each time period *t* given a target reservoir level $s_{k,t}^{N-1}$ and the marginal value of water $MVW_{k,t}^{N-1}$ estimated by the RLROM model. The objective function consists of four terms: The first and second terms represent the sum of the revenues from the spot transactions to both the US and the Alberta markets and the forward sales, where $P_{t,h}$ and $P_{f_{t,h}}$ are the spot and forward market prices at time step *t* and sub-time step *h*. The third term accounts for the trade-off between the short and long term values of water in storage as the difference between the target (end of period) storage calculated in iteration *N*-1 and the target storage calculated in the current iteration *N* multiplied by the marginal value of water calculated in iteration *N-1*. The fourth term penalizes the violation of the minimum and maximum storage limits ($\Delta S_{k,t}$), where c_k is the penalty for violating the storage limits specified in the RL model for reservoir *k*.

4.3. Solution Algorithm

This section presents a detailed description of the solution algorithm of the RLROM model outlined in the previous sections. First, a description of the RLROM solution algorithm using the tabular form for storing the Q-values is presented. This is followed by a description of the function approximation algorithm that can be used for larger state space problems. The RLROM model is implemented in the AMPL mathematical programming language (Fourer et al., 2003). A CPLEX solver (ILOG Inc.) implementing the simplex algorithm is used to solve the linear optimization problem of the GOM model.

4.3.1. RLROM Model Algorithm Using Lookup Tables

A flow chart illustrating the different steps of the model algorithm is presented in Figure 4.8. A detailed description of the implementation of the RL algorithm in a procedural form is described hereafter:

- Divide the state space $\{S\}$ to a finite number of discretized states that covers the range between the minimum and maximum reservoir storage.

- Define the number of stages T and the set of actions $\{A_t\}$.

- Use a graphical user interface (GUI) to process the data sets required in the model runs in the specified time steps and sub-time steps. These data include the load, price, and the transmission limits.

- Run a batch of GOM model jobs for each point on the state space grid for each stage in the planning period. Store the results of the model in a tabular form. These lookup tables inform the agent at each starting state and at each action what would be the transition state and the corresponding rewards.

- Run the Moment Matching technique and the Monte Carlo simulation described earlier to generate a specified number of scenarios Ω of the random variables: the natural inflow I_t^{ω}, forecasted load L_t^{ω}, and forecasted market price P_t^{ω}.

- At the start of the process, initialize the value function and the Q-values arbitrarily, or alternatively, set the values to zero for all states and state-action pairs. The RL agent moves forward in each iteration from stage 1 to the last stage T. To estimate the

Q-values, $Q_t(s',a)$, applying the Q-Learning update rule (Equation 4.5); the value function $f_{t+1}(s') = \max_{a_{t+1}} Q_{t+1}^{N-1}(s',a)$ at the end of period state s'_{t+1}, the rewards r_t, and the estimate of the Q-values $Q_t^{N-1}(s',a)$ are known from previous iteration.

- Set the model parameters including: the discount factor γ, the initial values of the learning rate α and the exploration rate ε according to the adopted formulas described earlier in this chapter.

- Set the number of cycles to N_{max}. Where N_{max} is chosen to be a large number that satisfies the convergence criteria, as described in detail in the following section.

- Starting at the first stage, initialize the algorithm by randomly sampling the state space (i.e. randomly choose a point in the state space grid s').

- Randomly sample the stochastic variables from the scenarios generated from their probability distribution.

- In the first iteration, the agent chooses action a_t randomly, as it has not learned yet any information about the Q-values. In subsequent iterations, the agent chooses the action a_t using the ε-greedy policy $\hat{\pi}_t(s')$ derived from the learned Q-values $Q_t^{N-1}(s',a)$ where: $\hat{\pi}_t(s') \in \arg\max_{a \in A} Q_t^{N-1}(s',a)$

- The agent interacts with the environment (GOM model results stored in lookup tables) and it receives a signal depending on the chosen action and on the sampled scenarios in the form of the next stage state transition $s'_{j,t+1}$ and a numerical reward $r_{j,t}(s',a)$.

- Apply the Q-Learning update rule (Equation 4.5) to estimate a new value for the Q-values, which can be presented in a general form as:

$$\textit{New Estimate} \leftarrow \textit{Old Estimate} + \textit{Step Size}\,[\textit{Target - Old Estimate}] \tag{4.19}$$

Store the new estimate of these Q-values as $Q_t(s',a)$.

- The agent moves to the selected state in the next stage, sets the target state s'_{t+1} in the first time period to be the initial state s'_t in the second time period. Repeat the procedure of sampling the action a_t using the ε-greedy policy and determine the reward signal and the transition state until the agent reaches the final stage.

- At the final stage T, the agent is at state s'_T. The agent repeats the same procedure as in the previous stages and receives the reward signal $r_T(s',a)$ and moves to the transition state s'_1. The Q-learning update equation for the estimate of the action-value function applies the following equation for estimating the future value function of the next stage:

$$f^N_{T+1}(s') = f_1^{N-1}(s') \tag{4.20}$$

- The agent starts a new iteration. In this new iteration and in subsequent iterations, until the termination of the algorithm, the agent is always using the information it learned so far to update the future value function estimates (i.e. reinforce its learning of the Q-values). In the beginning, the agent tends to explore more frequently, with a probability of (1-ε), to gain more knowledge about how good or bad it is at taking the actions. Later on, and as the age of the agent increases, the agent becomes more greedy and it chooses the best action, $\max_a Q_{t+1}(s',a)$, with a probability of ε. However, the agent also explores actions other than the greedy ones with a probability of (1-ε). As part of the convergence requirements (Sutton and Barto, 1998), the exploitation rate increases with the age of the agent and as the step size is decreasing (Figure 4.2).

- The above steps are repeated until convergence is achieved. The optimal value function $f(s')$ and the optimal generated policies $\pi_t^*(s') \in \arg\max_{a\in A} Q_t(s',a)$ are stored for all elements of the state space.

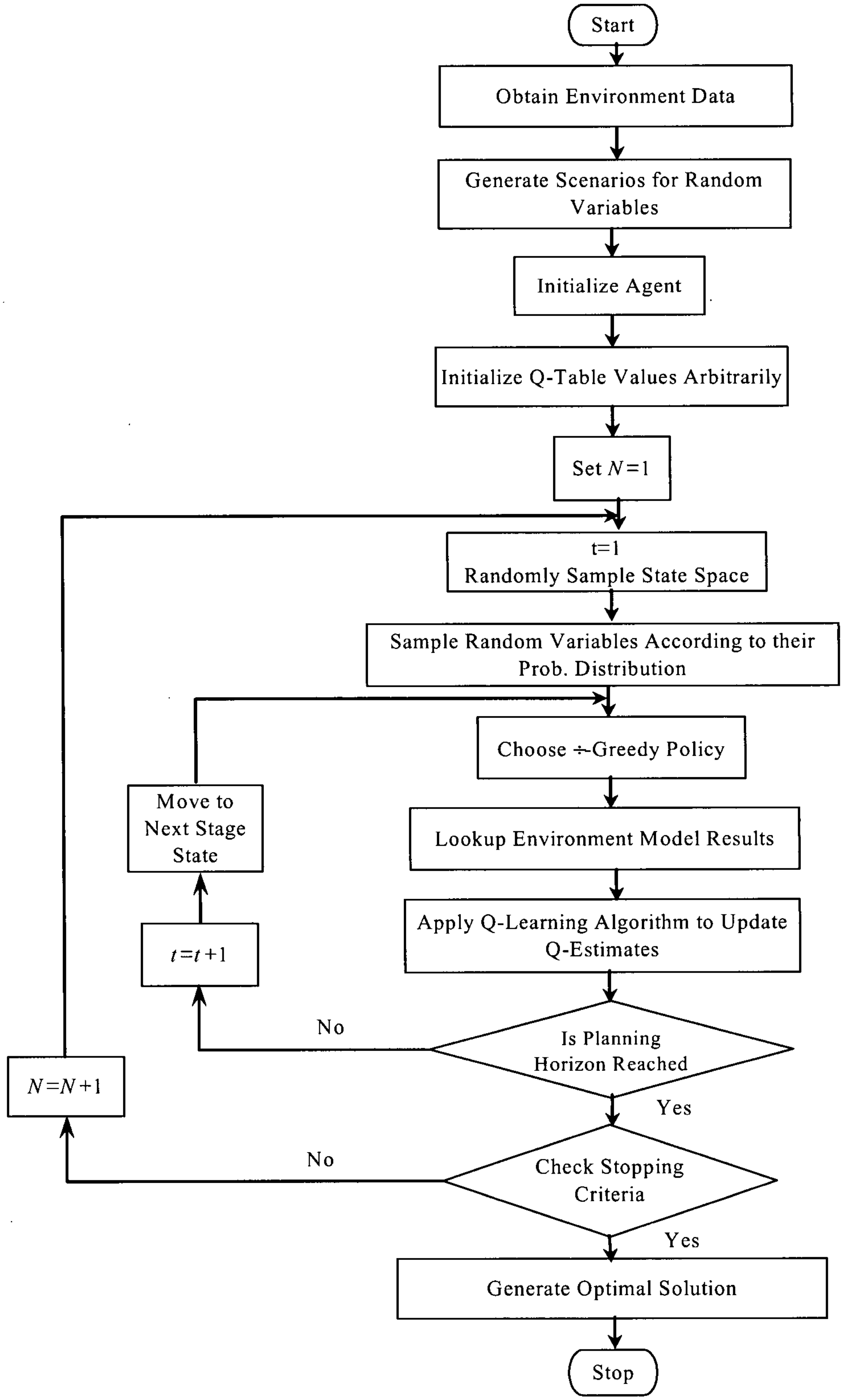

Start
Obtain Environment Data
Generate Scenarios for Random Variables
Initialize Agent
Initialize Q-Table Values Arbitrarily
Set N=1
t=1
Randomly Sample State Space
Sample Random Variables According to their Prob. Distribution
Choose ÷-Greedy Policy
Lookup Environment Model Results
Apply Q-Learning Algorithm to Update Q-Estimates
Is Planning Horizon Reached
No
Yes
t=t+1
Move to Next Stage State
Check Stopping Criteria
No
Yes
N=N+1
Generate Optimal Solution
Stop

Figure 4.8. RLROM model algorithm using lookup tables

4.3.2. The RL Model Algorithm Using Function Approximation

The algorithm presented above has the advantage of avoiding the calculation and storage of the transition probability matrix. However, as the state space and the decision variables increase, the use of the lookup tables to store the action-value function becomes impractical. This is mainly due to the fact that the memory required to store the Q-values becomes very large. Accordingly, the algorithm is modified to allow for the use of a larger number of state and decision variables. Function approximation using a piecewise linear function (PWL) approximation technique is used to overcome problems with the storage of the value functions in the RLROM model. In this case, the state space is a continuous functional surface rather than a *J* dimensional grid. One advantage of this method is that the target storage at every time period is not restricted to the grid points of the state space. Rather, it can be any value within the state space surface. The other advantage of this method is that it allows linking the GOM model to interact with the RL model on an on-line basis at each time step. At each time step, the agent passes the sampled scenarios of the random variables and the sampled state-action pair to the GOM model. The GOM model optimizes the operation of the reservoirs and sends back to the agent a signal in the form of a next stage transition and the rewards corresponding to the selected action. This flexibility, of having the target storage take any value, increases the chances of finding a feasible solution in the GOM model runs. This is unlike the case requiring that an optimal solution at the grid points be found, which in some cases results in infeasible runs for the GOM model. To overcome this problem, a much finer state space grid needs to be generated, which further increases storage requirements for these optimization problems. Therefore, function approximation results in a significant reduction in computer storage requirements and a more robust algorithm implementation of the proposed system.

Figure 4.9 presents a flow chart of the RL model algorithm using function approximation. The flow chart indicates the interaction of the RL agent with the model of the environment (GOM) online. Figure 4.10 displays a schematic representation of the RLROM model algorithm using PWL function approximation.

At each time period, the marginal value of water, $MVW_{t,j}(s')$, is calculated as the derivative of the value function $f(s')$ with respect to storage in reservoir j:

$$MVW_{t,j}(s') = \partial f_t(s') / \partial s_j \tag{4.21}$$

The marginal value of water is updated after each iteration, and its units is converted from \$/cms-day to \$/MWh using a conversion factor, $HK_{j,t}$, as function of reservoir storage: $HK_{j,t}(s) = G_{j,t}(s) / Q_{T_{j,t}}(s)$ (4.22)

where $G_{j,t}$ is the plant generation in MWh and $Q_{T_{j,t}}$ is the turbine discharge in cms-day.

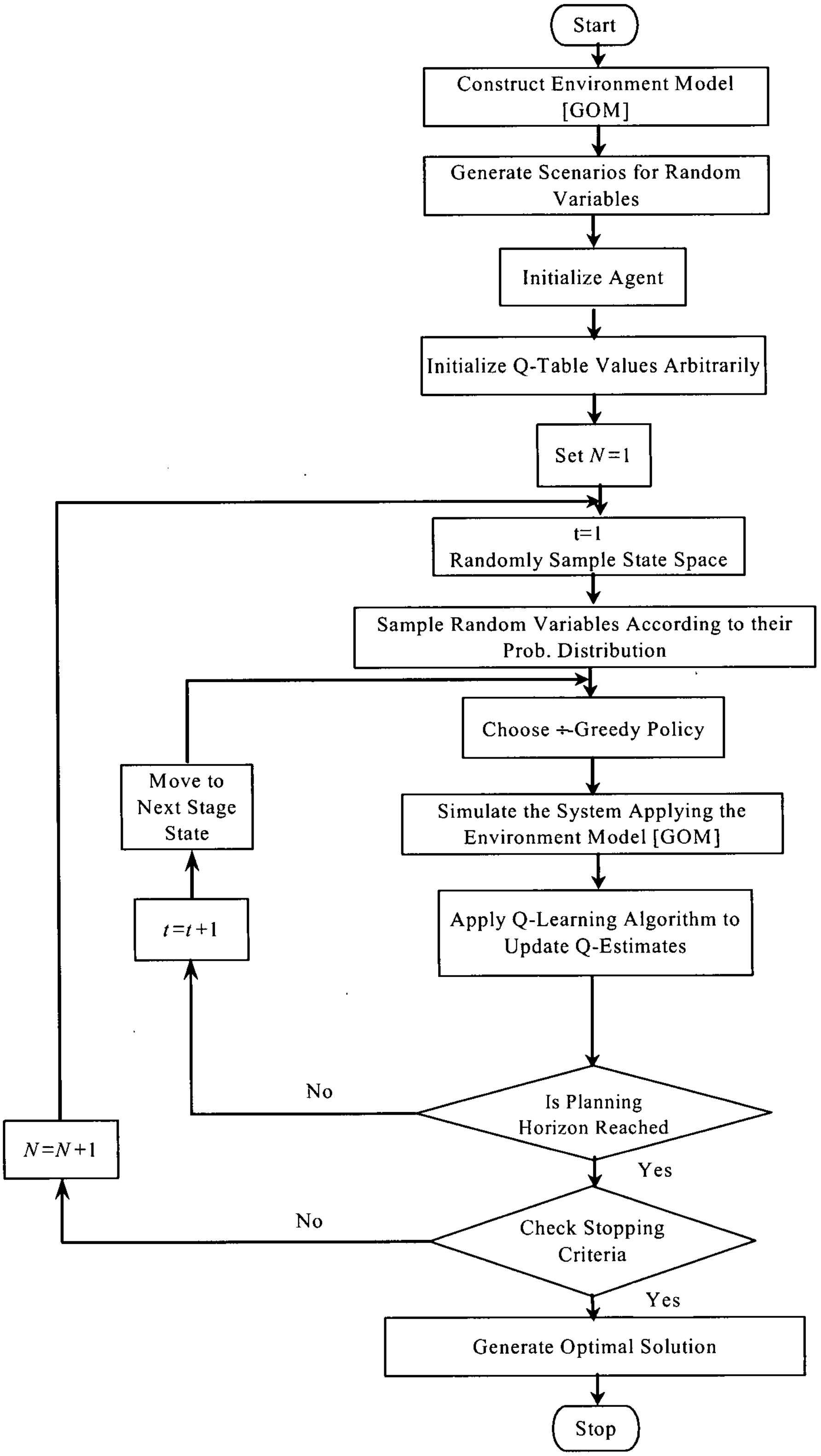
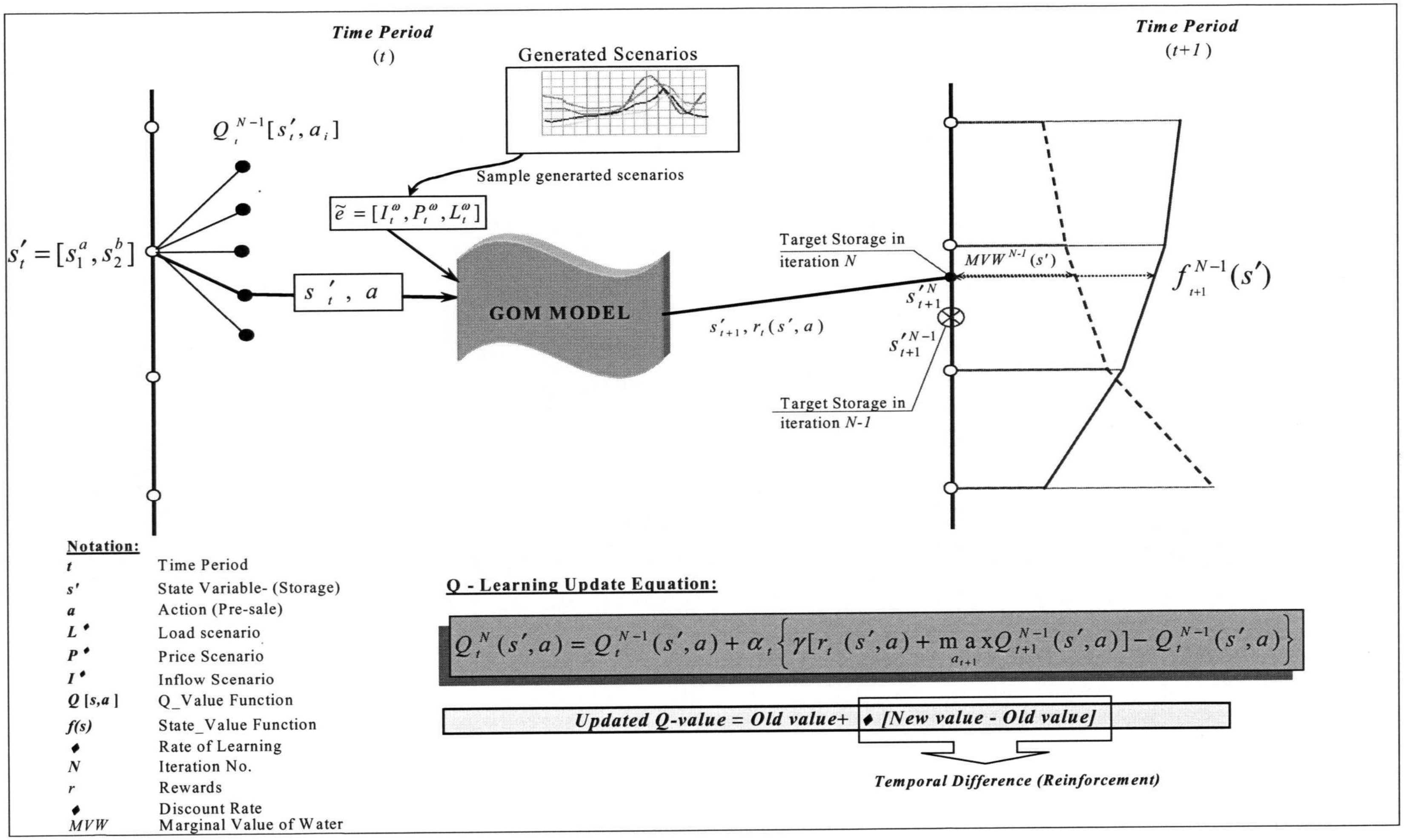

Start
Construct Environment Model [GOM]
Generate Scenarios for Random Variables
Initialize Agent
Initialize Q-Table Values Arbitrarily
Set $N=1$
t=1
Randomly Sample State Space
Sample Random Variables According to their Prob. Distribution
Choose ÷-Greedy Policy
Simulate the System Applying the Environment Model [GOM]
Apply Q-Learning Algorithm to Update Q-Estimates
Move to Next Stage State
$t=t+1$
No
Is Planning Horizon Reached
Yes
$N=N+1$
No
Check Stopping Criteria
Yes
Generate Optimal Solution
Stop

Figure 4.9. RLROM model algorithm using Function Approximation



Time Period
(t)
Generated Scenarios
Time Period
(t+1)
$Q_t^{N-1}[s_t', a_i]$
Sample generarted scenarios
$\tilde{e} = [I_t^{\omega}, P_t^{\omega}, L_t^{\omega}]$
$s_t' = [s_1^a, s_2^b]$
s_t', a
GOM MODEL
$s_{t+1}', r_t(s', a)$
Target Storage in iteration N
$MVW^{N-1}(s')$
$f_{t+1}^{N-1}(s')$
$s_{t+1}'^N$
$s_{t+1}'^{N-1}$
Target Storage in iteration N-1
Notation:
t Time Period
s' State Variable- (Storage)
a Action (Pre-sale)
L♦ Load scenario
P♦ Price Scenario
I♦ Inflow Scenario
Q [s,a] Q_Value Function
f(s) State_Value Function
♦ Rate of Learning
N Iteration No.
r Rewards
♦ Discount Rate
MVW Marginal Value of Water
Q - Learning Update Equation:
$Q_t^N(s',a) = Q_t^{N-1}(s',a) + \alpha_t\left\{\gamma[r_t(s',a) + \max_{a_{t+1}} Q_{t+1}^{N-1}(s',a)] - Q_t^{N-1}(s',a)\right\}$
Updated Q-value = Old value+ ♦ [New value - Old value]
Temporal Difference (Reinforcement)

Figure 4.10. Schematic representation of RLROM model algorithm using function approximation

4.3.3. Convergence

In stochastic dynamic programming, the optimality condition is reached when the solution is within a specified optimality tolerance. In general, the benefits of attaining the true optimal solution do not outweigh the time and cost of reaching the true optimal. The main concern in solving the stochastic reservoir optimization problem is to decide on how many iterations are sufficient to assure a good approximate solution.

Reinforcement learning algorithms rely on stochastic approximation in the evaluation process of the optimal value function $f^*(s')$. Stochastic approximation is an iterative procedure that produces a sequence of solutions in such a way that after a finite number of iterations the temporal difference of the expected values approach zero with probability 1 (Gosavi, 2003).

Accordingly, and as long as the Q-values are changing, we should continue to run the algorithm. As the age of the agent increases, the step size parameter α decreases. When the step size value becomes smaller than a specified value, the algorithm could be stopped. At this point the Q-values should stabilize and no change should occur for each state-action pair. After each iteration, of the RLROM runs, the absolute difference (ΔQ_t^N) between the estimated Q-value function in iteration N and iteration N-1 is calculated at each time period as follows:

$$\Delta Q_t^N = \left| Q_t^N(s',a) - Q_t^{N-1}(s',a) \right| \qquad \forall t, s', a \tag{4.23}$$

The computation terminates when the difference in the Q-values between successive iterations (ΔQ_t^N) remains constant for several iterations and consequently the Q-values are said to converge to the optimal solution. Gosavi (2003) also suggests other criteria to stop the algorithm when the policy does not change after a number of iterations, and this could be explored in the future.

4.4. Summary

The proposed RL approach outlined in the previous chapter, led a practical model, of the complex multireservoir system. Instead of modeling a single reservoir , the RLROM model was developed for the BC Hydro's two main river systems, the Peace and the Columbia Rivers. The model was formulated to establish an optimal control policy for these multiyear storage reservoirs and to derive the marginal value of water in storage. The RLROM model is presented with two solution algorithms: the first algorithm relies on the use of lookup tables to store the Q-values and the second algorithm, which allowed the extension to handle a larger scale multireservoir problem, relies on the use of the function approximation technique.

The RLROM model considers several stochastic variables: the domestic load, market prices, and the natural inflows. The use of the moment matching technique for generating scenarios of the load, inflow and price variables has the advantage of using a limited number of scenarios to represent the random variables' statistical properties, in particular the moments and the correlation of extensive historical time series records.

A large-scale hydroelectric reservoir optimization model (GOM) based on linear programming was integrated with the RLROM model. In this way, the optimization process was extended to include the other reservoirs on the Peace and on the Columbia Rivers: the Dinosaur, the Revelstoke, and the Keenleyside reservoirs. This integration allowed more efficient on-line interaction between the agent and the environment to be carried out. It also made it possible to capture the diurnal variation of the price and load on shorter time periods.

5. MODEL TESTING AND IMPLEMENTATION

5.1. Introduction

This chapter is structured as follows: **First**, a single reservoir optimization problem, representing a test bed for this research work was used to investigate capability of the RL approach to solve the reservoir operation planning problem and to gain experience with the RL technique. Three cases were considered to test the performance of the RL algorithm on problems of increasing size.

Second, a two reservoir problem was tested using the multireservoir model formulation presented in the previous chapter. The objective of this test case was to investigate the potential use of the GOM model off-line as a model of the real system. Lookup tables were used to store the feedback information from GOM and the estimated Q-values from the RL algorithm for a subset of the full state space of the two reservoirs.

Finally, the RLROM model was used to model the full state space and the function approximation RL algorithm was implemented for the BC Hydro two main reservoir systems. The GOM model was linked to run on-line within the RL algorithm. A case study is presented to demonstrate the capability of the model to solve the large-scale multireservoir operation planning problem. As both of the GMS and the Mica dams have multiyear storage capabilities, the model was run for a planning horizon of 36 months.

The optimized storage value function, marginal value of energy for both of the GMS and the Mica dams will be presented and discussed. In addition, examples of the optimal control policies proposed by the RLROM model will be presented. The model output includes the optimized control policies for: market transactions, plant generation, and turbine releases for the five main plants in the Peace and the Columbia River systems.

Initially, the model was run in training mode, where the RL agent learns the action-value function and the optimal control policies. Once the RL agent learns the optimal control policies, the model can then be used to control the monthly operations planning decisions. The use of the RL model to control the operation planning for reservoir systems will be presented and discussed.

5.2. Single Reservoir Model - Test Case

As a first step in investigating the capability of the RL approach to solve the reservoir operation planning problem, a single reservoir case was used as a test-bed. The problem was formulated and solved using the Q-Learning technique. The problem was also solved using the value iteration method of the stochastic dynamic programming (SDP) technique. The established optimized solution using the SDP model was used as a base line to evaluate the RL model results. Working on this problem provided useful insights about the RL approach. In addition, it was possible to gain experience on the sensitivity of the model results to the various parameters used in the formulation. The following sections present a description of the case study, SDP and RL test model formulation, establishing the RL model parameters, and the model results.

