THE UNIVERSITY OF MANITOBA

AN APPLICATION OF SUCCESSIVE LINEAR PROGRAMMING TO THE OPTIMIZATION OF AN INTERCONNECTED HYDRO UTILITY OPERATION

by

© KAROLJ REZNICEK

A Thesis

Submitted to the Faculty of Graduate Studies in Partial

Fulfillment of the Requirements for the Degree of Master

of Science

DEPARTMENT OF CIVIL ENGINEERING

Winnipeg, Manitoba

DECEMBER 1988

ISBN 0-315-51576-7

AN APPLICATION OF SUCCESSIVE LINEAR PROGRAMMING
TO THE OPTIMIZATION OF AN INTERCONNECTED
HYDRO UTILITY OPERATION

BY

KAROLJ REZNICEK

A thesis submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements of the degree of

MASTER OF SCIENCE

ACKNOWLEDGMENTS

I would like to express my gratitude to all people who provided me with support and understanding while working on this thesis.

I am specially indebted to Professor S.P. Simonovic for his advice, guidance and encouragement throughout my studies at the University of Manitoba. I am also thankful for the valuable comments of Mr A.D. Cormie and Mr P.E. Barritt-Flatt during the research which preceded this thesis. My gratitude goes to Professors I.C. Goulter and D.H. Burn for their assistance and suggestions pertaining this work.

I wish to acknowledge the financial support from Manitoba Hydro, approved by the Research and Development Committee, as well as the availability of Energy Management and Maintenance Analysis program and Manitoba Hydro data. I would also like to thank Mr. J.C. Grygier and Mr. J.R. Stedinger for providing me with the code of the Successive Linear Programming model developed at Cornell University.

I am grateful to my fellow graduate students in water resources at the University of Manitoba for being sincere friends and encouraging my work.

Special thanks goes to Wendy Seversen for helping me to edit and arrange this work.

Finally, I deeply appreciate the help and understanding of my parents throughout my studies.

ABSTRACT

The operation planning of a hydro-electric utility is a complex problem involving decisions about reservoir releases, energy supplies and many other production related problems during a certain planning period. Mathematical modelling (simulation and optimization techniques) is widely used to aid the decision making process. This work presents a deterministic Linear Programming (LP) based optimization model. The objective is to maximize the energy export benefits of the utility, while minimizing the costs of satisfying the domestic power demand over the planning period. For the specified reservoir inflow and power demand scheme, decisions about the energy production, export and import have to be made for each time step.

An iterative algorithm named EMSLP (Energy Management by Successive Linear Programming) was developed to solve the optimization problem. The EMSLP algorithm has two iteration levels: at the first level a stable solution is sought, and at the second the interior of the feasible region is searched to improve the objective function whenever its value decreases.

The EMSLP algorithm has been tested using the Manitoba Hydro system data. To evaluate the performance of the algorithm a comparative study has been made with the EMMA (Energy Management and Maintenance Analysis) program used in the Manitoba Hydro practice. The results of the comparison have shown a number of advantages of the EMSLP algorithm.

TABLE OF CONTENTS

	PAGE
Acknowledgements	i
Abstract	ii
Table of Contents	iii
List of Tables	vi
List of Figures	vii
Chapter 1: Introduction	1
1.1 Problem Statement	1
1.2 The Concept of Successive Linear Programing	3
1.3 Modelling approximations	4
Chapter 2: Review of Previous Work	6
2.1 Mathematical Modelling Applied to Hydro Power Generation	6
2.1.1 Dynamic Programming	7
2.1.2 Nonlinear Programming	13
2.1.3 Simulation	15
2.2 Linear Programming Applications	16
2.2.1 Deterministic Models	18
2.2.2 Stochastic Models	23
2.3 Description of the EMMA Program of Manitoba Hydro	25
2.4 Description of the Successive Linear Programming Model of J. Grygier	39

PAGE

Chapter 3: Energy Management by Successive Linear Programming	50
3.1 The Problem Formulation	50
3.1.1 The Constraint Set	52
3.1.2 The Objective Function	56
3.2 The EMSLP Algorithm	58
Chapter 4: Evaluation of EMSLP	63
4.1 The Case Study	63
4.2 The Results of Comparison with EMMA	67
4.2.1 Final Storage Value Variation	68
4.2.2 System Load Variation	68
4.2.3 Release Limit Variation	76
Chapter 5: Conclusions	92
5.1 Discussion of EMSLP-EMMA Comparison Results	92
5.2 Developments of the EMSLP Research	95
5.3 Differences between EMSLP and Grygier's SLP Model	95
5.4 Directions of Future Research	96
Appendix A: Notation	98
Appendix B: Bibliography	100
Appendix C: Tables of the Case Study Results	105

PAGE

Appendix D: The Fortran Program of EMSLP Applied to the Case Study 112

D.1 Program Structure 112

D.2 Variables 114

D.3 Input Data 120

D.4 Program Execution 121

D.5 Output Report 126

Appendix E: Source Code 128

Appendix F: Sample Input and Output File 178

LIST OF TABLES

		PAGE
Table 1.	The discretized load duration curve data	65
Table 2.	The inflow scheme	65
Table 3.	The energy price structure	65
Table C-1.	EMSLP results for varying storage benefits	106
Table C-2.	EMMA results for varying storage benefits	107
Table C-3.	EMSLP results for varying system demand	108
Table C-4.	EMMA results for varying system demand	109
Table C-5.	EMSLP results for varying release limits	110
Table C-6.	EMMA results for varying release limits	111

LIST OF FIGURES

		PAGE
Figure 1.	Load duration curve modelling	31
Figure 2.	EMMA flowchart	38
Figure 3.	Flowchart of the SLP algorithm (Grygier, 1983)	47
Figure 4.	EMSLP flowchart	60
Figure 5.	Schematic Illustration of the Case Study	64
Figure 6.	Comparison of reservoir levels for $3600/KCFS-day ending storage value	69
Figure 7.	Comparison of reservoir levels for $4000/KCFS-day ending storage value	70
Figure 8.	Comparison of reservoir levels for $4200/KCFS-day ending storage value	71
Figure 9.	Comparison of reservoir levels for $4500/KCFS-day ending storage value	72
Figure 10.	Comparison of reservoir levels for $4600/KCFS-day ending storage value	73
Figure 11.	Comparison of reservoir levels for $4700/KCFS-day ending storage value	74

		PAGE
Figure 12.	Comparison of reservoir levels for $4800/KCFS-day ending storage value	75
Figure 13.	Comparison of reservoir levels for system load equal to 0.3 times the original load	77
Figure 14.	Comparison of reservoir levels for system load equal to 0.5 times the original load	78
Figure 15.	Comparison of reservoir levels for system load equal to 0.8 times the original load	79
Figure 16.	Comparison of reservoir levels for system load equal to 1.0 times the original load	80
Figure 17.	Comparison of reservoir levels for system load equal to 1.5 times the original load	81
Figure 18.	Comparison of reservoir levels for system load equal to 2.0 times the original load	82
Figure 19.	Comparison of reservoir levels for a release limit of 5.67 m^3/sec	84
Figure 20.	Comparison of reservoir levels for a release limit of 8.57 m^3/sec	85
Figure 21.	Comparison of reservoir levels for a release limit of 11.35 m^3/sec	86

		PAGE
Figure 22.	Comparison of reservoir levels for a release limit of 14.19 m^3/sec	87
Figure 23.	Comparison of reservoir levels for a release limit of 19.86 m^3/sec	88
Figure 24.	Normalized objective function difference for storage value variation	89
Figure 25.	Normalized objective function difference for load variation	90
Figure 26.	Normalized objective function difference for release limit variation	91

CHAPTER 1.

INTRODUCTION

The optimization of hydro power production (in broader sense reservoir operation) has been a problem addressed by many researchers in the past three decades. Due to the complexity of the problem none of the numerous optimization techniques could model all the pertinent characteristics of a reservoir system operation (multiple-reservoirs, multiple time-periods, stochastic inflows, and nonseparable objective functions). However, this does not mean that there were no successful applications of operations research techniques to the problem. For real world problems, some simplifications are required to take into account the characteristics of the reservoir system, the available data, and the modelling goal. The simplified problem can be successfully modelled. The model can be used for planning and operation purposes keeping in mind the assumptions made.

An algorithm for optimal midterm operation of an interconnected hydro utility with a deterministic input is presented in this thesis.

1.1 PROBLEM STATEMENT

There are three major time horizons used in a hydro utility planning procedure: long, mid-, and short-term planning. Long-term planning involves making strategic decisions about the system for several decades in the future.

Midterm operation problems require decisions to be made in the system on a weekly or monthly basis over a yearly period. Short-term planning requires making decisions on a daily or hourly basis.

The midterm planning horizon is divided into multiple weekly or monthly time steps to cover the whole planning period. The hydro utility's goal in the case of Manitoba Hydro is to satisfy the domestic load throughout the planning period. In order to comply with its obligations the utility has to import energy in the periods of energy deficiencies, i.e., when there is insufficient water for release through the hydro power plants, or insufficient capacity in other domestic generating stations. On the other hand, additional benefit can be obtained by exporting the excess energy in periods when the production exceeds the domestic demand. The energy price structure on the power market can also allow some additional benefits by rationally scheduling the export-import policy (e.g., buying off-peak priced energy to satisfy the domestic load and storing water for on-peak priced production and possibly export).

To prevent an operation which would empty the reservoirs at the end of the planning period (i.e., to be greedy in achieving higher benefits) there is an assigned value of water for the last time period. The value reflects the future benefits from the stored water.

Deterministic planning means that the stream flows in the hydraulic system, the domestic load and the energy prices are known in advance for the whole planning period in each time step. These assumptions are far from realistic, because each of the above model inputs is uncertain. Since the model does not incorporate the uncertain future explicitly, it has to be evaluated implicitly by sensitivity analysis. The optimization has to be performed with different scenarios of stochastic data to evaluate their impact on the operation policy. The way to improve the operation policy is to update the forecasted input data whenever additional information is available and to run the model again. Therefore, the optimal policy obtained from the model is implemented only for the first time step. For the next time step, the model optimizes with the new forecast.

1.2 THE IDEA OF SUCCESSIVE LINEAR PROGRAMMING

The algorithm uses the technique of Linear Programming (LP) to optimize the operation. Like the other mathematical programming techniques LP requires the formulation of an objective function to be optimized and a constraint set to limit the feasible solution space. In the problem of hydro production optimization the objective is to maximize production benefits and minimize costs with respect to the constraints which describe the system. In addition, LP requires linear relationships in the objective function and the constraints, too. The problem of hydro production optimization is nonlinear. Some of the nonlinearities can be approximated by piece wise linearization to the desired degree of accuracy, and they do not represent a serious obstacle for the LP application. The major difficulty lies in the nonseparable character of hydro production function. The multiplication of release, storage, plant efficiency and conversion constant give the produced hydro energy. The release and storage are both decision variables in the model and they must be separated in order to apply the LP technique. There are numerous approximations which can be applied to linearize the relationship. The common to most of them is that the solution to the problem is obtained in a sequence of iterative LP solutions. The assumed values used to linearize the original function are updated after each iteration until the input is close enough to the output value. This technique to solve nonlinear problems using LP is called Iterative Linear Programming (ILP).

The algorithm presented in this thesis applies an approximation which belongs to a special class of ILP. The nonlinear function is approximated around a chosen point in the decision space by its first order Taylor series expansion. The algorithms developed on the basis of this approximation are called Successive Linear Programming (SLP) algorithms.

1.3 MODELLING APPROXIMATIONS

In addition to the already introduced simplified planning with deterministic input, modelling of the hydro utility operation requires a number of other facilitating assumptions. The calculation of the plant head and efficiency is performed simultaneously by determining their product called Energy Rate Function

(ERF). The calculation does not take into account the impact of discharge, i.e. ERF is assumed to be the function of the storage exclusively. In reality the efficiency and the tail water are both dependent on the discharge. In addition, the representative ERF value for a time step is assumed to be the average of the function value for the storage at the beginning and at the end of the time step.

It is also assumed that there is a linear relationship between the volume of the water stored in the reservoirs at the end of the planning period, and the future benefits from that storage. In reality the return per unit volume diminishes with the increase of the amount stored. In the model it is assumed to have a constant return per unit volume.

The model assumes that the generating capabilities are available for production during the whole planning period. In reality every plant must have outages due to maintenance. The operation planning has to take that into account. The problem of maintenance can be included into the model in the same manner as is done in the EMMA program, Manitoba Hydro (1986).

In case of multiple reservoir modelling the travelling times of water between the reservoirs can be ignored. This assumption can be valid only for midterm planning when the travelling time is negligible compared to the time step length. In cases when this is not true additional modelling is required (e.g., by introducing an artificial reservoir) to take into account the travelling time between the reservoirs.

CHAPTER 2.

REVIEW OF PREVIOUS WORK

2.1 MATHEMATICAL MODELLING APPLIED TO HYDRO POWER GENERATION

The problem of optimal management of hydro power generation has been actively studied by a large number of researchers in various academic and research institutions and electric utilities. Many successful applications of mathematical models have been made. The method chosen depends on the characteristics of the hydraulic and electric system, on the availability of data, and on the objectives and constraints specified.

The operation of a hydro utility is nonseparably connected to the problem of reservoir management. Hydro production is often one of the major purposes (if not the only) of building a reservoir. The hydro production of a power plant during a time period is a function of the released water and the forebay storage level. The forebay is usually a reservoir. Even if it is not, as in the case of run of river plant, the operation of the generation station can be largely influenced by the releases from an upstream reservoir. The only hydro power plants which are not related to reservoirs are the run of river hydro plants built on unregulated rivers, but these are rare.

The methods applied for reservoir management, and also used for hydro-electrical system operation, can be classified according to Yeh (1985) into four major groups.

a) Linear programming;

b) Dynamic programming;

c) Nonlinear programming;

d) Simulation;

Combinations of the above methods have also been reported in the literature. Since the application of Linear programming (LP) will be discussed in detail in Section 1.2, at this point the last three methods are addressed.

2.1.1 Dynamic Programming

Dynamic programming (DP) is a technique for optimization of multistage decision processes. It is used extensively to optimize water resources systems. The popularity of DP is due to the fact that the nonlinear and stochastic characteristics of water resources systems can be translated into a DP formulation without difficulties. DP is well suited to handle deterministic short term (daily, hourly), and stochastic mid term (monthly or yearly) operation problems (Larson and Keckler, 1969). A deterministic model for a power generation system with pumpback developed by Hall and Roefs (1966) also shows the applicability of the method to mid term planning. Young (1967) proposes a method to deal with the stochastic character of the inflows while optimizing with deterministic DP. Reservoir operating rules are obtained using a combination of stream flow generation and DP optimization of releases. The stochastic character of the inflows was taken into account by generating a long inflow sequence by a Monte Carlo technique. The release policy for this sequence was optimized by a deterministic forward DP. The reservoir operating rule is a regression function of the release to the storage, inflow and forecast of the next inflow. The generated/forecasted inflows and the optimal storages are used as a sample to estimate the coefficients of the regression function by the least square method. Applying the rule, the economic loss as a function of the release is minimized for annual usage of a single reservoir.

However, Yeh (1985) has stated that the major drawback of DP in its original form is the inability to handle big multiple reservoir systems. The memory and computing time requirements are the major limiting factors. Each reservoir requires at least one state variable (e.g., storage) which can have several values (in the discrete case) at every stage (e.g., time step). The possible number of combinations (state vectors) to be explored grow exponentially with the number of state variables at each stage. The computational burden is unbearable for a system of more than a few reservoirs. This problem is called the "curse of dimensionality". In this section several DP based models are presented based on the nature of the applied methodology rather than the chronological order of appearance.

The remedial measure to alleviate the "curse of dimensionality" is to decompose the complex multiple state variable problem into a series of sub-problems which can be solved recursively. The methods of dimension reduction beside the decomposition of the original problem also follow an iterative solution procedure. One of the methods is the Incremental DP (IDP) used by Larson and Keckler (1969), systematized and referred to by Heidari et al. (1969) as Discrete Differential DP (DDDP). The method starts with a trial state trajectory satisfying a specific set of initial and final conditions and applies the DP recursive equation to the neighborhood of this trajectory. At the end of each iteration step a locally improved trajectory is obtained and used as the initial trajectory for the next step. The procedure stops when no further improvement is identified, and it is assumed that a local optimum is found.

Another method to alleviate the curse of dimensionality is called Incremental DP with Successive Approximations (IDPSA). The concept is to decompose the multiple-state variable DP problem to a number of subproblems of one state variable and to optimize one at a time while the others have assumed state trajectories. In the following step another subproblem is optimized after the state vectors were updated with the previous solution. The procedure is repeated until the solution of the original problem converges. The method was first applied by Larson and Keckler (1969) for a multiple reservoir system. Nopmongcol and Askew (1976) combined the incremental DP and the DP with successive approximations. Their algorithm used IDPSA to obtain the input state trajectory combination for the two-at-a-time IDP execution. The results of both IDP and IDPSA can be influenced by the choice of the initial state trajectory, but this is a common problem for many other iterative procedures.

Stochastic DP (SDP) can take into account the uncertainty of the input data. One of data which is inherently random is the reservoir inflow, and its impact on the operational policy has to be considered. SDP models can directly incorporate this aspect of the analysis into the solution procedure. In the work of Daellenbach and Read (1976) a stochastic dynamic programming model of the Swedish State Power Board (Gustafsson, 1968) is described. All reservoirs and stream flows are aggregated and presented by a single reservoir and a single hydro station. The program derives water value curves as a function of reservoir level for the planning period of 52 weeks. The reservoir levels are optimized to have a minimal thermal energy production cost of the power system. The optimization is constrained by the requirement to satisfy the specified demand for the given marginal cost structure of thermal energy and the total amount of storable and non-storable stochastic inflow. The historic sequence of weekly observations of stream flows during the most recent 30 years is used as a sample to estimate the average water values. The model is used in conjunction with a simulation model, which helps to aggregate the stream flows and storage contents of the various river systems.

Turgeon (1980) compares two DP techniques applied to the problem of optimal operation of a multireservoir power system with stochastic inflows. One is the one-at-a-time method (also referred to as DPSA in the above discussion). The other is the aggregation/decomposition method. The first gives an optimal feedback operating policy for each reservoir. The feedback term implies the assumption that the turbine release from a particular reservoir is a function of the storage and inflow of that reservoir, exclusively. An assumption that the release is related to the storage in the other reservoirs, too, i.e. the open-loop solution, requires DPSA execution for every time step, which is costly in computer time. The second, aggregation/decomposition approach breaks up the original complex parallel reservoir/power plant system into two components. One component is the actual reservoir/power plant of the original complex system, while the other is an aggregate of all the remaining elements of the system. In this way a two state variable stochastic DP problem is formulated, which can be solved without dimensionality problems. The procedure is repeated for every reservoir separately, and the solutions are combined to result the solution of the original problem. The two methods were applied to, and compared on the basis of, a system of six reservoirs/power plants. In this evaluation, the aggregation/decomposition was proven to be better.

Reliability-constrained DP arised from the fact that long range reservoir operation has to trade off the return and the risk associated with not achieving it.

A probabilistic DP model with discounting was formulated to solve the stated problem. The probabilistic term stands for the independent, stochastic character of the inflows in the model. The problem has been solved either using the penalty function approach or the Lagrangian duality theory of nonlinear programming. However, there are substantial difficulties in formulating a multireservoir problem (e.g. interdependence of inflows). There are also no attempts to evaluate the severity and duration of failures to satisfy the targets. The applicability of the approach is limited to long term planning purposes.

For the problems where the objective function is separable and convex (in the case of minimization) and the system can be described solely by dynamic equations (i.e. linear dynamics, quadratic performance problem or LQP) an analytical solution can be obtained. The methodology can be generalized for multiple state variable problems without running into the dimensionality problem like in the classical discrete DP. For the problems where the above conditions do not hold, the objective function or the system dynamics equation can be expanded into Taylor series. In this way, around the initial estimate the requirements for the analytical solution are satisfied. The solution procedure for these non LQP problems is iterative. The method has the name of differential DP and it was introduced by Jacobson and Mayne (1970).

Turgeon (1981) presented an algorithm related to a DP approach. The task was to optimize releases from a system of hydro power plants located in series on the same river. The solution procedure was based on the principle of progressive optimality. The feature of the approach is that it does not require the discretization of the solution space. It can also handle discontinuous return functions, and the objective function does not have to be linearized nor approximated by a quadratic function.

To conclude, DP is capable of handling a large scale of problems in reservoir systems. According to the literature (Yeh, 1985) its major limitation is the curse of dimensionality and numerous efforts have been made to alleviate this problem.