5.2.1. Problem Description

The system considered in this case consists of one reservoir. The supply from the reservoir is mainly used for hydropower generation. The hydropower producer operating the system generates the electricity to satisfy the local demand and for sale in the open market. The energy yield, depending on the inflows, is variable from period to period throughout the year and is governed by the following probability distribution:

Table 5.1. Probability of inflow for the different periods

Inflow Volume (Energy units)	Probability											
Period	1	2	3	4	5	6	7	8	9	10	11	12
10	0.10	0.15	0.30	0.25	0.10	0.15	0.30	0.25	0.30	0.25	0.30	0.10
11	0.20	0.25	0.20	0.20	0.20	0.25	0.20	0.20	0.20	0.20	0.20	0.20
12	0.30	0.30	0.10	0.15	0.30	0.30	0.10	0.15	0.10	0.15	0.10	0.25
13	0.25	0.20	0.15	0.25	0.25	0.20	0.15	0.25	0.15	0.25	0.15	0.15
14	0.15	0.10	0.25	0.15	0.15	0.10	0.25	0.15	0.25	0.15	0.25	0.30

The power producer requires 10 units of energy over each period to satisfy its demand and could store as much as 12 units. Any energy stored but not used can be stored or sold in the following period. A marketer is willing to pay a premium price for the producer's energy according to the scale below, if the producer guarantees delivery at a certain time of the year:

Table 5.2. Forward market price

Units of Energy	0	1	2	3	4
Price ($)	0	1200	1000	800	800

If the producer contracts too much of its production, leaving less than 10 units for its own needs, then the shortfall must be made by purchasing energy on the spot market (spot buy) at $1500 per unit. Any energy held at the end of the month, which the storage facilities can't store will be sold on the spot market (spot sell) for $500 per unit. The producer limits transactions on the spot market to those that are absolutely necessary. The objective is to maximize the expected discounted profit over the foreseeable future, with an effective annual interest rate of 7%.

5.2.2. SDP mathematical formulation

The following notation was used in the mathematical formulation of the single reservoir problem:

t Time period, where $t \in 1,\ldots,T$ in month/s,

a Forward sale (contract), where $a \in A$ in units of energy,

s Reservoir storage, where $s \in S$ in units of energy,

i Inflow volume, where $i \in I$ in units of energy,

L Domestic load=10 units of energy,

U Upper storage limit in units of energy,

Spot_Buy/ Spot_Sell Spot market transactions (buy/sell respectively) in unit of energy,

Spot_Buy_Cost/ Spot_Sell_Rev Cost of buying / revenue from selling in $,

Exp_Spot_Buy_Cost/Exp_ Spot_Sell_Rev Expected value of buy cost/sell revenue in $,

Cont_Rev Forward sale (contract) revenue in $,

P_{spot_buy}/ P_{spot_sell} Price of spot buy / sell in $/unit of energy,

$P_f(a)$ Price of forward (contract) sale in $/unit of energy,

R Rewards in *$*,

γ Discount rate,

$TP(s_t,i_t,a,s_{t+1})$ Transition probability of moving to state s_{t+1} for a given state s_t, action a, and inflow i_t

$TPS(s_t,a,s_{t+1})$ Transition probability of moving to state s_{t+1} for a given state s_t and action a, where

$$Exp\_Spot\_Buy\_\mathrm{Re}v(s_t,a) = \sum\nolimits_{s_{t+1}} Spot\_Sell(s_t,i_t,a) * P_{spot-sell} * TP(s_t,i_t,a,s_{t+1})$$

$$TPS(s_t,a,s_{t+1}) = \begin{cases} \sum_{i_t} TP(s_{t+1} - s_t + L + a) & \forall \quad s_{t+1} \geq U \\ TP(s_{t+1} - s_t + L + a) & \forall \quad U > s_{t+1} > 10 \\ \sum_{i_t} TP(s_{t+1} - s_t + L + a) & \forall \quad s_{t+1} \leq 10 \end{cases}$$

and $f_t(s_t)$ is the value function for state s in time period t.

The mathematical formulation of the SDP problem is as follows:

Policy Rewards Calculation:

$$R(s_t,a) = -Exp\_Spot\_Buy\_Cost(s_t,a) + Exp\_Spot\_Sell\_\mathrm{Re}v(s_t,a) + Cont\_\mathrm{Re}v(s_t,a) \quad (5.1)$$

where:

$$Exp\_Spot\_Buy\_Cost\ (s_t,a) = -\sum_{s_{t+1}} Spot\_Buy(s_t,i_t,a) * P_{spot-buy} * TP\ (s_t,i_t,a,s_{t+1}) \quad (5.2)$$

$$Exp\_Spot\_Buy\_\mathrm{Re}v(s_t,a) = \sum_{s_{t+1}} Spot\_Sell(s_t,i_t,a) * P_{spot-sell} * TP(s_t,i_t,a,s_{t+1}) \quad (5.3)$$

$$Contract\_\mathrm{Re}v(s_t,a) = P_f(a) * a \quad (5.4)$$

Constraints:

Constraint on reservoir storage (state variable): $U \geq s_t \geq 10$ (5.5)

Constraint on forward sales (decision variable): $0 \leq a \leq 4$ (5.6)

Spot transactions (sell) constraint: if $s_{t+1} > U$ $Spot\_Sell_t = s_{t+1} - U$ (5.7)

Spot transactions (buy) constraint: if $s_{t+1} < 10$ $Spot\_Buy_t = 10 - s_{t+1}$ (5.8)

Hydraulic continuity constraint: $s_{t+1} = s_t + i_t - L_t - a$ (5.9)

and $U \geq s_{t+1} \geq 10$

Objective Function

$$f_t\ (s_t) = \max_a \left\{ R(s_t, a) + \gamma * \sum\nolimits_{s_{t+1}} [f_{t+1}(s_{t+1}) * TPS(s_t, a, s_{t+1})] \right\} \qquad (5.10)$$

The optimal policy π_t^* is:

$$\pi_t^*(s) = \arg\max_a \left\{ R(s_t, a) + \gamma * \sum\nolimits_{s_{t+1}} [f_{t+1}(s_{t+1}) * TPS(s_t, a, s_{t+1})] \right\} \qquad (5.11)$$

5.2.3. RL mathematical formulation

The same set of constraints, as described above in the SDP formulation was used in the RL model. However, the rewards and the objective function were calculated differently. As described earlier in chapter 3, the RL Q-Learning algorithm relies on sampling the state, action, and random variables to calculate the rewards instead of calculating the expected values of rewards as described above for the case of the SDP algorithm. Accordingly, there was no need to calculate the transition probabilities. Rewards in the RL formulation were calculated as follows:

Rewards Calculation:

$$r_t(s,a) = -Spot\_Buy\_Cost(s_t, i_t, a) + Spot\_Sell\_\mathrm{Re}v(s_t, i_t, a) + Cont\_\mathrm{Re}v(s_t, a) \qquad (5.12)$$

where:

$$Spot\_Buy\_Cost(s_t, i_t, a) = Spot\_Buy(s_t, i_t, a) * P_{spot-buy} \qquad (5.13)$$

$$Spot\_Buy\_\mathrm{Re}v(s_t, i_t, a) = Spot\_Sell(s_t, i_t, a) * P_{spot-sell} \qquad (5.14)$$

$$Contract\_\mathrm{Re}v(s_t, a) = P_f(a) * a \tag{5.15}$$

Objective Function

The Q-Learning update rule was used to calculate the state-action value function (Q-Value):

$$Q_t^N(s_t, a) = Q_t^{N-1}(s_t, a) + \alpha_t \left\{ r_t\ (s_t, a) + \gamma * \max_a Q_{t+1}^{N-1}(s_{t+1}, a)] - Q_t^{N-1}(s_t, a) \right\} \tag{5.16}$$

From this we get:

$$f_t(s_t) = \max Q_t^N(s_t, a)) \tag{5.17}$$

$$\pi_t^*(s_t) \in \arg\max_a Q_t^N(s_t, a) \tag{5.18}$$

where $Q_t^N(s, a)$ is the state-action value function at time period *t* and iteration *N*, *r*(*s*,*a*) is the sampled rewards for state *s* and action *a* as calculated in Equation 5.12.

The problem was formulated in AMPL and was solved using the Q-Learning algorithm described earlier in section 3.4.3.2. For comparative and evaluation purposes, the SDP problem was also solved using the value iteration algorithm.

Three cases were modeled for this test problem, as follows:

Case 1: Considering three states (10-12) and four stages.

Case 2: Considering three states (10-12) and twelve stages.

Case 3: Considering ten states (10-19) and twelve stages.

5.2.4. Solution Steps of Q-Learning Algorithm

The following is a presentation of a step by step procedure of the Q-Learning solution algorithm applied in solving the single reservoir problem:

Step 1 Set the model parameters including: the annual discount rate γ= 0.07, the exploration rate exponent ζ = 0.000045, and the learning rate exponent ψ= 0.61. The ζ and ψ constants are used to calculate the exploitation rate and the learning rate respectively (equations 4.6 and 4.7). Set the number of iterations N_{max} = 10,000.

Step 2 Initialize the state-action value function (Q-Tables) to zero for all state-action pairs, $Q_t(s,a) = 0$.

Step 3 Starting at the first stage; initialize the algorithm by randomly sampling the state space, i.e. randomly choose a point from the discretized reservoir storage (state space), for example $s_1^1 \in \{10,11,...,19\}$.

Step 4 Randomly sample the inflow variable according to the probability distribution in time period one, where $i_1 \in \{10,11,...,14\}$.

Step 5 The agent chooses action a_1 (forward sales or contracts) randomly, where $a_1 \in \{1,...,4\}$. In the first iteration, for all the stages $t \in \{1,...,12\}$ the agent chooses the action at random as it has not yet learned any information about the Q-values.

Step 6 The agent interacts with the model of the environment. The chosen action and the sampled inflow scenario are simulated. The agent receives a signal in the form of the transition to the next stage state s_2^1 based on the hydraulic balance equation (5.9) and a numerical reward $r_1\ (s,a)$ as calculated in equation (5.12). The rewards are calculated as the sum of the spot and contract sales.

Step 7 The learning rate α is calculated using equation (4.7).

Step 8 Apply the Q-Learning update rule (equation 5.16) to calculate a new estimate for the Q-value as follows:

$$Q_t^N(s_t, a) = Q_t^{N-1}(s_t, a) + \alpha_t \left\{ \left[r_t\ (s_t, a) + \gamma * \max_a Q_{t+1}^{N-1}(s_{t+1}, a) \right] - Q_t^{N-1}(s_t, a) \right\}$$

The first term of the equation represents the initial action-value function (Q-value) in this iteration (N=1) and at this time period (t=1), $Q_1^0(s_1, a)$. The second term represents the reinforcement achieved to the initial estimate of the Q-value. This is calculated as the difference between the new Q-value $(\gamma * r_1(s_1, a) + \max_a Q_2^0(s_2, a))$ and the initial estimate of the Q-value $(Q_1^0(s_1, a))$ multiplied by a step size (learning rate, α_t). This equation can be represented in general form as follows:

New Estimate ← *Old Estimate* + *Step Size* [*Target (New Estimate)* - *Old Estimate*]

The *New estimate* explained above is the sum of the rewards achieved in this time period t and the discounted value function at the transition state in time period $t+1$.

Store the *New Estimate* of the Q-values as $Q_1^1(s_1, a)$. At this point, for stage 1 and time period 1 we have a new state-action estimate (Q-Value) for the visited state-action pair. The other state-actions remain unchanged (zero). Next visit to this state-action pair, may be in iteration 50, the agent uses the stored Q-value $Q_1^1(s_1, a)$ as the *Old Estimate* and apply the same update rule to calculate the *New Estimate* $Q_1^{50}(s_1, a)$. The best policy (corresponding to the maximum Q-value) is always updated after each visit to a state and calculating the Q-values for the sampled state-action.

Step 9 The agent moves to the selected state in the next time period, s_2^1. The procedure of choosing random action a_t and determining the reward signal and the transition state is repeated until the agent reaches time period T.

Step 10 The agent starts a new iteration (*N*=2). At the first time period (*t*=1), sample the state space randomly, s_1^2. Calculate the exploitation rate ε applying equation 4.6. The agent chooses the action a_l using the ε-greedy policy. Accordingly, with probability ε, the agent is choosing the best action estimated so far, $\max_a Q_1^2(s_1, a)$. In the beginning, the exploitation rate is small and the agent tends to explore more frequently, with a probability of (1-ε), to gain more knowledge about how good or bad it is at taking the actions. Later on, and as the age of the agent increases, the exploitation rate increases and the agent becomes more greedy and it chooses the best action with probability ε. Steps 3 to 9 are repeated until convergence is achieved. However, starting from iteration 2 and in subsequent iterations, until the termination of the algorithm, the agent is always using the information it has learned so far to update the Q-value function estimates (i.e. reinforce its learning of the Q-values) applying the ε-greedy policy rather than randomly selecting the actions as in the first iteration. The optimal value function *f*(*s*) and the optimal generated policies $\pi_t^*(s) \in \arg\max_{a \in A} Q_t(s, a)$ are stored for all elements of the state space.

Prior to illustration and discussion of the results, the following section presents the process followed in establishing the parameters that were used in the RL algorithm.

5.2.5. Establishing the parameters of the RL Algorithm

In the initial tests of using the Reinforcement Learning technique to solve the single reservoir problem, a series of runs was performed to establish the appropriate settings of the RL algorithm parameters. The following sections present examples of the runs, which were performed for establishing the exploitation rate and the learning rate RL parameters. Case 2 (considering 3 states and 12 stages) was selected as an example to present the results for a fixed number of iterations $N = 8000$ in all runs.

5.2.5.1. Exploitation rate

The effect of changes in the exploitation rate parameter was investigated first using a constant exploitation rate parameter ε ranging from 0.2 to 1.0. For the rest of the cases a variable exploitation rate was used. The variable rate was expressed as a function of the age of the agent as outlined in Equation 4.6 with the value of parameter ζ ranging from 0.00001 to 0.0001. For both cases (constant and variable exploitation rate) a learning rate parameter $\psi = 0.5$ (Equation 4.7) was used. Figure 5.1 presents the results for this set of runs for both constant and variable exploitation rates. The maximum error in the value function is presented as a percentage of the maximum difference from the solution derived by the SDP model. The results of these runs indicate that varying the exploitation rate with time appears to provide the best performance.

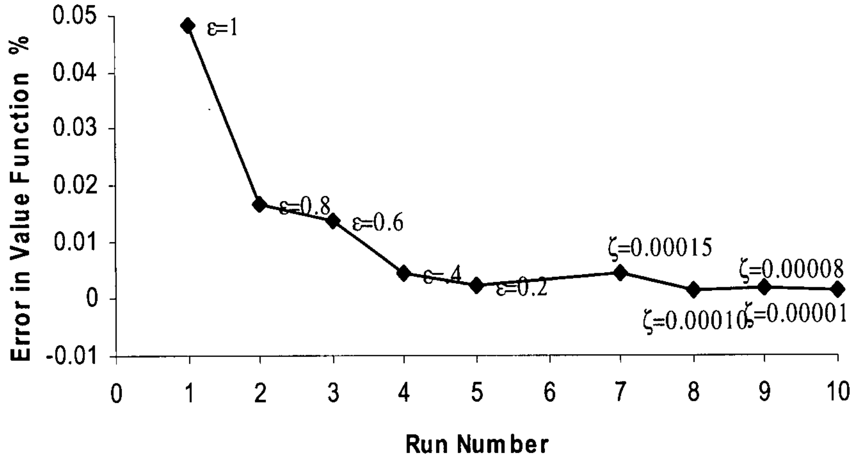


Figure 5.1. **Effect of varying the exploitation rate**

Examining the convergence behavior of the solution (as an example, at time period t=5 and state s=11), Figure 5.2 shows that the expected rewards increase with increasing experience of the learning agent. At the beginning of the iterations, the estimated value function increases significantly. However, the rate of increase gradually decreases until it diminishes, as the algorithm converges to the optimal solution. It is clear that when a greedy policy is consistently followed, it results in a poor quality solution, compared with other ε-greedy policies (using either a constant or variable exploitation rate with ε smaller than 1). Following the greedy policy, the value function starts to increase and levels off earlier and at a lower level compared to other policies (at about 0.95 of the optimal solution). On the other hand, the ε-greedy method is capable of reaching close to the exact solution. The ε-greedy method performs better as the agent continues to choose the action randomly at a small fraction of time and therefore it has a better chance to find a better approximation to the optimal solution. Figure 5.3 illustrates the robustness of the Q-Learning method over a wide range of values for the exploitation rates parameter ζ.

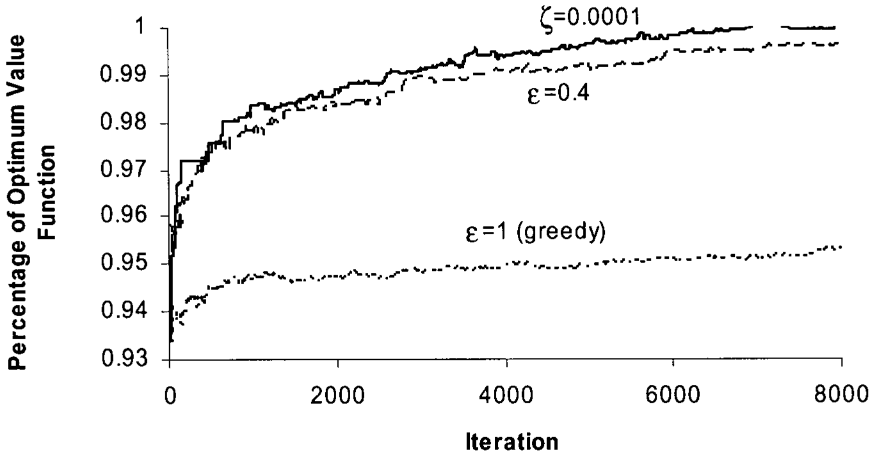


Figure 5.2. Convergence of Q-Learning solution applying different exploration-exploitation policies

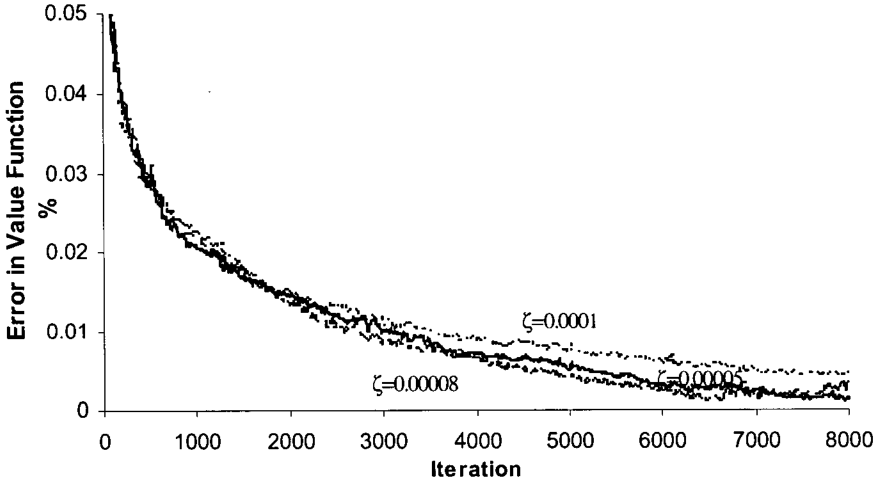


Figure 5.3. **Performance of different ε-greedy policies**

5.2.5.2. Learning rate

To establish the best value for the learning rate parameter, and to determine the sensitivity of the solution to the variation in the learning rate, sets of runs were performed using constant learning rates α of 0.15 and 1.0. Also, a polynomial learning rate was used (Equation 4.7) with the exponent ψ ranging in value from 0.5 to 1.0. The same exploitation rate was used throughout the runs assuming ζ= 0.0001. Figure 5.4 demonstrates that by gradually reducing the learning rate as the number of iterations increases, the model convergence to a better quality solution (lower error in the value function). The best results were obtained by setting ψ=0.6 and the accuracy of the solution deteriorates if ψ is either increased or decreased. Figure 5.4 shows that the error in the value function could have a V shape relationship with the learning rate, which stresses the importance of experimentation with these parameters prior to the adoption of the values for use in the RL algorithms.

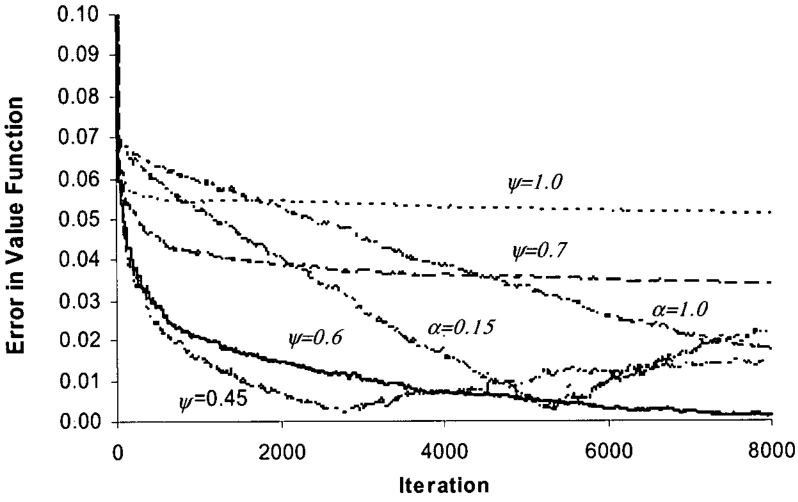


Figure 5.4. Performance of different learning rates

5.2.6. Results

To examine the quality of the solution obtained using the RL approach for the single reservoir problem, the results of the three test cases described earlier were evaluated. First, the RL model parameters for each of the three test cases were established using a variable exploitation rate parameter ζ and learning rate parameter ψ. The following table presents the parameters and the number of iterations used for each of the test cases.

Table 5.3. Test cases study parameters

Test Case	Iterations	Exploitation rate parameter	Learning rate parameter
		(ζ)	(ψ)
Case 1	15,000	0.000065	0.59
Case 2	15,000	0.000050	0.58
Case 3	20,000	0.000045	0.61

The convergence of the value function is presented in Figure 5.5. The results indicate that for all 3 test cases, the RL model was capable of converging to an optimal solution with a very little difference from the solution derived by the SDP model. The error in the value function is defined as:

$$Error = \max\left|f_t^*(s) - f_t(s)\right| \quad (5.19)$$

where $f^*(s)$ is the value function obtained using the SDP model.

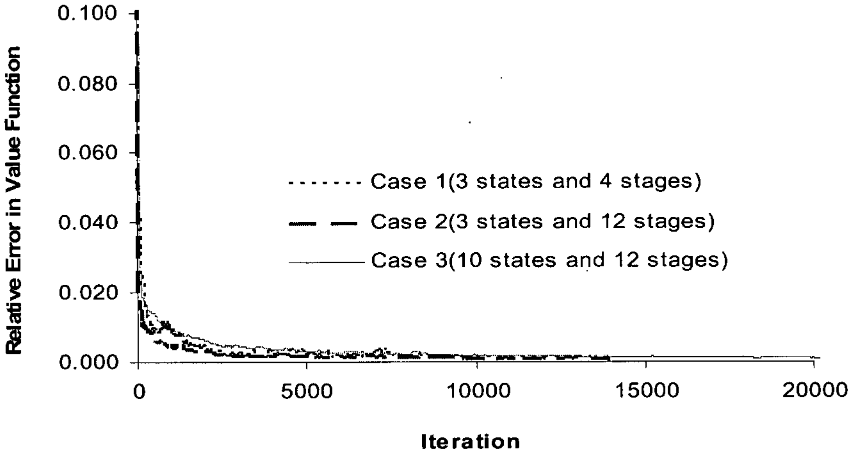


Figure 5.5. Convergence of the value function

The maximum relative error in the value function derived by the RL model for each of the three cases was: 0.0009, 0.0005, and 0.0009 respectively. For each stage, the mean relative error (*MRE*) in the RL solution was calculated and listed in Table 5.4.

The *MRE* was calculated as:

$$MRE_t = \frac{1}{n_j}\sum_1^{n_j}\frac{\left|f_t^*(s) - f_t(s)\right|}{f_t^*(s)} \quad (5.20)$$

The results indicate that the mean relative error does not exceed 0.04%.

Table 5.4. Mean relative error (%) for the single reservoir RL model

Period	1	2	3	4	5	6	7	8	9	10	11	12
Case 1	0.03	0.04	0.03	0.04	-	-	-	-	-	-	-	-
Case 2	0.02	0.02	0.03	0.02	0.03	0.03	0.01	0.03	0.02	0.04	0.03	0.02
Case 3	0.03	0.02	0.03	0.03	0.02	0.02	0.03	0.03	0.03	0.03	0.03	0.04

The optimal value function derived by RL Q-Learning and SDP models were superimposed and compared. Figure 5.6 presents the study results for stages 6 and 12 of the single reservoir test case 3. The results indicate a close match of the value function obtained by the RL and the SDP models.

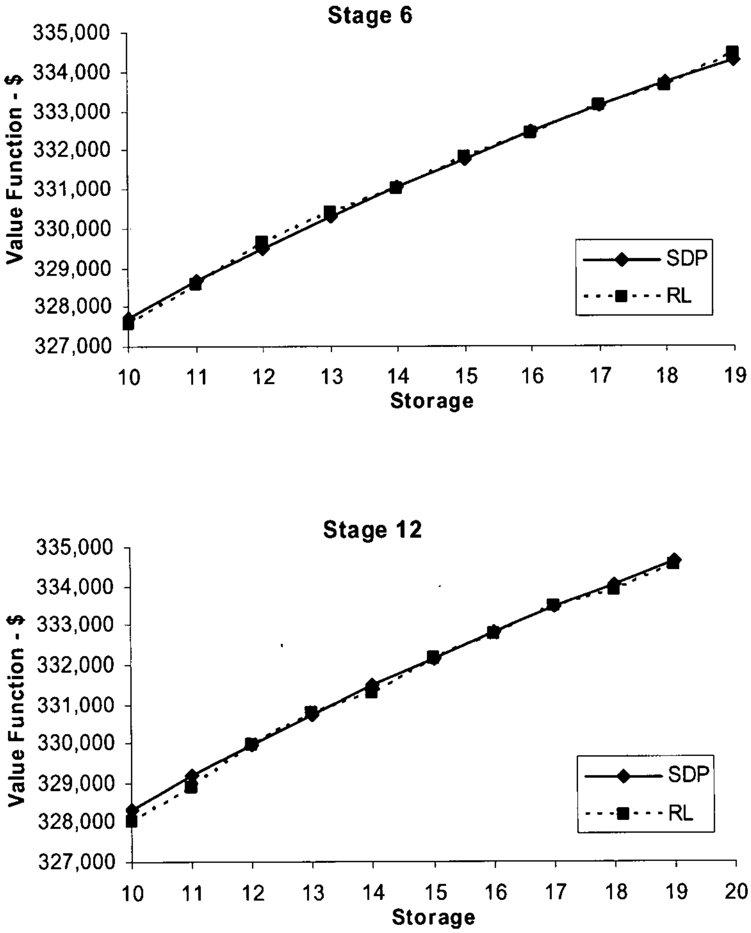
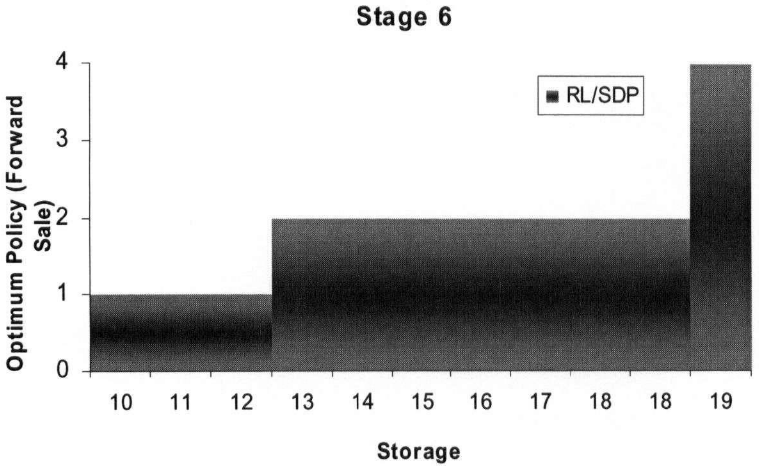


Figure 5.6. Value function obtained by RL and SDP at time periods 6 and 12

Figure 5.7 displays the optimal forward sale policies derived by both of the RL and SDP models for stage 6 and stage 12. The operating policies suggested by the SDP and RL model were identical. As the results presented suggest, it is more profitable to contract more energy, as the storage in the reservoir increases.



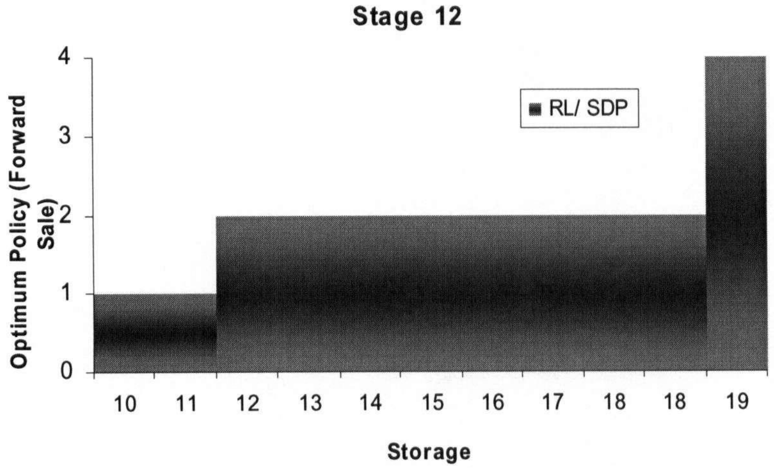


Figure 5.7. Optimal policies derived by the RL and SDP models at time periods 6 and 12

The results obtained indicate that for all of the three test cases the performance of the RL is stable and provides a good approximation to the optimal solution. The results also, interestingly, show that the number of iterations required to reach convergence are not affected by the size of the state space. This also provides a good measure of the robustness of the RL approach. Consequently, the single reservoir test case offers some useful insights on the potential use of the RL algorithms in handling the larger scale multireservoir problem, which will be discussed next.

5.3. Two Reservoir Model - Test Case

This test case was used as a building block for the development and the implementation of approach to the large-scale multireservoir problem. This case study provided a way to test the performance of the algorithm in handling larger scale problems. The RL Q-Learning model was run for a finite number of iterations until it converged. Convergence was assessed in terms of the difference between the current solution and the previous iteration solution, as suggested by Gosavi, 2003. The derived RL model optimal solutions were then compared with the SDP model results.

5.3.1. System Modeling

The system considered in this test case consisted of the GMS generating plant at the Williston reservoir and the MCA generating plant at the Kinbasket reservoir on the Peace and Columbia Rivers respectively. The state variables include a subset of the full GMS and MCA storage volumes. The GMS storage state variable was discretized to 30 increments covering the range from 260,000 to 410,000 cms-day, and 5 increments for Mica ranging from 245,000 to 285,000 cms-day. Accordingly, the state space for each time period consists of 150 storage combinations for both reservoirs. In addition, a provision was made in the formulation to account for the system electricity demand in the state space. Table 5.5 presents the forecasted system electricity load and the peak load in the four time periods considered in the model runs, each consisting of three months.

Table 5.5. System load and peak demand

Period	**1**	**2**	**3**	**4**
Load - MWh	4,983	5,362	5,804	5,872
Peak - MW	8,820	9,720	10,458	10,219

To test the performance of the proposed RL multireservoir model, the historical inflow data for water years 1964 -1968 were used in five scenarios of the random inflow variable. The monthly inflow data to the reservoirs on the Peace and the Columbia system and the assumed scenario probability associated with each of the inflow sequences are presented in Table 5.6:

Table 5.6. Peace system inflow scenarios (cms)

Scenario	Prob.	Peace				Columbia			
ω		Period							
		1	2	3	4	1	2	3	4
1	0.10	1180	787	354	326	1000	570	308	272
2	0.20	1100	644	278	258	673	582	330	267
3	0.30	788	551	358	307	694	437	357	288
4	0.25	1150	579	275	499	759	501	291	285
5	0.15	823	692	342	208	718	492	312	264

Monthly price forecasts for the US and Alberta markets were used to represent the market conditions in the northwest. Table 5.7 presents the heavy load hours (HLH) and light load hours (LLH) price forecast corresponding to the five inflow water years.

Table 5.7. US and Alberta market price forecast ($/MWh)

ω	P_{US}^{ω}		P_{AB}^{ω}	
	HLH	LLH	HLH	LLH
1	32.78	32.93	34.53	34.69
2	41.41	40.61	43.62	42.78
3	36.08	36.35	38.01	38.29
4	39.20	38.97	41.30	41.06
5	34.18	34.53	36.01	36.37

The range of the forward transactions decision variable was discretized to 5 decisions. Each decision was distributed between the heavy load hours and light load hours.

Table 5.8. Range of forward transactions

Period		**1**	**2**	**3**	**4**
LLH-IMP	Minimum	57	32	48	48
GWh	Maximum	287	158	239	239
HLH-EXP	Minimum	97	107	84	111
GWh	Maximum	484	535	419	553

where IMP and EXP are the imports and exports respectively.

5.3.2. Results and Analysis

The problem formulation was described in detail in the previous chapter. The RL model was run on a three month time step for four time periods. To simulate the behavior of the real system under different inflow and release conditions, the GOM model was used off-line in batch mode prior to performing the RL model runs. The GOM model was run on a daily time step with three sub-time steps during the weekdays and one time step during the weekend. The purpose of running GOM on a finer time step than the RL model was to capture variations in the load and prices that occur in a typical day. First, a batch of 15,000 runs was carried out to cover all possible combinations of: the storage state variable, inflow, and forward sale (decision variable) for the four time periods (30*5*5*5*4=15000). For each run, the GOM model generates the optimal system rewards and the transition state, which are then stored in lookup tables for use in the RL model runs.

Values for the RL model parameters of $\psi = 0.5$ and $\zeta = 0.000025$ were adopted for the model runs. The model was set to run until it converged, which was achieved after 80,000 iterations as shown in Figure 5.8.

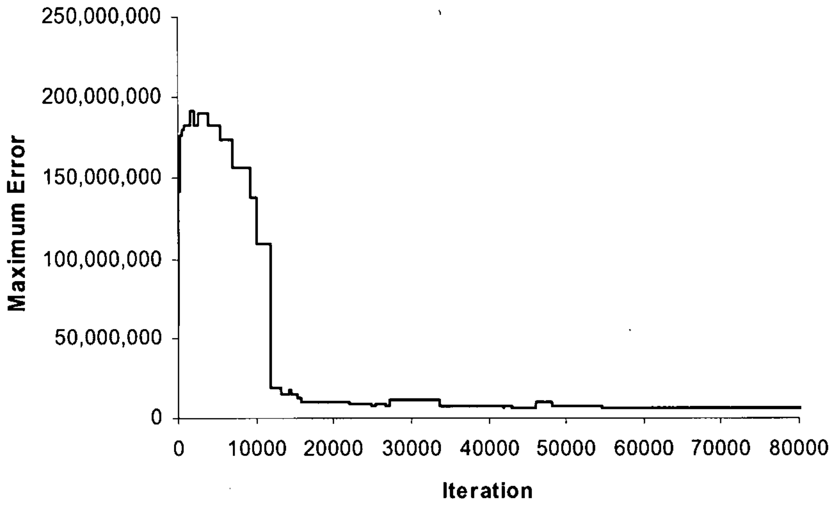


Figure 5.8. Convergence of the RL model

The RL model results were compared to the optimal solution derived by the SDP model. Figures 5.9 presents a sample of the optimal value functions estimated by the RL and the SDP models. The storage value function for different GMS storage levels is given on the abscissa for five MCA storage levels. The graph demonstrates the capability of the RL model to converge to near optimal solution with a mean relative error not exceeding 0.0256. The results also indicate the stability of the model for the different time periods. The MRE for each of the four stages is given in the following table:

Table 5.9. Percentage of Mean relative error for the two reservoirs RL model

Period	1	2	3	4
M.R.E (%)	2.35	2.56	2.44	2.37

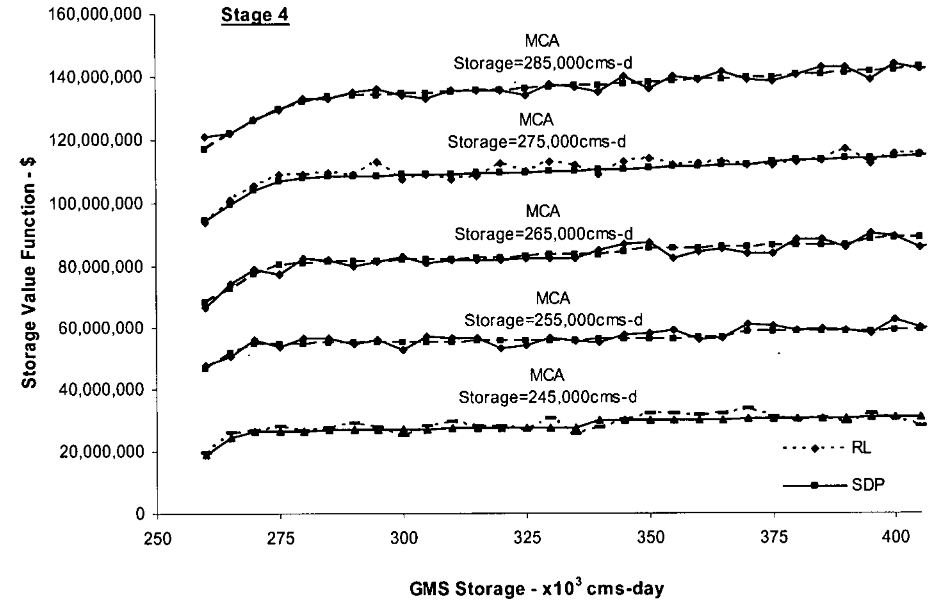


Figure 5.9. GMS storage value function as a function of MCA storage

The optimal operating policy obtained by the RL model was also evaluated. Assuming different initial storage levels for GMS and MCA and for several inflow scenarios, a number of simulation runs were carried out using the optimal policies at each stage for both the RL and the SDP model. For the different inflow scenarios and starting conditions, the simulation results were found to be identical to those derived by the SDP model. Two examples of the simulation run results are illustrated in Figure 5.10.

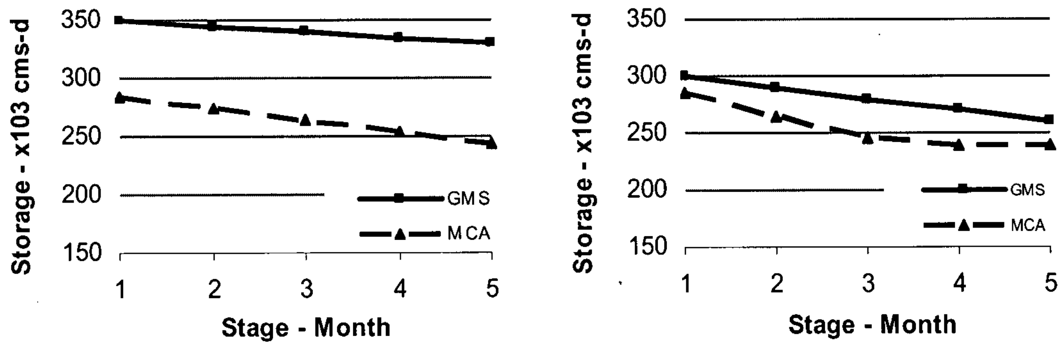


Figure 5.10. Simulated optimal operation planning policy

Results from the two reservoir test model clearly demonstrated the capability of the RL model in solving the problem and obtaining a good approximation of the optimal value function. Using the GOM model, in batch mode has the merit of providing the feedback information needed for the RL model in an off line setting. This resulted in a speed up of the RL model runs. However, some infeasibility difficulties were encountered with this mode of GOM runs, as the model could not always find the optimal solution at the grid points for the storage state space as it had a coarse grid structure. This problem has occurred since the feasible optimal solution may exist at any point in the state space that is not necessarily at an exact point of the coarse grid. Consequently, there was a need to use a finer grid structure. For the state space used in this study, a batch of 15,000 GOM runs was performed. To cover the whole state space, the amount of data management and CPU time would increase significantly and it would become impractical for use in this study. This problem was dealt with during the implementation phase of the RLROM as will be described in section 5.4, through the use of function approximation instead of lookup tables.

5.3.3. Efficiency of the RL Algorithm

A series of runs were conducted to assess the efficiency of the RL model in terms of the computational speed. The run time (CPU time) of the RL model was compared to that of the SDP model. Assuming *s* state discretization, *t* time periods, *a* decision variables, *e* scenarios of random variables, then the problem size becomes (*s.t.a.e*). The two models were run for different problem sizes ranging from 900 to 178200. Figure 5.11 displays the variation in computational time needed for the RL and the SDP models with problem size. The figure clearly demonstrates that the SDP model run time appears to be increasing quadratically as the size of the problem increases whereas the RL model has a linear run time relationship with the problem size. These results suggest that the RL model is more efficient in terms of computational time as the size of the problem increases larger than 150000.

If the SDP were to be used to solve the two reservoir problem covering: the whole state space assuming 59000 states, 36 time periods, 10 scenarios, and 5 actions, the problem size is in the order of 10^8. Using a polynomial regression relationship established from the results presented in Figure 5.11 indicates that the estimated time for SDP to solve a problem of 10^6 in size would be impractical. It can be concluded from this analysis that the RL overcomes the dimensionality problem through the potential use of sampling and function approximation techniques as will be presented next.

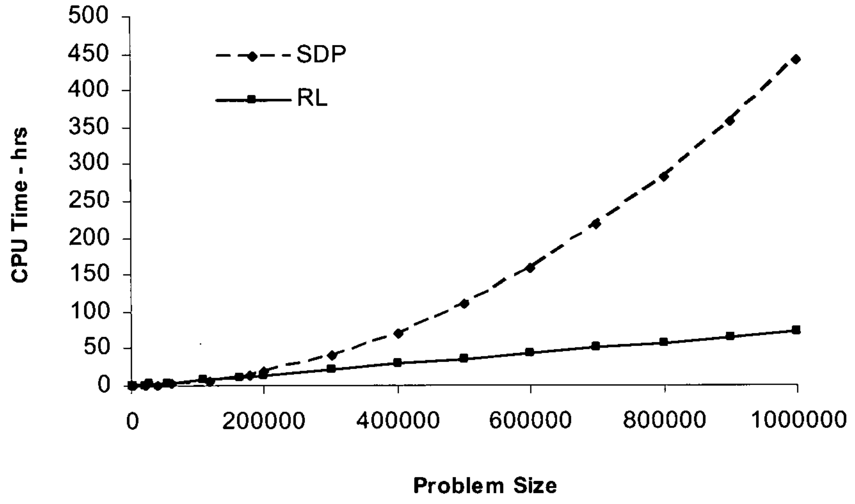


Figure 5.11. Comparison of the RL and SDP CPU time

5.4. RLROM Multireservoir Model Implementation - Case Study

The storage and generation facilities on the Peace and Columbia Rivers dominate the system operation planning function at BC Hydro. It is well recognized by the system operation planners that proper planning of the system should consider the coordinated operation of these two systems, and that they should be integrated together in one model. This section presents the results of implementation of the RLROM model on BC Hydro's main reservoir systems on the Peace and the Columbia Rivers while addressing the uncertainties that are usually encountered by the system operator, namely: the natural inflow, the system electricity load, and the market price. A case study will be presented to illustrate the capability of the RLROM model to provide answers to this complex operation planning problem. The main outcomes of the RLROM model are: the optimal control planning policy for the system, the estimated value of water in storage, and the marginal value of water in the main reservoirs of the BC Hydro system. The capabilities of the RL model to handle the dimensionality problem are also demonstrated.

Several enhancements were made to the RL model and to the approach that was presented in the previous section. The moment matching scenario generation methodology described in the previous chapter was used to generate a finite set of scenarios that portrays the properties of the random variables. To deal with the larger state space, the RL algorithm with function approximation was used instead of lookup tables. The storage value function will be represented by a continuous piecewise linear function, $f(s')$. In addition, the formulation of the RLROM model was modified to allow for on-line interaction with the GOM model at each stage in the iterations. This setup has provided the GOM model with the flexibility to find the optimal solution at any point on a continuous state space. In addition, this setup has avoided the infeasibility problems that were encountered in the GOM batch runs using the grid structure of the state space as described in section 5.3.

5.4.1. Description of the Case Study

This case study considers the operation planning of BC Hydro's hydroelectric generating plants on the Peace River and on the main stem of the Columbia River. These two systems account for about 65% of the total BC Hydro generation capacity of about 11,000 MW. The production schedules for all other hydro projects were fixed at their normal operation. The most significant of these resources include hydro projects on the Bridge River, Campbell River, and other small hydro systems and thermal generation plants such as Burrard and Island Cogeneration plant (ICG).

The hydroelectric system on the Peace River is located in the northern interior of British Columbia and it consists of: (1) The W.A.C. Bennett dam (Williston Reservoir) and the G.M. Shrum generating station (GMS) and (2) The Peace Canyon dam (Dinosaur Reservoir) and the Peace Canyon generating station (PCN). The Peace system supplies approximately one third of BC Hydro's annual energy production.

The powerhouse of the GMS generating station is equipped with 10 generating units with a total capacity of 2,730 megawatts (MW). The average annual inflow to the Williston reservoir is about 1,080 m^3/s and its drainage area is about 68,900 km^2. Williston reservoir is a multiyear storage reservoir and is the largest in the Province, with a live storage capacity of 39,471 million m^3. The reservoir reaches its lowest operational level (645 m) in April and May and its maximum water level (672 m) in September and October.

The Peace Canyon dam is located 23 kilometers downstream of the Bennett dam with a generating capacity of 694 MW with 4 generating units. The Dinosaur reservoir inundates 9 km^2 with a very small local natural inflow and is normally operated between elevations 500 m and 502.9 m. The GMS and PCN generating stations are typically operated in a hydraulic balance and the flow is tightly controlled during the winter season to manage downstream ice conditions on the Peace River.

The Columbia basin is situated in the southern interior of British Columbia. The main hydroelectric facilities on the Columbia River comprise: (1) Mica dam (Kinbasket reservoir) and Mica generating station (MCA), (2) Revelstoke dam and reservoir and the Revelstoke generating station (REV), and the Hugh Keenleyside dam (Arrow lakes reservoir) and Arrow lakes generating station (KNA).

Mica and the Hugh Keenleyside projects were constructed under the Columbia River Treaty and are operated to maximize the mutual benefits to Canada and the US, with respect to flood control and power generation. The Mica generating station consists of 4 generating units with a generating capacity of 1805 MW. The Kinbasket reservoir, which is the second largest multiyear storage reservoir within BC Hydro system, has a live storage capacity of 14,800 million m^3. It receives an average annual inflow of 586 m^3/s from a drainage basin of 21,000 km^2. The Revelstoke dam is located about 130 km south of the Mica Dam. The Revelstoke generating station also has 4 generating units with a total generating capacity of 2000 MW. The drainage area upstream of the Revelstoke dam is about 5,500 km^2, and the average annual local inflow is 221 m^3/s, which is predominantly snowmelt driven. The maximum operating elevation is 573.0 m with a normal daily fluctuation of about 1.0 m. The Arrow lakes are about 230 km long downstream of the Revelstoke Dam. Their drainage basin is about 10,000 km^2. The inflows to the Arrow lakes are regulated by the Mica Dam. Arrow lakes generating station has 2 generating units with a capacity of 185 MW.

Throughout this section, the Peace and the Columbia main five reservoirs/dams/ generating stations will be referred to as: GMS, PCN, MCA, REV, and KNA.

5.4.2. Input Data

This section presents a summary of the historical/ forecasted data that were used as inputs in the case study. Also, the physical and the operating limits for the five reservoirs considered in this case study are outlined. The case study is set to run on a monthly time step, which was then subdivided to three sub-time steps for weekdays (WK) and one time step for the weekend (WE). Weekends include Sundays and holidays. The weekday time steps were split to: peak (WKPK), high (WKHI), and the low (WKLO) sub-time steps with 2, 14, and 8 hours respectively. The subdivision into sub-time steps enhances representation of the hourly variation in system load and market prices. Table 5.10 presents the monthly hours in each sub-time step for typical months in a non-leap year.

Table 5.10. Monthly Hours in each sub-time step

Month	Sub-time step				Total
	WKPK	WKHI	WKLO	WE	
October	52	364	208	120	744
November	48	336	192	144	720
December	50	350	200	144	744
January	52	364	208	120	744
February	48	336	192	96	672
March	52	364	208	120	744
April	48	336	192	144	720
May	50	350	200	144	744
June	52	364	208	96	720
July	52	364	208	120	744
August	50	350	200	144	744
September	50	350	200	120	720

5.4.2.1. Inflows

The characteristics of the local natural inflows to the five reservoirs on the Peace and Columbia Rivers are given in Table 5.11. The historical records considered covers 60 water years for the period of 1940-2000. Figure 5.12 presents the average monthly inflow and the cumulative monthly inflows. The inflows, which mainly result from snowmelt, typically increase in April and May, peak in June and July, and taper off in January. Figure 5.13 presents the normalized cumulative distribution of the average monthly inflows for the five reservoirs in this study.

Table 5.11. Annual average reservoir local inflow characteristics in cms

River	Reservoir	Dam		Average	Minimum	Maximum	Std.Dev.
Peace	Williston	G.M. Shrum	GMS	12,940	9,158	17,197	1,750
	Dinosaur	Peace Canyon	PCN	321	1	1,618	353
Columbia	Kinbasket	Mica	MCA	6,891	5,578	8,726	678
	Revelstoke	Revelstoke	REV	2,821	2,112	4,030	379
	Arrow	Keenleyside	KNA	4,256	2,141	5,518	834

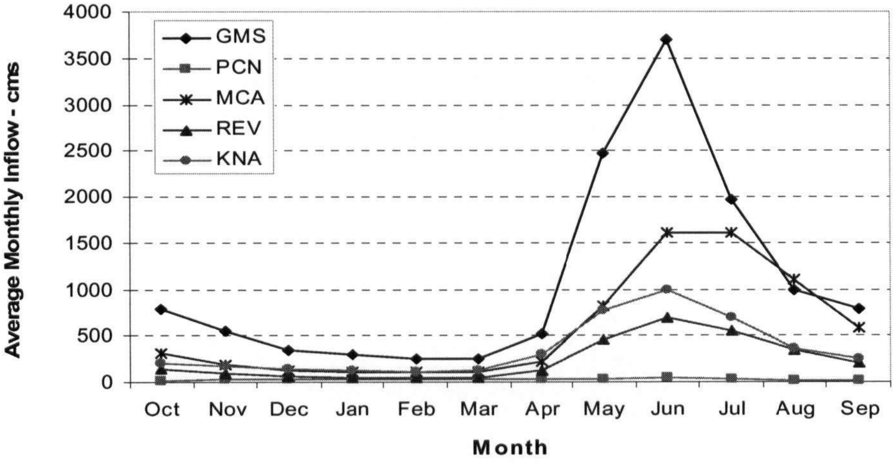


Figure 5.12. Average monthly reservoirs local natural inflows

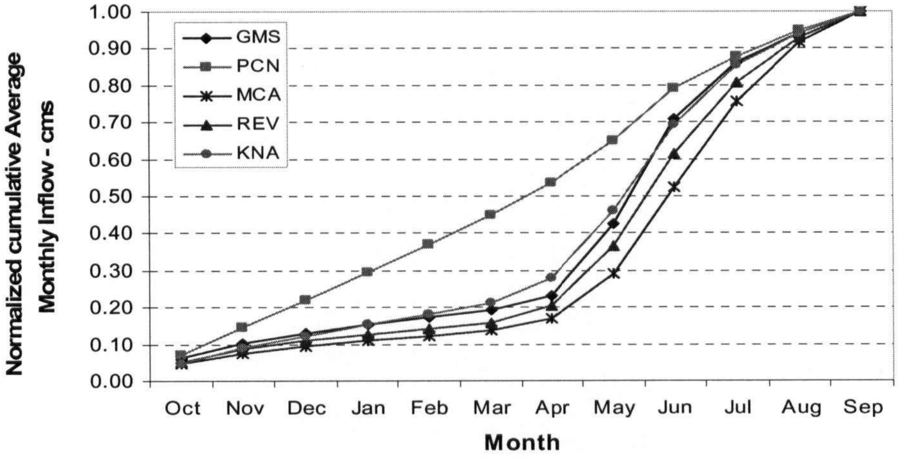


Figure 5.13. Normalized cumulative monthly distribution of the average annual reservoirs local natural inflows

5.4.2.2. Market Prices

Table 5.12 and Figure 5.14 present the average monthly market price forecast for each sub-time step. It was assumed that the peak and the high weekday prices are equal and that the weekend market prices were equal to the (WKLO) weekday sub-time step.

Table 5.12. Average monthly market price in each sub-time steps ($/MWh)

Month	WKPK	WKHI	WKLO	WE
October	54.23	54.23	52.39	52.39
November	58.95	58.95	54.92	54.92
December	63.02	63.02	49.57	49.57
January	66.13	66.13	53.08	53.08
February	56.43	56.43	51.93	51.93
March	54.86	54.86	50.60	50.60
April	54.17	54.17	46.11	46.11
May	50.78	50.78	36.54	36.54
June	50.12	50.12	36.83	36.83
July	54.63	54.63	43.20	43.20
August	75.13	75.13	45.76	45.76
September	56.89	56.89	50.51	50.51
Minimum	50.12	50.12	36.54	36.54
Average	57.94	57.94	47.62	47.62
Maximum	75.13	75.13	54.92	54.92

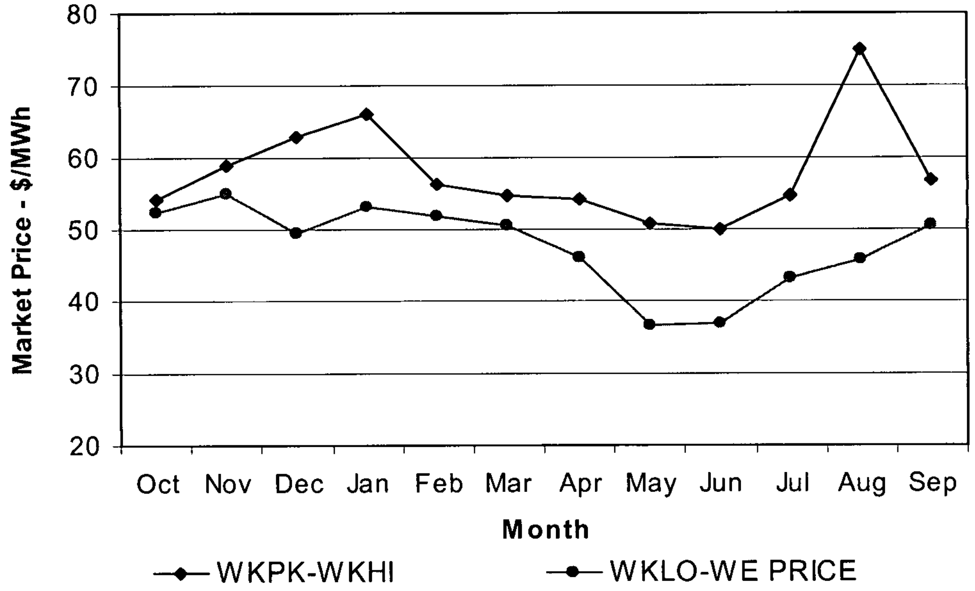


Figure 5.14. Average monthly market price in each sub-time step

5.4.2.3. Electricity System Load

The average monthly electricity load forecast considered in the case study is presented in Table 5.13. This electricity load should be met mainly by the generation from the five hydropower plants on the Peace and Columbia Rivers and other system resources. The other system resources were considered fixed; however, they were shaped according to the historical operation pattern. Figure 5.15 displays the average monthly percentage of the annual load, based on 15 years of BC Hydro historical records.