2.1.2 Nonlinear Programming

Nonlinear programming (NLP) methods have not been applied to water resources systems analysis as often as LP or DP. This is primarily due to the fact that these methods are much less efficient in using computer time and memory than the others. In addition, the mathematics is much more complicated, and the methods do not lend themselves easily to stochastic problem solutions. The remedial measure is to include a sensitivity capability in the algorithm. Of course the application of these methods has its advantages, too. NLP can handle non-separable functions (e.g. hydro production) and nonlinear constraints.

For the general problem where the objective and constraints are both nonlinear the penalty and/or barrier solution methods could be one of the choices

(Yeh, 1985). Assuming convexity of the constraints the problem can be solved by applying the Lagrangian dual procedure (Yeh, 1985).

If the problem is simpler, in the sense that the constraints are linear functions of the decision variables and only the objective function is nonlinear, one of the solution techniques is the gradient projection method proposed by Rosen (1960). The feature of the method is that it implements the feasible direction algorithm without solving an LP at each iteration step. This is possible since the set of active constraints is changing at most by one element at a time and the required projection matrix can be calculated from the previous one by an updating procedure.

Another method for the same class of problems (linear constraints, nonlinear objective function) is the reduced gradient method. The method was used by the Tennessee Valley Authority for scheduling weekly releases (TVA, 1976). Rosenthal (1981) applied a modification of the reduced gradient methodology to optimize a nonlinear nonseparable objective function with a linear network flow constraints. An unusual feature of the algorithm is the integer programming subproblem whose function is to obtain the superbasic set and the search directions needed in the reduced gradient method.

A summary comment on the NLP methods could be that the major obstacle for their application is the rate of convergence and the overall high computer requirements.

2.1.3 Simulation

Simulation is a mathematical modelling technique aimed at providing a response of the system for a certain input. The input includes decision rules which provide guidelines for the operation. The decision maker can examine the consequences of different operation scenarios for an existing or planned system. Simulation is extensively used in water resources. Some of the known models are HEC-3, HEC-5, SIM I and II. For a more detailed review of models see Yeh (1985). The advantage of simulation is that it can be more flexible, versatile and detailed in the system description than the optimization techniques. On the other hand, optimization looks to all possible decision scenarios, while simulation is limited to a finite number of input decision alternatives.

The adopted operating rules used as input into simulation models are summarized by Loucks and Sigvaldason (1982). They suggest that the operating policies may include some of the following general concepts: target storage volumes, allocation zones within the reservoir, flow ranges, and conditional rule curves dependent on the expected natural inflows.

The combined use of optimization and simulation models is a common idea. Loucks et al. (1981) suggest use of optimization to screen a great number of feasible plans and to explore the remaining ones in more detail by applying a simulation model. The general tendency in recent years is to incorporate an optimization scheme into the simulation model. One of these models is developed by Sigvaldason (1976).

The practical application of optimization techniques in water resources management is not so widespread due to the complexities of the water resources systems and the existence of noncommensurable objectives. In this regard, simulation is an effective tool for studying the operation of the complex water resource system incorporating the experience and judgment of the planner or design engineer into the model.

1.2 LINEAR PROGRAMMING APPLICATIONS

Linear Programming (LP) has been one of the most widely used mathematical programming techniques for optimization of water resources systems. The technique refers to a special class of problems where the objective function and the constraints are both linear or can be approximated by a linear relationship. The major advantage of this technique over the others is that the solution algorithm efficiently identifies the global optimum and there is a mathematical proof for the existence of an optimal solution. LP software packages are widely available, and this feature makes its application especially attractive. The planner has to concentrate only on the problem formulation and does not have to master every detail of the LP solution procedure. The fact that LP problems can be solved very efficiently gave special incentive to structure nonlinear problems as linear optimization models. The nonlinearities may be resolved either by approximation (e.g., piecewise linearization of concave function to be maximized), or by approximation and iteration (e.g., linearization of a nonseparable function). The optimization of complex objective functions can be solved by piecewise linearization and applying a variant of simplex method called separable programming (Daellenbach and Read, 1976).

The major obstacle for applying LP to the hydro utility operation problem is the nonseparable character of the hydro production function. Recently Can et al. (1982) described three methods to overcome the nonlinearity. The first method is simlar to that applied in EMMA program of Manitoba Hydro: assume a constant head during the time step and iteratively improve the assumption using the LP solution. The second method calculates upper and lower bounds on the basis of forecasted inflows. The head is assumed to be constant for specified intervals in the hydro production calculation. The third method utilizes separable programming to find the approximate optimal solution. The stage-storage curve is piecewise linearized and two new variables are introduced to transform the hydro production function in a separable form.

However, it has to be noted that with any applied linearizing approximation the identified solution is not necessarily the global optimum as in the case of linear problems.

The LP models can be divided into two big groups: deterministic and stochastic. The short description of some of the recently developed models follow.

2.2.1 Deterministic Models

Daellenbach and Read (1976) describe a deterministic LP model used by the Pacific Gas and Electric (PG&E) Company of San Francisco (Miller and Thompson, 1971, 1972). The program utilizes the increasing marginal thermal costs and decreasing marginal efficiency of hydro-generation due to head loss by piece-wise linear approximations. Each reservoir is represented individually. PG&E uses a composite marginal fuel cost curve for the whole system. Its shape and location depends on the level of thermal shut-down: the higher the shut-down level, the higher the marginal fuel cost. The level of shut-down is estimated from the daily system load curve after subtracting the power from the noncontrollable energy sources (e.g., contracted import, nuclear power and base loaded units), an intelligent guess of the hydro production and adding the contracted export load. On the basis of this analysis a preschedule of the thermal shut-down level for each month is estimated with regard due to its effects on the size of the transmission losses and the spinning reserve requirements. The analysis made at PG&E indicates that the nonlinearity of the composite fuel cost curve can be approximated adequately by six to seven linear segments, which reflect not only the characteristics of the existing plants but also breaks in the cost of the fuel used. The objective function contains also the cost of the import energy less the exported energy if any. Hydro production is given a zero cost coefficient. The objective is to minimize the total cost over the planning period. Every power source is constrained by a number of technical and behavioral limitations, but most of the constraints are related to the modelling of reservoir and hydro plant operation. They include constraints on storage levels, flow continuity, release limits, and for reservoir head variation due to its nonlinear effect on the result. There is a minimum target level for each reservoir to be met at the end of the planning period. The model is used to aid the decision process of long term allocation of power sources in PG&E.

Takeuchi and Moreau (1974) have developed a method for finding optimal operating policies for a multiunit water resource system that extends over two river basins and serves multiple demands. The problem of determining optimal values for control variables within a monthly interval (for a set of initial state variables) is formulated as a convex piece-wise LP problem. The objective is to minimize the monthly value of the loss function (i.e., immediate losses) and to minimize the expected value of the economic efficiency losses over all future months. The economic efficiency losses are the unknown function of the end-of-month state variables. That function can be estimated from the stochastic DP problem solution within which the LP problem is nested. Special techniques are applied to obtain a large number of solutions to similar LP problems which are needed as input for the stochastic DP problem to find an approximate overall solution. The previous task involves the use of simulation in a recursive algorithm. Simulation is also used to test the derived policy using the actual stream flow data. The method was developed and tested to study the further development of a water resource system.

Draper and Adamowski (1975) have applied LP as a screening or allocation model to provide information on system operation and response. This information is later used in the preliminary design of hydroelectric power producing facilities. The objective was to maximize the ability to generate continuous system power. The constraints involved storage limits and power requirements. The inflow scheme consisted of synthetically generated data. The nonlinear power response is approximated by linear power-discharge relationship for three different storage volumes.

Dagli and Miles (1980) formulated a model with the objective to maximize the sum of average monthly hydrostatic heads of four power plants on the same river over a yearly time horizon. Requirements were set to supply water for irrigation, as well as maintaining river flows downstream of the reservoirs. The authors applied a deterministic LP modelling procedure with updating, called adaptive planning (AP). The idea of AP is to optimize the operations of the system on the basis of deterministic stream flow forecast. The obtained result is applied only for the first time step. To determine the operation of the system in the next time step, the program is run again with the updated stream flow forecast. In this way new additional information is added to the optimization. The obtained solution is not necessarily optimal but it is very close to the optimum. The model was used for long term planning to determine the operating policies for a set of four dams, each of them associated with a hydro-electric plant.

Bechard et al. (1981) developed a deterministic linear-separable programming model to optimize the operation of the reservoirs located in the Ottawa River basin. LP is used to perform the basic optimization steps which are later used in the complex multi objective decision analysis. The model has the objective of reducing flood damages and maximizing energy production benefits. The basic approach is the multiple-objective optimization by weighting coefficients to trade off the two objectives. By applying different weights a trade-off curve can be obtained and later used by the decision maker to identify the best compromise solution. However, certain difficulties were encountered due to the different optimization time horizons of the two objectives. The energy objective requires one year period since the load and reservoir elevation have a yearly repetitive cycle. The flooding objective requires a time horizon of only three or four months of the flooding season. The problem was solved by applying a hierarchical approach. The long-term, yearly optimization was performed with respect to the energy objective only. The mid-term model of about 16 weeks included both objectives and took into account the results from the long term optimization. The hierarchical structuring was achieved by using the long-term optimal storages as targets to be met by the mid-term model. The continuity between the mid-term and the short-term model of about ten daily steps was provided in a similar way. All three models are to be run sequentially with updated deterministic streamflow forecasts. The hydro-electric system was represented in detail by two types of plants (run-of-river and with reservoir) and three type of channels (controlled, free and generating). The hydro power production is modelled with piece-wise linearization and an iterative solution procedure to handle its non-separable nature. The model can be used for operation planning. It can also be applied as a tool to determine the effect of future development in the basin or the impact of modifying one or more operating constraints.

Pereira and Pinto (1983) described a methodology to coordinate the mid- and short-term scheduling of hydro-thermal systems. The technique is able to incorporate the electrical problems encountered in the short-term planning into a constraint which is added to the mid-term scheduling problem. This constraint refers to the weekly target variable in the mid-term problem. In this way a feedback is achieved between the short- and mid-term planning with only a few modifications required in the specialized algorithms used at each level. The performance of the model was tested on a case study of the Brazilian Northeast Network.

2.2.2 Stochastic LP Models

Stochastic LP models are developed to incorporate the nondeterministic character of the input data (e.g., stream flows, cost coefficients etc.). The need for modelling uncertainty is well described by Daellenbach and Read (1976). It is emphasized that the planning based on the expected values (e.g., streamflows) essentially assumes that the costs of the positive and negative deviations from these averages as well as the probability of such deviations are perfectly symmetrical, and independent from period to period. None of these assumptions correspond to reality. The uncertainty of the input data can be taken into account in deterministic modelling through sensitivity analysis. However, the procedure does not consider explicitly the stochastic character of the input data and may not lead to satisfactory results.

There are several methodologies to be used for characterization of nondeterministic parameters in LP models. A brief review follows.

The two-stage or stochastic programming with recourse is described by a practical example presented in Loucks et al. (1981). This method is able to deal with constraints which include random variables. In the work by Yeh (1985) the importance of distinguishing the decision stages is emphasized to understand the method. At the first stage the activity levels are determined. At the second stages, after the occurrence of the random event, a correction follows minimizing the negative effects of the activity at the first stage. In a water resource system the decisions taken in the first stage can be described as the target levels. At the second stage the minimization of the losses of not meeting the set targets is performed. The objective function has two parts: one where the effect of the target values is evaluated and the other which gives the expected value of losses not meeting the targets from the first part. In order to solve this problem by LP the probability distribution of the random event(s) has to be discretized. This results in addition of multiple constraints pertaining to the second part of the objective function to the set of constraints which limit the optimization of the first part of the objective. The discretized problem can be solved simultaneously although there are two decision stages. In case where the discretization is not possible a nonlinear deterministic problem can be formulated. The major shortcoming of the method is that it requires the evaluation of the recourse action by an adequate estimation of losses from the effect of random variation. There are also dimensionality problems due to the additional constraints and variables introduced by the discretization of the distribution function of the random event.

An alternative method to represent uncertainty in an LP model is chance-constrained programming. The method refers to problems with one or more random coefficients in the constraint set (either on the right or left hand side). In these situations, instead of applying the expected value of the random variable as the RHS, chance constraints can be written to define the probability of failure of that constraint. Chance-constraints can be converted into deterministic equivalents under the condition that the probability distribution of the random variable is known. Chance constrained models did not find application to hydro power optimization.

In conclusion, it should be emphasized that the major task of every decision making under uncertainty is to try to derive a deterministic equivalent of the stochastic problem. In cases where this is not possible the alternative is to apply a Monte Carlo simulation to assess the impact of random effects on the operation.

2.3 DESCRIPTION OF THE EMMA PROGRAM OF MANITOBA HYDRO

The EMSLP algorithm described in this thesis is tested using the Manitoba Hydro data. In order to evaluate the performance of the algorithm the results were compared to the EMMA program runs (Reznicek and Simonovic, 1988a). EMMA was made available for this research by the courtesy of Manitoba Hydro, for which the author is specially grateful. To understand the differences in the results obtained by the two algorithms, EMMA program has to be understood, too. Therefore, the details of EMMA algorithm and the system on which it is used, are presented in the following section.

Manitoba Hydro is responsible for the operation of the integrated power systems of Manitoba Hydro and Winnipeg Hydro. The generation system is composed of thirteen hydro power and three thermal plants. The total capacity is 4250 MW of which more than 90 % is hydro. The task of operations planning is to derive a schedule of reservoir releases, hydro generation, thermal generation and energy exports-imports with the goals to meet the forecasted system demand (energy and peek capacity requirements), maintain system reliability, and operate economically (Barritt-Flatt and Cormie, 1988).

The hydraulic system of Manitoba Hydro is very specific. There are a few big, shallow reservoirs among which Lake Winnipeg is the biggest. Due to its vast area the level changes very slowly even if an excessive amount of water is released during the time step. The operation range is very small, only several metres. The down stream hydro power plants on the Nelson river do not have the capabilites to store large amounts of water. They mostly operate as run-of-river plants. The situation on the Winnipeg River is similar. The power production in the series of plants depends on the release from Lake of the Woods and Lac Seul. The above facts are important for the assumptions incorporated in the EMMA program regarding the iterative solution procedure. The head variations in the system during the planning period are small. The developed algorithm is well suited for these specific conditions of the hydraulic system.

The EMMA program was developed by the Computer Services Division and the Energy Resources Section of Manitoba Hydro to support the decision making process (Manitoba Hydro, 1986). A deterministic LP optimization model was formulated to determine operation plans for hydro generation, thermal generation and inter-connections with maintenance of these facilities. The planning horizon is about a year (depending on the purpose of running the model) which permits tradeoffs among current and future costs and revenues. The program has the capability to formulate the LP problem based on the input data, to solve it using an iterative solution procedure and to present the results in a form of reports. Since the actual problem is formulated by the computer the program is very flexible. Any configuration aggregated from the real Manitoba Hydro system can be optimized. The power of the model lies in the possibility to optimize with a different level of details depending on the set goal. The program is written professionally and due to its flexibility, the program can have a very wide range of applications not necessarily related to hydro-thermal power system operation (e.g., irrigation system).

The model's deterministic nature enables representation of the hydro-electric system in great detail. The stochastic aspect of the input data is dealt implicitly by performing the optimization with different stream flows and energy load, i.e., conducting a sensitivity analysis. The model is used in the manner of adaptive planning (Dagli and Miles, 1980). The operating plan is optimized on a regular basis as new information is available. The input data is updated using new forecasts of precipitation, river flows, domestic energy loads, and export market prices. Practically, only the policy determined for the first time step is implemented, while the others have the role to provide impact on that policy from the aspect of long term planning. Consequently the time horizon in EMMA is divided into shorter steps in the near future and longer ones for the more distant future. The shorter near future steps can be also attributed to the more certain forecasts for this period.

The introduction of an optimization model provided not only a shorter time frame of forecasting, analysis, review and implementation of the up-to-date operations plan but also enhanced the comprehensiveness of the planning. This aspect of the modeling with EMMA is extremely important for the operation of the Manitoba Hydro system. The relative importance of the system components can change depending on the current status of the reservoirs, the characteristics of forecasted flows and predicted loads. The system operation has to reflect these changes accurately. In order to ensure this condition, the model was built with the aim of accurately representing the system components.

The modeling done in EMMA can be separated in three parts:

a) hydraulic system;

b) electrical system; and

c) maintenance system.

The hydraulic system is composed of reservoirs, lakes and rivers. Reservoirs for hydro-electric production have usually two outlets: a spillway and a penstock through the turbines. The lakes can be drained either through a control structure or by natural, i.e., nonregulated outlet. The storage of the lakes and reservoirs can be discretized to segments to approximate the nonlinear storage-stage curve by piece-wise linearization. The last segment in the last time step can be further divided into sub-units of the discrete interval called figments to better describe the price-volume relationship. Time delays introduced by open channel flow can be modelled by introducing a fictitious (or dummy) lake of an appropriate stage-storage and outlet rating characteristics. Beside the above mentioned elements the model can describe natural inflow to lakes and rivers and consumptive withdrawals. The elements can be combined in any desirable fashion. There are few restrictions in configuring the run-of-river generation stations. Explicit constraints can be formulated to:

a) ensure that the sum of "figments" is equal to the storage of the last segment;

b) provide mass balance for lakes;

c) set the final and initial lake stage to be equal;

d) limit the outflow from a powerhouse or spillway of a generation station as a function of the upstream storage; and

e) limit the outflow from a control structure as a function of the upstream storage.

The electrical system consists of energy generating and transmitting elements with the purpose of satisfying the domestic and contracted export energy load. The domestic system load is specified in each time step by a deterministic load duration curve. The load duration curve represents the intensity of the load during a time step reorganized in a descending order. In other words for a certain load the curve specifies the time (or the fraction of the time step) when it is going to be exceeded. The curve is approximated in the model by a number of strips and an instantaneous peak load. The width of the strip represents the fraction of the time step while the length represents the average load. The load duration curve modelling is represented in Figure 1. The load must be satisfied during each strip. The energy price varies within a time step, which is the reason for designating on and off peak strips. Thus, the cost and revenue functions are different for the on and off peak strips, with on peak strips having a higher energy price than the off peak ones.

The energy transferred from or to the neighboring utilities is dealt with in two different ways. The export or import can be either firm or interruptible. The firm purchase or sale of energy means that it must be satisfied 100 % of the contracted time, and therefore it is incorporated in the constraint set. The amount of energy imported from or exported to the interruptible market depends on the decisions made in the optimization process limited by the available tieline capacity.