Table 5.13. Average monthly electricity system load in each sub-time step

Month	WKPK	WKHI	WKLO	WE	Total (GWh)
October	339	2,374	1,357	783	4,853
November	348	2,437	1,393	1,044	5,222
December	380	2,659	1,519	1,094	5,652
January	400	2,798	1,599	922	5,719
February	366	2,562	1,464	732	5,124
March	369	2,583	1,476	852	5,280
April	310	2,168	1,239	929	4,645
May	304	2,128	1,216	875	4,523
June	313	2,189	1,251	577	4,330
July	312	2,184	1,248	720	4,464
August	305	2,135	1,220	879	4,539
September	307	2,148	1,227	736	4,418

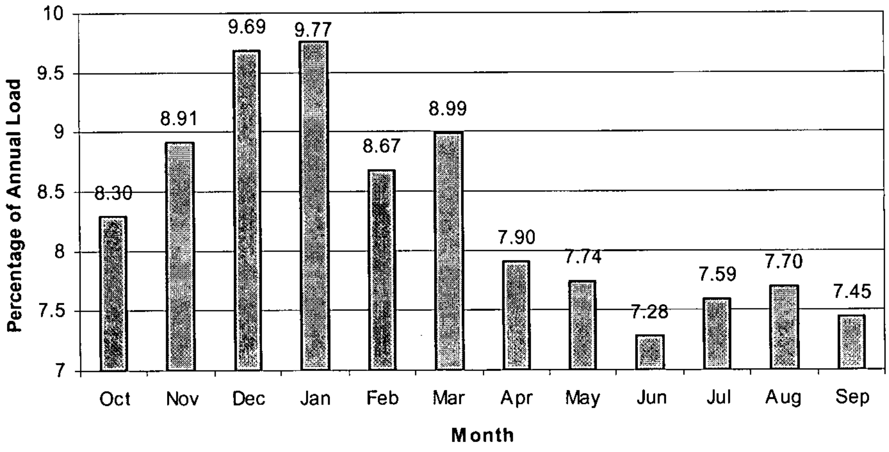


Figure 5.15. Historical monthly distribution of % of system load

5.4.2.4. Transmission limits

The monthly transmission limits used in the study for the imports and exports to the US and Alberta markets are given in Tables 5.14 and 5.15. These limits represent the capacity of the transmission lines between BC and the US and Alberta based on historical records.

Table 5.14. BC-US transmission limits

Month	Import (MWh)				Export (MWh)			
Month	WKPK	WKHI	WKLO	WE	WKPK	WKHI	WKLO	WE
October	1,800	1,800	1,800	1,800	2,600	2,600	2,600	2,600
November	1,500	1,500	1,500	1,500	2,600	2,600	2,600	2,600
December	1,500	1,500	1,500	1,500	2,600	2,600	2,600	2,600
January	1,500	1,500	1,500	1,500	2,600	2,600	2,600	2,600
February	1,500	1,500	1,500	1,500	2,600	2,600	2,600	2,600
March	1,500	1,500	1,500	1,500	2,300	2,300	2,300	2,300
April	1,800	1,800	1,800	1,800	2,300	2,300	2,300	2,300
May	1,800	1,800	1,800	1,800	2,300	2,300	2,300	2,300
June	1,800	1,800	1,800	1,800	2,300	2,300	2,300	2,300
July	1,926	1,949	1,957	1,960	2,600	2,600	2,600	2,600
August	1,926	1,948	1,956	1,931	2,600	2,600	2,600	2,600
September	1,926	1,942	1,948	1,930	2,600	2,600	2,600	2,600

Table 5.15. BC-Alberta transmission limits

Month	Import (MWh)				Export (MWh)			
	WKPK	WKHI	WKLO	WEHI	WKPK	WKHI	WKLO	WEHI
October	560	560	560	560	700	732	744	700
November	640	640	640	640	725	741	747	725
December	560	560	560	560	725	757	769	725
January	560	560	560	560	750	764	769	767
February	560	560	560	560	725	757	769	725
March	640	640	640	640	725	741	747	725
April	560	560	560	560	560	586	595	569
May	560	560	560	560	560	573	578	562
June	560	560	560	560	560	586	595	560
July	435	435	435	435	560	582	590	587
August	435	435	435	435	580	606	615	584
September	435	435	435	435	560	586	595	565

5.4.2.5. Forward Sales

Based on historical records for the period of January 2000 to June 2005, a range of the pre-sales in the US and Alberta markets was used in the case study. Table 5.16 lists the minimum and maximum monthly forward sales for each sub-time step. The negative sign means pre-export and the positive sign means pre-import.

Table 5.16. Forward transactions (Pre-export and Pre-import)

Month	Minimum (GWh)				Maximum (GWh)			
	WKPK	WKHI	WKLO	WE	WKPK	WKHI	WKLO	WE
October	71	498	169	105	-50	-353	-27	-17
November	42	293	134	84	-57	-396	-73	-46
December	36	249	68	42	-41	-285	-98	-61
January	40	281	97	61	-8	-59	10	6
February	51	359	118	74	-1	-10	2	1
March	86	601	179	112	63	441	116	73
April	80	560	133	83	20	141	73	46
May	82	574	176	110	16	109	95	60
June	95	663	199	124	12	83	42	26
July	31	214	143	90	-45	-315	-118	-74
August	16	115	148	93	-66	-460	-36	-22
September	82	571	205	128	-77	-539	-67	-42

5.4.2.6. Operation Constraints

The operation and the physical limits related to reservoirs storage, turbine discharge, plant discharge, and plants generation are given in the Table 5.17.

Table 5.17. **Reservoirs operation and physical limits**

		GMS	PCN	MCA	REV	KNA
Storage - cms-d	Max	475,000	2,695	285,000	60,500	104,825
	Min	55,000	2,440	125,000	58,500	7,850
Turbine Discharge - cms	Max	1,858	2,025	1,140	1,743	845
	Min	0	0	0	0	0
Plant Discharge- cms	Max	1,959	1,982			
	Min	43	283*			
Generation - MW	Max	2,730	700	1,800	2,000	190
	Min	58	50	0	0	0

* PCN minimum ice flow releases are 1100cms in December and 1500 cms in January.

To represent the Columbia River Treaty operation, the monthly average releases from KNA were used in this case study as listed in Table 5.18.

Table 5.18. **Average outflow releases from Keenleyside**

Month	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep
KNA Outflow -cms	1101	1410	1664	1812	1226	761	712	718	830	1678	2159	1742

5.4.3. Scenario Generation

5.4.3.1. Inflow, Price, and Load Scenarios

The moment matching scenario generation technique was used to develop a number of scenarios that represent the same statistical properties of the historical data records for the inflow, price, and load variables. Based on 60 years of records, the first four moments (mean, standard deviation, skewness, and kurtosis) and the correlation of the monthly inflow for the Peace and Columbia Rivers were estimated. The Peace inflows represent the sum of the GMS and the PCN local inflows. The Columbia inflows represent the aggregated local inflows of MCA, REV, and KNA. The statistics presented in Tables 5.19 and 5.20 define the properties of the Peace and the Columbia inflows 24 random variables.

Table 5.19. Monthly Peace system inflow statistics

Statistics		Peace Inflow											
		Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep
Mean - cms	μ	817	570	375	324	279	270	545	2512	3751	2002	1012	804
SD -cms	σ	243	179	95	93	71	72	235	685	906	613	337	234
Skew	γ	0.24	0.54	-0.02	0.822	0.342	0.67	1.08	0.22	0.76	0.36	1.25	0.30
Kurtosis	κ	1.56	1.80	1.50	2.18	1.62	1.94	2.67	1.55	2.07	1.63	3.06	1.59

Table 5.20. Monthly Columbia system inflow statistics

		Columbia Inflow											
		Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep
Mean –cms	μ	659	474	337	285	264	283	637	2044	3298	2845	1811	1030
SD - cms	σ	172	144	80	69	69	80	190	490	645	606	340	234
Skew	γ	1.20	0.74	0.39	0.71	1.23	0.72	0.28	0.39	0.73	0.14	1.19	1.65
Kurtosis	κ	2.93	2.05	1.65	2.00	3.00	2.01	1.58	1.66	2.03	1.52	2.90	4.23

A correlation analysis between the different variables was also carried out for the moment matching method. However, the results indicate a weak correlation between the Peace and Columbia inflows, and this weak correlation protects against drought conditions at the system level.

Operations planners at BC Hydro carried out a regression analysis to establish a relationship to relate the electricity prices in the heavy load hour (HLH) and light load hour (LLH) prices with the annual inflow volume at the Dalles dam (located on the Columbia River in the US). This relationship was then expressed in the form of price multipliers to inflate or deflate the average electricity market price forecast. Therefore, an additional random variable was added to the moment matching algorithm to generate scenarios representing the Dalles inflow volumes, and a set of the HLH and LLH price multipliers were then used to generate the market price scenarios. Table 5.21 lists the statistics of the Dalles inflow volumes.

Table 5.21. Inflow volume statistics at the Dalles Dam

Statistic		Dalles Inflow
Mean (bm^3)	μ	171
SD (bm^3)	σ	32
Skew	γ	-0.03
Kurtosis	κ	1.50

As shown in Figure 5.16 a low correlation exists between the Peace and the Columbia inflow at KNA and the inflow volume at the Dalles during the fall and winter months. On the other hand, a higher correlation exists between the Columbia inflow at KNA and the Dalles inflow volume during the summer months as compared with other months of the year.

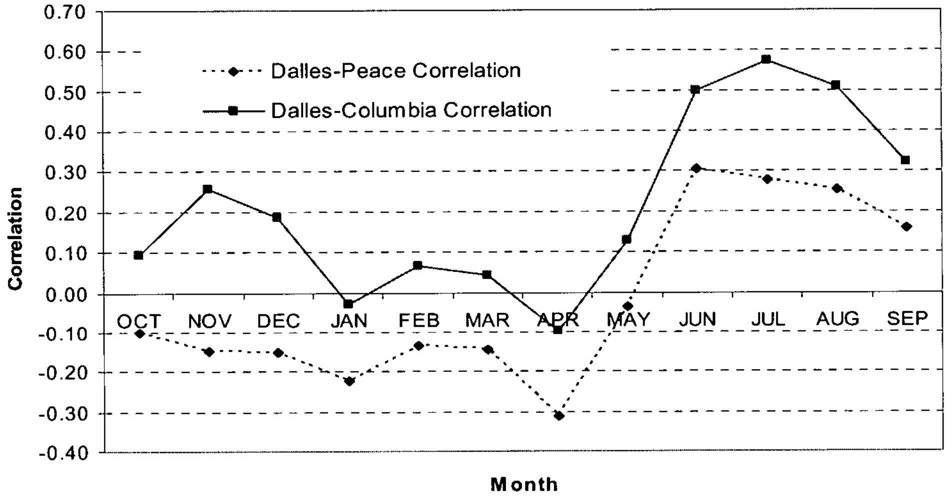


Figure 5.16. Correlation between the Dalles inflow volume and the monthly Peace and Columbia inflows

To express the monthly variability of the load forecast, the 5th, 50th, and 95th percentiles of the annual electricity load forecast as presented in Table 5.13 were considered. Table 5.22 presents the results of using the three-point approximation method of a continuous variable as described by Pearson and Tukey, 1965, to estimate the mean and standard deviation of the electric system load.

Table 5.22. Results of the three point approximation of the mean and the variance

Statistic	Percentile	Load (GWh)
5th Percentile	P_5	56,046
Median	P_{50}	58,769
95th Percentile	P_{95}	61,480
Mean	μ	58,767
Standard Deviation	σ	1,652

A Monte Carlo simulation was then used to generate 10,000 load scenarios. Figure 5.17 presents the system electric load cumulative probability distribution function for a log-normal distribution. The probability distribution was divided into 60 increments to match the number of historical inflow data sets for the Peace, the Columbia Rivers, and at the Columbia River at Dalles. These sixty load values were then used to estimate the first four moments and the load correlation with the other random variables.

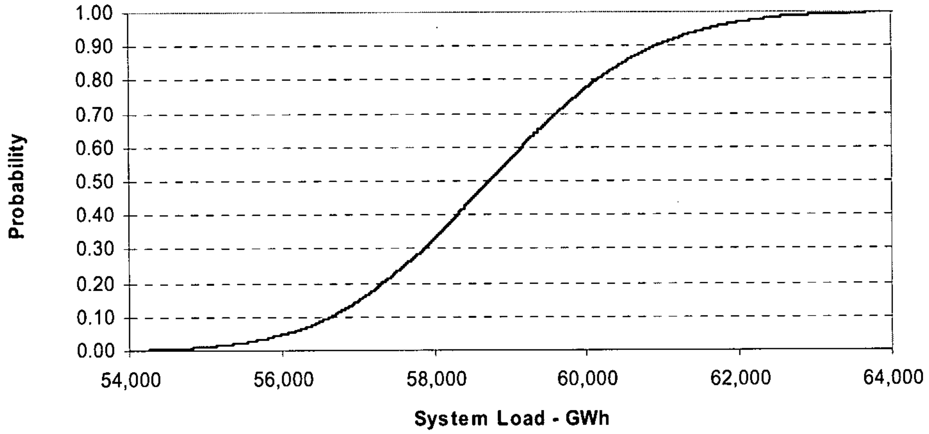


Figure 5.17. Probability distribution function of forecasted system load

5.4.3.2. Results of the Moment Matching Algorithm

The moment matching algorithm developed by Kaut et al. (2003) was used to generate the set of scenarios used in this study. The input data included the first four moments and the correlation matrix of the 26 variables as described above (twelve month Peace and Columbia inflows, the Dalles inflow volume, and the electricity load forecast). The algorithm was run to generate a number of scenario sets: ranging from 10 to 100 scenarios for each of the 26 variables. To evaluate the quality of the generated scenarios, a comparison between the statistics of the generated scenarios and the historical data was carried out. The absolute mean relative error (MRE) metric was used in this analysis. Figure 5.18 presents the mean relative error (MRE) for the estimated four moments. Overall, the graph indicates that the generated scenarios preserve the statistical properties of the original data for the 26 variables with high degree of accuracy. The results of the mean are a 100% match for the mean of the historical data, even for the low number of 10 scenarios case. For the other statistics (standard deviation, skewness, and kurtosis), Figure 5.18 demonstrates that the MRE decreases with the increase in the number of generated scenarios. The MRE decreases to less than 5% as the number of scenarios reaches 30 and it further reduces to lower values as the number of generated scenarios increases, but with a lower rate of decrease.

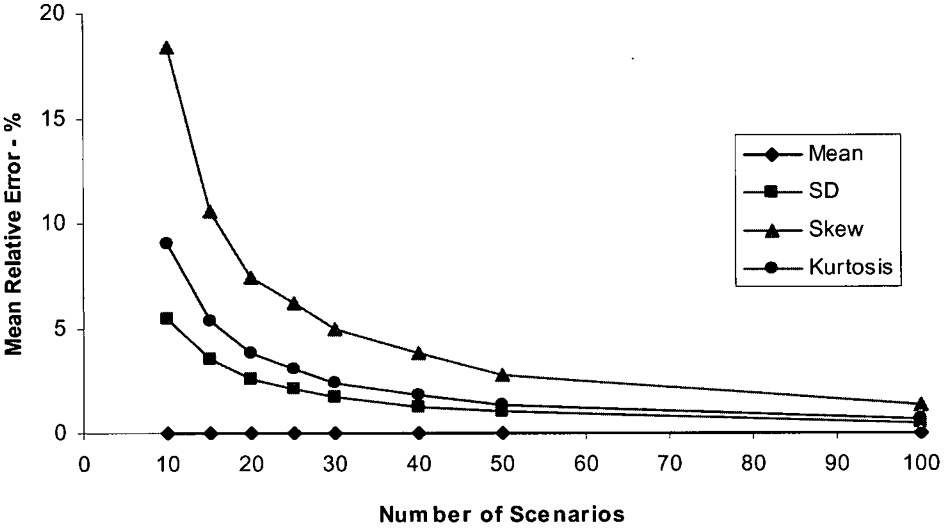


Figure 5.18. Testing the quality of the different numbers of generated scenarios

As an example, a comparison of the statistics of the 30 generated scenarios of the Peace River inflows with the historical records are displayed in Figure 5.19. The graph clearly demonstrates the high degree of accuracy of the estimated four moments of the generated scenarios.

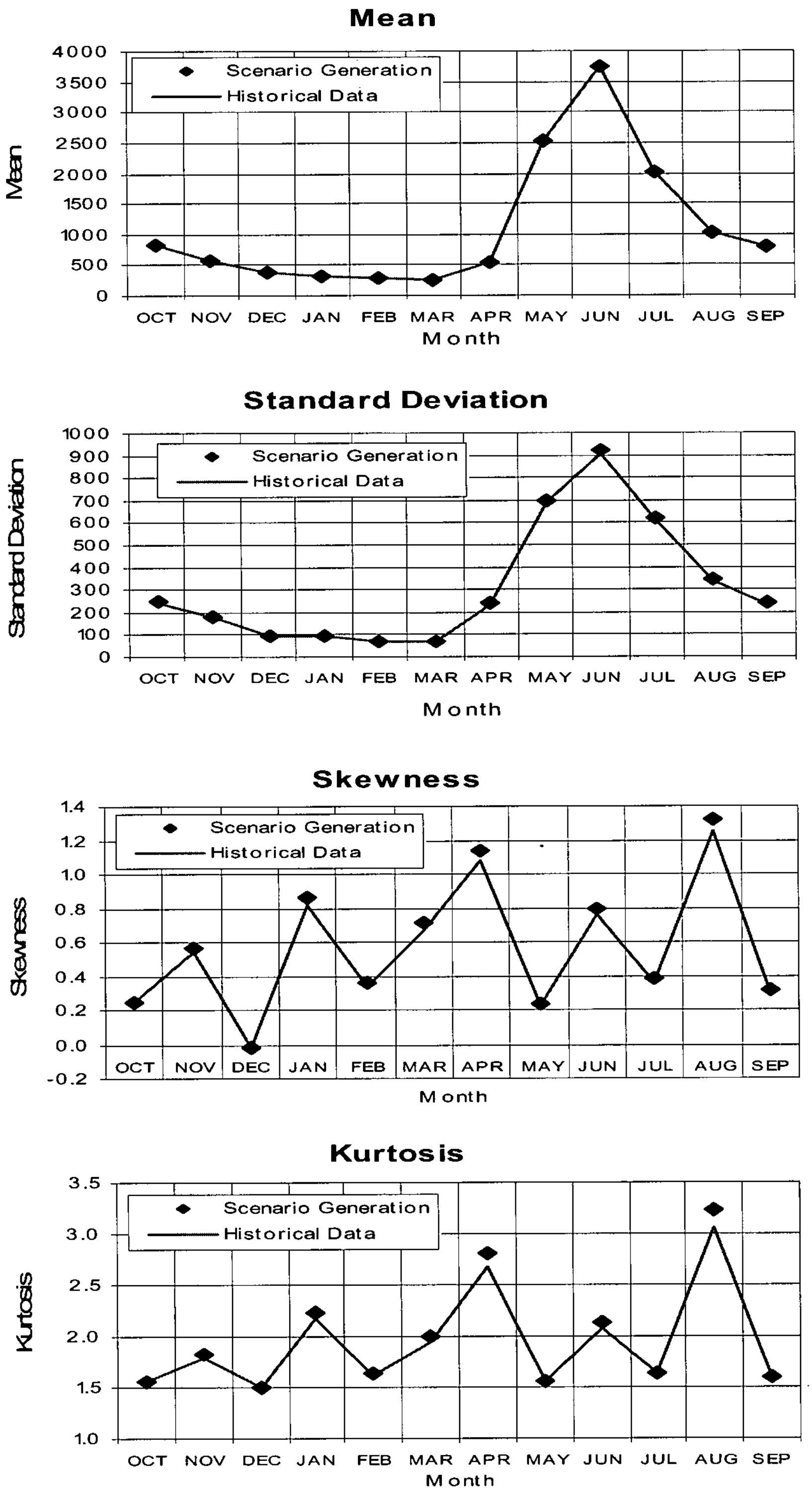


Figure 5.19. Comparison between the Peace inflow historical data and the generated scenarios statistics

5.4.4. Results of the RLROM Model Runs

The input data for the case study presented in section 5.4.2 and the set of scenarios of the random variables described in section 5.4.3 were used to run the RLROM model. The model was set to run on a monthly time step for thirty six months starting from the beginning of the water year in October. The results of using the model for the case study are presented and discussed in the following sections. To assess the quality of the operation planning policies derived by the model, the results are presented and analyzed. Finally, the model was run in control mode and the results of a simulation run are presented and discussed.

The following parameter setting was adopted in this case study: the exploitation exponent parameter $\zeta = 0.0125$, learning rate exponent $\psi = 0.845$, and annual discount rate of 8%.

5.4.4.1. Value of water in storage

As stated in the previous chapter, the objective function maximizes the expected net revenue from the spot and forward transactions. Figures 5.20 to 5.22 present three dimensional (3-D) views of the value function in Canadian dollars, as a function of GMS and MCA storage volumes. The 3-D plots present the shape of the value of water in storage surface function for different months of the year, namely: December, May, and August. The choice of the three months was meant to represent: the winter, spring (freshet), and summer respectively. The graphs clearly demonstrate the dependence of the value function on the storage level in both of GMS and MCA reservoirs. It can also be seen that this dependence on the storage level varies from one month to the other. For example, the value function in May is less sensitive to the storage level in the other reservoir. It can also be seen that the minimum value of water in storage occurs during the freshet when the storage level is rapidly increasing, and the market price is low.

The slope of the value function surface represents the dollar value of each unit of storage in dollars per cubic meter second-day ($/cms-d). The derivative of the value function with respect to the storage volume in MCA represents the marginal value of water in MCA. Similarly, the derivative of the value function with respect to GMS storage represents the marginal value of water in GMS. The marginal value of water is also called the shadow price or the Lagrange multiplier of the mass balance equation for a reservoir (Equation 4.9).

The steeper slopes of the storage surface value functions for December and August, as shown in Figures 5.20 and 5.22, indicate a high marginal value of water ($/cms-d) in those months compared with the corresponding values in May. This is mainly attributed to the high market prices and high demand during the winter months as shown in Figure 5.14 and Figure 5.15. Whereas in summer, the demand is low and the inflow hydrograph is in a receding trend, and the high marginal values is a result of the higher market opportunities during the heavy load hours (HLH) as shown in Table 5.12. On the contrary, high inflows, low demands, and low market prices during the freshet results in flatter slopes of the storage value function as shown in Figure 5.21.

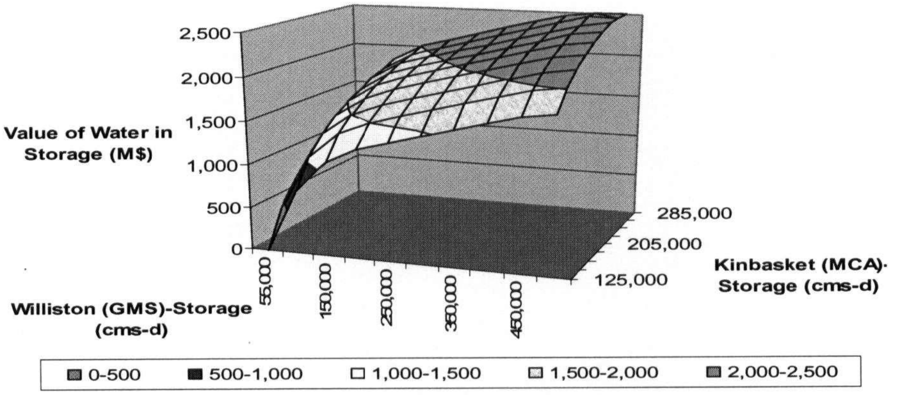


Figure 5.20. Storage value function in December

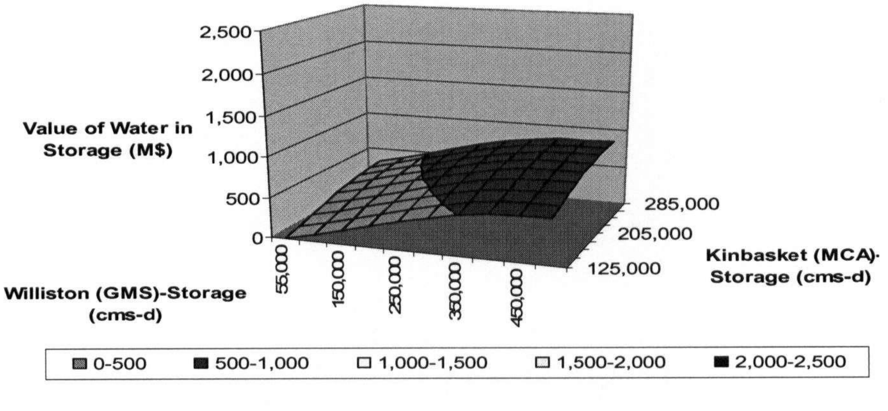


Figure 5.21. Storage value function in May

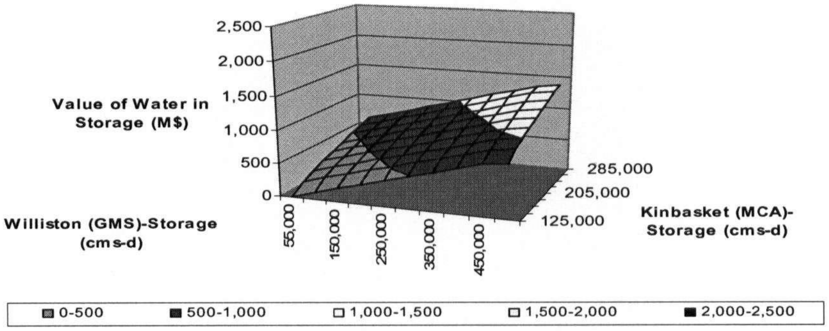


Figure 5.22. Storage value function in August

To examine the value function, the December storage value is plotted, as an example, in a 2-D view as shown in Figure 5.23 for different increments of GMS storage. The results show the decreasing slope of the value function with the increase of storage in Mica. This indicates the effect of the available storage volume on the value of water in Mica reservoir: as the Mica reservoir level rises the marginal value of the water decreases. Also, the graph demonstrates the decreasing marginal value of water in Mica as more storage is available in GMS. The decreasing gap between the curves of the different GMS storage increments indicate a reduction in the marginal value of water in GMS for a given storage level in Mica as shown in Figure 5.23.

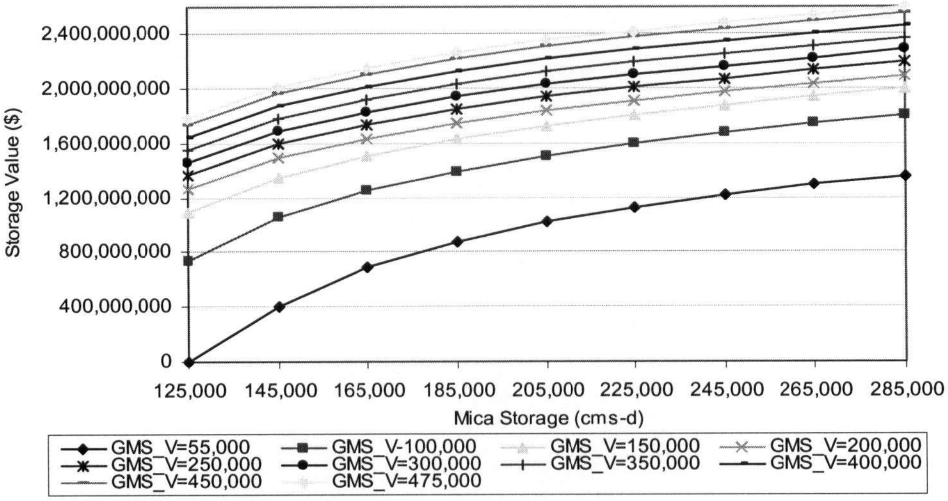


Figure 5.23. Mica Storage value function in December

Figure 5.24 presents the monthly value of water in storage for different GMS reservoir storage increments with a storage level at 50% in Mica reservoir (205,000 cms-d). The results indicate that the highest storage values occur in the period of November to January and the lowest storage values in the period of April to June. With low storage in GMS in December and January, the graph demonstrates that the marginal water values (slope of the first segment of the storage value function) tend to be higher because: the electricity demand is high, inflows are low, and the Peace River ice flow constraint is imposed in December and January.

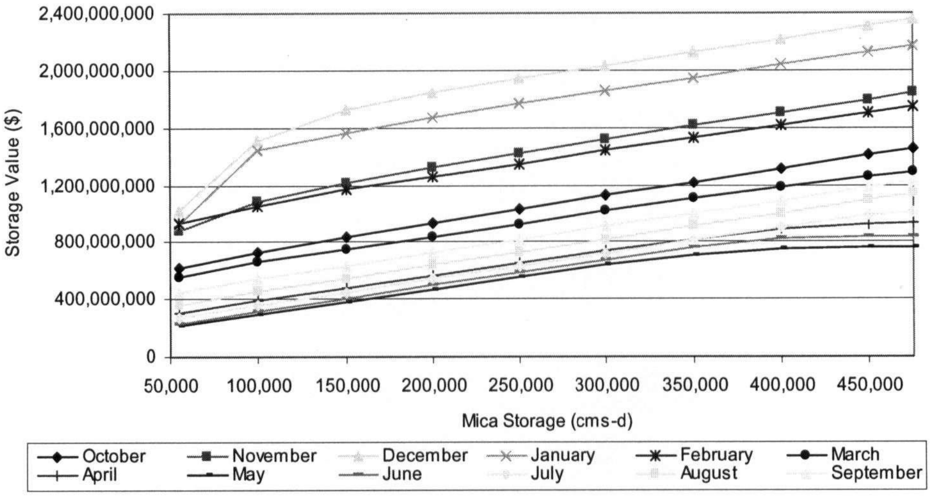


Figure 5.24. Monthly GMS storage value function with Mica reservoir at 50% storage level

5.4.4.2. Marginal value of energy

One of the most significant pieces of information obtained from the model results is the marginal value of energy, expressed in $ per megawatt-hours ($/MWh). As stated earlier, the significance of this information is that the operation planning dispatch decisions are based on a comparison of the energy value with market prices. The marginal value of energy is obtained by converting the marginal value of water in $/cms-d to $/MWh using the HK factor as expressed in Equation 4.22.

Tables 5.23 and 5.24 present the monthly range of the marginal value of energy for MCA and GMS in ($/MWh). For each month, the marginal value of energy is given for the combination of empty or full MCA storage (125,000-285,000 cms-d) and empty or full storage in GMS (55,000-475,000 cms-d). For the case of empty MCA reservoir, the summary results in Table 5.23 clearly demonstrate the impact of the storage level in GMS on the marginal value in MCA in almost all months with the exception of the period from April to June where the MCA marginal value of energy is not sensitive to the storage level in GMS. The distinctly high marginal values of energy in certain months could be attributed to the cost of satisfying the high KNA outflow releases required in those months by the Columbia treaty operation (average KNA outflow values were presented in Table 5.18).

When MCA reservoir is full, it can be seen that the marginal value of energy is less sensitive to the available storage level in GMS. The same independence in MCA marginal values of energy is noticeable during the freshet even for empty GMS storage. However, the influence of GMS storage on MCA marginal value of energy during the freshet is most noticeable in the middle range of MCA storage as shown in Figure 5.28.

Table 5.23. MCA monthly range of marginal value of energy

Month	GMS Storage	MCA Marginal Value ($/MWh)	
		Empty	Full
	cms-d	125,000cms-d	285,000 cms-d
October	55,000	295.87	47.14
	475,000	173.32	43.66
November	55,000	338.65	48.6
	475,000	224.7	44.93
December	55,000	264.9	49.49
	475,000	141.2	43.85
January	55,000	335.18	44.67
	475,000	126.65	37.36
February	55,000	186.78	36.69
	475,000	86.61	27.03
March	55,000	83.5	25.88
	475,000	53.79	14.96
April	55,000	48.72	7.62
	475,000	42.7	4.68
May	55,000	50.04	3.33
	475,000	49.72	1.72
June	55,000	60.56	14.35
	475,000	53.62	10.18
July	55,000	106.71	36.47
	475,000	103.53	35.02
August	55,000	161.64	45.29
	475,000	138.72	44.19
September	55,000	199.16	46.5
	475,000	141.63	44.51

The summary results presented in Table 5.24 show the higher GMS marginal value of energy when the reservoir is at low storage levels during the period September to January. This is mainly due to the high cost associated with meeting the Peace ice flow constraint in December and January (1100 and 1500 cms respectively). The lowest GMS marginal value occurs in the months of April and May where the marginal value drops to values close to zero $/MWh, when both of GMS and MCA are full.

Table 5.24. GMS monthly range of marginal value of energy

Month	MCA Storage	GMS Marginal Value ($/MWh)	
	cms-d	Empty 55,000cms-d	Full 475,000 cms-d
October	125,000	231.78	41.87
	285,000	96.70	41.74
November	125,000	440.05	42.12
	285,000	267.82	41.78
December	125,000	461.40	41.21
	285,000	297.25	41.49
January	125,000	349.27	37.84
	285,000	54.81	37.00
February	125,000	200.78	26.96
	285,000	64.85	24.34
March	125,000	94.50	11.77
	285,000	50.91	8.66
April	125,000	47.53	1.77
	285,000	45.16	0.20
May	125,000	48.89	0.86
	285,000	48.42	0.40
June	125,000	53.64	23.42
	285,000	47.93	11.58
July	125,000	65.76	42.06
	285,000	51.83	40.63
August	125,000	64.14	42.45
	285,000	55.46	40.88
September	125,000	128.93	42.50
	285,000	61.15	42.19

Figures 5.25 to 5.28 present typical examples of the marginal value of energy for MCA and GMS in winter, spring, and summer months - December, May, and August. Figure 5.25 demonstrates the sensitivity of the marginal values of energy in MCA to the available storage in the reservoir particularly when the reservoir drops below 205,000 cms-d. This sensitivity is more prominent when the GMS reservoir storage is at or below 150,000 cms-d.

When MCA and GMS reservoirs are at their lowest levels (i.e. at 125,000 and 55,000 cms-d respectively), MCA marginal value of energy increases rapidly. The high marginal value is due to the penalty for violating the minimum storage constraint to meet the system electricity load or the Columbia treaty operation outflow constraint at KNA. In general, the results suggest that at the low range of GMS and MCA storage levels, the marginal value of energy is higher than the average LLH market price in December. In such a case it would be more valuable to store water than to release it.

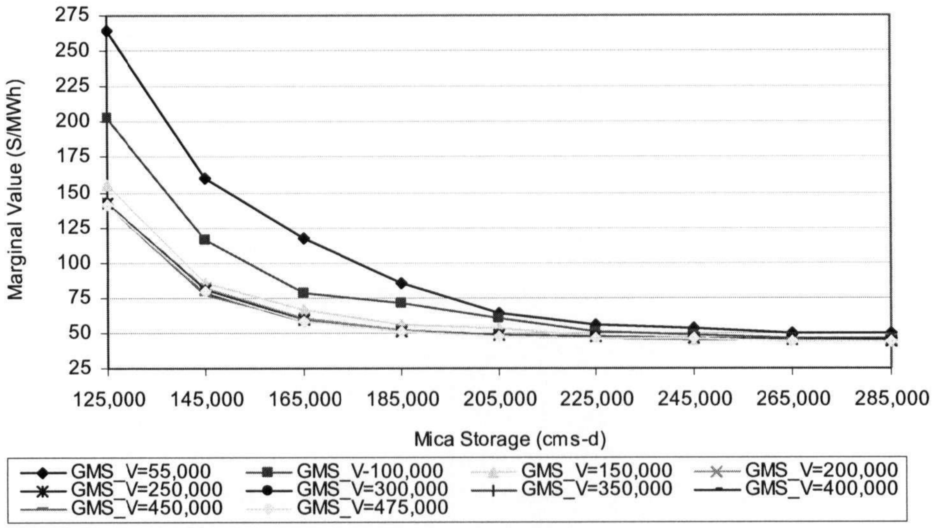


Figure 5.25. MCA marginal value of energy in December

On the other hand, Figures 5.26 and 5.27 show that the marginal value of energy in GMS is more sensitive to the variation in MCA storage below a GMS storage volume of 250,000 cms-d. The dependence of GMS marginal value of energy on MCA storage is apparent when the storage in MCA is at or below 185,000 cms.d. The higher value of GMS energy in December is mainly attributed to the higher electricity load and the high market prices accompanied with the lower inflow conditions in winter as compared with the freshet period. The high outflow releases downstream of Peace Canyon dam (PCN) in December and January, are required to maintain the winter ice flow constraint and this also contributes to the high marginal values particularly when MCA is at or below 145,000 cms-d. Figure 5.27 demonstrates that the variation of the GMS marginal value for different MCA storage is decreasing as GMS reservoir is above 50% of its full storage volume (250,000cms-d). This is apparent in the decreasing gap between of the marginal value of energy curves for different values of MCA storage as GMS storage increases.

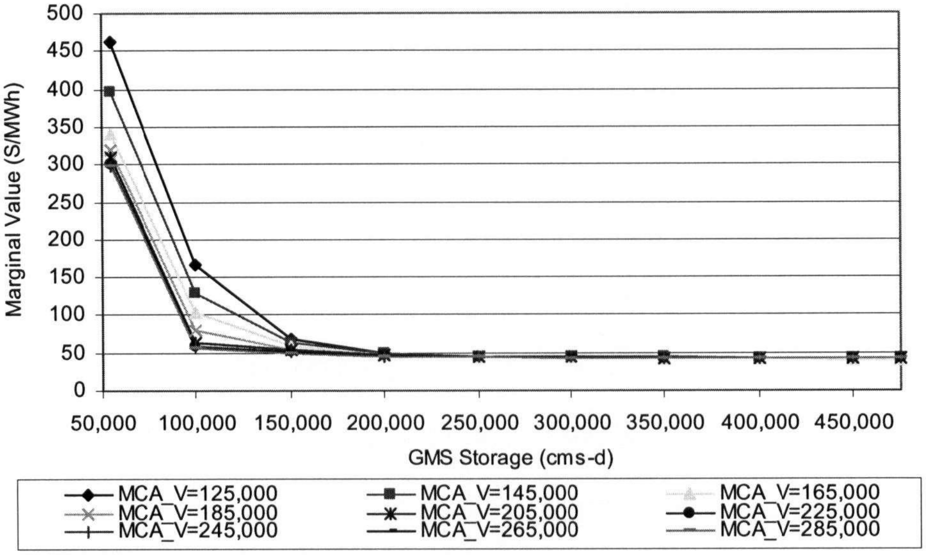


Figure 5.26. GMS marginal value of energy in December

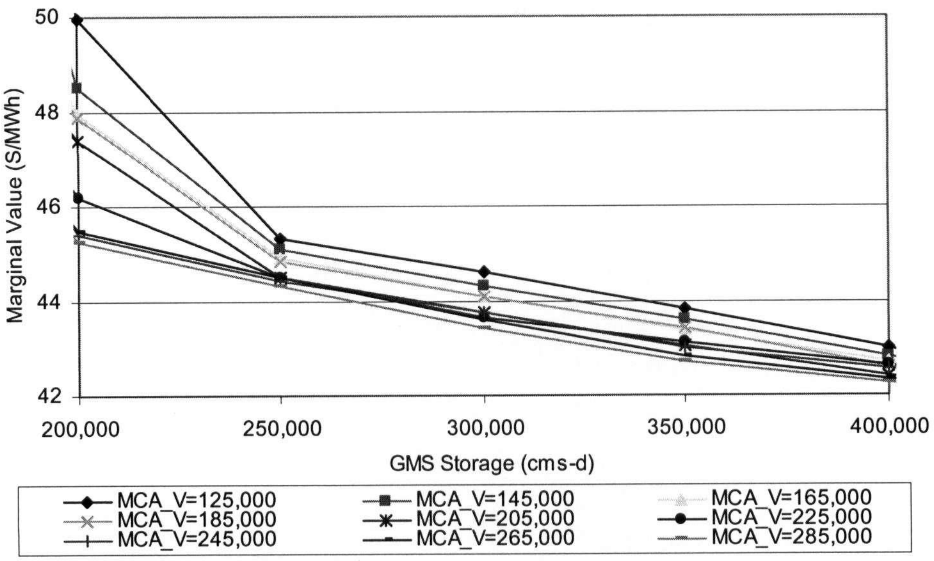


Figure 5.27. GMS marginal value of energy in December

Figures 5.28 and 5.29 display the marginal value of energy for MCA and GMS in the month of May. In general, the marginal value of water is low during this time of the year for the full range of storage in both of MCA and GMS reservoirs. The highest MCA and GMS marginal values of energy occur when the MCA and GMS reservoirs are empty and the value of energy is about 50.0 $/MWh. On the other hand, the value of energy drops to almost zero when the reservoirs are at full pool level. The low marginal value of water is attributed to the fact that additional storage in these reservoirs would have a high probability of spill during this period. Figure 5.28 demonstrates the significance of the change in GMS storage levels on the MCA marginal values of energy in the range from 165,000 to 265,000 cms-d. Figure 5.29 portrays the influence of variation in MCA storage on GMS marginal value of energy, in particular with GMS storage in the range from 250,000 to 400,000 cms-d.

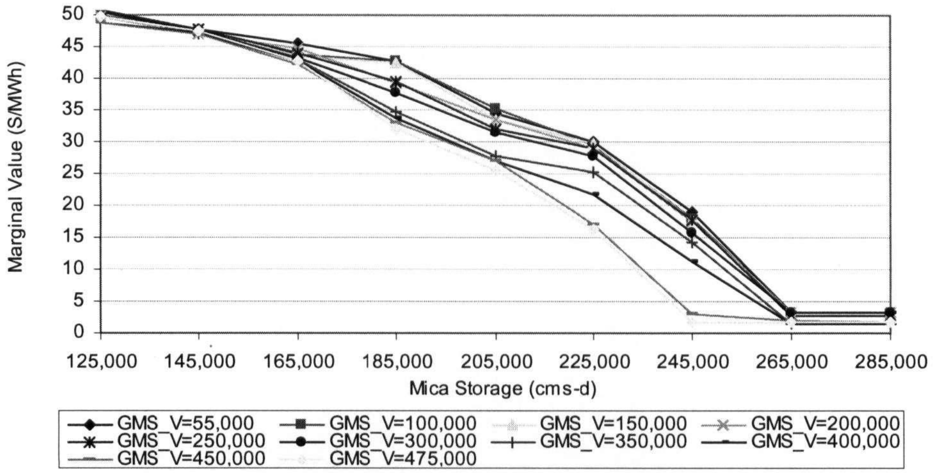


Figure 5.28. MCA marginal value of energy in May

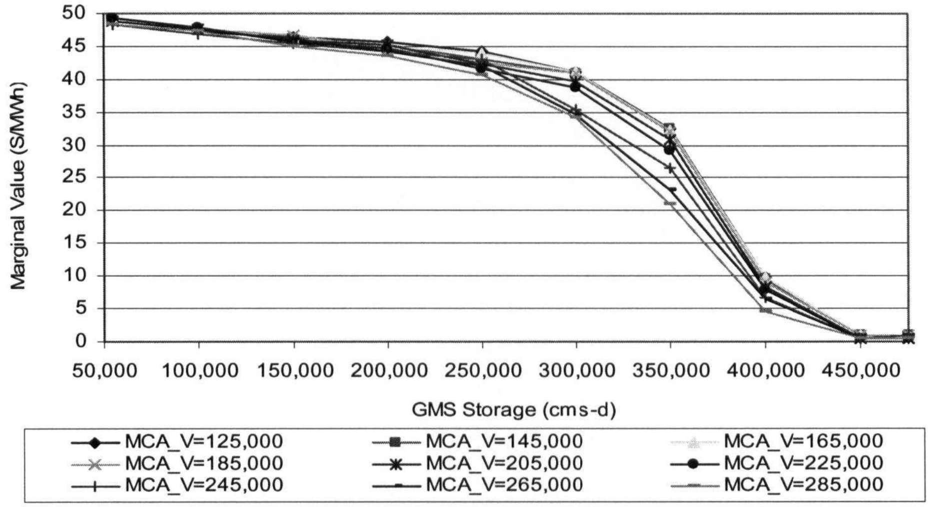


Figure 5.29. GMS marginal value of energy in May

Figure 5.30 demonstrates the marginal value of energy for GMS in August. The higher inflows and lower electricity demand in August, as compared with December, result in a noticeably lower marginal value when both of MCA and GMS are at high storage levels. Despite the lower load in August, as compared to the winter months, the high marginal values when the reservoirs are low could be attributed to the high California market opportunities in August, particularly during heavy load hours (see

Table 5.12 for market price structure). Figure 5.30 illustrates the significant influence of MCA storage level on the GMS marginal value of energy particularly when the GMS storage drops below 300,000 cms-d. Above 300,000 cms-d, the marginal value is less dependant on the GMS storage level.

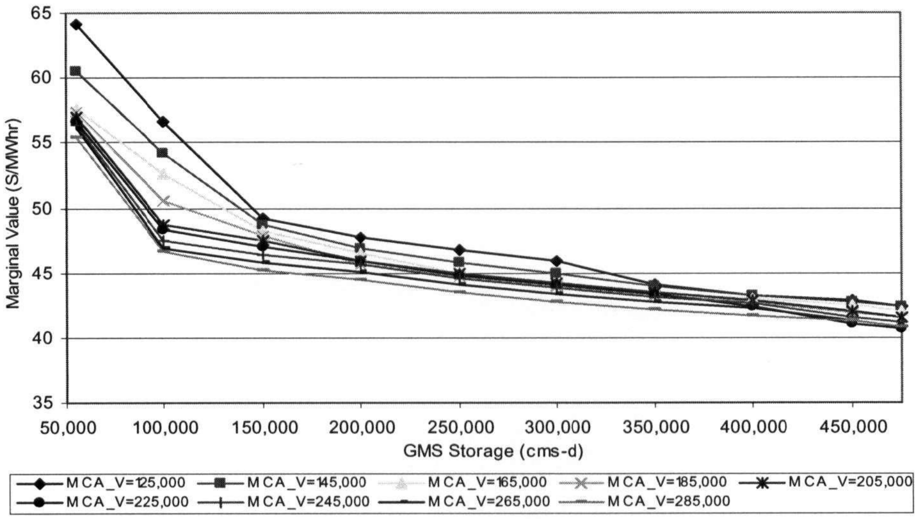


Figure 5.30. GMS marginal value of energy in August

Figures 5.31 and 5.32 display the monthly MCA marginal value of energy for different GMS storage increments with MCA storage at 125,000 and 285,000 cms-d respectively. Figure 5.31 displays on the secondary y-axis the average Keenleyside (KNA) outflow releases, as stipulated by the Columbia River Treaty. The impact of these high release requirements in the winter months combined with high electricity load, high market prices, and low inflows are reflected in the high marginal values during those months. This effect is less noticeable in the summer as the load is not high and higher inflows are available as compared with winter months.

The high marginal value of energy in the summer is mainly due to better market opportunities, particularly in August and September as shown earlier in Table 5.12. Figures 5.31 and 5.32 demonstrate that, whatever the available storage in GMS and MCA reservoirs, the lowest marginal values occur in April and May. Obviously, high natural inflows and lower electricity demand are the main reasons. Also both figures point to the increased sensitivity of the MCA marginal values to the incremental change in GMS storage in the fall and winter months.

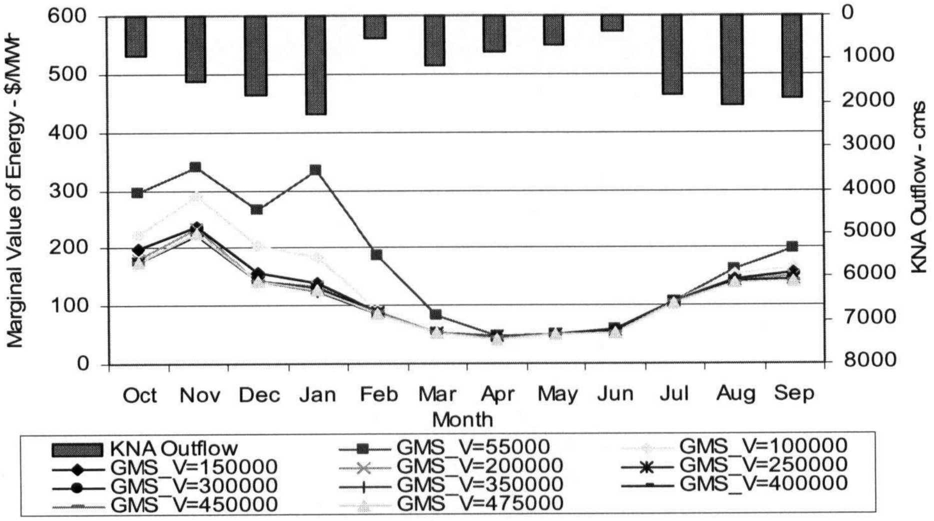


Figure 5.31. **MCA monthly marginal value of energy for a low storage level at MCA (MCA storage=125,000cms-d)**

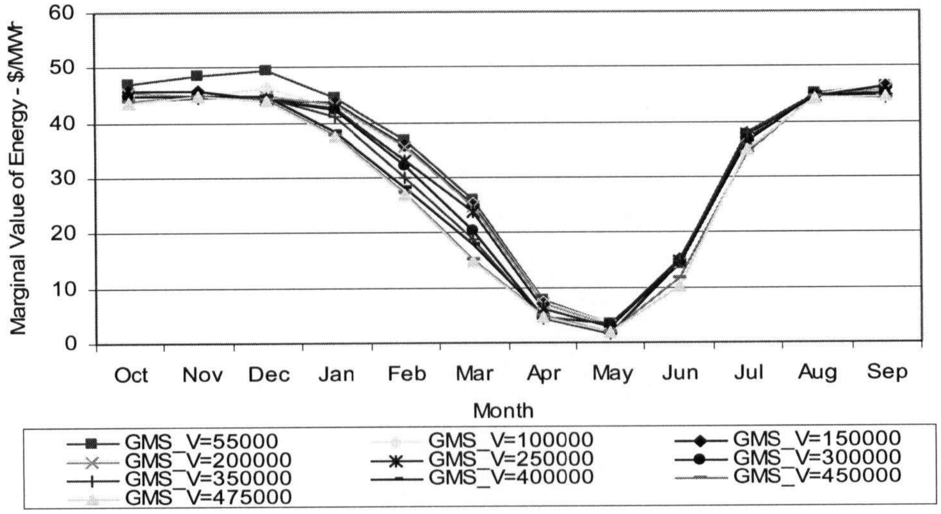


Figure 5.32. **MCA monthly marginal value of energy for a high storage level at MCA (MCA storage=285,000cms-d)**

Figure 5.33 demonstrates the effect of the high Columbia Treaty release requirements on the KNA marginal value of water in $\$/m^3$ when the MCA reservoir is at low and at high storage levels. The marginal values are extracted from the shadow price of the hydraulic continuity constraint (storage mass balance) of the GOM model. The results indicate that when MCA storage level is low, the high flow releases result in significantly high marginal values of water at KNA. These marginal values propagate and cause an increase in the MCA marginal of water/energy as presented in Figure 5.31, in particular when the storage level at MCA is low.

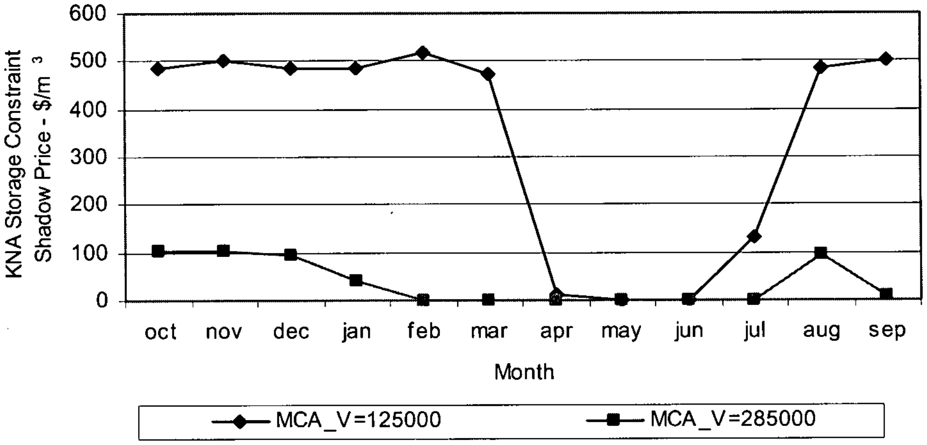


Figure 5.33. Marginal value of KNA storage constraint

Figure 5.34 displays the monthly results of GMS marginal values of energy for a GMS storage level of 55,000 cms-d. Similar to MCA's marginal values, the results indicate higher marginal values during the winter months. In addition, the results indicate the dependence of the marginal values in GMS on the available storage in MCA during the winter and fall months. The highest GMS marginal values occur in November, December, and January in response to the high releases that are required to satisfy the Peace River ice flow constraint and to high market prices. The graph also indicates that GMS marginal value is independent of MCA storage in the spring and in early summer months as the storage level in the reservoir increases and when the demand is low.

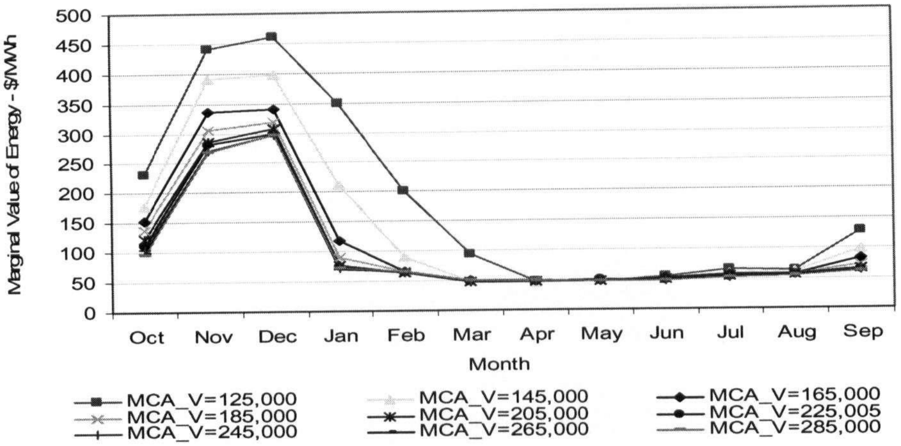


Figure 5.34. **GMS monthly marginal value of energy for an empty GMS reservoir storage (GMS storage=55,000cms-d)**

Figure 5.35 displays the monthly results of GMS marginal values of energy for a full GMS storage of 475,000cms-d. Similar to MCA, the GMS lowest marginal values occur in April and in May. However, when the GMS reservoir is full, the marginal values are not sensitive to changes in MCA storage levels almost all year around. The graph also demonstrates that the marginal values during July and September are in the same order of magnitude as those in October and December.