The electrical system is described by formulating the following constraint types:

a) supply and demand - to ensure that for every load duration curve strip, the system load plus the firm exports are supplied from hydro generating stations, thermal plants and imports;

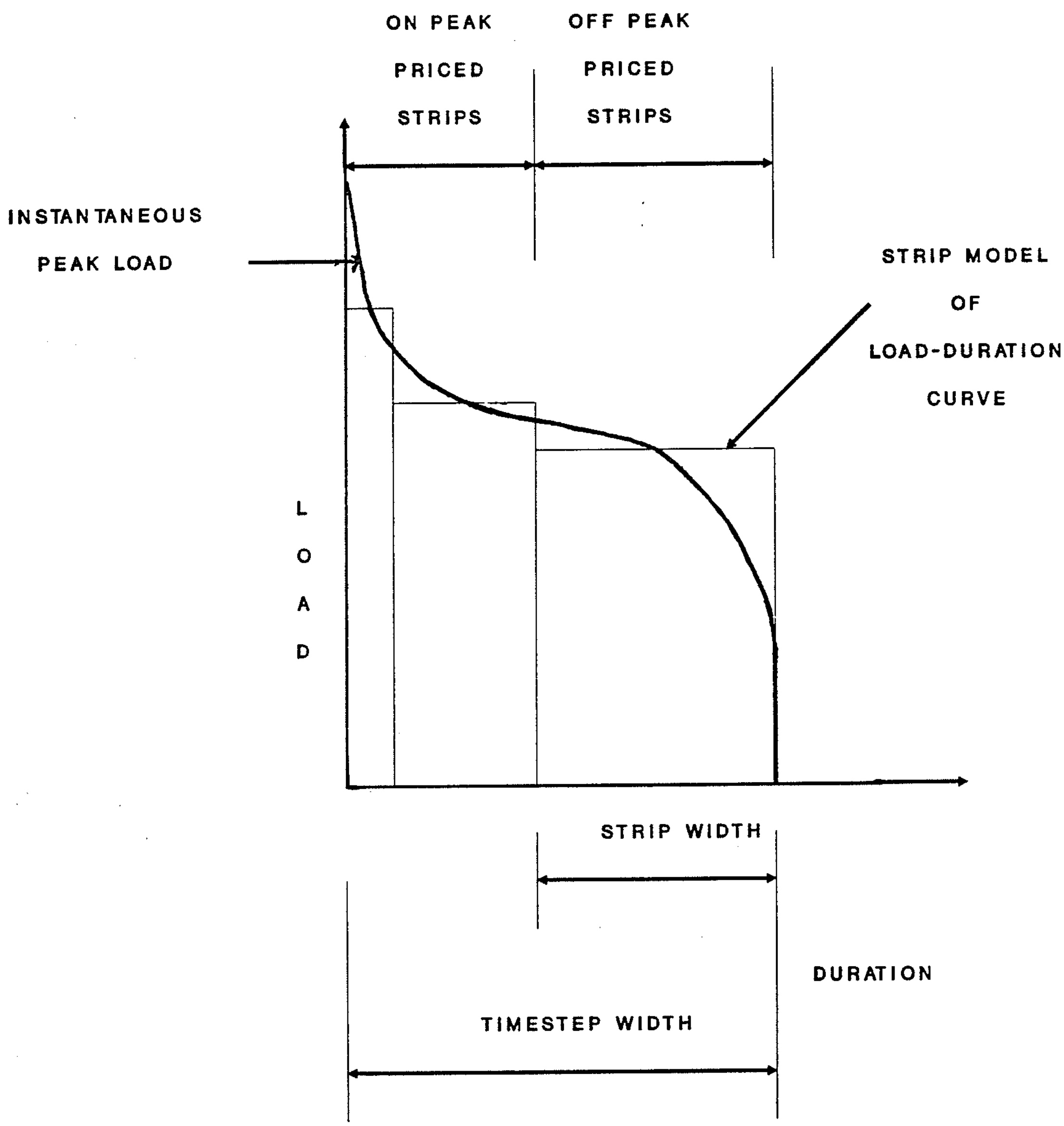


Figure 1. Load duration curve modeling (Barritt-Flatt and Cormie, 1988)

b) hydro production - to convert the powerhouse discharge during a time step using the production coefficients into energy for each load duration curve strip;

c) run-of-river - to ensure flow continuity by load duration curve strip with an upstream generating station;

d) hydro shaping - to ensure that the capacity utilized in the load duration curve decreases moving from on to off peak, or remains unchanged for the base loaded generation station;

e) thermal shaping - the same as d) but for thermal plants;

f) tieline total load - to ensure that the export and import time does not exceed the total time;

g) import fraction limit - to limit the total interruptible import energy to a percentage of the required system energy for each time step;

h) contract energy limit - to place a minimum on the total of all the contract energy variables in all strips of all the load duration curves; and

i) waste heat - to limit the maximum capacity of the thermal plant when the thermal pollution of the water source downstream of the plant exceeds either the specified maximum temperature change or the set maximum temperature.

The maintenance scheduling in EMMA is performed by imputting an annual maintenance plan based on the station requirements and system operation requirements and reviewing the same during the optimization process. The effect of maintenance in the optimization is reflected by reducing the capacity and energy capability of the station during the time step when it is scheduled. The possible constraints for this issue are:

a) crew scheduling - to ensure that for a specified period crew holidays are accounted for;

b) required maintenance - to ensure that for each plant sufficient maintenance is done to meet the set requirements;

c) maintenance space - to ensure that after maintenance is accounted for and the forced outage is subtracted there is enough capacity in the system to meet the peak load;

d) available capacity - to ensure that the available capacity of the hydro or thermal plant is not exceeded in a load duration strip; and

e) crew availability - to ensure that the maintenance done by a crew within a time step does not exceed its availability in hours.

Except the mentioned major constraint groups there is a set of constraints pertaining to energy grouping. These constraints are needed to maintain a balance of generation in the electrical transmission system. Any energy variable in the formulation can be included or excluded in these constraints by assigning an appropriate membership coefficient. The right hand side of the constraint may be a constant or a function of the domestic load for the interval covered by the constraint. The constraints are:

a) energy grouping by strips - to place a minimum value (as a function of system load) on the total of the defined energy variables on the particular load duration curve strip;

b) energy grouping by time steps - to place a minimum value (as a function of system load) on the total of the defined energy variables in the particular time step; and

c) energy grouping by period - to place a minimum value (as a function of system load) on the total of the defined energy variables for the chosen period of study.

The objective function contains all the LP variables with the assigned appropriate cost coefficients. The cost coefficient can have the following possible meanings (Barritt-Flatt and Cormie, 1988):

- storage coefficient
    - may reflect flood damage for the upper segments
    - for the last time step it can denote benefits from the future energy production
    - a symbolic penalty coefficient to provide filling the lower segments of the reservoir before the upper ones
- release
    - the net benefit or cost of any release assignment

- generated energy
    - fuel cost of the production and/or
    - the fixed cost of the plant maintenance
- import and export energy
    - the interruptible energy market structure
    - the contracted price of the firm energy transfer
- scheduled maintenance
    - non-economic preferences in the assignment of maintenance

The problem of optimizing the operation of an interconnected hydro-thermal utility is inherently nonlinear. There are different nonlinear relationships in the problem: the stage-storage reservoir curve, the load duration curve, the hydro production function, cost curves, etc. The LP solution technique can be applied exclusively to a linear objective function subject to linear constraint set. It is therefore required to substitute the original nonlinear relationships by a linear approximation.

If the nonlinear relationship can be described as a univariate function (one variable is function of the other) the problem is fairly simple: the function may be piece-wise linearized. However, special care must be taken to ensure that the linear approximation is satisfactory particularly as representing the original relationships in the critical ranges.

The problem is somewhat different when the relationship has to be described by a multivariate nonseparable function, as in the case of hydro production given by the following expression:

$$E = \gamma * Q * H * T * e(Q,H) \tag{1}$$

the produced energy (**E**) is a function of discharge (**Q**), head of the power plant (**H**), and efficiency (**e**) multiplied by the specific weight of water (γ), and the observed time period length (**T**). It has to be noted that the efficiency is the function of discharge and head.

The discharge and the plant head are both directly and indirectly decision variables, whose level has to be determined in the LP solution. Formulated as an LP, this problem cannot be modelled directly. Thus, an iterative algorithm has to be followed.

EMMA resolves the difficulty by assuming a constant value for the production coefficients (PC):

$$PC = \gamma * T * H * e \tag{2}$$

Applying this assumption, the operations problem is linearized and can be solved by an LP package.

The algorithm of EMMA program can be summarized as follows:

(a) start with a set of assumed production coefficients (for each of the variables describing the produced energy);

(b) solve the LP problem;

(c) calculate the production coefficients using the obtained LP solution;

(d) compare the calculated coefficients with the assumed ones. If the difference is less than the specified tolerance level for each of them then stop, the solution has converged. If not, make an assumption for the next iteration in the following manner. If the difference between the assumed and calculated coefficient is less than 30% of the assumed one, accept the calculated coefficient as the assumption for the next iteration. If not, the assumption for the next iteration is the assumption for the previous iteration corrected by 30% in the direction of the calculated one. Go to step (b).

The flowchart of the algorithm is shown in Figure 2.

Experience has shown that the production coefficients for a network of hydro power plants converge rapidly for well constrained problems. In cases where the problem is ill-defined convergence may not be obtained. Typically, a convergence tolerance of a few percent is used.

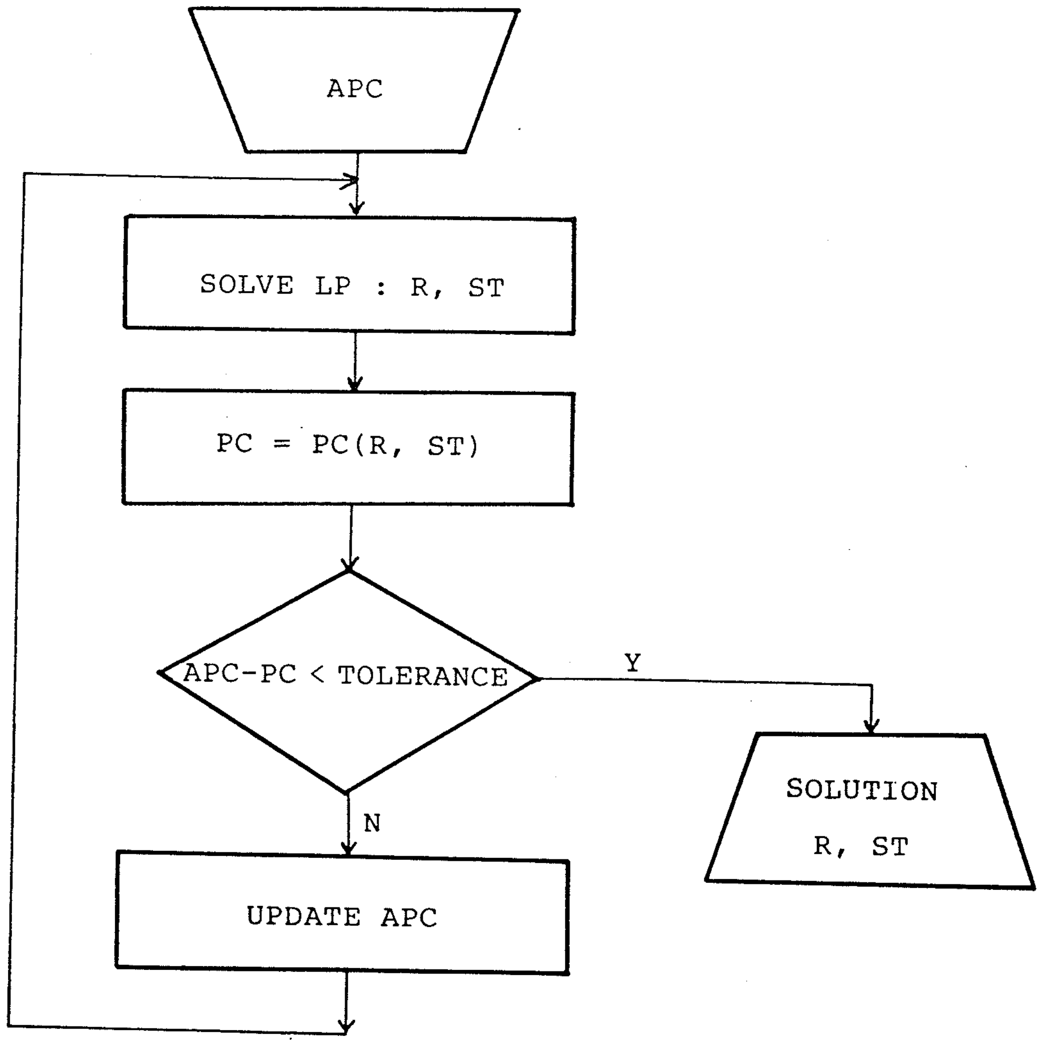


FIGURE LEGEND:

APC - assumed production coefficient set
PC - calculated production coefficient set
R - calculated release vector
ST - calculated storage vector

Figure 2. EMMA Flowchart

2.4 DESCRIPTION OF THE SUCCESSIVE LINEAR PROGRAMMING MODEL OF J. GRYGIER

Grygier (1983) compared three algorithms for optimizing the operation of a multi reservoir hydrosystem over a medium term. Besides a combination of LP-DP and an optimal control algorithm the performance of a Successive Linear Programming (SLP) algorithm was explored. Grgyier's work is the basis of the research which is the topic of this thesis. His SLP algorithm is pertinent to thedevelopment of EMSLP algorithm, and is therefore presented in detail in this section.

The Grygier algorithm attempts to maximize the value of energy generated by a hydropower system over the planning period, plus the expected future benefit from the remaining water in the reservoirs at the end of the planning period. The major assumption is that the produced energy can be sold on the market with no limitations. The energy production during each time step is separated according to the price into on-peak and off-peak energy. The on-peak production is, however, maximized to a certain number of hours in the time step. It is also assumed that the price for the water stored at the end of the planning period is constant. The objectives, besides energy production, are incorporated into the constraint set. For example, storage levels can be bounded above and below to allow for recreation or flood control. The time of flow between the reservoirs is ignored due to the time scale of optimization (monthly time step). The only energy source is hydro and thus it is treated implicitly in the formulation (i.e., there is no energy variable in the formulation). The energy is expressed through the hydro production function.

The constraint set includes:

a) flow continuity - to maintain conservation of mass in the reservoir;

b) minimum and maximum storage bounds - to take care of the physical characteristics of the reservoirs and to be used in the solution search;

c) upper bounds on sum of on- and off-peak releases - to reflect the maximum production capability during the time step, i.e., impose turbine flow limits;

d) upper bound on on-peak release - to limit the on-peak production; and

e) minimum energy - to ensure that a set minimum energy is produced during the time step since there is no load duration curve to be satisfied by the system.

Even with the introduced assumption of constant water price the problem is nonlinear due to the hydro production function, and the reservoir stage-storage relationship. The linearization procedure pursued in this algorithm resolves both simultaneously.

The major obstacle for applying LP to the stated problem is the non-separable character of the hydro production function. Some remedial measures were analyzed previously by Can et al. (1982). SLP utilizes a substantially different approach.

The idea in SLP algorithm presented by Grygier is to apply a first order Taylor series approximation to the hydro production function around the chosen storage and release values. The application of the Taylor expansion to linearize nonlinear problems and to solve them by LP iteratively is discussed by Palacios-Gomez et al. (1982). The linearization procedure has the following form.

The energy equation (1) can be reformulated to:

$$E = ERF*R \tag{3}$$

where ERF stands for energy rate function, and is expressed as

$$ERF = \gamma *H*e \tag{4}$$

and R designates the release

$$R = Q*T \tag{5}$$

The assumption is that ERF is only a function of the head, i.e., of the storage and that it is not dependant on the discharge. This assumption can be supported with the reasoning that the discharge changes many times during the time step and its mean value (calculated in the LP solution) is a poor approximation to be used in the efficiency calculation for the whole time step. Thus,

$$ERF = ERF(ST) \tag{6}$$

To account for the storage value change during the time step, it is assumed that the value of the ERF for the time step t is the average of the function value for the initial and final storage:

$$ERF_t = 0.5*(ERF(ST_{t-1})+ERF(ST_t)) \tag{7}$$

The energy equation for the T-th time period has the form:

$$E_t = ERF_t*R_t \tag{8}$$

or:

$$E_t = 0.5*(ERF(ST_{t-1})+ERF(ST_t))*R_t \tag{9}$$

It can be noticed that in Eq. (9) there is a multiplication of release and a function of storage (ERF) which makes the expression nonlinear and nonseparable. The complexity of Eq. (9) can be reduced by removing the ERF function from the product. The simplification can be achieved by applying a first order approximation of ERF in Eq. (9) instead of its real form. The approximation of a function by a Taylor series is possible in the vicinity of some chosen point. In the model these are the estimated values of storage at the beginning of the time step (i.e., at the end of the previous time step) $\hat{ST}_{t-1}$ and at the end of the time step $\hat{ST}_t$. The form of Eq. (9) after introducing the first order Taylor approximation for ERF is:

$$E_t = 0.5[ERF(\hat{ST}_{t-1}) + ERF(\hat{ST}_t) + DERF(\hat{ST}_{t-1}) * (ST_{t-1} - \hat{ST}_{t-1})$$

$$+ DERF\,(\hat{ST}_t) * (ST_t - \hat{ST}_t)] * R_t \qquad (10)$$

where DERF denotes the first derivative of ERF with respect to ST. In Eq. (10) there is now a product of two linear decision variables instead of a linear and a nonlinear one. However, the nonlinearity is still present, although in a simpler form. The remedial measure to remove the product of storage and release is to apply the approximation introduced by Loucks (1981):

$$ST_t * R_t = \hat{ST}_t * \hat{R}_t + (ST_t - \hat{ST}_t) * \hat{R}_t + \hat{ST}_t * (R_t - \hat{R}_t) \qquad (11)$$

where $\hat{R}_t$ is the estimated i.e. known release value. It can be noticed that this approximation is very similar to the one introduced for ERF except that it deals with a multivariate function.

Finally, combining Eqs. (10) and (11) :

$$E_t = 0.5*\{[ERF(S\hat{T}_{t-1}) + ERF(S\hat{T}_t)] * R_t + DERF(S\hat{T}_{t-1}) * (ST_{t-1} - S\hat{T}_{t-1})$$

$$* \hat{R}_t + DERF\ (S\hat{T}_t) * (ST_t - S\hat{T}_t) * \hat{R}_t\} \qquad (12)$$

The linearization of the energy production represented by Eq. (12) requires assumptions, i.e., estimates not only for storage but also for release.

It is important to note that the ERF function incorporates the efficiency as well as the stage-storage relationship. Thus, there is no need to model this relationship separately. For each power plant a differentiable ERF function has to be derived. This can be done by determining a regression curve on the available operation data of the plant. The ERF functions and the estimates for releases are the part of the input into the model. The estimates for the storages are calculated using the estimated releases, given the inflow scheme and the flow continuity equation.

The algorithm developed on the basis of the above problem formulation and linearization technique, can be described in the following steps:

a) initialize the control variables (e.g., storage variability (VARYMX), iteration counter (ITER=0)) and calculate the coefficients of the LP problem on the basis of the input data (except for the right hand side (RHS) of the storage bounding contraints and coefficients of the hydro production in the objective function and the minimum energy constraint);

b) calculate the coefficients of the hydro production in the minimum energy constraint and the objective function using: the ERF and DERF (the first derivative of ERF with respect to storage) values of the estimated storages, the estimated storages and the estimated releases;

c) calculate the RHS of the storage bounding constraints allowing the storage not to vary more than VARYMX around the estimated value;

d) solve the set LP problem, i.e, obtain the values for storage and release for each time step, the objective function value; then set ITER=ITER+1; decrease the value of VARYMX by multiplying it with VARFC1 factor (<1);

e) if the objective value is bigger than the one obtained in the previous iteration (in the case of first iteration, the previous solution is zero) then accept the calculated solution as the best so far, and use the calculated storages and releases as input estimates for the next iteration; go to step b;

f) otherwise decrease the value of VARYMX by multiplying it with a factor (VARFAC < 1);

e) check whether VARYMX is less than the set tolerance level VARMIN; if yes then stop the program execution;

g) otherwise go to c.

The flowchart of the algorithm is presented in Figure 3.

The algorithm has a special emphasis on obtaining the highest objective possible. This fact is integrated into the directions for the iterative procedure. The algorithm takes different courses depending on whether there is an improvement in the objective function compared to the previous iteration or not. If the objective improves the newly obtained solution is used to recalculate the coefficients related to the hydro production. If not, the coefficients remain unchanged, the feasible space of the storage values is decreased by changing the right hand sides of the storage bounding constraints and the LP is resolved. In this way a new, higher objective function value can be identified. The reasoning for the above action can be explained in a following way.