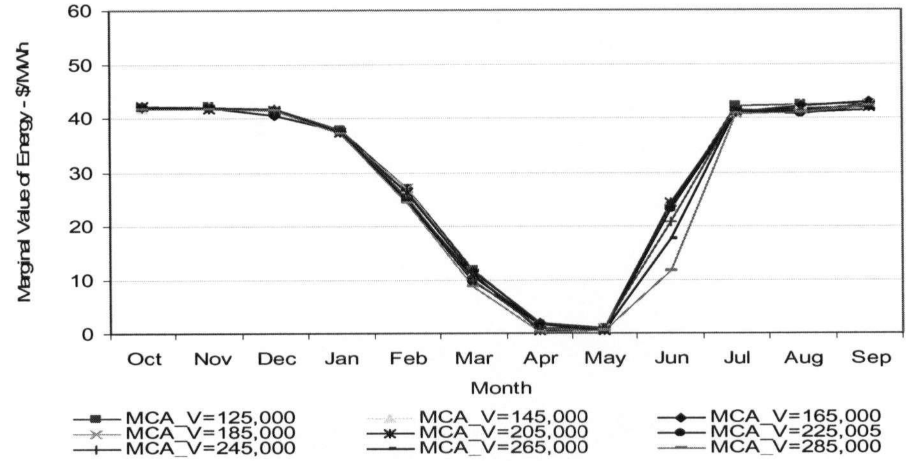


Figure 5.35. **GMS monthly marginal value of energy for a full GMS reservoir storage (GMS storage=475,000cms-d)**

To demonstrate the variation in the GMS marginal value of energy in the different months, the cases of empty and full MCA storage are presented. For these two cases, Figures 5.36 and 5.37 portray the details of the monthly variation in GMS marginal values of energy for the range of storage increments at GMS. The figures demonstrate the significant difference in GMS marginal value of energy for GMS storage below a level of 150,000 cms-d, for low and full MCA storage. This difference is apparent for most of the year, with the exception of the period from April to June. Figures 5.38 and 5.39 illustrate a magnified region for the results for GMS storage values above 150,000 cms-d. Comparing the results presented in the two figures, it can be seen that higher marginal values are noticeable in GMS when the MCA storage is low. This effect on GMS marginal value varies from month to month and decreases where GMS storage level increases. It also shows that the effect of MCA storage on GMS marginal value is minimal when the GMS storage level is high. In both cases, the highest GMS marginal values occur in the period from October to January. The lowest marginal values occur during the months of April and May.

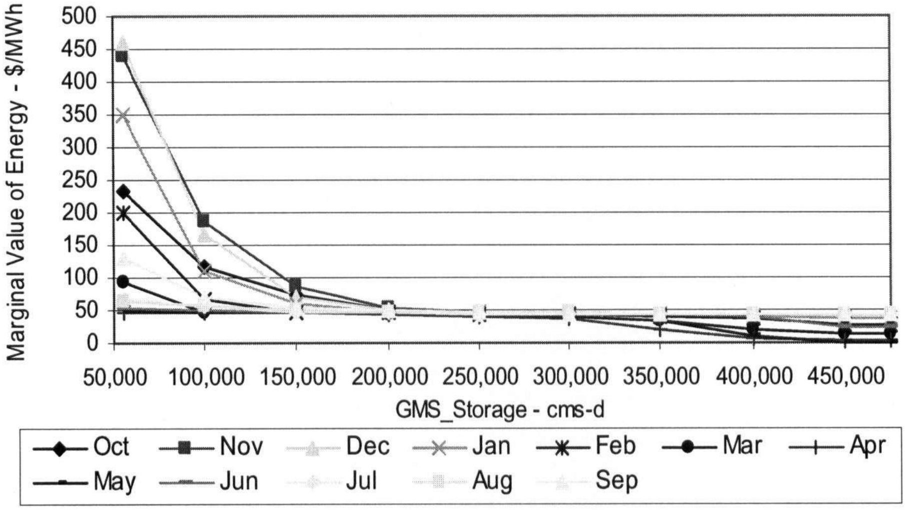


Figure 5.36. GMS monthly marginal value of energy for a low storage level at MCA (MCA storage=125,000cms-d)

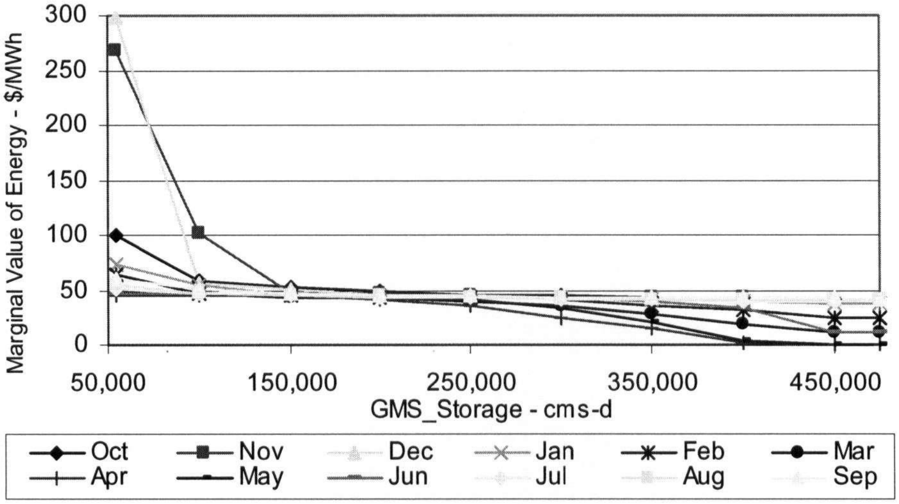


Figure 5.37. **GMS monthly marginal value of energy for a high storage level at MCA (MCA storage=285,000cms-d)**

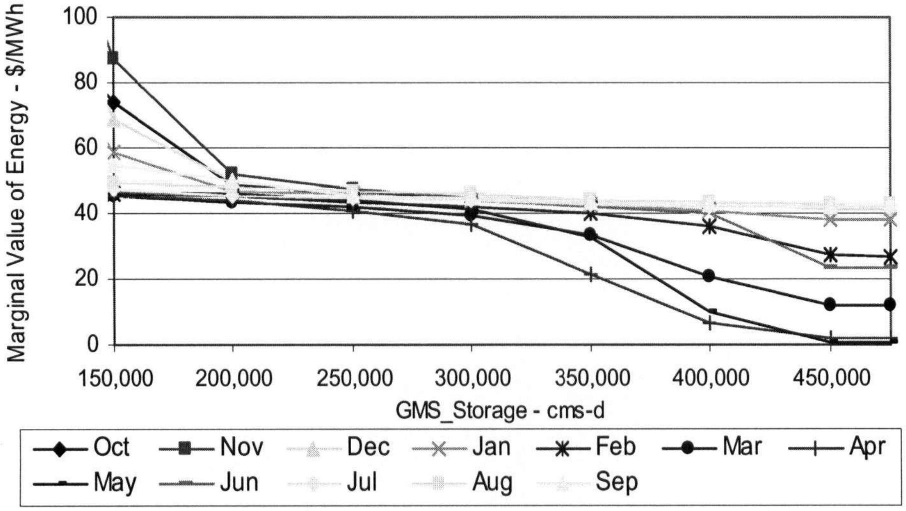


Figure 5.38. **GMS monthly marginal value of energy for a low storage level at MCA (MCA storage=125,000cms-d)**

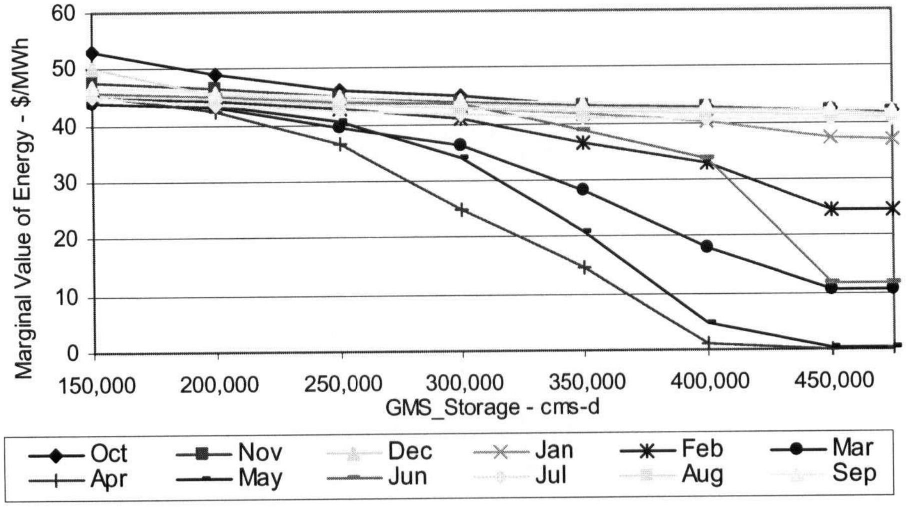


Figure 5.39. **GMS monthly marginal value of energy for a high storage level at MCA (MCA storage=285,000cms-d)**

5.4.4.3. Optimum Control Policy Analysis

In this section, the optimized control policies for operation planning derived by the RLROM model are presented and discussed. The optimized control policy includes the net market transactions and the operation planning of the five main reservoirs on the Peace and the Columbia Rivers, including the optimized plants' generation and turbine releases. To illustrate the optimal control policies proposed by the RLROM model, the model was run using the market prices, electricity system load, and inflow scenario data presented in Table 5.25.

Figures 5.40 and 5.41 present 3-D views of the net market transactions in January and in August. It can be seen in Figure 5.40 that the model recommends a net import policy when the storage volumes in MCA and GMS are low. For a MCA storage of 125,000 cms-d, the results recommend a net import transaction regardless of the storage level at GMS. As the storage levels at MCA and at GMS exceed 200,000 cms-d, the model suggests a net export policy since the marginal value of energy drops below the market price and it would be more profitable to export.

Figure 5.41 shows the high export policies recommended by the model in August as the market price in California is high in the heavy load hours. The GMS/MCA marginal values of energy for the case of full reservoirs are $44.51 and $42.19. When both reservoirs are at the 50% storage level (i.e GMS and MCA storage are: 205,000 cms-d and 200,000 cms-d), the marginal values of energy are: $45.93 and $50.70 respectively. Given that the market prices for HLH/ LLH August market prices are $91 and $53, it can be observed that the market prices are higher than the marginal value of energy even during LLHs. This explains why the suggested export policies by the RLROM model are almost at the limit of the inter-tie transfer capability of both of the US and Alberta markets which is usually set at 2500 GWh in August. When the storage levels in GMS and MCA are low, the model recommends a mix of import and export depending on the hour of the day. Nevertheless, the system is still a net exporter even with the low storage levels in GMS and MCA, as the electricity demand at this time of the year is relatively low.

Table 5.25. Market price, system load, and natural inflow scenario data

Month	Market Price ($)				System Load				Local Inflow				
	WKPK	WKHI	WKLO	WEHI	WKPK	WKHI	WKLO	WEHI	GMS	PCN	MCA	REV	KNA
October	65	65	60	60	7,682	7,116	5,474	6,215	810	23	279	127	179
November	71	71	63	63	8,693	7,901	6,152	6,819	372	16	244	125	223
December	76	76	57	57	9,121	8,253	6,426	7,377	236	17	118	59	125
January	80	80	61	61	8,940	8,217	6,500	7,743	367	30	86	38	103
February	68	68	60	60	8,906	8,238	6,371	7,014	197	18	90	39	103
March	66	66	58	58	8,209	7,692	6,048	6,634	189	19	79	36	99
April	65	65	53	53	7,382	7,005	5,549	6,031	371	20	320	183	429
May	61	61	42	42	6,949	6,659	5,296	5,616	3,421	51	582	323	548
June	60	60	42	42	6,710	6,428	5,070	5,519	5,001	61	2,064	897	1,264
July	66	66	50	50	6,747	6,418	5,030	6,038	1,236	18	1,259	436	544
August	91	91	53	53	6,977	6,675	5,175	5,556	859	19	994	305	331
September	69	69	58	58	6,986	6,683	5,077	5,611	497	11	538	186	231

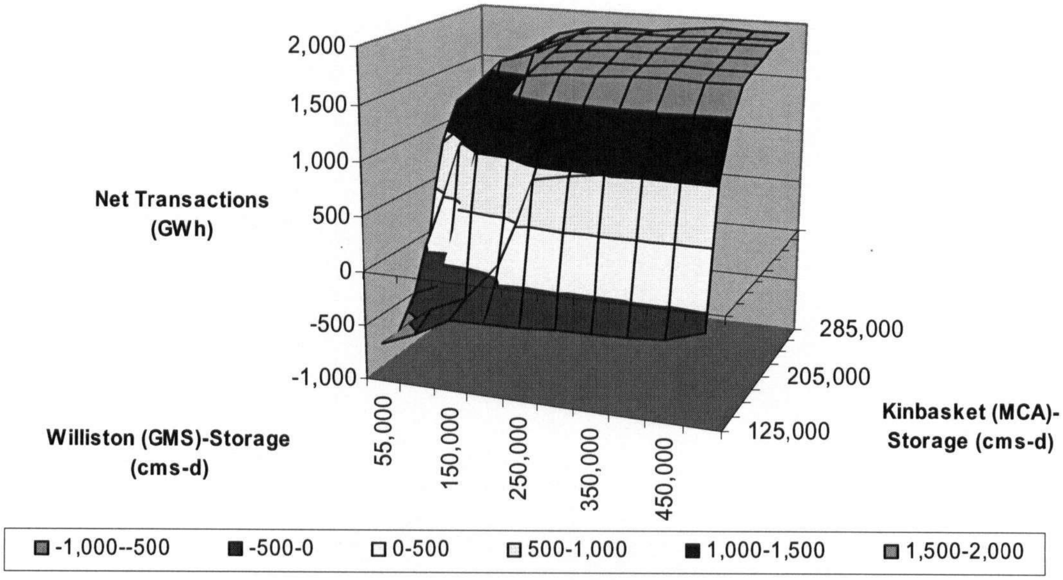


Figure 5.40. January net market transactions

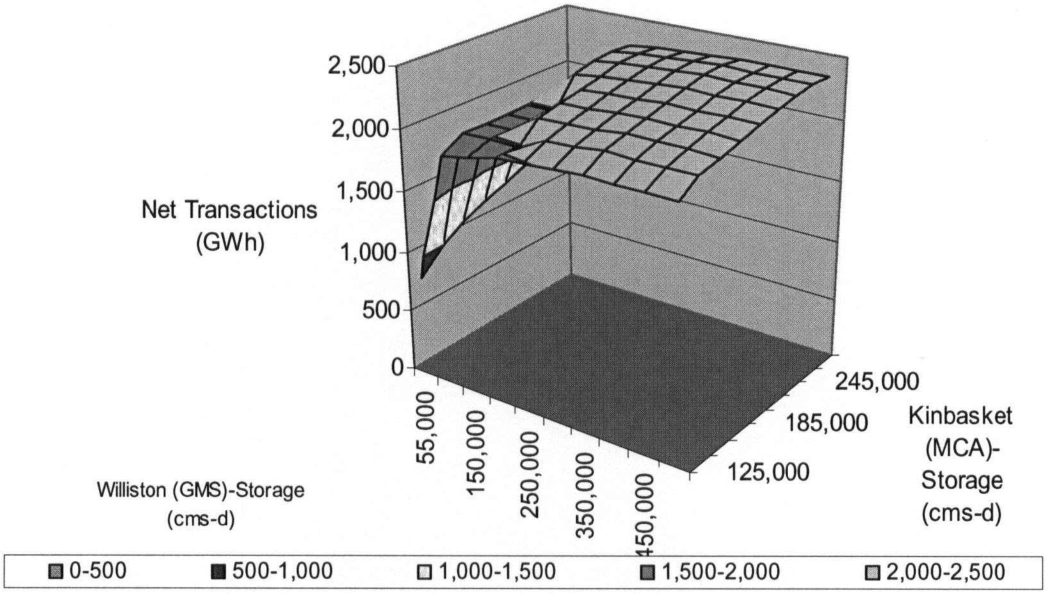


Figure 5.41. August net market transactions

For the cases when the storage levels in GMS are either low or full, Figures 5.42 and 5.43 portray the monthly net market transactions, for different MCA storage volumes. The results show that the net market transactions are largely dependent on the storage volume available in each reservoir. Figure 5.42 demonstrates that for a low GMS storage level, the system will switch to a net importer of energy mode for the whole range of MCA storage in March and up to storage level of 265,000 cms-d in February. For other months, the net transactions are largely dependent on the available storage in MCA. When GMS reservoir is full, Figure 5.43 shows that the system would be a net importer of energy in October, January, March, and April, when the storage level in MCA is low. Figure 5.43 also illustrates that when the MCA reservoir is full, the highest export occurs in the period from August to February, whereas the lowest export transactions occur in the period from March to June. This is attributed to the high HLH and LLH price margins in the summer and winter periods, as compared with those in the spring. The high level of net export transactions occurs in the winter months for this scenario run as the load is lower than the available resources and there is a room to export.

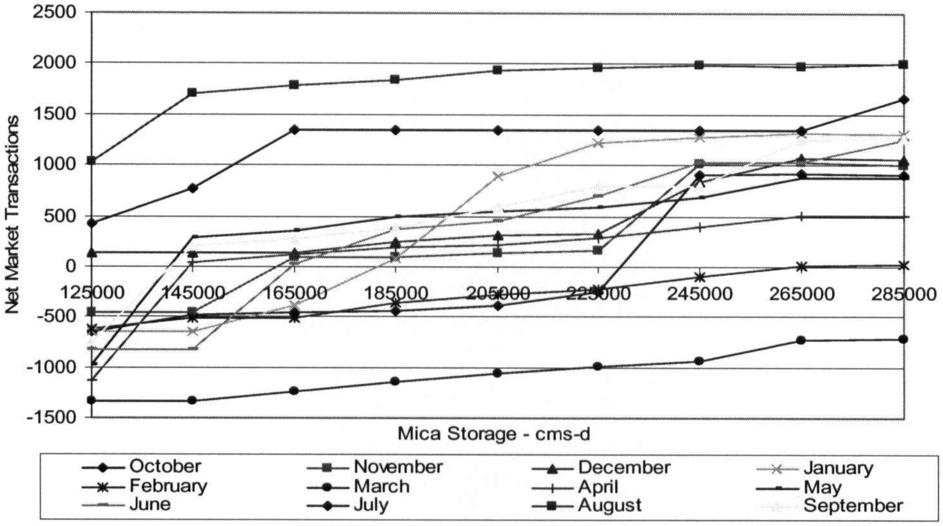


Figure 5.42. Monthly net market transactions for GMS storage of 55,000 cms-d

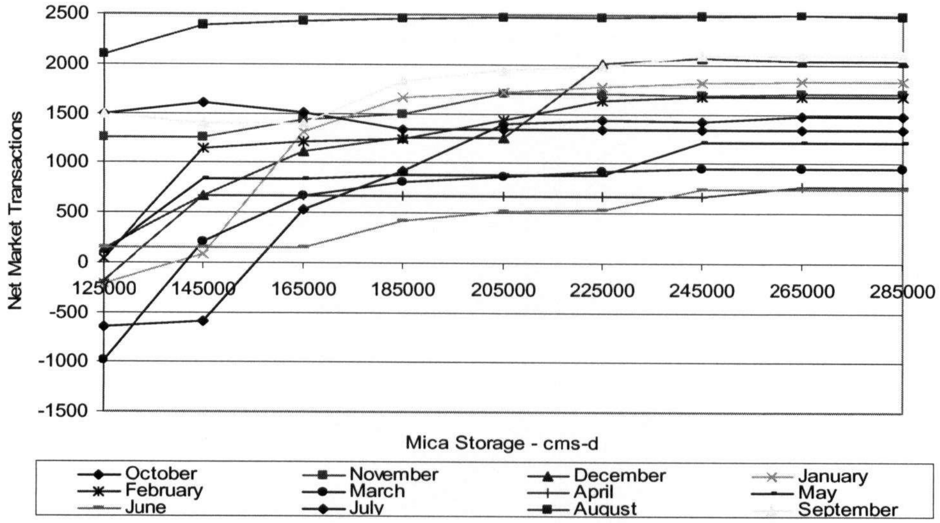


Figure 5.43. Monthly net market transactions for GMS storage of 475,000 cms-d

Figures 5.44 and 5.45 display the monthly variation in net market transactions with different storage levels in GMS and MCA. Figure 5.44 demonstrate that for low MCA and GMS storage levels, the system would be a net importer of energy for most of the year. When the GMS reservoir is full, the system would be a net exporter of energy in all months except for January and March. Considering the case of a full MCA reservoir, Figure 5.45 shows that the system would be a net exporter of energy in all months with the exception of March when storage in GMS is low. In addition, Figure 5.45 illustrates that when the MCA reservoir is full, the net transactions are sensitive to the amount of GMS storage in all months except for August and January.

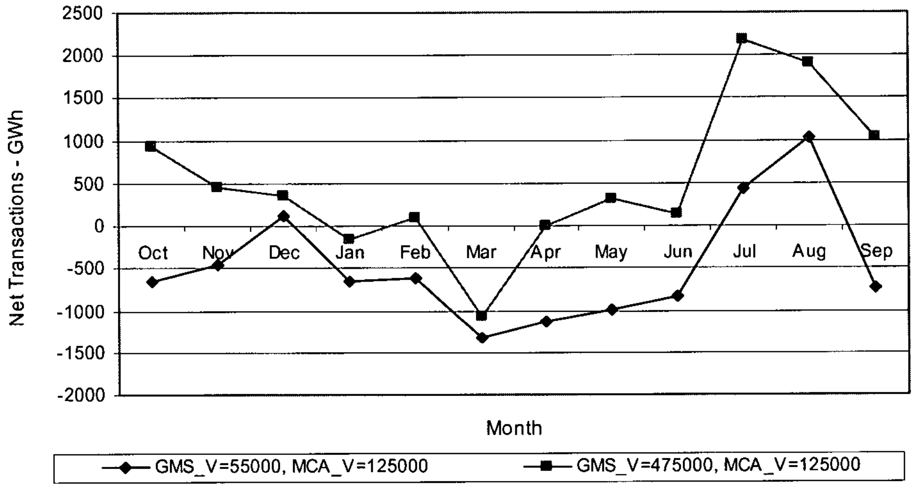


Figure 5.44. Monthly net market transactions for MCA storage of 125,000 cms-d

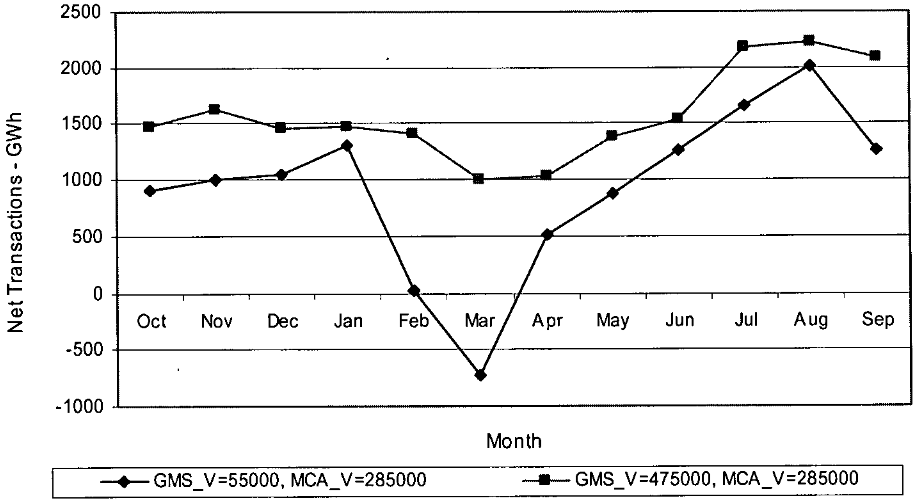


Figure 5.45. Monthly net market transactions for MCA storage of 285,000 cms-d

As indicated in the previous chapter, the RLROM optimizes the operation of the five main plants on the Peace and the Columbia Rivers. The results for these plants for the months of January are presented next. Figures 5.46 and 5.47 display the five plants generation in January for different MCA storage when GMS reservoir is at low storage and high storage levels respectively. Figure 5.46 shows that when the storage levels at MCA and at the GMS reservoirs are low, GMS generation was maintained at about 1300 GWh in order to release water that is needed to meet the Peace River ice flow constraint in January (1500 cms-d), as illustrated in Figure 5.48.

Figure 5.47 illustrates that when the GMS storage is full, MCA generation increases gradually as the available storage in MCA increases. The Revelstoke generation increases, as MCA generation increases, to maintain the hydraulic balance with MCA reservoir as shown in Figures 5.48 and 5.49. Keenleyside is generating up to the maximum limit to minimize the spill flow needed to meet the Columbia Treaty release requirements. The spill flow from Keenleyside is the difference between the required treaty releases and the maximum turbine discharge.

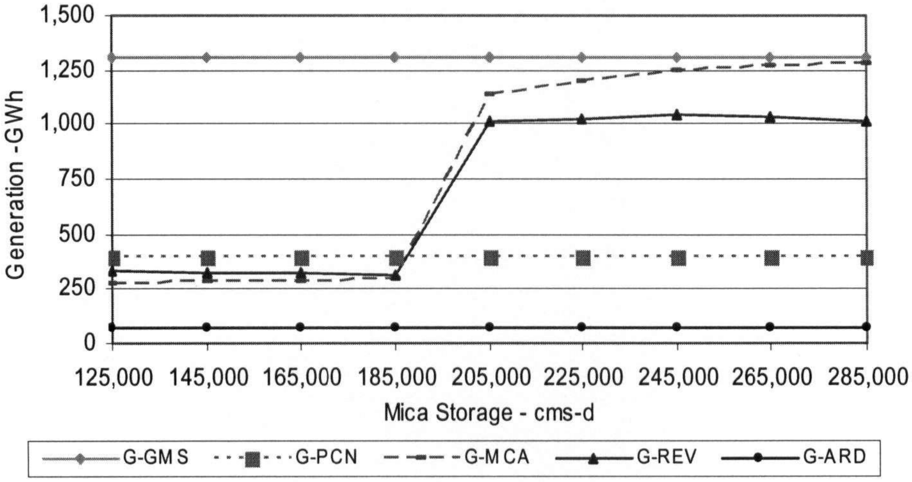


Figure 5.46. Plants generation in January at different MCA storage increments with GMS storage at 55,000 cms-d

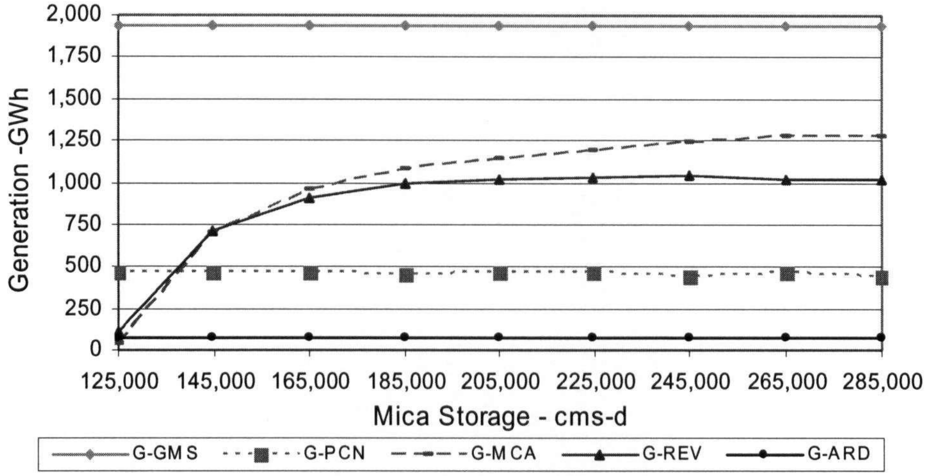


Figure 5.47. Plants generation in January at different MCA storage increments with GMS storage at 475,000 cms-d

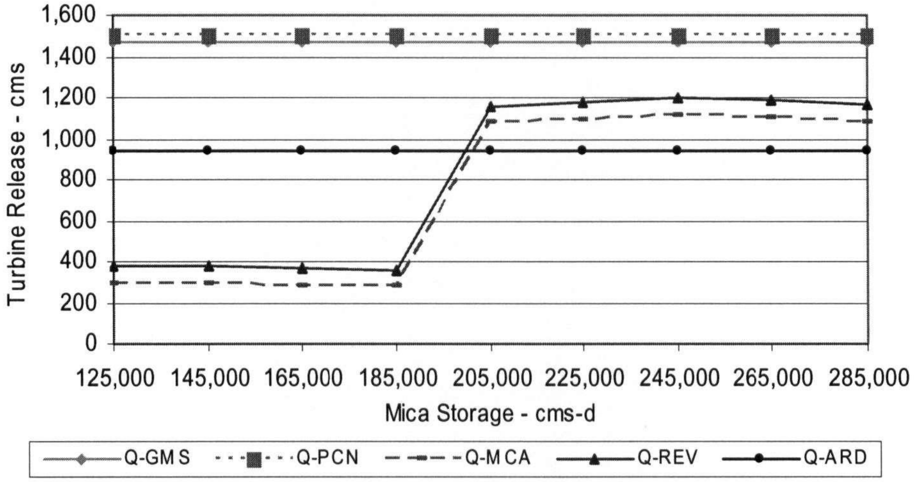


Figure 5.48. Plants average turbine releases in January at different MCA storage increments with GMS storage at 55,000 cms-d

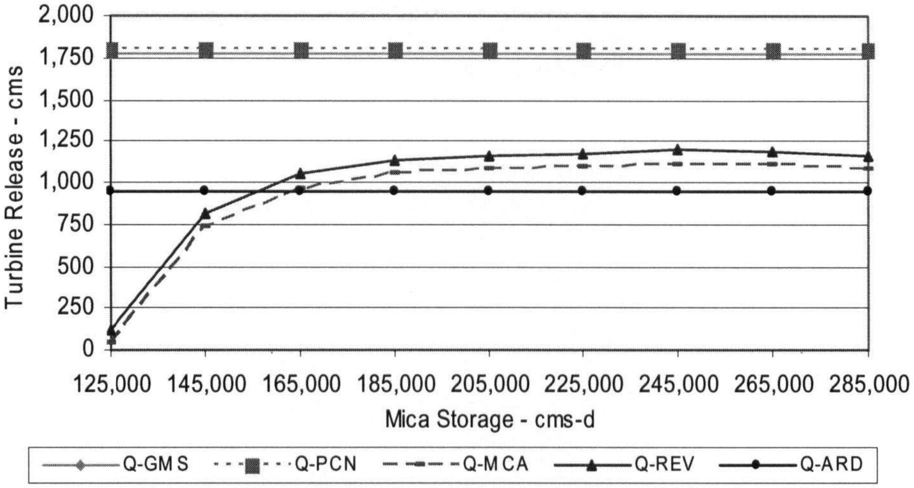


Figure 5.49. Plants average turbine releases in January at different MCA storage increments with GMS storage at 475,000 cms-d

5.4.4.4. Using the RLROM model in Control Mode

Once the RL agent learns the optimal control policy, it can then be used to control the operation planning of the different reservoirs given the inflow, market prices and electricity load at any given time period. The RLROM model is also capable of predicting the operation policy for an extended period of time (for example for several years) based on a given forecast. To achieve this, the RLROM model was modified to run in control mode, rather than in learning mode, using the learned optimal control policies. At any time period and for a given starting storage level, the target storage is linearly interpolated between the target points of the state space. The model uses the learned target storage volumes, optimal forward sales, and the marginal values of water to control the system operation. Given this information and the forecasted inflow, market price, and the electricity load, the model solves the optimization problem at each time period and returns the optimal hydro generation schedules, turbine releases, and market transactions. To illustrate the results of implementing the RLROM model in control mode, a data set of inflows, price, and load forecast and initial storage (or forebay) levels conditions were considered in the analysis.

Figure 5.50 presents the monthly generation of the five main plants in the study. The graph demonstrates that despite the lower electricity load in the months of August and September the plants' generation is high to take advantage of the export opportunities available in these months. On the other hand, in June the plants' generation reduces to low levels, and the model recommends the system to store more water during the freshet and early summer months, as there is little market opportunity, and to generate heavily during the winter and late summer months to take advantage of high market prices during these periods.

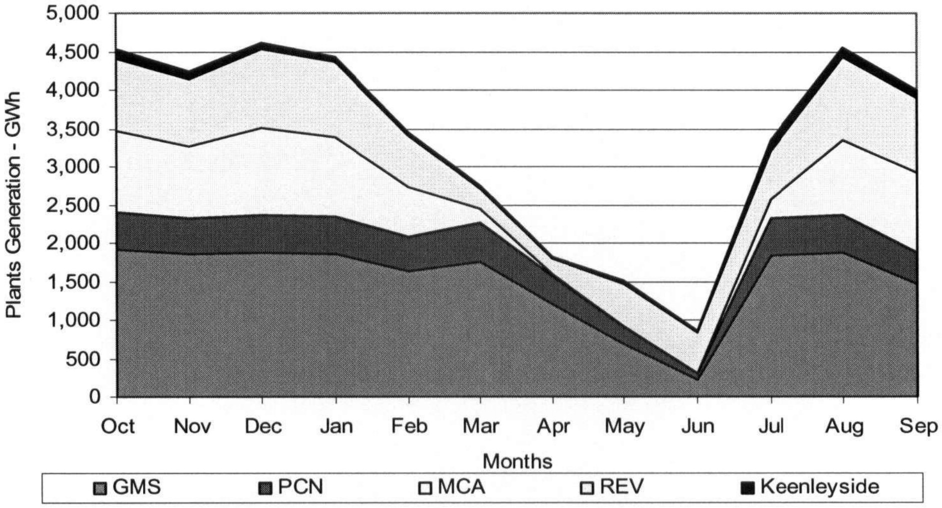


Figure 5.50 Monthly plants generation

Figure 5.51 displays the forecasted heavy load hours (HLH) and light load hours (LLH) market price and the marginal value of energy for GMS and MCA in $/MWh. For the periods from October to February and from August to September, the marginal value of energy for GMS is lower than the LLH market price. Consequently, it is more profitable to release water from GMS than to store it. This is reflected in the high generation policy adopted in these months. In the period from March to July, it is more profitable to generate less at LLH, as the marginal value of energy is higher than the market price and to generate more during HLH where the market price is higher than the marginal value of energy. The lower load and market prices explain the low generation during those months as it is more profitable to import and store water in periods of good market opportunities in the summer and in the winter months. The results presented in Figure 5.51 also indicate that the marginal value of energy for MCA is slightly higher than that for GMS for most of the year. The marginal values of energy for MCA in the period from March to July are even higher than the HLH market prices. This explains the MCA low generation pattern during this period as shown in Figure 5.50.

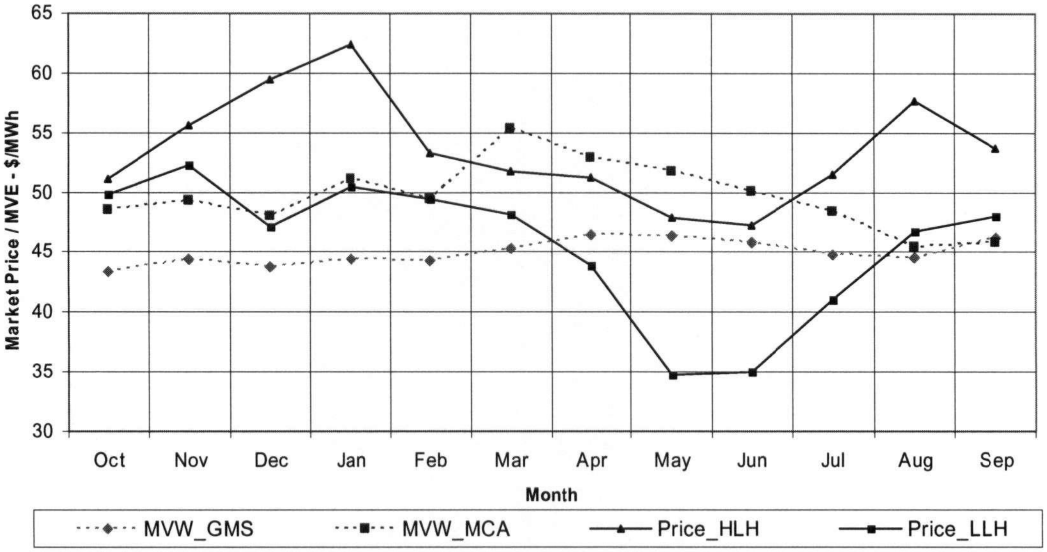


Figure 5.51. Market price and marginal value of energy

Figure 5.52 illustrates the load resource balance, where it can be seen that the system is in a net import mode for the period extending from February to July, while it is in a net export mode for the rest of the year.

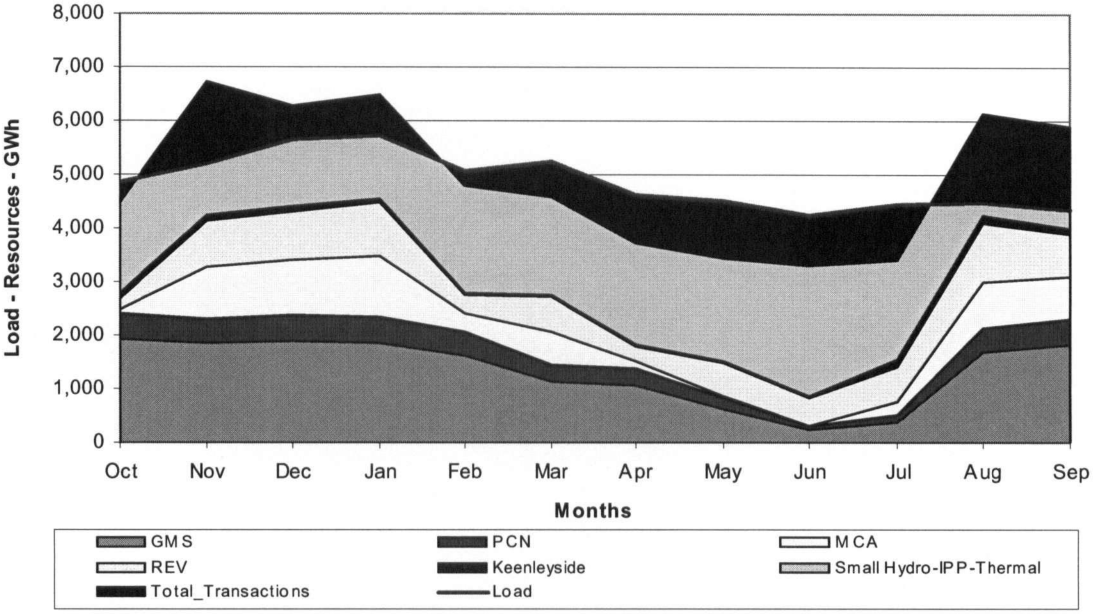


Figure 5.52. Load resource balance

Figure 5.53 shows that about 51% of the total energy is generated by the Peace River system (GMS and PCN) and 49 % was generated by the Columbia River system (REV, MCA, and KNA). About 80% of the Peace generation was produced by GMS and the remaining 20% from PCN. The generation percentages from MCA, REV, and KNA plants on the Columbia system were 42%, 52%, and 6% respectively. A detailed analysis of the results shows that combining the Peace and the Columbia five main plants, together generate about 65% of the energy needed to meet the system load. These percentages closely match the percentages of actual generation of the BC Hydro system (Making the Connection, 2000), which gives confidence in the model results.

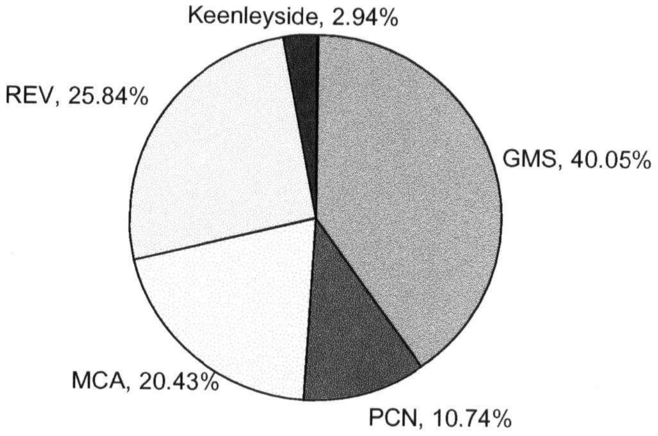


Figure 5.53. Annual distribution of the five main plants generation

To study the effect of changes in the market price forecast on the optimized trading policies, three price scenarios (average, low, and high) were considered with an average annual price of \$46, \$41, and \$51 respectively. Using the moment matching algorithm, three samples were generated for the stochastic variables and were used in this analysis. The results are displayed in Figure 5.54, and it suggests that, for the low price scenario, the system would be importing and exporting in different months of the year, but it would be in a net importing mode (4695 GWh). Futher analysis of the results shows that the system was a net exporter in January and in August, and a net importer for the rest of the months. This is attributed to the fact that the LLH market prices are lower than the marginal value of energy in all months and the HLH was also lower than the marginal value during the spring and fall periods where the system load is low and the market price is low. On the other hand, for the case of the high price scenario, the system would be in net exporter mode. The system is a net exporter in all months with the exception of March to June. As discussed earlier, when the marginal value of energy is higher than the market price, the model tend to release more water for export purposes.

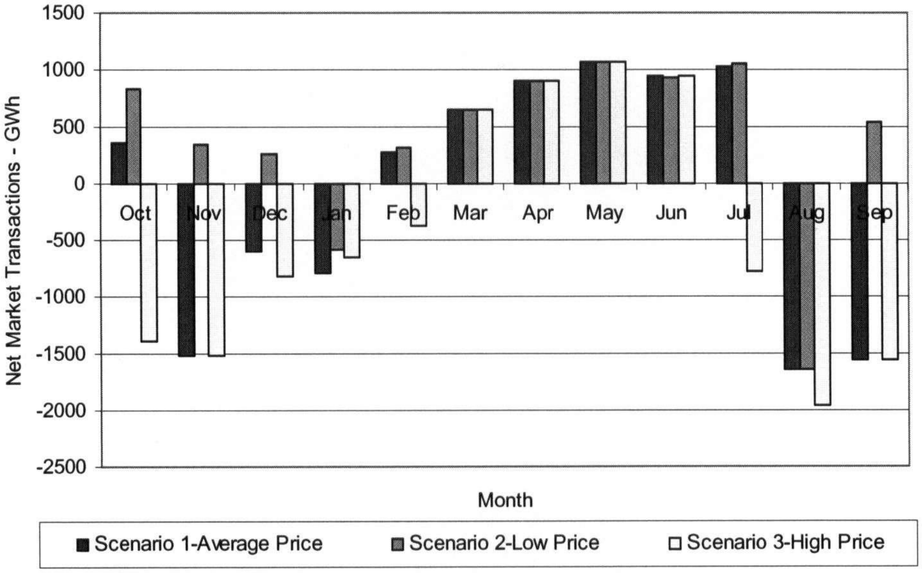


Figure 5.54. Net market transactions for three price scenarios

5.5. Summary

The RL approach methodology, as discussed in chapter 4 was first tested for a small scale single reservoir operation problem, and then on a two reservoir operation problem. The test and convergence results were encouraging and demonstrated the potential capability of the RL approach to handle a larger scale multireservoir operation planning problem. The RL model was then expanded and linked to interact with the generalized optimization model (GOM) as the environment that represents (simulates) the real system. A piecewise linear function approximation technique was adapted to approximate the present value functions. Using this function approximation technique allowed the handling of a larger state space and gave the flexibility to solve the optimization problem at hand. The target storage levels were approximated as a continuous function rather than grid points in the state space, as is usually done in the conventional SDP formulation. Three random variables; natural river inflows, electricity load, and market prices; were considered. A scenario generation-moment matching technique was adopted to generate the inflow, load, and price scenarios. In this way, the serial and cross correlations of the inflows, load, and price random variables were considered in the scenario generation process.

The RLROM model was used to optimize the operation of the BC Hydro main reservoir systems on the Peace and the Columbia Rivers. Two state variables for the GMS and MCA storage were considered. The objective of the model was to estimate the value of water in storage, marginal value of water/energy, and the optimal operation planning decisions, or control policies, including market transactions. In addition, the model optimizes the operation of the five main plants: GMS, PCN, MCA, REV, and KNA. The model was run on a monthly time step with a number of sub-time steps to capture the heavy load hour (HLH) and light load hour (LLH) variation in load and in market prices. The RLROM model results were presented and discussed, including the storage value function and the marginal value of energy/water. The optimized control operation policy established was also presented and discussed. A modified version of the model was developed which allows for use of the results from the learning phase in control mode. Examples of the results of using the RLROM model in the control mode were also presented.

The case study results indicate the dependence of the marginal value of energy in each reservoir on the available storage in other reservoirs. However, there are periods of time and ranges of storage levels where this dependence is not significant. Also, it was shown that the marginal value of energy is largely affected by the constraints imposed on the system operation. The impact of the ice flow constraint in the Peace River on the GMS marginal value of energy was clearly demonstrated. Similarly, the influence of the Columbia Treaty operation on the marginal value of energy for MCA was presented and discussed. The results of using the RLROM model in system control mode indicate that the GMS and the MCA reservoirs are typically drawn down by April to May, just prior to the spring freshet. The drawdown process generally begins in September and October when inflows are low. The reservoir levels begin to increase in May when the turbine discharges are reduced due to lower system demands and increased local inflow. It reaches its highest annual levels in the August to September period.

6. CONCLUSIONS AND RECOMMENDATIONS

6.1. Summary

Strategic planning of multireservoir operation involves taking decisions with uncertain knowledge of future supply (natural inflows), demand (domestic electricity load), and market conditions (market prices). In this research, different modeling approaches and optimization techniques for the operation of multireservoir systems operation have been reviewed. An explicit stochastic modeling approach was found to be the most appropriate way to address the different sources of uncertainty. However, a major obstacle to this approach, that needs to be addressed and resolved, is the high dimensionality of the problem.

The research carried out, for this thesis, concluded that stochastic dynamic programming (SDP) is still a very powerful technique for capturing the essence of the sequential, nonlinear, and stochastic reservoirs optimization problems. However, SDP requires that the values of the transition probabilities and the transition rewards (model of the system) are to be calculated. For large-scale systems that involve several stochastic variables, constructing the system model can be a very difficult task. It is for this reason that DP is said to be plagued by the curse of dimensionality. One possibility to tackle these problems is through the use of machine learning techniques from the field of artificial intelligence (AI), particularly the Reinforcement Learning (RL) technique.

This thesis has explored the use of RL artificial intelligence approach to solve the large-scale operations planning problem of a hydroelectric power multireservoir system. RL is a computational approach for learning from interaction with an environment and from the consequences of actions to derive optimal control strategies. RL has adapted key ideas from various disciplines namely: machine learning, operations research, control theory, psychology, and neuroscience (Sutton and Barto, 1998). The application of the RL technique in the water resources systems domain is relatively new. However, the advantages that RL offers in dealing with large-scale problems, make it a promising area of research in that field. Modern reinforcement learning techniques can be applied to both trial and error learning, without a formal model of the environment, and to planning activities with a formal model of the environment, where an estimate of the state-transition probabilities and immediate expected rewards could easily be evaluated. A detailed review of the RL approach and its main algorithms was conducted and presented in this thesis.

A RL based approach is adopted in this research work to develop a stochastic large-scale medium/ long term multireservoir operation planning model (RLROM). The objective of the model is to develop operating policies that maximizes the expected revenues from energy transactions in a competitive market, while considering the uncertainties, such as future inflows to the reservoirs and the integrated system operation and physical constraints. In this research, the dispatch of the hydropower generation from the different available resources was based on the incremental production costs, which were based on the concept of the marginal value of water in storage. Accordingly, the value of water ($/cms-day) /energy value ($/MWh) was derived as a function of the storage volume of the two major multiyear storage reservoirs on the Peace and Columbia Rivers (GMS and MCA). The uncertainties in the main random variables were expressed in the form of scenarios. A scenario generation-moment matching approach was adapted to generate a set of scenarios for the inflow, market prices, and electricity loads, that preserved the statistical properties, in particular the moments and the correlations of these multiple stochastic variables.

A hydroelectric reservoir optimization model (GOM) based on a linear programming algorithm, which had been previously developed and was in use by BCHydro, was integrated with the RLROM model. Instead of interacting with the real system, the RL agent used the GOM as a model of the environment to provide feedback on its actions and it used this information dynamically to learn the optimum control policy.

6.2. Conclusions

The developed RLROM model was successfully used to handle a large-scale multireservoir operation planning problem. The developed model was applied to BC Hydro's main reservoirs on the Peace and the Columbia Rivers. The RLROM model was designed to run for a multiyear planning horizon (36 months for the case study considered). This allowed the development of realistic planning policies, given the large amount of multiyear storage in the Peace and Columbia River reservoir systems. The results demonstrated that the model was capable of providing realistic answers to the strategic planning questions, including: What is the value of water in storage in each of the multiyear storage reservoirs? What is the marginal value of water /energy in each of the multiyear storage reservoirs? What are the optimal system control decisions including when to buy or sell energy and how much to buy or sell; and how much energy to generate from each of the five plants in the system. The RLROM model, which is presented in this thesis, considers the stochastic variables: domestic load, market prices, and natural inflows. The moment matching technique, which was used for generating scenarios of the inflow, price, and electricity load variables has the great advantage of using only a limited number of scenarios to represent the properties, correlations and cross correlations of an extensive time series based on historical records.

The marginal value of water/energy for MCA and GMS obtained from the RLROM model represent key information that controls the strategic store/release decisions. In addition, this information feeds to other BC Hydro medium and short term optimization models (for example: GOM and STOM). The shorter term models use this information together with the target storage in the trade-off decisions between the shorter term and longer term benefits.

At the beginning of the runs, the model operates in a training mode and the agent is progressively learning the optimal value function and the optimal control policies for the system. Once the model results convergence, the agent knows the value function and the optimal control policies. At this point, the RLROM model can be used on a continuous time bases to maintain optimum control over any time period. Another useful and practical application of the model is to examine the system behavior under certain forecasts of the random variables and to estimate the corresponding marginal value of energy. This sensitivity analysis is useful for understanding the effects of the changes in any of the random variables on the operation policy and on the marginal value of water/energy. For example, being able to estimate the effects of multiyear droughts on system operation is of particular significance to strategic planning for operation of the system.

Reinforcement Learning offered two key advantages in handling the large-scale multireservoir control problem:

(1) RL provides a way to avoid the curse of modeling. By using ***sampling*** (simulation), RL can be applied to large-scale problems without an explicit evaluation and enumeration of all the transition probabilities and their expected immediate rewards.

(2) RL overcomes the curse of dimensionality through the use of ***function approximation*** methods. Those methods require a limited number of elements to specify a value function, instead of possibly millions of states.

A hydroelectric reservoir optimization model (GOM) based on linear programming was integrated with the RLROM model. In this way, the optimization process was extended to include the other reservoirs on the Peace and the Columbia namely, the Dinosaur, Revelstoke, Keenleyside reservoirs, in addition to the main ones at GMS and Mica. This link, has allowed an efficient on-line interaction between the agent and the environment (the operator and the model) during the iterations. Also, it made it possible to capture the effects of the diurnal variation of the price and load on shorter time periods. In addition, the optimal plant generation schedules which are based on the optimal unit commitment and loading (represented in GOM by piecewise linear plant generation functions) is captured in the RLROM model.

The results of the case study implemented using BC Hydro system demonstrated the following:

- The major influence of the available storage in the GMS and MCA storage reservoirs are on the storage value, marginal value, and optimum control policy. However, the study also demonstrated that there are periods of time and ranges of storage levels in GMS and MCA where this influence is not significant.

- The Peace Canyon River minimum ice flow constraint has a major effect on the release policies and on the marginal values in GMS. Similarly, for the Columbia River system the required Columbia Treaty operation influences the marginal value of energy in MCA due to the high releases required from the Keenleyside dam during certain periods of the year.

- The model backs-off plants generation levels to store water, during light load hours (LLH), to take advantage of the difference in heavy load hours (HLH) and light load hours (LLH) price margins and it exports more when there is a market opportunity with high market prices. During the summer, when the domestic loads are low, however, the model is heavily exporting to take advantage of the good market opportunity, particularly in California.

- The moment matching scenario generation results are very encouraging. A sensitivity analysis on the effect of the number of generated scenarios on the mean relative error (MRE) in the moments and correlation determined the appropriate number of scenarios that should be used in the RLROM model runs. The number of generated scenarios, which were derived, demonstrated that this is a very practical approach to obtaining good results, without excessive amounts of computing.

In summary, it can be concluded that the RL approach is a very effective approach to the strategic planning for the operation of large-scale hydroelectric power generation systems. The developed methodology has the potential to significantly improve the operational planning for the integrated BC Hydro system.

6.3. Future Work

There are a number of ways in which the techniques, developed in this thesis, could be enhanced and areas into which it could be expanded. The following is a proposed list of future related research work that could be carried out to extend and enhance this work:

- Investigate the use of multi-agent (independent/cooperative agents) and to use parallel processing for running the model to speed up the learning process and to decrease CPU time.

- Investigate other RL approaches including SARSA, Dyna-Q and other RL techniques that could potentially integrate the planning and learning process.

- Future research is needed to generalize the use of the function approximation approach, outlined in this research. Neural networks are a possible alternative to the currently adopted piecewise linear function approximation technique.

- Automate the process of selecting the values of the RL parameters (learning rate and exploration/exploitation rate).

- Further develop the RL stochastic optimization model to be used for on-line real time control model.

- Extend the model to include other state variables, such as additional plants and check their effects on the marginal values of energy.

- Model the Columbia River Treaty operation constraints in more detail.

- Include regional transmission limits, when modeling regional loads and resources in the BC Hydro system.

REFERENCES

Abdulhai B., R. Pringle, and G. J. Karakoulas, "Reinforcement learning for True Adaptive Traffic Signal Control," *Journal of Transportation Engineering*, ASCE, 129 (3), 278-285, 2003.

Arnold, E., P. Tatjewski, and P. Wolochowicz, "Two Methods for Large-Scale Nonlinear Optimization and Their Comparison on a Case Study of Hydropower Optimization," *Journal of Optimization Theory and Applications*, 81(2), 221-248, 1994.

Barros, M., Tsai F., Yang, S.-L., Lopes, J., and Yeh, W. "Optimization of Large-Scale Hydropower System Operations," *Journal of Water Resources Planning and Management*, ASCE, 129(3), 178-188, 2003.

BC Hydro, "*Making the Connection: the BC Hydro electric system and how it is operated*," 2000.

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

Benders, J. F., "Partitioning Procedures for Solving Mixed Variables Programming Problems," *Numerishe Mathematik*, 4, 238-252, 1962.

Bertsekas, D. and J. Tsitsiklis, *Neuro-Dynamic Programming*, Athena Scientific, Belmont, Mass., 1996.

Bertsekas D. P., "Neuro-Dynamic Programming," *Encyclopedia of Optimization,* Kluwer, 2001.

Bhattachaya B., A. H. Lobbrecht, and D. P. Solomatine, "Neural Networks and Reinforcement Learning in Control of Water Systems," *Journal of Water Resources Planning and Management*, ASCE, 129(6), 2003.

Birge, J. R., "Decomposition and Partitioning Methods for Multistage Stochastic Programs," *Oper. Res.*, 33, 989-1007, 1985.

Birge J. R. and Mulvey J. M., "Stochastic Programming," In *Mathematical Programming for Industrial Engineers*, eds. M. Avriel and B. Golany (Dekker, New York, 1996), 543-574, 1996.

Can, E.K. and M.H. Houck, "Real Time Reservoir Operations by Goal Programming," *Journal of Water Resources Planning and Management*, ASCE, 110(3), 297-309, 1984.

Cario M., B. Nelson, "Modeling and Generating Random Vectors with Arbitrary Marginal Distributions and Correlation Matrix," Technical Report, Department of *Industrial Engineering and Management Sciences*, Northwestern University, Evanston, Illinois, 1997.