Since the objective function is essentially a nonlinear function the true optimal solution (the extreme point) does not have to lie in the corner of the feasible space. However, these are the only points the LP can identify as optimal ones. Therefore, the simple iterative application of LP can result in a suboptimal

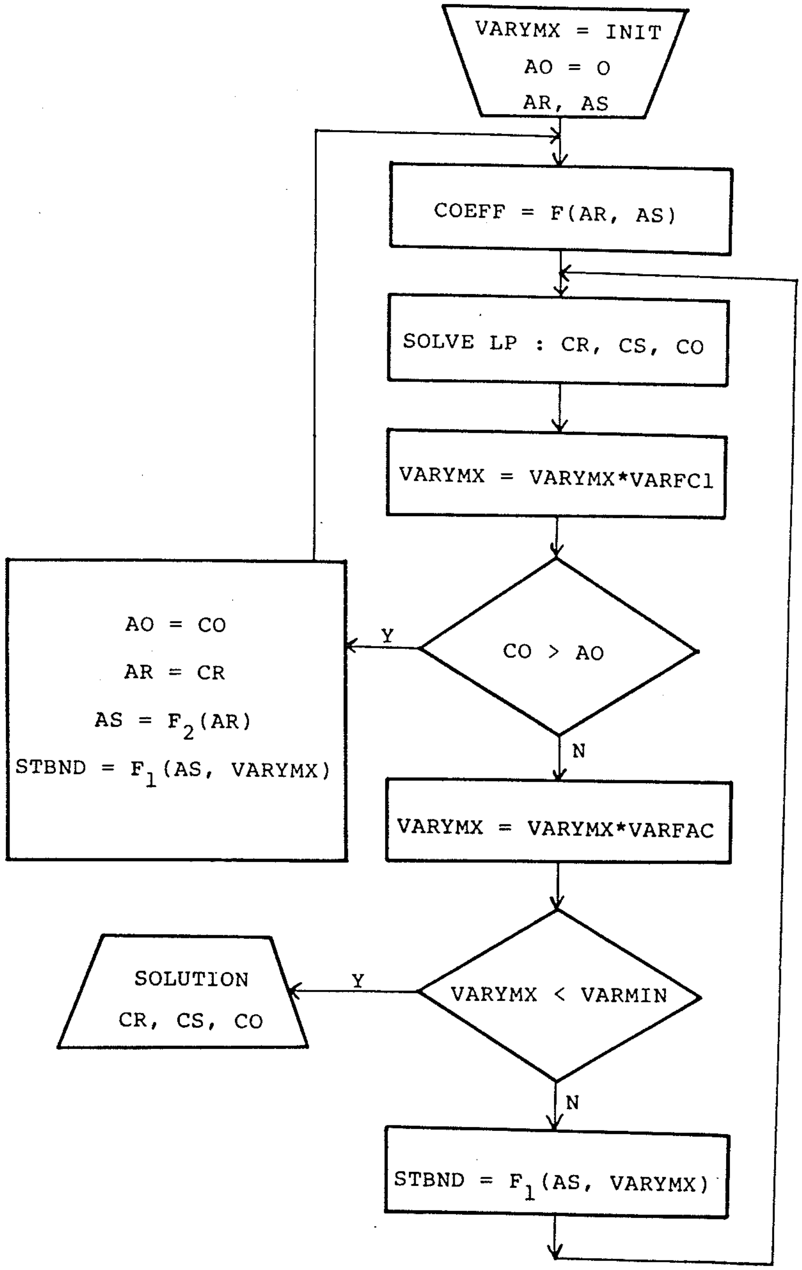


FIGURE LEGEND:

AO	- accepted objective function value
AR	- accepted release vector
AS	- accepted storage vector
COEFF	- hydro-production constraint coefficient
CO	- calculated objective function value
CO	- calculated objective function value
CR	- calculated release vector
CS	- calculated storage vector
INIT	- initial storage variability
STBND	- storage bound vector
VARFAC	- factor (< 1)
VARFC1	- factor (< 1)
VARMIN	- minimum storage variability
VARYMX	- storage variability

Figure 3. Flowchart of the SLP algorithm (Grygier, 1983)

solution. The specialty of the algorithm lies in its capability to search for an optimum in the interior of the feasible region whenever there is a drop in the objective function value. This feature is enabled by decreasing the feasible solution space around the obtained storage trajectory. In practical terms, this means that LP is able to examine new feasible points within the decreased space. Previously, these points were in the interior of the solution space and ignored by the LP, butnow are corner points and thus checked for optimum.

It is interesting to notice that the storage variability VARYMX is either unchanged or has a decreased value going from one iteration to the other. In other words it is decreasing during the program execution. This property is used as a criteria to terminate the program execution. When the value of VARYMX is less than a specified tolerance the program stops, assuming that the optimum was reached. The convergence of the iterations is ensured by making the solution space narrower and narrower around the identified trajectories. This eventually achieves the condition that the last two solutions differ less than the set tolerance.

Another feature of the decreasing solution space is that it prevents the algorithm from bouncing back and forth between two extreme points infinitely and enables the intermediate maximum to be identified.

Grygier (1983) claims that local optimum is always achieved and that for the examined applications the global optimum was also identified. The algorithm was used to optimize the 12-month operation of hydrosystems consisting of a single reservoir, two reservoirs in series, and three in parallel (with one being a pumped-storage facility). In comparison with the other algorithms examined, Grygier (1983) claims that SLP is the easiest to implement even though it is not the fastest method.

CHAPTER 3

Energy Management by Successive Linear Programming (EMSLP)

3.1 PROBLEM FORMULATION

The EMSLP algorithm developed during the research (Reznicek and Simonovic, 1988b) has the task of optimizing midterm operation planning of an interconnected hydro utility for a deterministic future. The operation involves scheduling reservoir releases to obtain hydro power, and managing energy transfer through the interconnections. The utility has to satisfy the domestic power demand described by the load duration curve in each time step of the planning period. The load duration curve is approximated by a number of strips in which the load is assumed to be constant. The energy price varies during the time step and therefore a different energy price can be assigned to each load duration curve strip. In periods of deficiency or if the energy price structure makes it rational the demand is satisfied from import. On the other hand when the energy market, reservoir storage and domestic load conditions make it desirable the energy can be exported to increase the benefits of the utility. The operation has to comply with the physical characteristics of the system and the operational licenses. The model has the following decision variables :

a) hydro energy $HE_{s,t}$;

b) export energy $EE_{s,t}$;

c) import energy $IE_{s,t}$;

d) turbine release R_t ;

e) spilled release S_t ; and

f) reservoir storage ST_t .

where s denotes load duration curve strip and t the time step number. It has to be emphasised that the above variables describe a single reservoir storage and release, hydro energy from one power plant, and export-import through one tieline. In case of a system where there are many of these elements another subscript has to be added to denote the specific element in the system.

There are two different ways to impose bounds on the variables. One is "simple bound" on the variable which is the same throughout the program execution and does not depend on the values of other variables in the decision making process. In this model the releases can have a lower bound to satisfy minimum flow conditions and upper bound to limit turbine flow or comply with the down stream discharge limits. These bounds are modelled without writing an explicit constraint. The LP routine takes care of them implicitly. The other way is to write explicit constraint in the problem formulation. The relationships which have to be described and involve more than one variable are formulated as explicit constraints. Constraints are also formulated if the "simple bounds" change from one iterative solution to the other.

3.1.1 The Constraint Set

The hydro production constraint describes the energy production in the hydro power plant. The EMSLP uses the linearization procedure formulated by Grygier (1983) and described thoroughly in Section 1.4. Rewriting Eq. (12) in terms of the above defined decision variables and sorting the unknowns to the left and the constants to the right of the equality sign, the hydro production constraint in the t-th time step has the form:

$$
\begin{aligned}
-2*\sum_{s} (HE_{s,t}) + [ERF(\hat{ST}_{t-1}) + ERF(\hat{ST}_t)]*(24/1000)*R_t + DERF(\hat{ST}_{t-1}) \\
\hat{R}_t(24/1000)*ST_{t-1} + DERF(\hat{ST}_t)*\hat{R}_t*(24/1000)*ST_t \\
= [DERF(\hat{ST}_{t-1})*\hat{ST}_{t-1} + DERF(\hat{ST}_t)*\hat{ST}_t]*\hat{RT}_t*(24/1000)
\end{aligned}
\tag{13}
$$

On the left hand side the energy in the time step is represented by a summation of the amounts allocated to each strip of the load duration curve. The conversion factor 24/1000 is needed to obtain the energy in GWh. The constraint is written for each time step and for each power plant if there is more than one in the system. The coefficients of the constraint are recalculated for every iteration when a new estimated storage trajectory is accepted.

The flow continuity constraint for the t-th time step gives the mass balance in a reservoir:

$$-ST_{t-1} + ST_t + R_t + S_t = I_t \tag{14}$$

The storage at the end of a time step has to equal the storage at the beginning of the next time step plus the inflow minus the turbine and spilled release. It is interesting to note that this constraint provides the link between the decision variables of different time steps. The storage variables of two adjacent time steps are directly involved. However, through the continuity constraints for the other time steps they are all indirectly related to each other. In case of a multiple reservoir system the constraint has additional terms depending on the system configuration (e.g., for reservoirs in series upstream release is downstream inflow).

Tieline load constraint for every load duration curve strip s and time step t:

$$IE_{s,t}*RATIO/(IEF*EEF) + EE_{s,t} \leq EML_{s,t}/EEF*24/1000*DPS_t \tag{15}$$

where RATIO denotes the export and import tieline capacity ratio, IEF and EEF are the import and export efficiencies respectively, $EML_{s,t}$ is the maximum export load in s during t and DPS_t is the number of days in the t-th time step. This constraint limits the amount of exported or imported energy depending on the tieline capacity and the length of time covered by that particular strip. The export and import variables describe the interruptible energy sales while the contracted export or import has to be included in the load duration curve. This is a viable way of modelling since the contracted energy requirement is known in advance and has to be satisfied without violations. This means that it can be treated in a same way as the domestic demand. Therefore it is possible to incorporate the two known requirements into one, namely the domestic demand.

The supply and demand constraint for every load duration curve strip s and time step t has the form of:

$$HE_{s,t}+IE_{s,t}-EE_{s,t} = L_{s,t}*(24/1000)*DPS_{t}*W_{s,t} \tag{16}$$

where $L_{s,t}$ denotes the system demand in s during t and $W_{s,t}$ is the load duration curve width of s in t. This constraint ensures that the domestic hydro production plus the import minus the export satisfy the energy demand in the particular strip of the load duration curve.

The minimum and maximum storage constraints bound the storage variable to comply with the physical characteristics of the reservoir, the operation license, and some other potential objectives as recreation, flood control, etc. These requirements can be modelled by placing a simple bound on the storage variable. However, the explicit constraint formulation is needed to model the change of these bounds from one iterative solution to the other. The change of the storage bounds modifies the feasible solution space and enables the search for an objective in the interior of the original solution space. The constraints are imposed on minimum storage in the t-th time step as follows:

$$ST_t \geq MAX(STMIN_t\ ,\ ST_t\text{-}VARYMX\ ) \tag{17}$$

maximum storage in the t-th time step

$$ST_t \leq MIN(\ STMAX_t\ ,\ ST_t+VARYMX\ ) \tag{18}$$

where $STMIN_t$ and $STMAX_t$ are the minimum and maximum allowed storage in t respectively and VARYMX is the allowed storage variability. In Eq. (17) it is required that the storage has to be greater than either the predefined minimum (i.e. the "simple" bound) or it must not be less than VARYMX from the estimated storage. The MAX operator ensures that the more stringent criteria is satisfied, and therefore, that both are satisfied. The upper bound is calculated in a similar manner. It is required that the storage has to be less than the simple bound, and also less than the estimated storage increased by the value of VARYMX. The value of the storage variability (VARYMX) and the estimated storages may change from iteration to iteration. Therefore the right hand sides of the constraints are recalculated before each iteration.

The following constraint relates the hydro energy to release in the t-th time step

$$\sum_{s} (HE_{s,t}) - ERF(STMAX_t)*R_t*(24/1000) \leq 0 \tag{19}$$

The relation of release to the produced hydro energy is not formulated explicitly in the hydro production constraint Eq. (13). Besides the hydro energy and release the left hand side of Eq. (13) also contains the storage variables. To stress the importance of the relationship an upper bound is imposed on the energy production in a time step. The production is limited to be not more than the released water multiplied by the maximum possible value of the energy rate function (i.e., the production rate when the reservoir is full).

3.1.2 The Objective Function

The objective is to maximize the interruptible energy export and the final storage volume while minimizing the production costs of satisfying the system demand (hydro energy production, import, spill costs). The benefit from the domestic energy consumption is not included in the objective, since it is constant and defined by the system demand. The mathematical form of the objective function is:

$$\text{Maximize}\{\sum_{t}[\sum_{s}(-HC_{s,t}*HE_{s,t}+EB_{s,t}*EE_{s,t}-IC_{s,t}*IE_{s,t})$$

$$-SC_{t}*S_{t}+B_{t}*ST_{t}]\} \qquad (20)$$

where $HC_{s,t}$, $EB_{s,t}$, $IC_{s,t}$, SC_t and B_t are the cost coefficients of hydro energy, export energy, import energy, spill, and storage variables, respectively. The hydro energy has an assigned cost of running the plant and since its major task is to satisfy the domestic load those benefits are not included as explained above. However, if the energy is exported it brings benefits to the system and thus has a positive coefficient in the objective function. Similarly the imported energy decreases the objective. As noted earlier, the energy in each strip can have a different price. Besides the energy variables the storage variable for the final time step is also included in the objective with a positive coefficient. In this way the release of all the water from the reservoirs at the end of the planning period and with the associated disregard for the future use of the system is prevented. However, it is assumed that the benefits are linearly related to the stored water although this is not so in reality. The water has an indirect value as a "fuel" for hydro production during the planning period and in addition, it has a value at the end of the planning period to take into account the benefits from the future production. Thus, the algorithm tends to release either the water through the turbines or store it for future production. Spill occurs only when it is physically necessary. Thus, there is no explicit need to penalize spill (i.e. to include the spill variables into the objective), although the program has that possibility. The turbine releases are omitted from the objective function because their effect is taken into account indirectly through the hydro energy variables.

In conclusion it can be said that the problem formulation is similar to the one existing in EMMA, although there are differences. EMSLP cannot model thermal plants and maintenance. There are no obstacles to adding these capabilities to the formulation, but they were omitted because the research emphasis was on the hydro production modelling and guiding the iterative solution procedure. Due to the different linearization technique and ideas introduced in the solution procedure the constraint set is different to that in EMMA. The major difference is in the hydro production constraint (13) which reflects the SLP approach to the problem. The storage bound [Eq. (17,18)] and hydro energy release relation constraints are added as a part of the original work done in the modelling.

2.2 THE EMSLP ALGORITHM

The simple modification of the ideas from Grygier's work was not possible for the optimization problem of the interconnected utility described earlier. A new algorithm EMSLP was developed to solve the problem (Reznicek and Simonovic, 1988 b). The EMSLP algorithm has the following steps.

(a) Set the LP problem according to the input data, and set the initial storage variability VARYMX. Calculate and accept the initial solution based on the estimated releases (the estimated storages are calculated using the flow continuity equation). Calculate the hydro production constraint coefficients from the solution.

(b) Solve the LP problem.

(c) Compare the calculated storages with the accepted ones, and if the difference is smaller than the tolerance, stop. The solution is obtained.

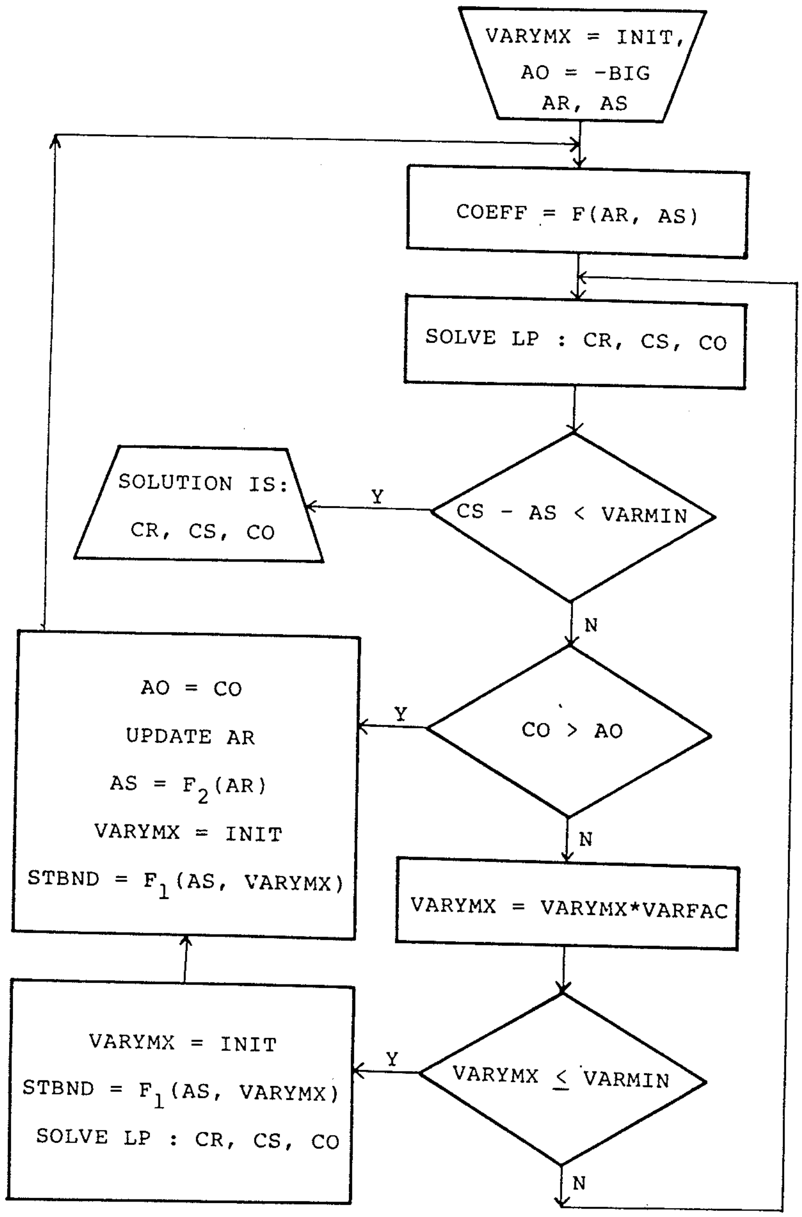
(d) If the calculated objective function value is better than the previously accepted one, then accept the calculated solution, but limit the change in the release policy to 30% of the previous accepted solution. With this release policy used as the estimate recalculate the coefficients in the hydro production constraint. Reset VARYMX to its initial input value and go to step (b).

(e) Otherwise decrease the value of VARYMX, and if it is still greater than the set minimum (VARMIN), go to step (b).

(f) If not, then use the first worse objective after the last improvement and the appropriate solution, as if it is better than the accepted one. Go to step (d).

Note that whenever the value of VARYMX is changed, the bounds on the storage variables are changed.

The flow chart of the algorithm is presented in Figure 4.



AO - accepted objective function value
AS - accepted storage vector
COEFF - hydro proudction constraint coefficients
CS - calculated storage vector
INIT - initial storage variability
VARMIN - minimum storage variability
VARYMX- storage variability

AR - accepted release vector
BIG - a very big number
CD - calculated objective function value
CR - calculated release vector
STBND - storage bound vector
VARFAC - factor (< 1)

Figure 4. EMSLP Flowchart

The algorithm has two iteration levels. At the first level a search for a stable solution is performed. At the second level the improvement of the objective function value is sought, whenever the objective function value drops between the two iterations. The search is performed by exploring the interior of the feasible region using the decreased storage variability VARYMX in the solution procedure. If the search for the better solution at the second level terminates unsuccessfully, the algorithm returns to the first level and accepts the initially identified worse solution. The search terminates on the first iteration level when a stable solution is identified.

The coefficients in the hydro production constraints are recalculated only at the first iteration level. At the second level, the lower and upper bounds on the storage volume are changed. The initial wide range is decreased with every iteration at the second level, approaching the accepted storage trajectory.

In Grygier's algorithm the storage variability is gradually decreased during the iterative process from the starting value to the set tolerance level when the program run terminates. According to the Grygier, the search ends with the local optimum. The application of the same algorithm to the problem of interconnected hydro utility led to suboptimal solutions, substantially inferior to the EMMA runs. Therefore a new algorithm was sought.

EMSLP guides the iterative procedure in a manner different from that in Grygier's algorithm. The change in the release policy from one iteration to the other is limited to be not more than a fraction (specifically 30%) of the accepted policy. Due to the application of the limited change, the convergence and stability of the iterative process is substantially improved. The storage variability has in EMSLP a somewhat decreased role. It is used only in the search for a better optimum at the second iteration level. The value of the variable does not necessarily decrease during the program execution. It is reinstalled to the original one at the end of the search on the second level. Therefore the storage variability cannot be used as a convergence criteria as in Grygier's algorithm. Instead, EMSLP checks whether the identified storage trajectory is close enough to the estimated solution, used as input into the iteration. The search terminates only if this condition is satisfied. These are the major differences between the two algorithms and proved to be fruitful for the problem of interconnected utility. On the other hand, EMSLP retained the feature of Grygier's algorithm to search the interior of the feasible region to possibly identify a better objective function value. As noted earlier, the need for this search arises from the nonlinear character of the objective function.

CHAPTER 4.

Evaluation of EMSLP

4.1 THE CASE STUDY

A small hydro-electric system based on the Manitoba Hydro system data was designed to test the performance of the newly formulated EMSLP algorithm. The bench mark for comparison were the results obtained by running the EMMA program for the same case study. In order to enable the comparison the case study had to be formulated to suit both of the models.

The system consisted of a single reservoir, power plant, and a tieline which enabled to import energy to satisfy the defined load and to export it if desired. The case study is schematically shown in Figure 5. The reservoir size was chosen to examine the impact of the head variation on the solution. The initial reservoir stage at the beginning of the planning period was set to 90 m (295 ft). The maximum stage was 91.5 m (300 ft) and the minimum 85.4 m (280 ft), with the stage-storage slope of 8 million m^3 m^{-1} (1 KCFS day ft^{-1}). The optimization time horizon consisted of five monthly time steps. The load duration curve of the power demand was discretized to two segments: one for on, and one for off peak demand in each of the time steps as shown in Table 1.

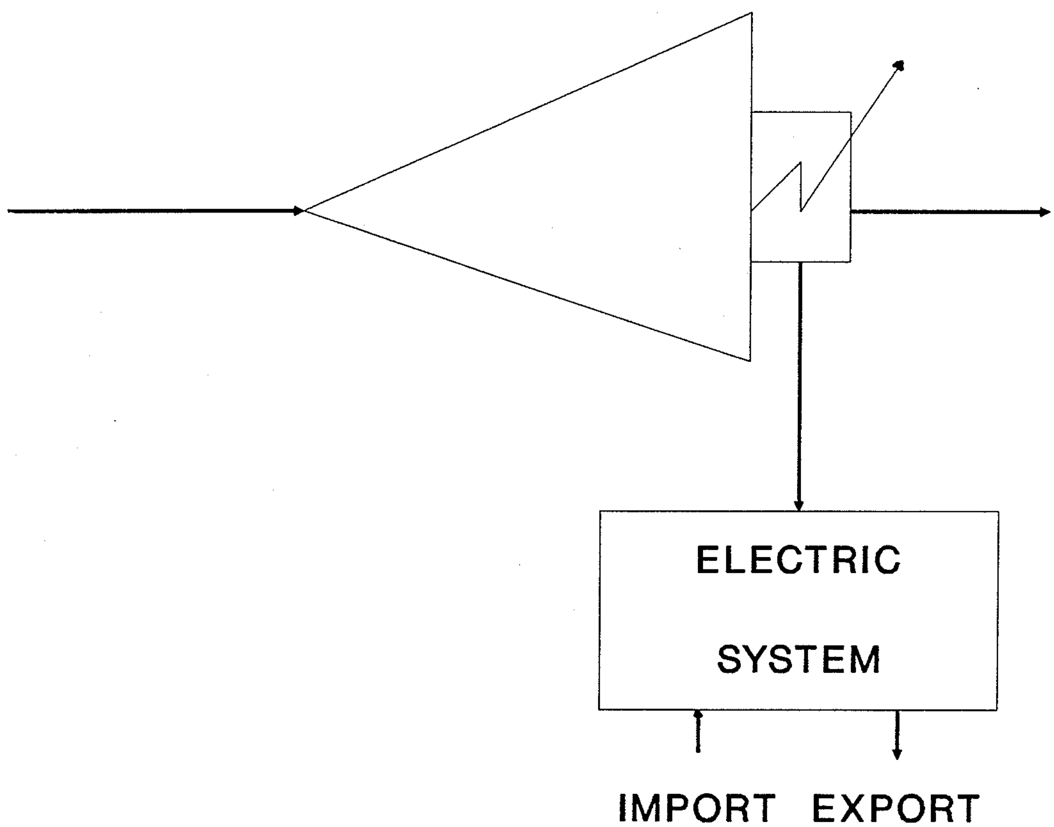


Figure 5. The Case Study

Table 1 The discretized load duration curve

DEMAND (MW)	TIME STEP 1	2	3	4	5
ON PEAK	6	5	6	7	6
OFF PEAK	4	3	4	5	5

The reservoir inflow had a winter pattern typical for Manitoba or other Northern Rivers, as shown in Table 2.

Table 2 The inflow scheme

INFLOW ($m^3\ s^{-1}$)	TIME STEP 1	2	3	4	5
	11.35	9.93	7.09	5.67	4.26

The cost coefficients for the objective function are shown in Table 3, and were chosen to resemble a realistic case existing in the Manitoba Hydro practice (Table 3).

Table 3 The energy price

PRICE ($ GWH^{-1})	TIME STEP 1	2	3	4	5
On peak import	20000	22000	20000	21000	21000
Off peak import	12000	12000	12000	14000	12000
On peak export	14000	18000	14000	18000	18000
Off peak export	8000	8000	8000	10000	9000

The convergence tolerance was set to 5% of the production coefficients (in EMMA). The minimum value of the storage variability for the search of the second level is also used as a convergence tolerance level at the first iteration level. Therefore, the tolerance in the storage variability is set to have 5% accuracy of ERF (in EMSLP). In this way, the convergence thresholds are made identical for both models.

In order to compare the performances of the models the optimization problem had to be simplified to suit both of them. EMMA models the efficiency of the plant as a function of discharge, while EMSLP makes efficiency dependent on the storage. The compromise has been made to take a constant value for efficiency. EMMA also has the capability to calculate the tail water elevation depending on the discharge from the reservoir, which finally affects the net head of the plant. EMSLP treats the tail water elevation as a constant. Therefore a constant tail water level was set to 46.67 m (153 ft). It is interesting to note that with applying the above assumption for the efficiency the ERF function used in the EMSLP model is reduced to be solely a function of the head. Since the tail water is constant the head depends only on the storage in the lake. Finally, the conclusion is that ERF is a linear function of the storage since the stage-storage relationship is assumed to be linear.

In order to solve the problem both programs had nine variables in each of the five time steps. EMMA generated six constraints in each time step, while

EMSLP had nine.

Initially a high head plant was examined. That case had a possibility of significant storage level changes during the planning period. The release trajectories of the EMSLP solutions had up to 400 % higher objective function values than the ones EMMA has identified as optimal. The difference is due to the different linearization process applied to approximate the hydro production function. EMMA's approximation is not suitable for the high head variation during the planning period. To recognize the fact that reservoirs in Manitoba have a very small operation range, the problem was changed to a low head variability case. The operation range was decreased from 30 to 6 m (100 to 20 ft). As it was expected, the decrease in the operational flexibility decreased the differences between the results, too.

4.2 THE RESULTS OF COMPARISON WITH EMMA

The algorithms were compared for a range of different input data on the basis of release policy, iteration number and objective function value. In the input data the value of the final storage, generation release limit, and the imposed system load were varied.

4.2.1 Final Storage Value Variation

The value of the storage in \$ per 2.45 million m^3 (1 KCFS-day) was varied from 3600 where it had no impact on the solution to 4800 where it had an overwhelming impact on the release policy.

Tables C-1 and C-2 in Appendix C contain the results obtained by EMSLP and EMMA, respectively. The reservoir levels obtained by the two algorithms are different (see Figures 6-12). EMSLP tends to store the water at the first time step and to release it later when the price of electricity is higher. EMMA releases a big amount at the beginning time steps to obtain a short term benefit. The energy production of EMSLP is higher than of EMMA due to a higher head maintained during the planning period. The difference can be attributed to the more significant role of the head in the EMSLP modelling of hydro production.

EMSLP responded to the increase in storage value at the value of \$ 3700 by storing more water at the end of the planning period than the required lower operation bound (85.4 m or 280 ft). In EMMA the storage value did not affect the solution even at the \$ 4200 level. Both algorithms kept a full reservoir at the 91.5 m (300 ft) stage when the benefit was set to \$ 4800.

4.2.2 System Load Variation

The original load followed a pattern of a typical five month winter demand. The values were proportionally varied by multiplying the original load with factors