Changchit, C., and M. P. Terrell, "CCGP Model for Multiobjective Reservoir System," *Journal of Water Resources Planning and Management*, ASCE, 115(5), 1989.

Crawely, P. and G. Dandy, "Optimal Operation of Multiple Reservoir Systems," *Journal of Water Resources Planning and Management*, ASCE, 119(1), 1-17, 1993.

Coloroni, A., and G. Fronza, "Reservoir Management Via Reliability Programming," *Water Resources Research*, 12(1), 1976.

Crites, R. H. and Barto, A. G., "Improving elevator performance using reinforcement learning," In D. S. Touretzky, M. C. Mozer, M. E. H., editor, *Advances in Neural Information Processing Systems: Proceedings of the 1995 Conference*, pages 1017-1023, Cambridge, MA. MIT Press, (1996).

Dantzig, G. B. and G. Infanger, "Intelligent Control and Optimization under Uncertainty with Application to Hydro Power," *European Journal of Operational Research*, 97, 396-407, 1997.

Davis, R. E. and R. Pronovost, "Two Stochastic Dynamic Programming Procedures for Long Term Reservoir Management," paper presented at IEEE Power Engineering Society summer meeting, IEEE, San Francesco, California, July 9-14, 1972.

Devroye, Luc, *Non-Uniform Random Variate Generation*, Springer-Verlag, New York, 1986.

Dorigo, M., E. Bonabeau., and G. Theraulaz, "Ant Algorithms and Stigmergy," *Future Generation Comput. Systems*, 16, 851-871, 2000.

Dupacova J., Giorgio Consigili, and Stein W. Wallace, "Scenarios for Multistage Stochastic Programs," *Annals of. Operations Research*, 100:25-53, 2000.

Druce, D. J., "Decision Support for Short Term Export Sales From a Hydroelectric System," In *Computerized Decision Support Systems for Water Managers*, Labadie J., L.E. Brazil, I. Corbue, and L.E. Johanson, (eds)., ASCE, 490-497, (1989).

Druce, D. J., "Incorporating Daily Flood Control Objectives Into a Monthly Stochastic Dynamic Programming Model for a Hydroelectric Complex," *Water Resources Research*, 26(1), 5-11, 1990.

Ernst Damien, Mevludin Glavic and Louis Wehenkel,. "Power Systems Stability Control :Reinforcement Learning Framework," Accepted for publication *in IEEE Transactions on Power Systems*, http://www.montefiore.ulg.ac.be/~ernst/IEEEtrans-2003.pdf, 2003.

Evan-Dar, E. and Y. Mansour, "Learning Rates for Q-Learning," *Journal of Machine Learning Research*," 5, 1-25, 2003.

Gasser, Michael, "Reinforcement learning: exploitation and exploration", http://www.indiana.edu/~q320/Notes/rl4.html, 2004.

Giles, J. and W. Wunderlich, "Weekly Multipurpose Planning Model for TVA Reservoir Systems," *Journal of Water Resources Planning and Management*, ASCE, 107(2), 1-20, 1981.

Goldberg, D., "*Genetic Algorithms in Search Optimization and Machine Learning*," Addison-Welsy Publishing Company, Inc., Reading, Massachusetts, 1989.

Gosavi A., "*Simulation-Based Optimization*," Kluwer Academic Publishers, 2003.

Grygier, J. and J. Stedinger, " Algorithms for optimizing Hydropower System Operation," *Water Resources Research*, 21(1), 1-10, 1985.

Foufoula-Georgiou E. and P. K. Kitanidis, "Gradient Dynamic Programming for Stochastic Optimal Control of Multidimensional Water Resources Systems," *Water Resources Research*, 24(8), 1345-1359, 1988.

Fourer, R., D. M. Gay, B. W. Kenigham, "*AMPL: A Modeling Language of Mathematical Programming*," The Scientific Series Press, 2003.

Halliburton, T. "Development of a Medium Term Planning Model for BPA, NSR Information, 1997.

Hiew, K., J. Labadie, and J. Scott, "Optimal Operational Analyses of the Colorado-Big Thompson Project," in *Computerized Decision Support Systems foe Water Managers*, J. Labadie et al., (eds.), ASCE, New York, 632-646, 1989.

Higle J. L. and Sen S., "Stochastic Decomposition: an algorithm for two-stage linear programs with recourse," *Mathematics of Operations Research*, 16:650-669, 1991.

Hogan, A. J., J. G., Morris, and H. E. Thompson, Decision Problems Under Risk and Chance Constrained Programming: Dilemmas in the transition, *Management Science*, 27(6), 698-716, 1981.

Howson, H. R. and N. G. F. Sancho, A New Algorithm for the Solution of Multistate Dynamic Programming Problems, *Math. Programming*, 8, 104-116, 1975.

Hoyland K., M. Kaut and S. W. Wallace, "A Heuristic for Moment-Matching Scenario Generation," *Computational Optimization and Applications*, 24(2-3), 169-185, 2003.

Hoyland K., and S. W. Wallace, "Generating Scenario Trees for Multistage Decision Problems," *Management Science*, 47(2), 295-307, 2001.

Huang, W-C., R. Harboe, and J. J. Bogardi, "Testing Stochastic Dynamic Programming Models Conditioned on Observed or Forecasted Inflows," *Journal of Water Resources Planning and Management*, ASCE, 117(1), 28-36, 1991.

Infanger G., Planning Under Uncertainty: Solving Large-Scale Stochastic Linear Programs, Boyd and Fraser, Danvers, 1994.

Jacobson, H. and Q. Mayne, *Differential Dynamic Programming*, Elsevier, New York, 1970.

Jacobs, J. G., J. Grygier, D. Morton, G. Schulz, K. Staschus, and J. Stedinger, "SOCRATES: A System for Scheduling Hydroelectric Generation Under Uncertainty," in *Models for planning under uncertainty*, Vladimirou et al. (eds.), Baltzer Publishers, Bussum, The Netherlands, 1995.

Jalali, M. R., A. Ashfar, and M. A. Marino, "Improved Ant Colony Optimization," *Scientia Iranica*, 13(3), 295-302, 2006.

Johnson, S.A., J. R. Stedinger, C.A. Shoemaker, Y. Li, and J. A. Tejada-Guibert, "Numerical Spline Stochastic Dynamic Programming," *in Proceedings of the 16th Annual Conference on Water Resources Planning and Management*, ASCE, 137-140, 1989.

Johnson, S.A., J. R. Stedinger, C.A. Shoemaker, Y. Li, and J. A. Tejada-Guibert, " Numerical Solution of Continuous-State Dynamic Programs Using Linear and Spline Interpolation," *Oper. Res.*, 41(3), 484-500, 1993.

Kaelbling P. Leslie, M. L. Littman, and A. W. Moore, "Reinforcement Learning: A Survey," *Journal of Artificial Intelligence Research*, 4, 237-285, 1996.

Kaut M. and S. W. Wallace, "Evaluation of Scenario-Generation Methods for Stochastic Programming," http://work.michalkaut.net/CV_and_study/SG_evaluation.pdf, 2003

Kelman, J., J. R. Stedinger, L. A. Cooper, E. Hsu, and S. Yuan, "Sampling Stochastic Dynamic Programming applied to reservoir operation," *Water Resources Research*, 26(3), 1990.

Ko, S-K, D. Fontane, and J. Labadie, "Multiobjective Optimization of Reservoir Systems Operations," *Water Resources Bulletin*, (28(1), 111-127, 1992.

Kumar, D. N. and M. J. Reddy, "Ant Colony Optimization for Multi-Purpose Reservoir Operation," *Water resources Management*, 20(6), 879-898, 2006.

Kracman, D. R., D.C. McKinney, D. W. Watkins, and L.S. Lasdon, "Stochastic Optimization of the Highland Lakes System in Texas," *Journal of Water Resources Planning and Management*, ASCE, 132(2), 62-70, 2006.

Labadie, J. W., "Dynamic Programming with the Micro-Computer," in *Encyclopedia of Microcomputers*, A. Kent and J. Williams, eds., Vol. 5, 275-337, Marcel Dekker, Inc., New York, 1990.

Labadie, J. W., "Reservoir System Optimization Models," *Water Resources Update Journal*, 107, Universities Council on Water Resources, 1998.

Labadie, J. W., "Optimal Operation of Multireservoir Systems: State-of-the-Art Review," *Journal of Water Resources Planning and Management*, ASCE, 130(2), 93-111, 2004.

Lamond, B. F. and A. Boukhtouta, "Optimizing Long-Term Hydropower Production Using Markov Decision Processes," *International Transactions in Operational Research*, 3, 223–241, 1996.

Lamond, B. F. and Boukhtouta, A., "Water reservoir applications of Markov decision processes," *Handbook of Markov Decision Processes: Methods and Applications*, Eds. Feinberg and A. Schwartz, Kluwer, 537–558, 2001.

Lamond, B. F., "Stochastic optimization of a hydroelectric reservoir using piecewise polynomial approximations," *INFOR*, 41(1), 51–69, 2003.

Lamond, B. F. and A. Boukhtouta, "A Neural Approximation for the Optimal Control of a Hydroplant with Random Inflows and Concave Revenues," *Journal of Energy Engineering*, ASCE, 131(1), 72–95, 2005.

Larson, R., *State Increment Dynamic Programming*, Elsevier, New York, 1968.

Loganathan, G. and D. Bhattacharya, "Goal-Programming Techniques for Optimal Reservoir Operations, *Journal of Water Resources Planning and Management*, ASCE, 116(6), 820-838, 1990.

Loucks, D. and P. Dorfman, " An Evaluation of Some Linear Decision Rules in Chance-Constrained Models for Reservoir Planning and Operation," *Water Resources Research*, 11(6), 1975.

Loucks, D.P., Stedinger, J. R., and Haith, D. A., *Water Resources System Planning and Analysis.* Prentice-Hall Inc., Englewood Cliffs, New Jersy 1981.

Loretan M., "Generating Market Risk Scenarios Using Principal Component Analysis: Methological and Practical Consideration," In *The Measurement of Aggregate Market Risk*, CGFS Publications No. 7, 23-60. Bank for International Settlements, http//www.bis.org/publ.ecsc07.htm, 1997

Lund, J. and I. Ferreira, "Operating Rule Optimization for Missouri River Reservoir System," *Journal of Water Resources Planning and Management*, ASCE, 122(4), 287-295, 1996.

Luri, P. M. and M. S. Goldberg, "An Approximate Method for Sampling Correlated Variables from Partially Specified Distributions," *Management Science*, 44(2), 203-218, 1998.

Martin, Q. W., "Optimal Daily Operation of Surface Water Systems," *Journal of Water Resources Planning and Management*, ASCE, 113(4), 453-470, 1986.

Murray, D. and S. Yakowitz, "Constrained Differential Dynamic Programming and its Application to Multireservoir Control," *Water Resources Research*, 15(5), 1017-1027, 1979.

Mousavi, S., M. Karamouz, and M. B. Menhadj, "Fuzzy-State Stochastic Dynamic Programming for Reservoir Operation," *Journal of Water Resources Planning and Management*, ASCE, 130(6), 460-470, 2004.

Nash, G., "Optimization of the Operation of a Two-Reservoir Hydropower System, Ph.D. Thesis, Department of Civil Engineering, University of British Columbia," 2003.

Parish, Rudolph S., "Generating Random Deviates from Multivariate Pearson Distributions," *Computational Statistics and Data Analysis*, 283-295, 1990.

Pearl, J., "*Heuristics: Intelligent Search Strategies for Computer Problem Solving*," Addison-Wesley, 1984.

Pearson, E.S. and J.W. Tukey, "Approximate means and Standard Deviations Based on Distances between Percentage Points of Frequency Curves," *Biometrika*, 52 (3/4), 533-546, 1965.

Pennanen T. and Koivu M., "Integration quadratures in Discretization of Stochastic Programming," E-Print Series, http://www.speps.info, 2002.

Pereira, M. V. F., "Stochastic Optimization of a Multireservoir Hydroelectric System: A Decomposition Approach," *Water Resources Research,* 21(6), 779-792, 1985.

Pereira, M. V. F., "Optimal Stochastic Scheduling of Large Hydroelectric Systems," Electrical Power & Energy Systems, 11(3), 161-169, 1989.

Pereira, M. V. F. and Pinto, L.M.V.G., "Multi-stage Stochastic Optimization Applied to Energy Planning," *Mathematical Programming*, 52(3), 359-375, 1991.

Pereira, M. V. F, Campodonico, N., Kelman, R., "Application of Stochastic Dual DP and Extensions to Hydrothermal Scheduling," *PRSI Technical Report*, http://www.psr-inc.com.br/reports.asp, 1999.

Pflug G. C., Scenario Tree Generation For Multiperiod Financial Optimal discretization," Mathematical Programming, 89(2), 251-271, 2001.

Piekutowski, M.R., Litwonowicz, T. and Frowd, R. J., "Optimal Short-Term Scheduling for a Large-Scale Cascaded Hydro System," *IEEE Transactions on Power Systems*, 9(2), 292-298, 1993.

Picacardi, C., and R. Soncini, "Stochastic Analysis Program (SAP) for SDP Optimization," Dep. Of Environ. Eng., Cornell Univ., Ithaca, N.Y., 1991.

Pronovost, R. and J. Boulva, "Long-Range Operation Planning of a Hydro-Thermal System Modeling and Optimization," Meeting the Canadian Electrical Association, Toronto, Ont. March 13-17, 1978.

Raman, H. and V. Chandramouli, "Deriving a General Operating Policy for Reservoirs Using Neural Networks," *Journal of Water Resources Planning and Management*, 122(5), 342-347, 1996.

Reis, L. F. R., G. A. Walters, D. Savic, and F. H. Chaudary, "Multi-Reservoir Operation Planning Using Hybrid Genetic Algorithm and Linear Programming (GA-LP): An Alternative Stochastic Approach," *Water Resources Management*, 19(6), 831-848, 2005.

Reznicek, K., and Cheng, T.C.E., "Stochastic Modeling of Reservoir Operations," *European Journal of Operational Research*, 50, 235-248, 1991.

Robbins, H. and S. Monro, "A Stochastic Approximation Method," *Ann Math. Statict.*, (22), 400-407, 1951.

Romisch W. and H. Heitsch, "Scenario Reduction Algorithms in Stochastic Programming," Computational Optimization and Applications, 24(2-3), 187-206, 2003.

Rotting, T. A. and Gjelsvik, A., "Stochastic Dual Dynamic Programming for Seasonal Scheduling in the Norwegian Power System," IEEE Transactions on Power Systems, 7(1), 273-279, 1992.

Russell, S. O., and P. E. Campbell, "Reservoir Operating Rules with fuzzy programming," *Journal of Water Resources Planning and Management*, ASCE, 122(3), 165-170, 1996.

Saad, M., A. Turgeon, P. Bigras, and R. Duquette, "Learning Disaggregation Technique for the Operation of Long-Term Hydrotechnical Power Systems," *Water Resources Research*, 30(11), 3195-3202, 1994.

Singh, S. and D. Bertsekas, "Reinforcement Learning for Dynamic Channel Allocation in Cellular Telephone Systems," *Advances in Neural information Processing Systems, Conf. Proc.*, MIT Press, Cambridge, Mass., 1996

Shawwash Z.K., T. Siu, and S. Russel, "The BC Hydro Short Term Hydro Scheduling Optimization Model" *IEEE Transactions on Power Systems*, 2000.

Shawwash Z.K. "*A Decision Support System for Real-Time Hydropower Scheduling in a Competitive Power Market Environment*," PhD Thesis, Dept. of Civil Engineering, UBC, 2000.

Sutton, R.S., "Reinforcement Learning," In Rob Wilson and Frank Keil (Eds.), *The MIT Encyclopedia of the Cognitive Sciences,* MIT Press, 1999, ftp://ftp.cs.umass.edu/pub/anw/pub/sutton/Sutton-99-MITECS.pdf.

Sutton, R. S. and A. G. Barto, "*Reinforcement Learning – An Introduction,*" MIT Press, Cambridge, Massachusetts, 1998.

Sutton, R.S., "Learning to Predict by the Methods of Temporal Difference," *Machine Learning*, Kluwar Academic Publishers, 3: 9-44, 1988.

Takeuchi, K., " Chance-Constrained Model for Real-Time Reservoir Operation Using Drought Duration Curve," *Water Resources Research*, 22(4), 1986.

Tejada-Guibert, J., J. Stedinger, and K. Staschus, "Optimization of the Value of CVP's Hydropower Scheduling," *Journal of Water Resources Planning and Management*, ASCE, 116(1), 53-71, 1990.

Tejada-Guibert, J. A., S. Johnson, and J. R. Stedinger, "Comparison of Two Approaches for Implementing Multireservoir Operating Policies derived Using Stochastic Dynamic Programming," *Water Resources Research* 29(12), 3969-3980, 1993.

Tejada_Guibert, J. A., S. Johnson, and J. R. Stedinger, "The Value of Hydrologic Information in Stochastic Dynamic Programming Models of a Multireservoir System," *Water Resources Research* 31(10), 2571-2579, 1995.

Tesauro, G. J., "Temporal difference learning and TD-Gammon," *Communications of the ACM* (38), 58-68, 1995.

Tilmant, A., E. Persoons, M. Vanclooster, " Deriving efficient reservoir operating rules using flexible stochastic dynamic programming," *Proc. 1st Int. Conf. on Water Resources Management*, WIT Press, U.K. 353-364, 2001.

Tsitsiklis, J. N, "Asynchronous Stochastic Approximation and Q-Learning," Machine Learning (16), 185-202, 1994.

Turgeon A., " Optimal operation of Multireservoir Power Systems with Stochastic Inflows," *Water Resources Research*, 16(2), 275-283, 1980.

Turgeon, A., " Optimal Short-Term Hydro Scheduling From the Principle of Progressive Optimality," *Water Resources Research*, 17(3), 481-486, 1981.

Turgeon, A., "An Application of Parametric Mixed-Integer Linear Programming to Hydropower Development," *Water Resources Research*, 23(3), 1987.

Turgeon A., "Solving a Stochastic reservoir management problem with multilag autocorrelated inflows," *Water Resources Research*, 41(12), W12414, 2005.

Unver, O. and L. Mays, "Model for Real-Time Optimal Flood Operation of a Reservoir System," *Water Resources Management*, 4, 21-46, 1990.

Valdes, J. B., Montburn-Di Filippo, J., Strzepek, K. M., and Restepo, P. J., "Aggregation-Disaggregation Approach to Multireservoir Operation," *Journal of Water Resources Planning and Management*, 118(4), 423-444, 1992.

Van Slyke, R. and R.J.-B. Wets, " L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Programming, *SIAM J. Appl. Math.*, 17, 638-663, 1969.

Vogel, R. M. and J. R. Stedinger, "The value of streamflow models in overyear reservoir design applications," *Water Resources Research*, 24(9), 1483-1490, 1988.

Wardlaw R. and M. Sharif, " Evaluation of Genetic Algorithms for Optimal Reservoir System Operation," *Journal of Water Resources Planning and Management*, 125(1), 25-33, 1999.

Watkins, C. J. C. H. *Learning from Delayed Rewards.* Ph.D. Thesis. King's College, University of Cambridge, Cambridge, UK, 1989.

Wen-Ceng Huang, Yuan Lun-Chin, and Lee Chi-Ming, "Linking Genetic Algorithms with Stochastic Dynamic Programming to Long-Term Operation of a Multireservoir System," *Water Resources Research*, 38(12), 1304, 2002.

Wilson, G., "Reinforcement Learning: A New Technique for the Real-Time Optimal Control of Hydraulic Networks," *Proc. Hydroinformatics Conf., Balkema*, The Netherlands, 1996.

Yakowitz S., "Dynamic Programming Applications in Water Resources," *Water Resources Research*, 18(4), 673-696, 1982.

Yeh, W. W-G. , " Reservoir management and operation models: A State-of-the-art Review," *Water Resources Research,* 21(12), 1797-1818, 1985.

Young, G. K., "Finding Reservoir Operation Rules," *J. Hydraul. Div.*, ASCE, 93(HY6), 297-321, 1967.

Zhang, W. and Dietterich, T. G., "High-performance job-shop scheduling with a time--delay TD (l) network," *Advances in Neural Information Processing Systems: Proceedings of the 1995 Conference*, pages 1024--1030, Cambridge, MA. MIT Press, 1996.

APPENDIX A *Q*_LEARNING NUMERICAL EXAMPLE

Let us define a Markov chain with two states s_1, s_2 and at each state two actions a_1, a_2 are allowed:

States = $\{s_1, s_2\}$

Actions: $A(s_1) = \{a_{11}, a_{12}\}$, $A(s_2) = \{a_{21}, a_{22}\}$

Rewards: R(s,a) = $r_t(s_1, a_{11})$, $r_t(s_1, a_{12})$, $r_t(s_2, a_{21})$, $r_t(s_2, a_{22})$

The transition probability matrix (TPM) associated with action 1 and 2 are P_t^1 and P_t^2:

$$P_t^1 = \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix} \qquad P_t^2 = \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix}$$

The transition reward matrix (TRM) associated with action 1 and 2 are R_t^1 and R_t^2:

$$R_t^1 = \begin{bmatrix} 6 & -5 \\ 7 & 12 \end{bmatrix} \qquad R_t^2 = \begin{bmatrix} 10 & 17 \\ -14 & 13 \end{bmatrix}$$

The objective is to maximize the expected rewards. A discount factor of 0.8 is assumed in this example. The learning rate α is defined as: $B/n(s,a)$, where B is a constant assumed 0.1 and n is the number of state-action pair visits. The exploration rate ε is assumed C/t, where C is a constant assumed 0.9 and t is the number of time periods. In the beginning, the learning agent tends to explore more and select non-greedy actions. As such, it randomly select with a probability ε a different action from the action suggested by the policy learned so far $\pi(s)$, where $\pi(s) = \arg\max_a Q^\pi(s,a)$.

As the number of episodes is getting larger, the learning agent will gradually turn to be greedy and select the best action estimated at state s most of the time with probability 1- ε. Calculations of the first ten steps are presented in Table A.1. The following is a detailed description of the calculation procedure in this *Q*-Learning algorithm example:

Period 1:

- initialize all the state-action value function (*Q*-Table) to 0
- randomly select an initial state *s*, the selected state is s_1
- calculate the exploration rate = *C/t* = 0.9/1.0 = 0.9
- randomly sample an action *a* with probability 0.9, the selected action is a_1
- update the number of visits to state s_1 and action a_1, $n(s_1, a_1) = 1$
- simulate action 1 and observe the transition to next state and the associated rewards. The next state is s_2 and the reward $r(s_1,a_2)$ is -5.
- the learning rate $\alpha = B/\mathrm{n}(s_1, a_1) = 0.1/1.0 = 0.10$
- update the Q-table using the equation: $Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$

$$Q(s_1, a_1) \leftarrow 0 + 0.1[-5 + 0.8^* \max\{0,0\} \ - 0] = -0.50$$

Period 2:

- the current state is s_2. Calculate the exploration rate $\varepsilon = C/t = 0.9/2.0 = 0.45$
- sample the actions to select the best (greedy) action with probability 0.45 and a random action with probability 0.55. The selected action is a_2
- update the number of visits to state s_2 and action a_2, $\mathrm{n}(s_2, a_2) = 1$
- simulate the action and observe the transition to next state and the associated rewards. The next state is s_2 and the immediate reward $r(s_2,a_2)$ is 13
- the learning rate $\alpha = B/\mathrm{n}(s_2, a_2) = 0.1/1.0 = 0.10$
- update the *Q*-table:

$$Q(s_2, a_2) \leftarrow 0 + 0.1[13 + 0.8^* \max\{0,0\} \ - 0] = 1.30$$

.***Period 10***:

- the current state is s_2. Calculate the exploration rate $\varepsilon = C/t = 0.9/10.0 = 0.09$

- sample the actions to select the best (greedy) action with probability 0.91 and a random action with probability 0.09. The best action at state s_2 is a_1 as (0.7>0.607),. select action a_1

- update the number of visits to state s_2 and action a_1, $\mathrm{n}(s_2, a_1) = 2$

- simulate the action and observe the transition to next state and the associated rewards. The next state is s_1 and the immediate reward $r(s_2,a_1)$ is 7

- the learning rate $\alpha = B/\,\mathrm{n}(s_2, a_2) = 0.1/2.0 = 0.05$

- update the Q-table:

$Q(s_2, a_1) \leftarrow 0.7 + 0.05[7 + 0.8*\max\{-0.103, 3.016\} - 0.7] = 1.136$

Table A.1. Sample calculations for the Q-Learning example

Time Period	State S_i	Probability Matrix				Reward Martix				Exploration Rate	□-Greedy Action Selection	Visits		Simulate Action & Observe Feedback		Learning Rate	Q-Values	
		a1		a2		R1		R2				a1	a2	Trans. state	reward	□	a1	a2
		S1	S2	S1	S2	S1	S2	S1	S2									
1											0.2879124			0.709933				
	s1	0.7	**0.3**	0.9	0.1	6	**-5**	10	17	0.9000	**a1**	1	0	**s2**	-5	0.100	-0.500	**0**
	s2	0.4	0.6	0.2	0.8	7	12	-14	13			0	0				**0**	**0**
2											0.1668252			0.4037938				
	s1	0.7	0.3	0.9	0.1	6	-5	10	17	0.4500		1	0				-0.5	**0**
	s2	0.4	0.6	0.2	**0.8**	7	12	-14	**13**		**a2**	0	1	**s2**	13	0.100	0	**1.30**
3											0.2584773			0.3560084				
	s1	0.7	0.3	0.9	0.1	6	-5	10	17	0.3000		1	0				-0.5	**0**
	s2	**0.4**	0.6	0.2	0.8	**7**	12	-14	13		**a1**	1	1	**s1**	7	0.100	0.70	**1.3**
4											0.0484106			0.9748682				
	s1	0.7	0.3	0.9	**0.1**	6	-5	10	**17**	0.2250	**a2**	1	1	**s2**	17	0.100	-0.5	**1.804**
	s2	0.4	0.6	0.2	0.8	7	12	-14	13			1	1				0.7	**1.3**
5											0.2377849			0.1620204				
	s1	0.7	0.3	0.9	0.1	6	-5	10	17	0.1800		1	1				-0.5	**1.804**
	s2	0.4	0.6	**0.2**	0.8	7	12	**-14**	13		**a2**	1	2	**s1**	-14	0.050	**0.7**	0.607
6											0.9477175			0.6368037				
	s1	**0.7**	0.3	0.9	0.1	6	-5	10	17	0.1500	**a1**	2	1	**s1**	6	0.050	-0.103	**1.804**
	s2	0.4	0.6	0.2	0.8	7	12	-14	13			1	2				**0.7**	0.607
7											0.1759355			0.8324002				
	s1	0.7	0.3	**0.9**	0.1	6	-5	**10**	17	0.1286	**a2**	2	2	**s1**	10	0.050	-0.103	**2.286**
	s2	0.4	0.6	0.2	0.8	7	12	-14	13			1	2				**0.7**	0.607
8											0.8248643			0.1913321				
	s1	0.7	0.3	**0.9**	0.1	6	-5	10	17	0.1125	**a2**	2	3	**s1**	10	0.033	-0.103	**2.604**
	s2	0.4	0.6	0.2	0.8	7	12	-14	13			1	2				**0.7**	0.607
9											0.9071856			0.9812344				
	s1	0.7	0.3	0.9	**0.1**	6	-5	10	**17**	0.1000	**a2**	2	4	**s2**	17	0.025	-0.103	**3.016**
	s2	0.4	0.6	0.2	0.8	7	12	-14	13			1	2				**0.7**	0.607
10											0.9071856			0.0694871				
	s1	0.7	0.3	0.9	0.1	6	-5	10	17	0.0900		2	4				-0.103	**3.016**
	s2	**0.4**	0.6	0.2	0.8	**7**	12	-14	13		**a1**	2	2	**s1**	7	0.050	**1.136**	0.607