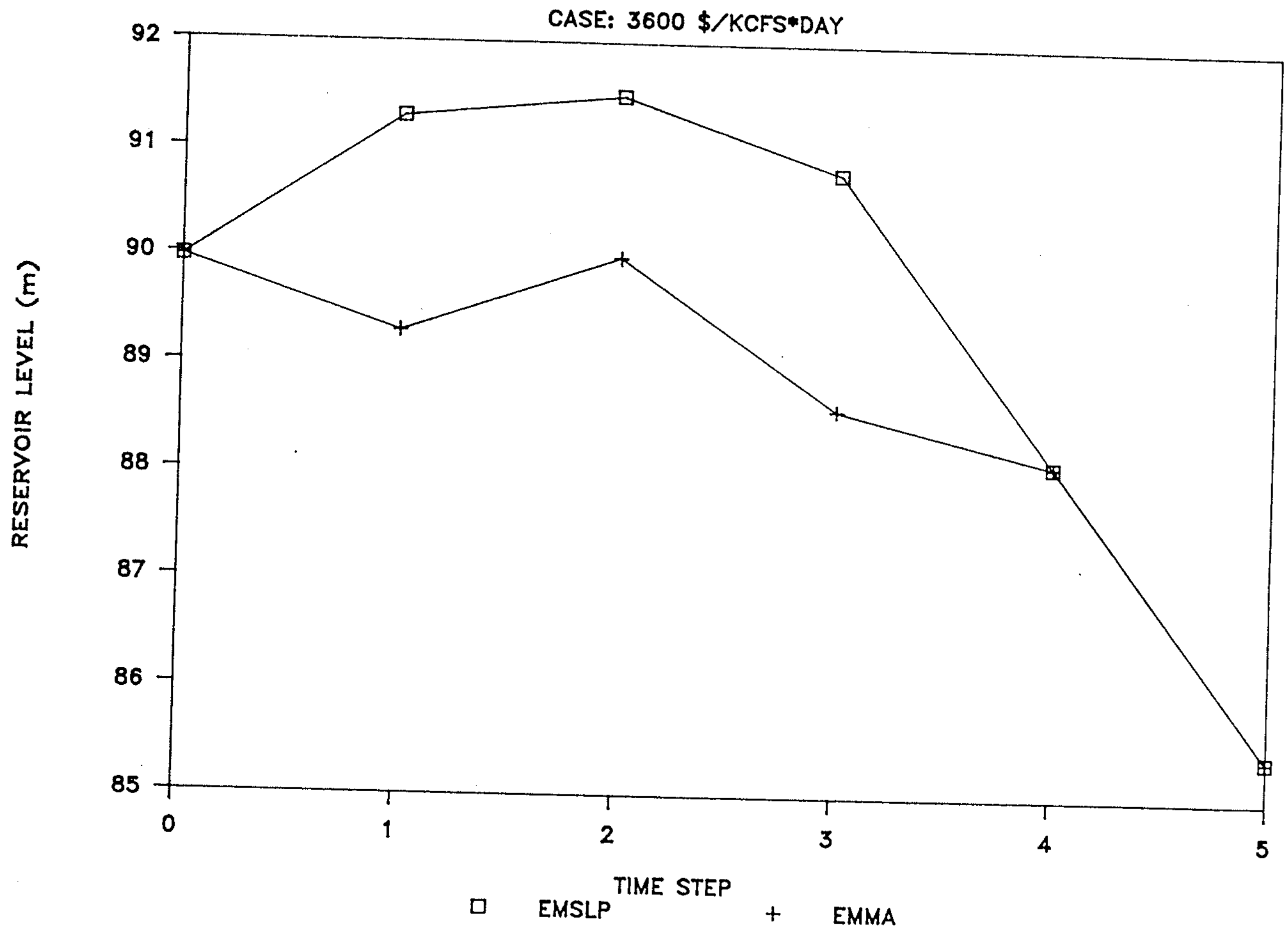


Figure 6. Comparison of reservoir levels for $3600/KCFS-day ending storage value

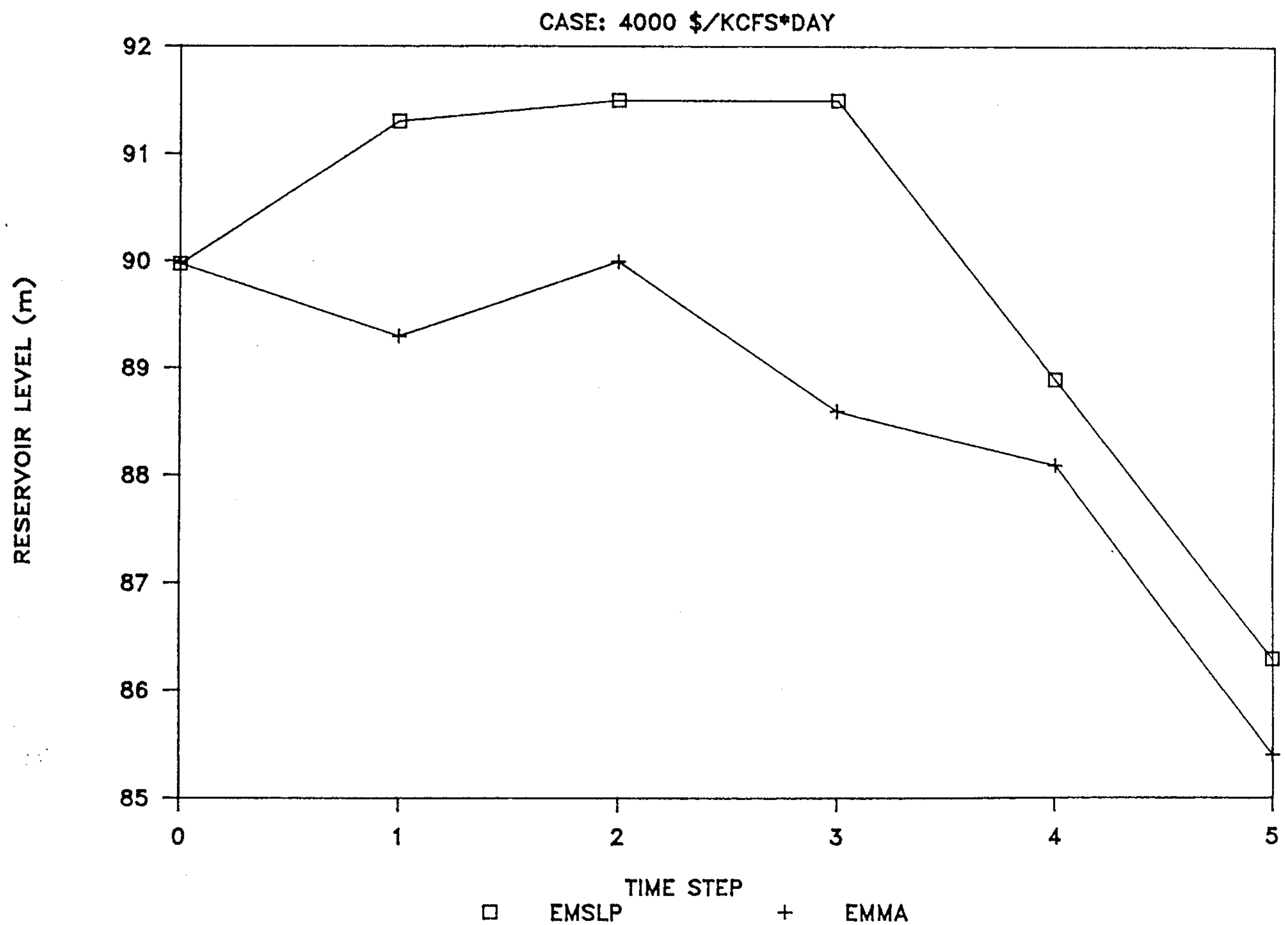


Figure 7. Comparison of reservoir levels for $4000/KCFS-day ending storage value

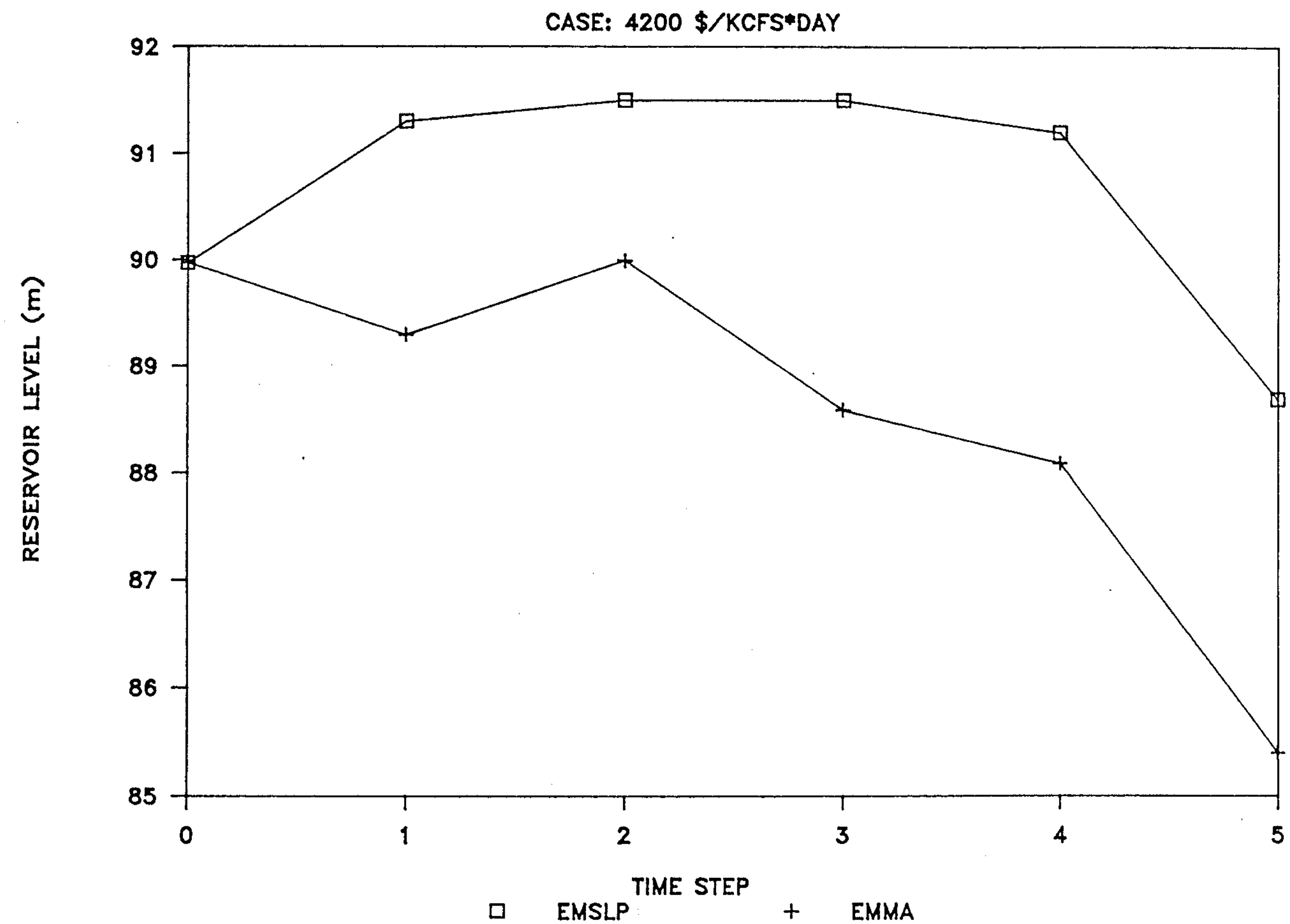


Figure 8. Comparison of reservoir levels for $4200/KCFS-day ending storage value

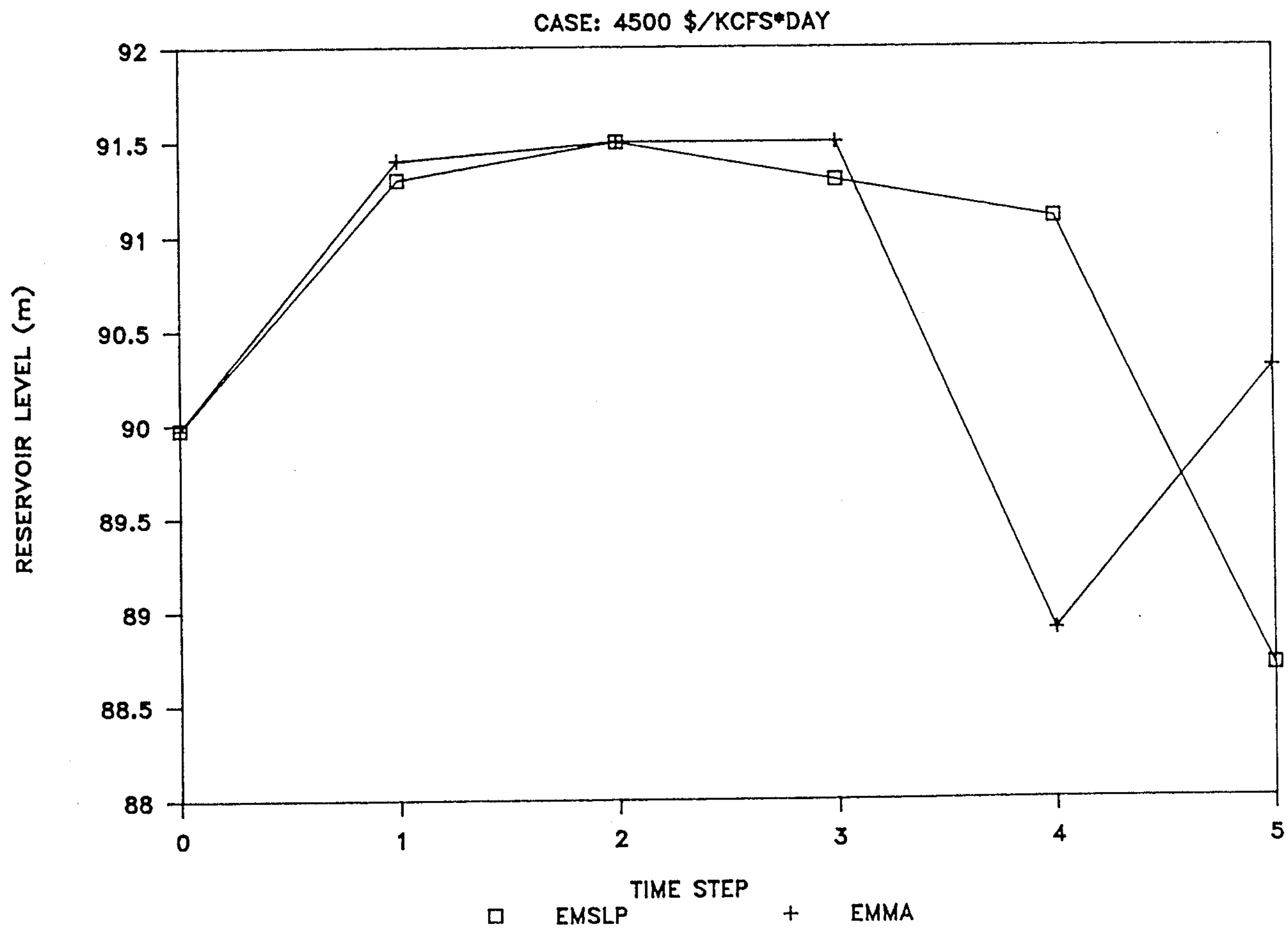


Figure 9. Comparison of reservoir levels for $4500/KCFS-day ending storage value

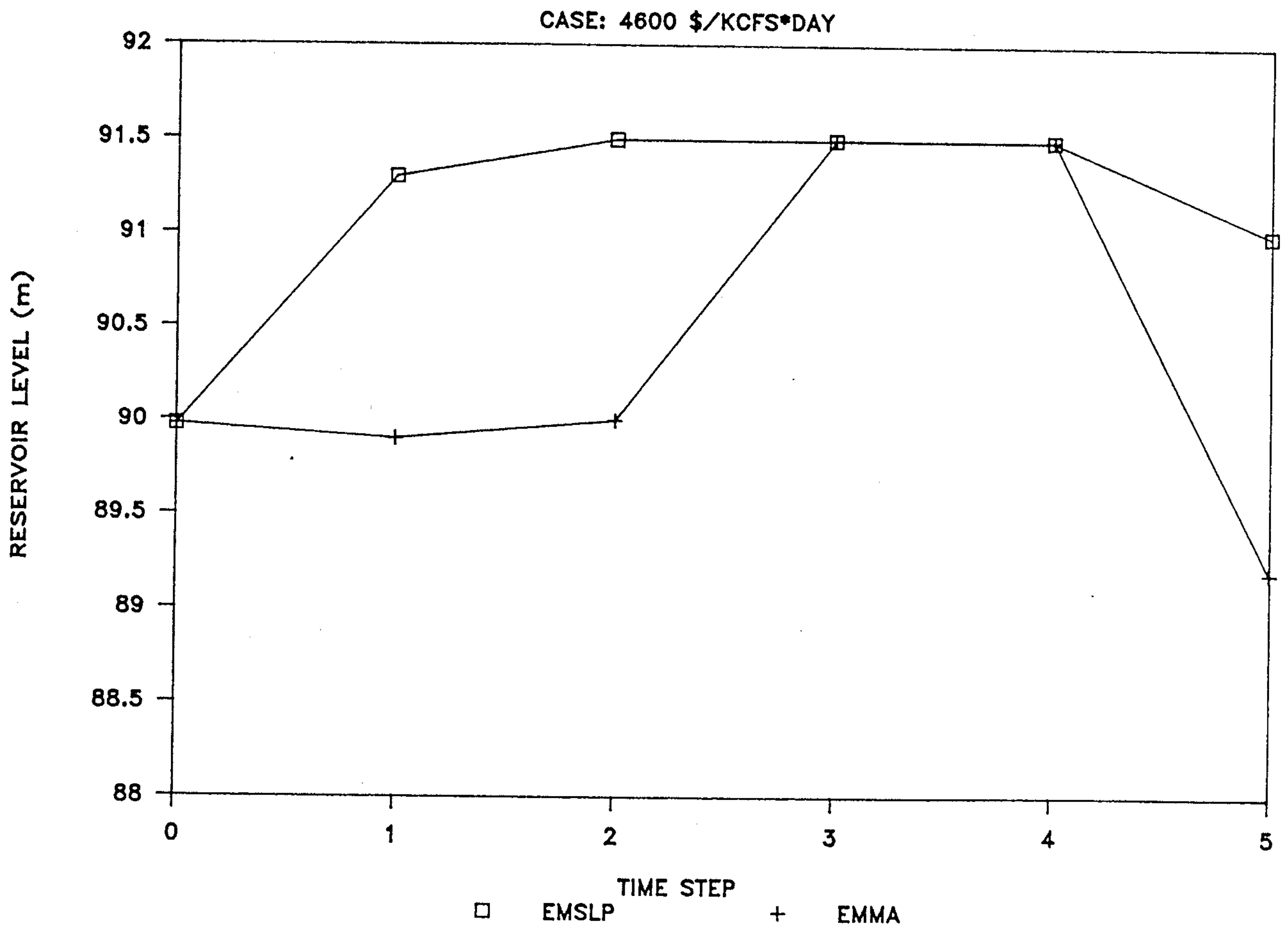


Figure 10. Comparison of reservoir levels for $4600/KCFS-day ending storage value

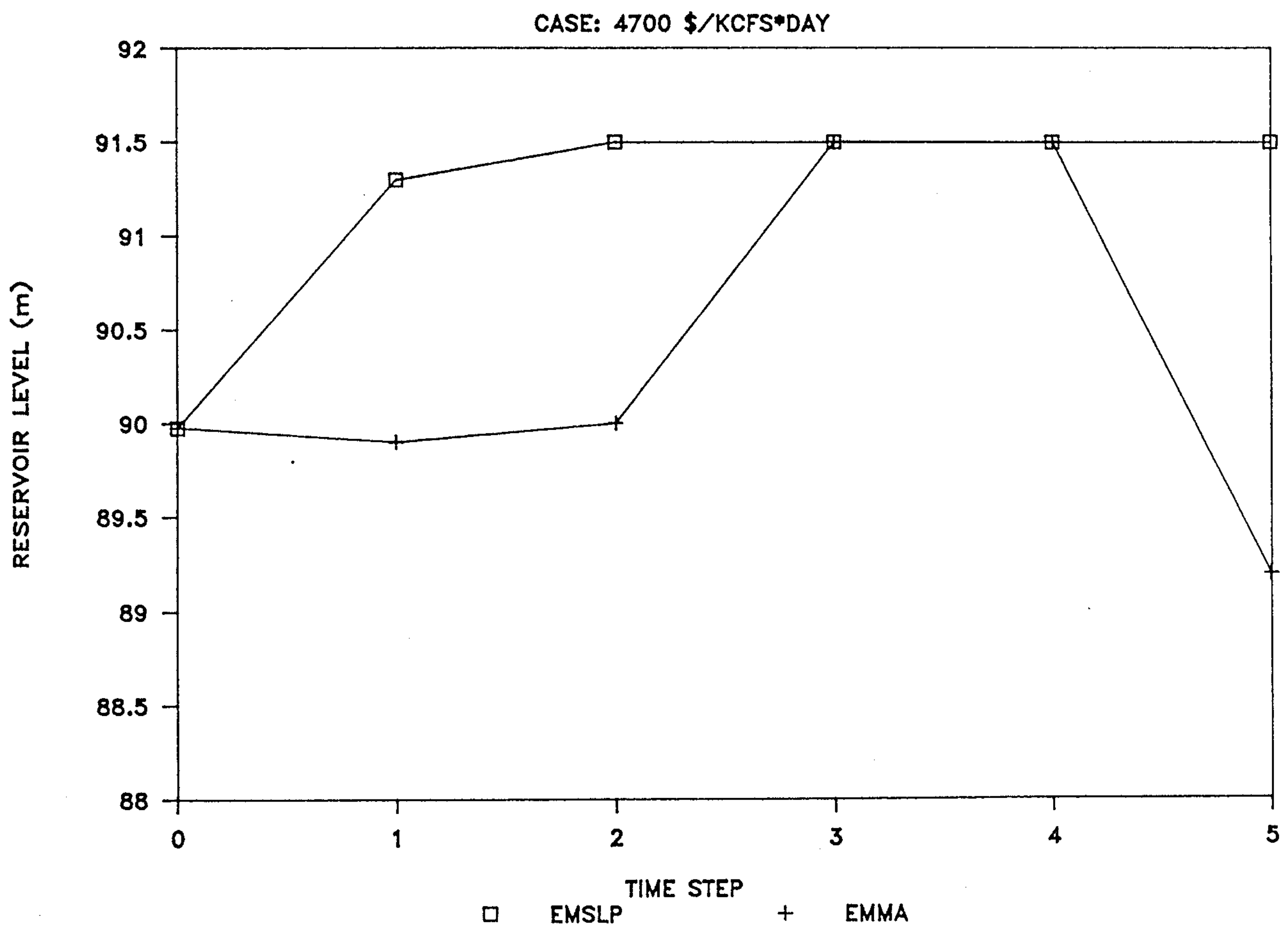


Figure 11. Comparison of reservoir levels for \$4700/KCFS-day ending storage value

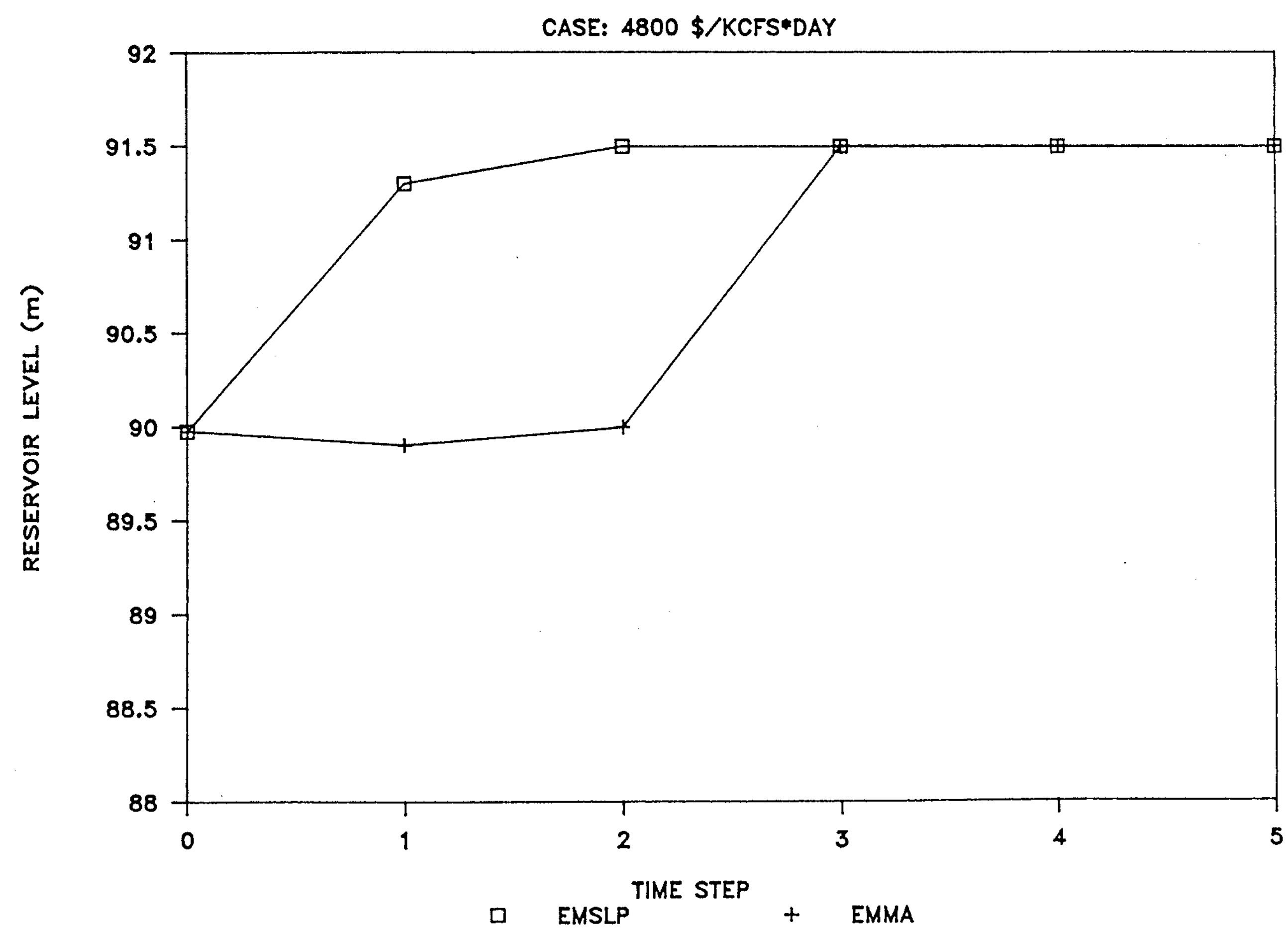


Figure 12. Comparison of reservoir levels for $4800/KCFS-day ending storage value

from 0.3 to 2. The results are shown in Tables C-3 and C-4 in Appendix C and Figures 13-18.

In cases of factors of 0.3 and 0.5, both algorithms identified very similar release policies. This similarity is easily explained: the load was low so it could be satisfied from the domestic production, and the remaining water was saved for the future production. The 0.8 case was also very similar in results, although the EMSLP released less water, and had a 5% higher objective value. The release policies significantly differ in all the other cases. In these others, load is high and the domestic production is not sufficient, import is needed. The available water is released to meet the demand in both of the models but in a different manner. The same tendency could be noticed for the storage value variation: EMMA has large releases in the initial time steps, and in the final ones it can release only the amount of the monthly reservoir inflow. Releases obtained by EMSLP have followed the import price structure: whenever the cost is high the release is high and vice versa. The objective function values differ in about 5% with a tendency for a decrease when the load increases over the original value. The decrease in the differences is due to the overwhelming impact of the import cost.

4.2.3 Release Limit Variation

The turbine release capability was varied from 5.67 to 19.86 m^3/s (0.2 to 0.7 KCFS). The results are presented in Tables C-5 and C-6 in Appendix C and Figures 19-22.

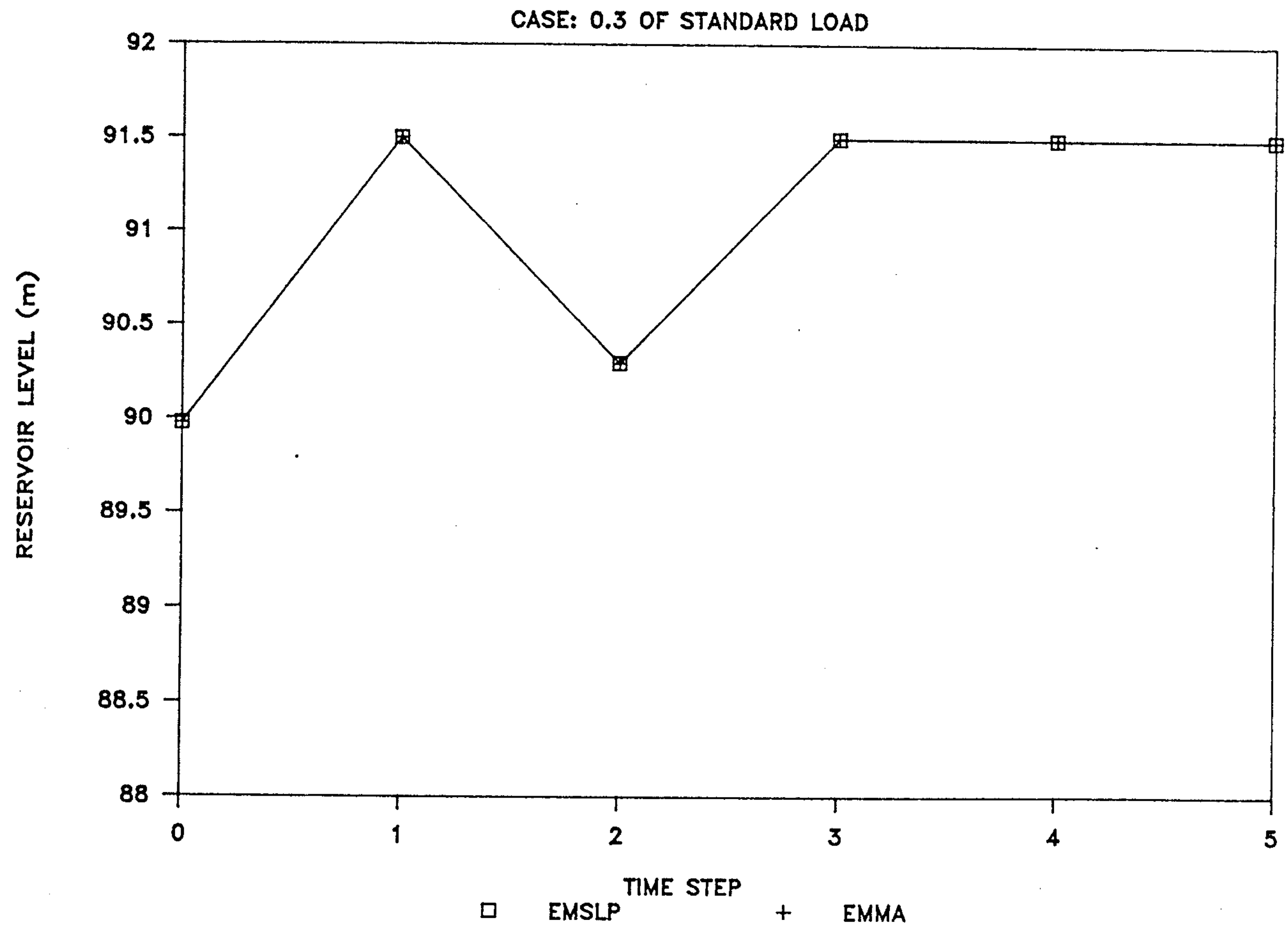


Figure 13. Comparison of reservoir levels for system load equal to 0.3 times the original load

Figure 14

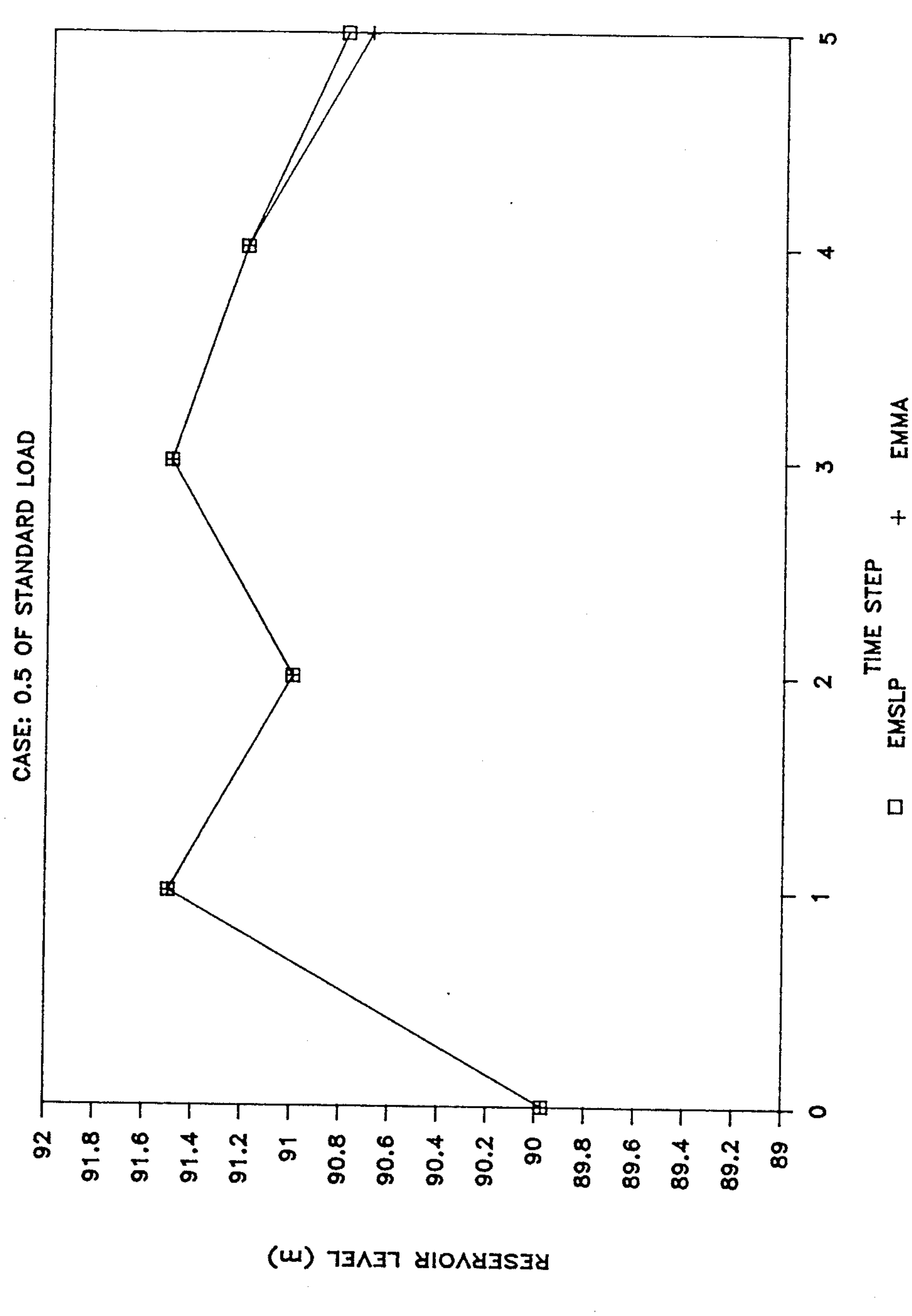


Figure 14. Comparison of reservoir levels for system load equal to 0.5 times the original load

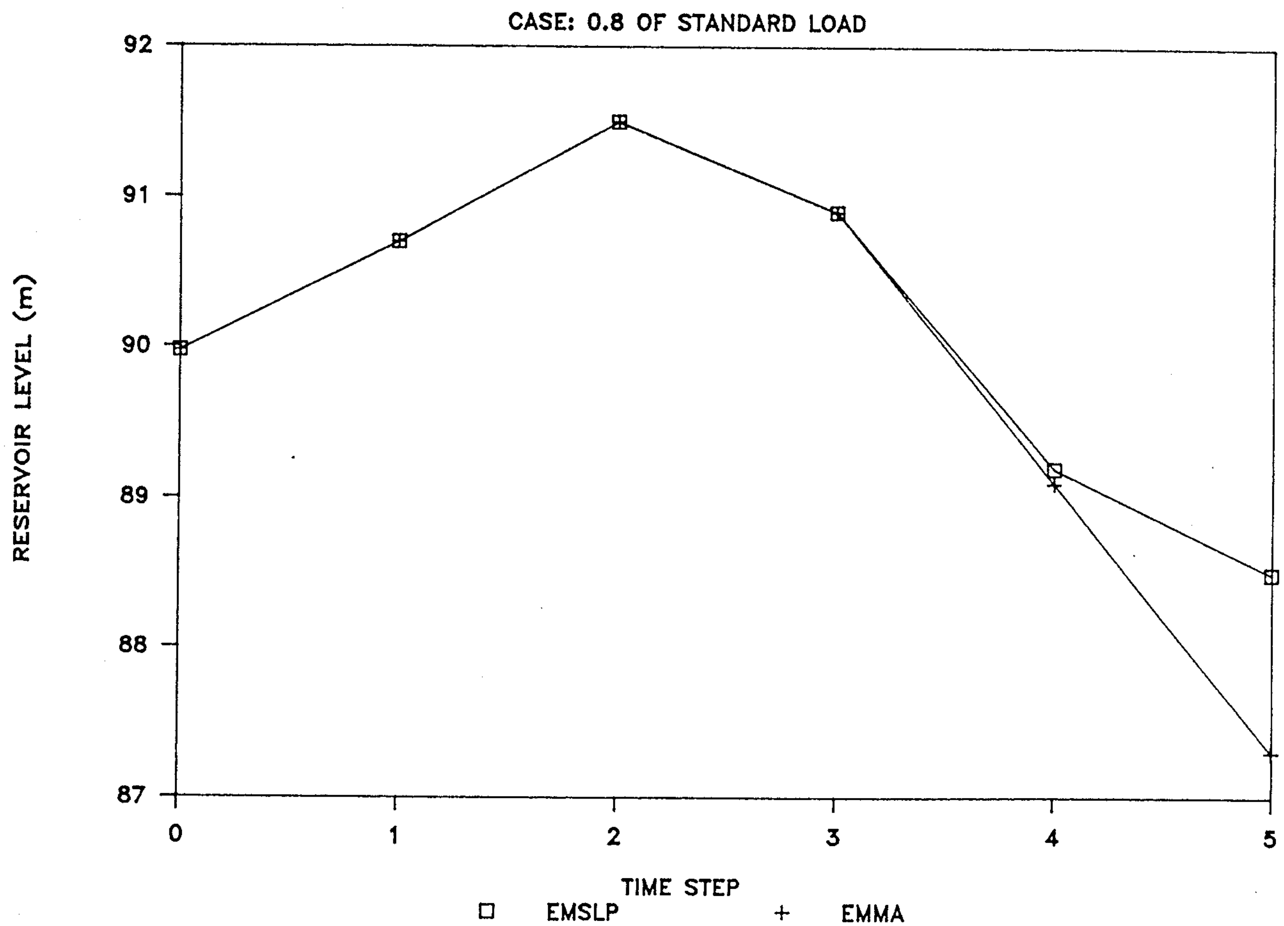


Figure 15. Comparison of reservoir levels for system load equal to 0.8 times the original load

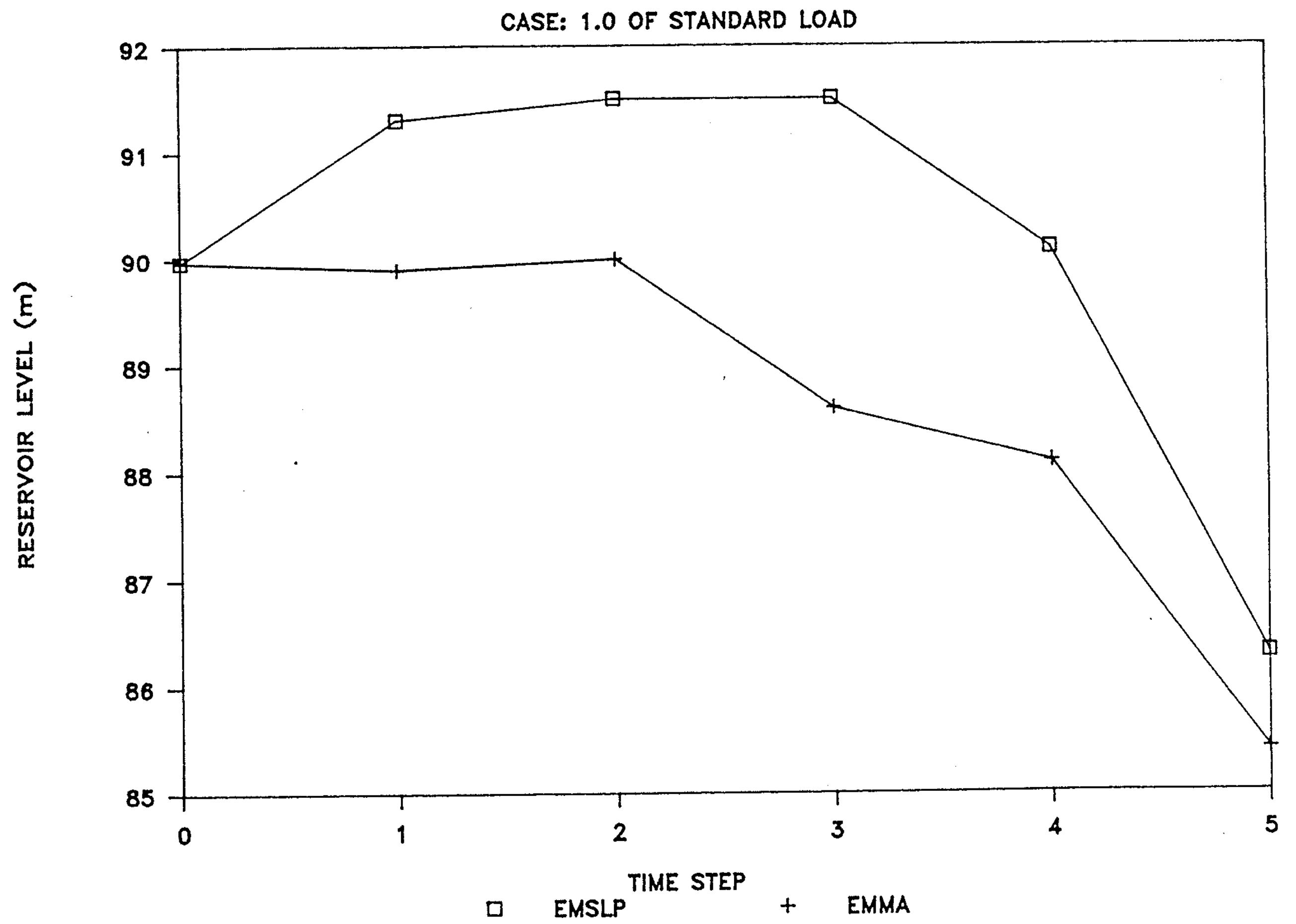


Figure 16. Comparison of reservoir levels for system load equal to 1.0 times the original load

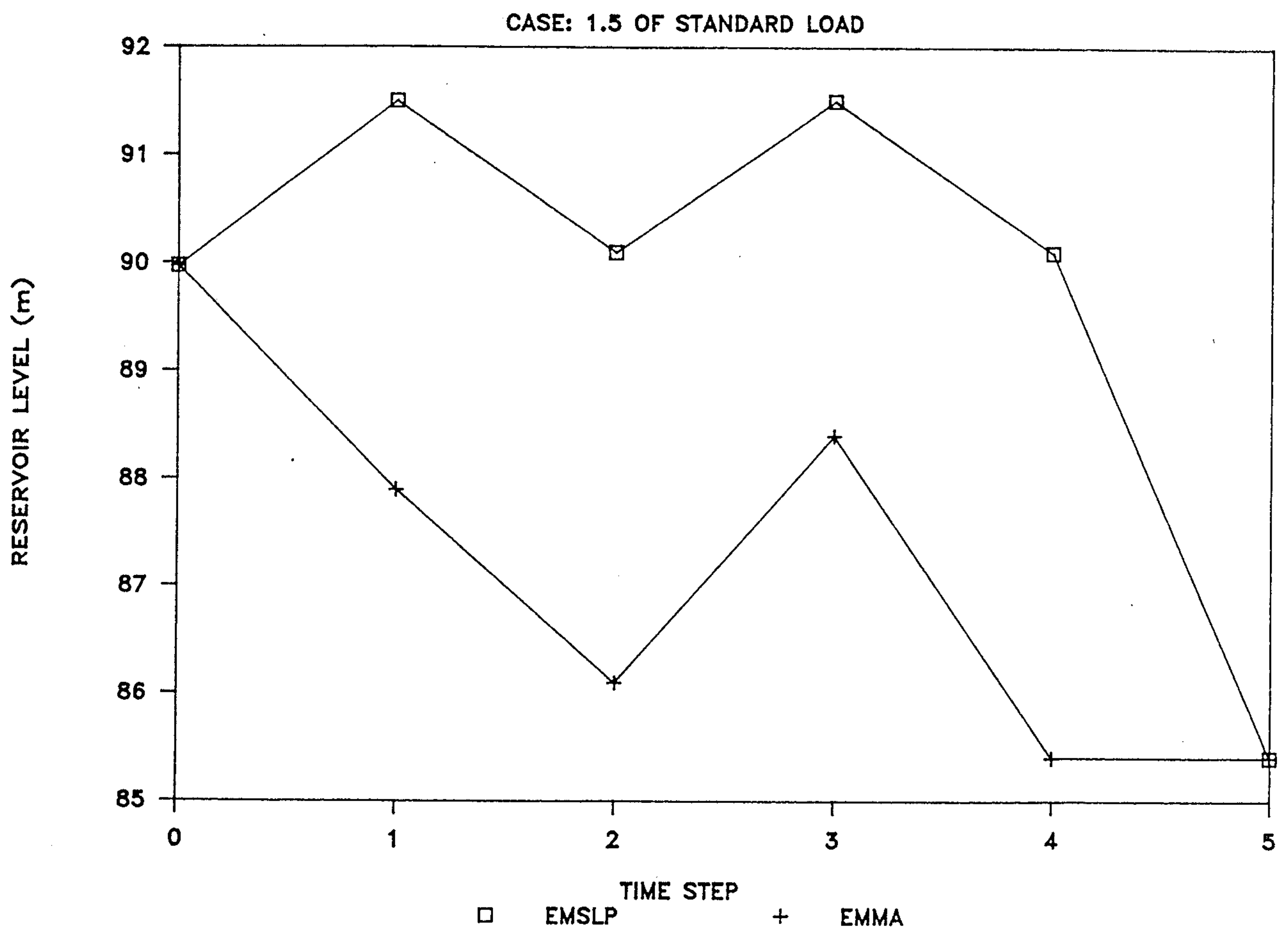


Figure 17. Comparison of reservoir levels for system load equal to 1.5 times the original load

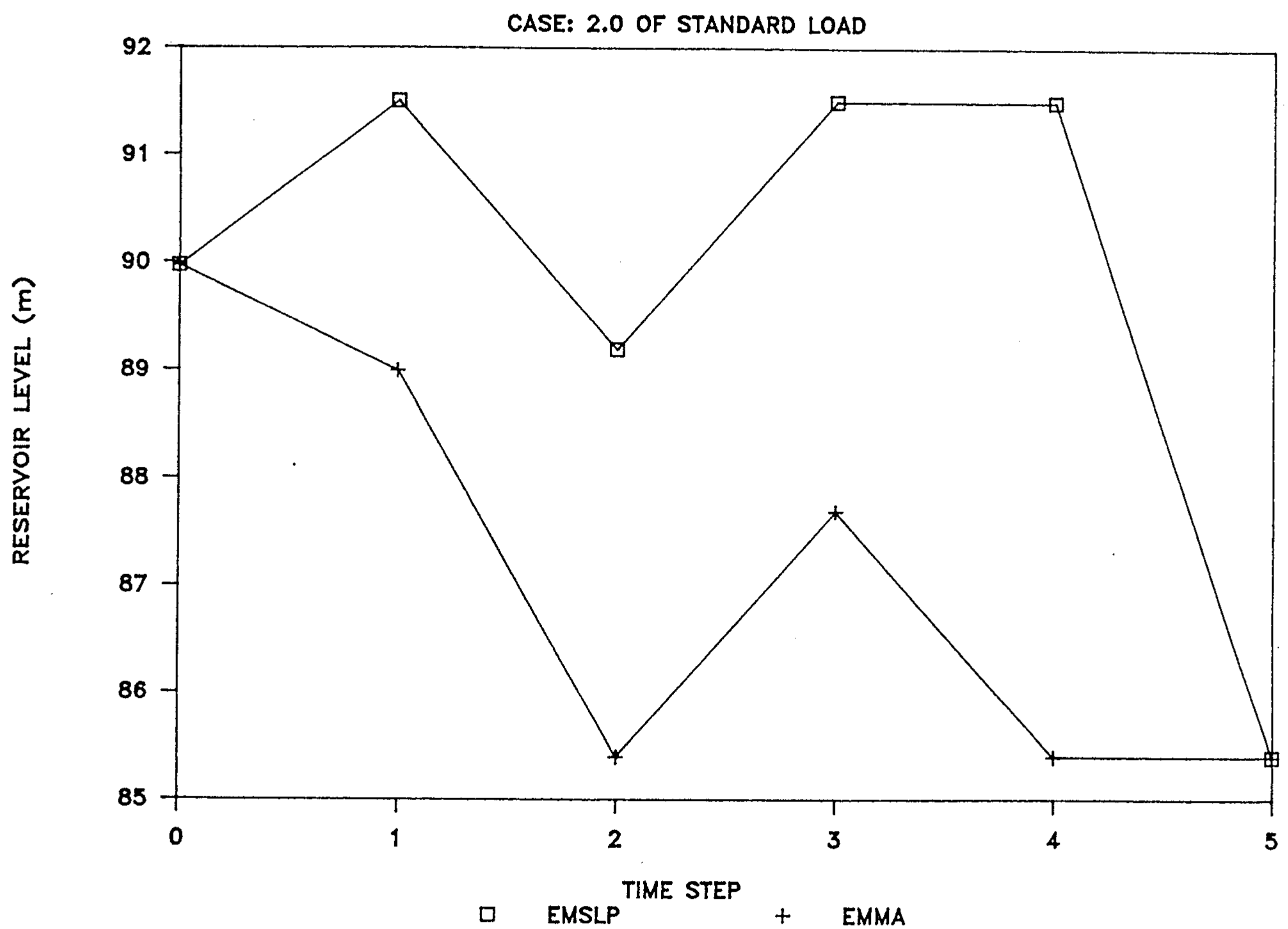


Figure 18. Comparison of reservoir levels for system load equal to 2.0 times the original load

Tables C-5 and C-6 show that in the 19.86 case the release capability is not constraining the release policies. In the 5.67 case spill occurs since the water can be neither stored in the reservoir, nor released through the turbines. EMMA spills a huge amount of water in the first time step. This is irrational, since the level of the reservoir after the first time step is even lower than the initial 90 m (295 ft). The rational action would be to maintain a full reservoir at the 91.5 m (300 ft) level. EMSLP maintains a full reservoir till the end of the planning period, and spills only the excess water in each of the time steps when the need arises. This difference gives the higher objective value in EMSLP. The 8.57 limit eliminates the need for spill. The solutions are very similar, since the releases are bounded by the limit. In the rest of the cases the impact of the release limit is not so dominant, and significantly different release policies were identified. The objective functions differ about 5% in favour of EMSLP.

Finally, the difference in the objective function values between EMSLP and EMMA programs normalized to the EMMA results are presented in Figures 24 to 26. Figure 24 represents the comparison for storage value variation. Figure 25 shows the normalized difference for different system loads. Figure 26 represents the comparison of objectives for the release limit variation experiments.

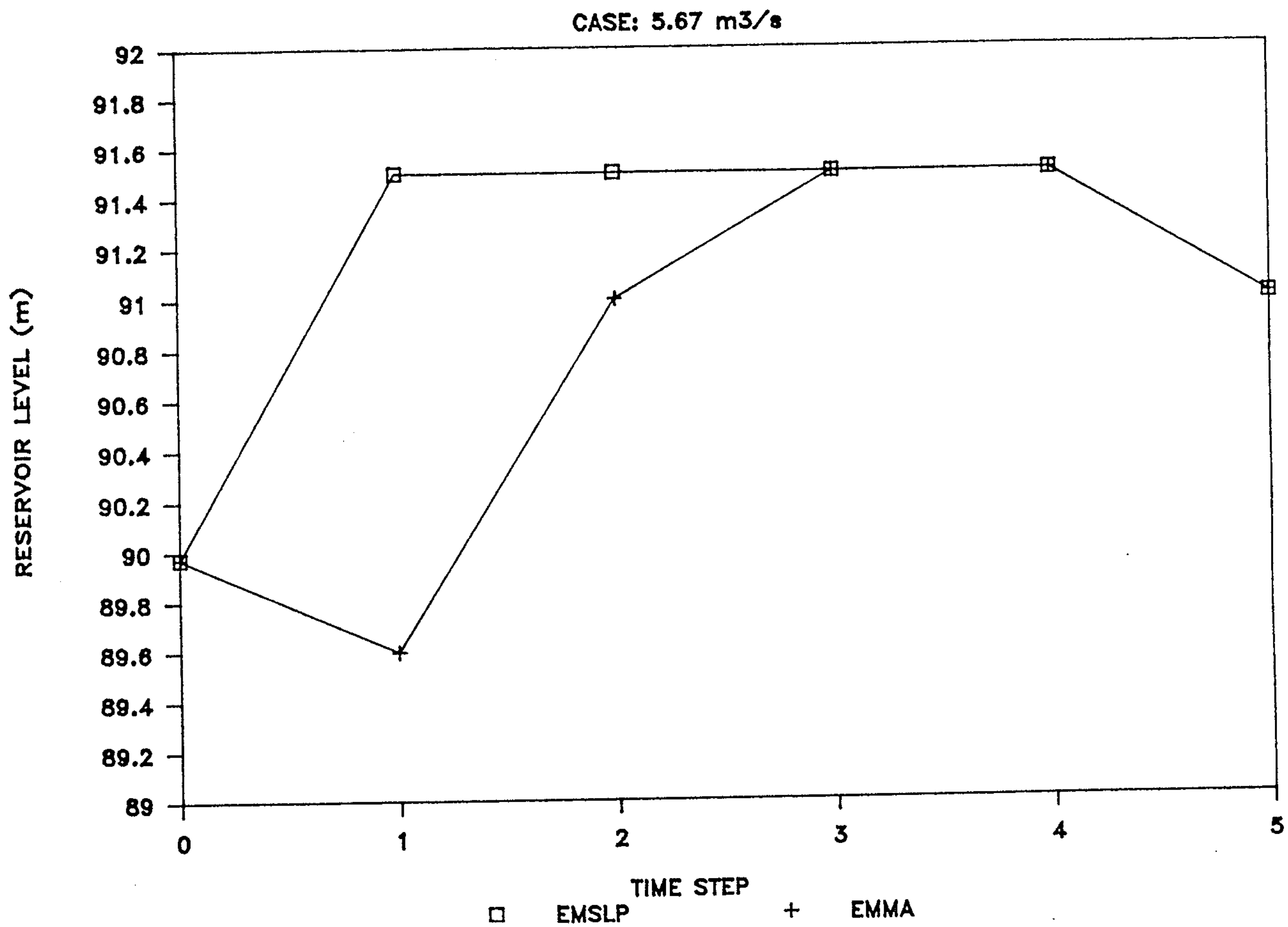


Figure 19. Comparison of reservoir levels for a release limit of 5.67 m^3/sec

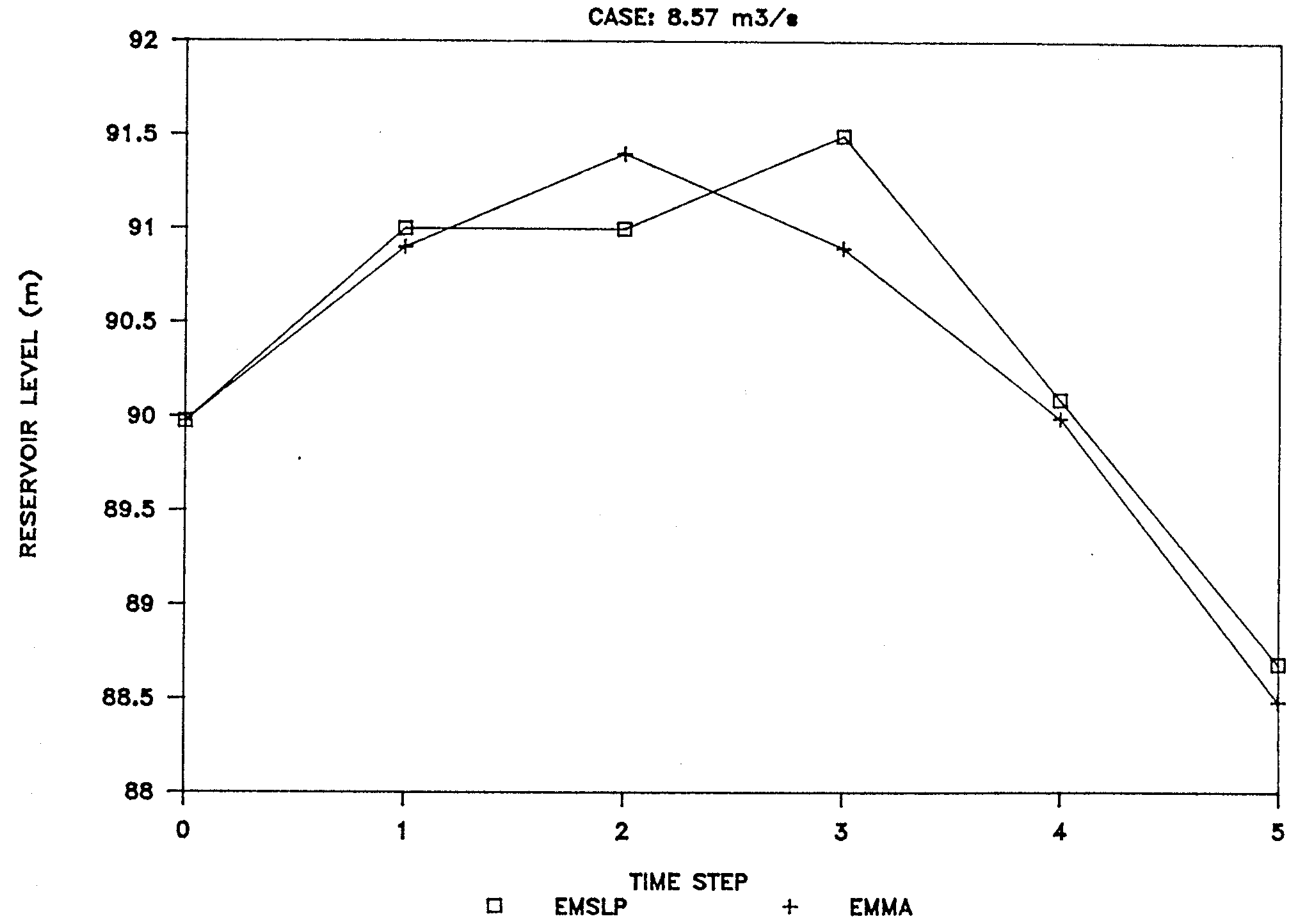


Figure 20. Comparison of reservoir levels for a release limit of 8.57 m^3/sec

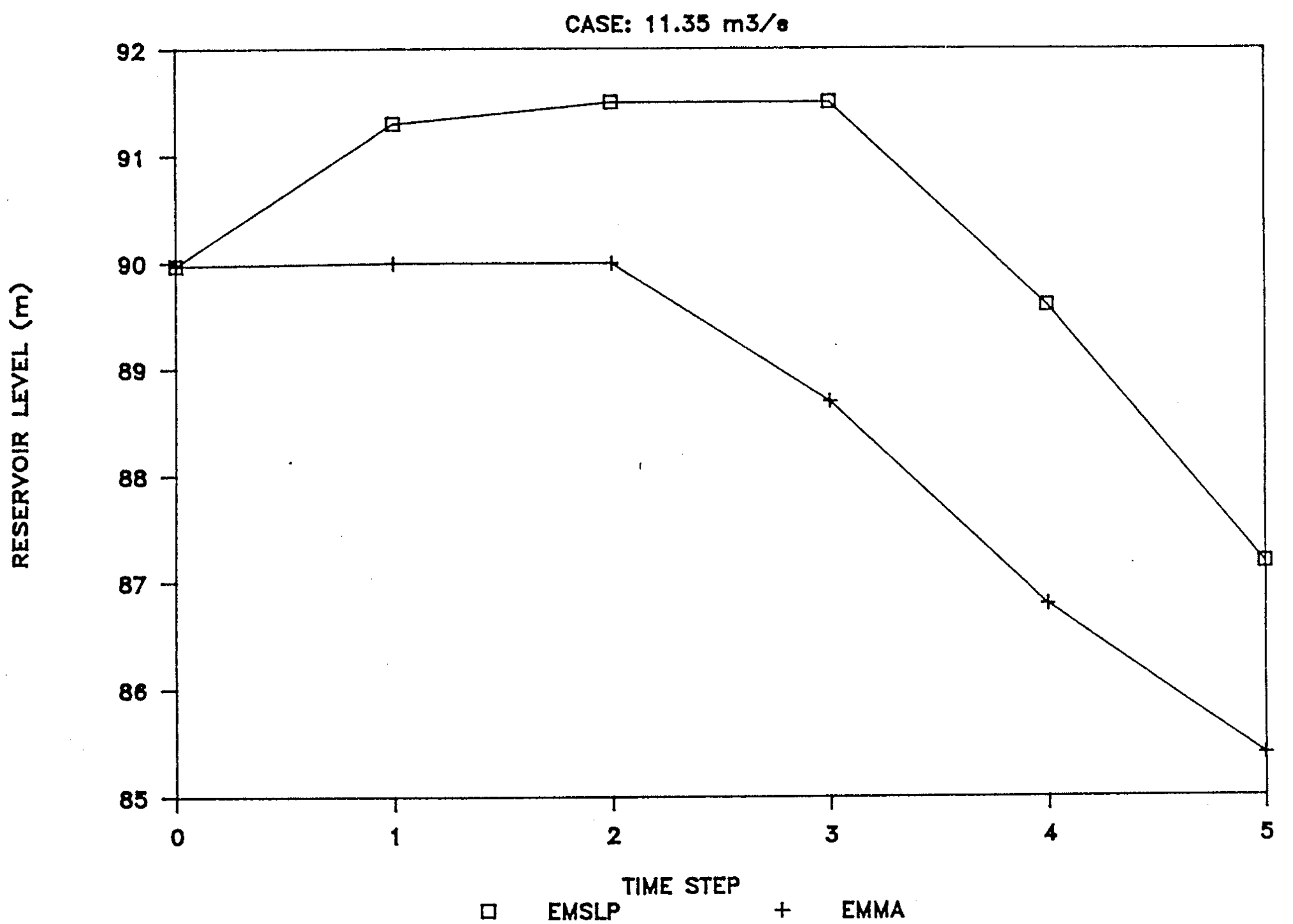


Figure 21. Comparison of reservoir levels for a release limit of 11.35 m^3/sec

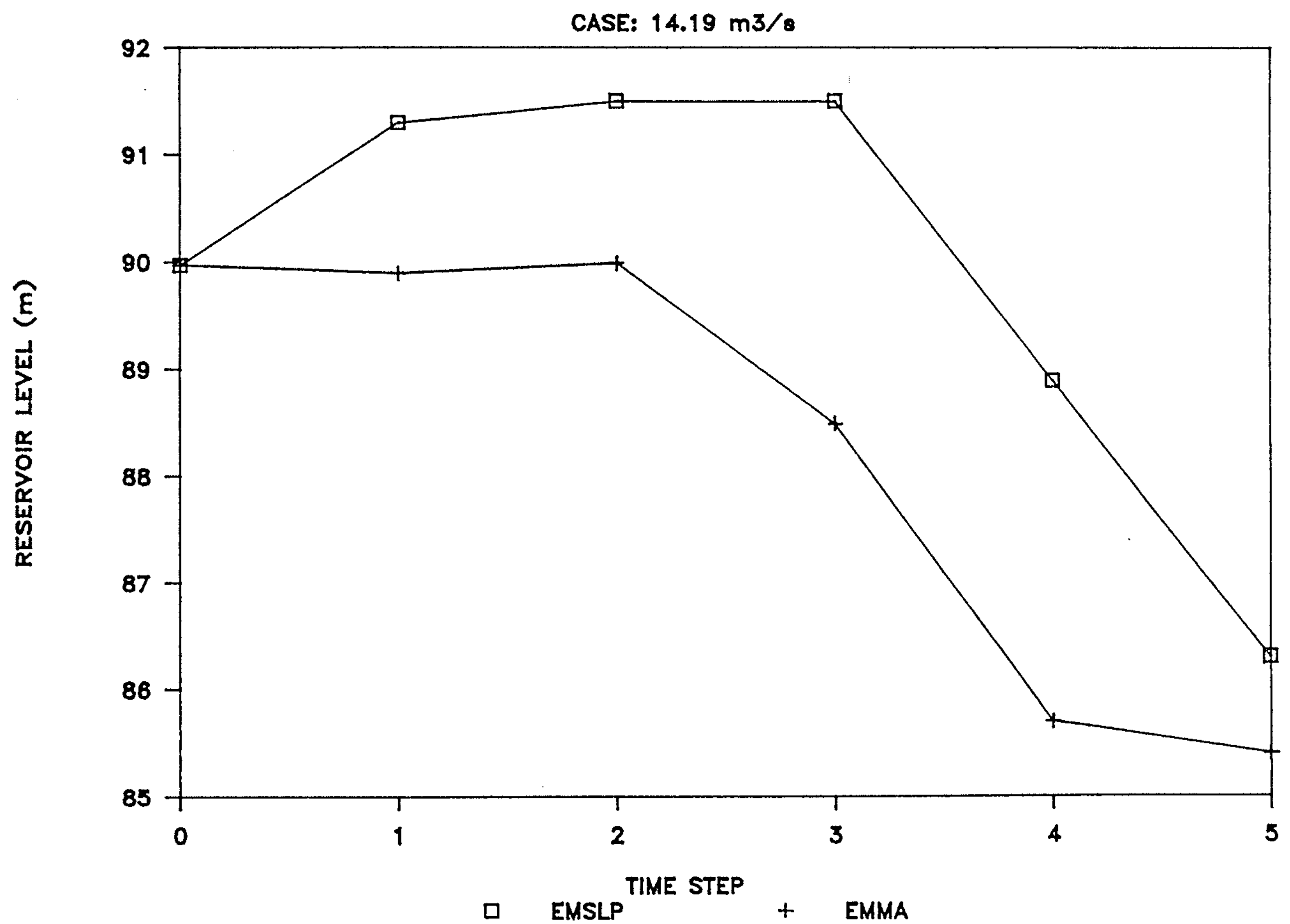


Figure 22. Comparison of reservoir levels for a release limit of 14.19 m^3/sec

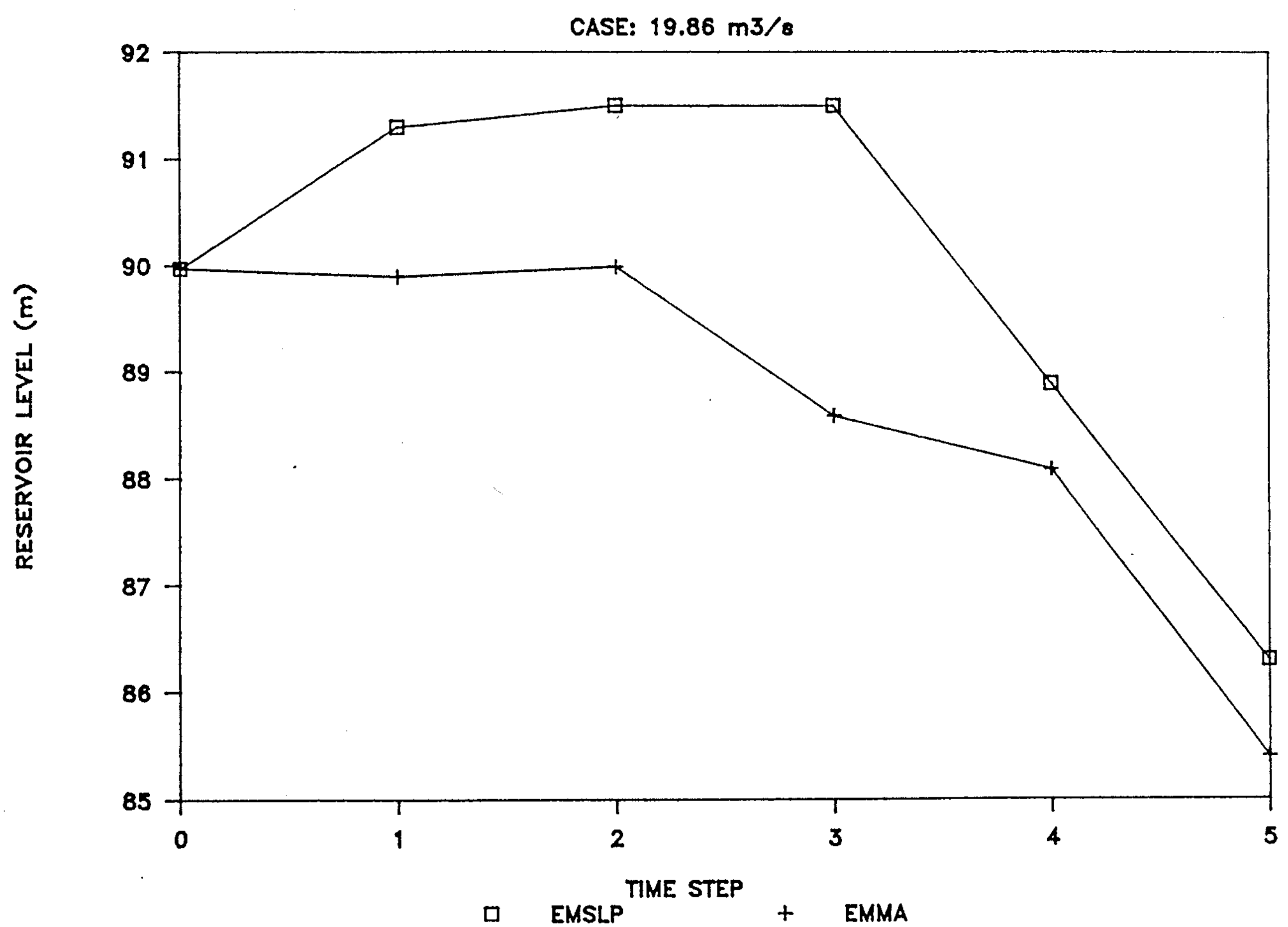


Figure 23. Comparison of reservoir levels for a release limit of 19.86 m^3/sec

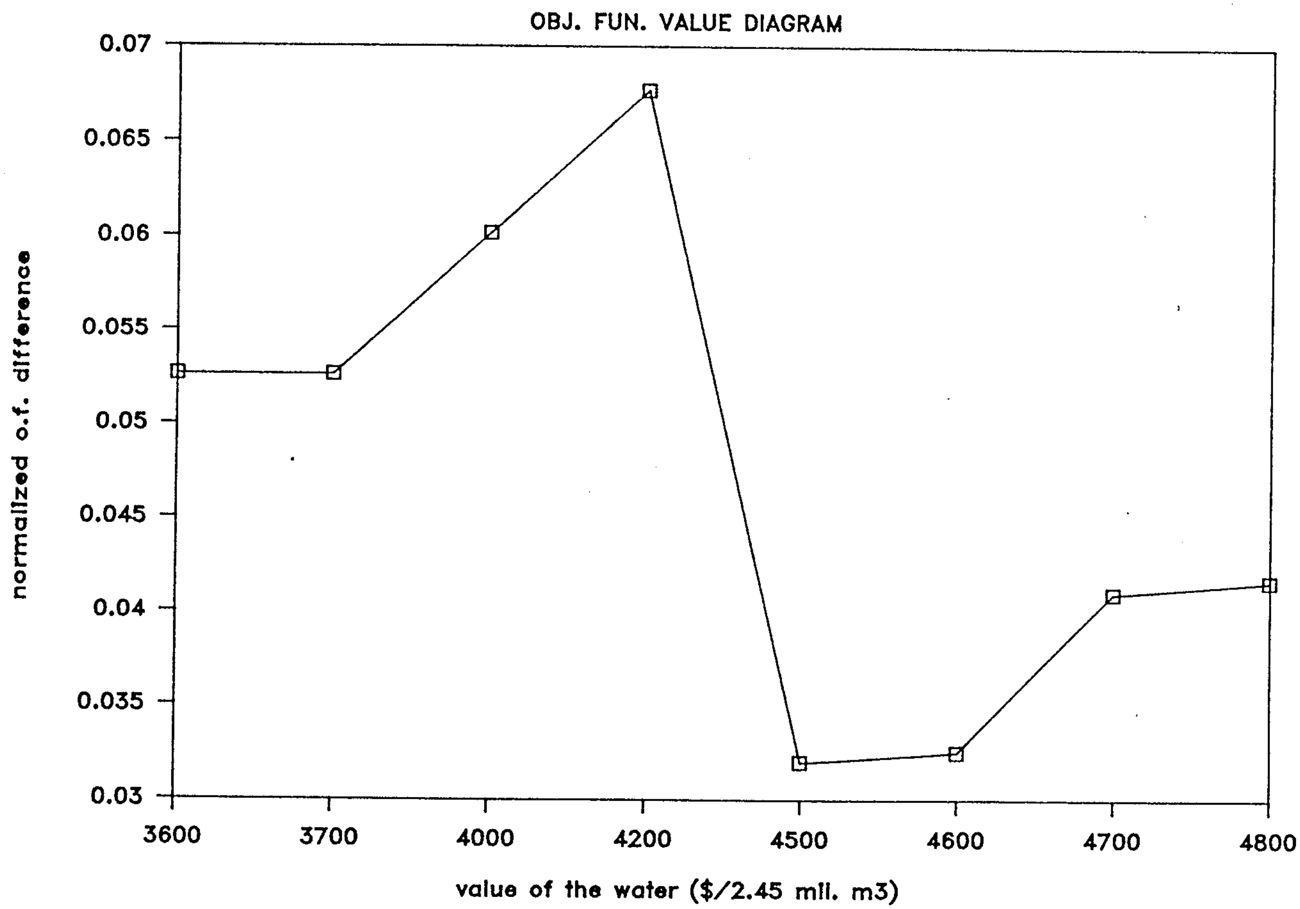


Figure 24. Normalized objective function difference for storage value variation

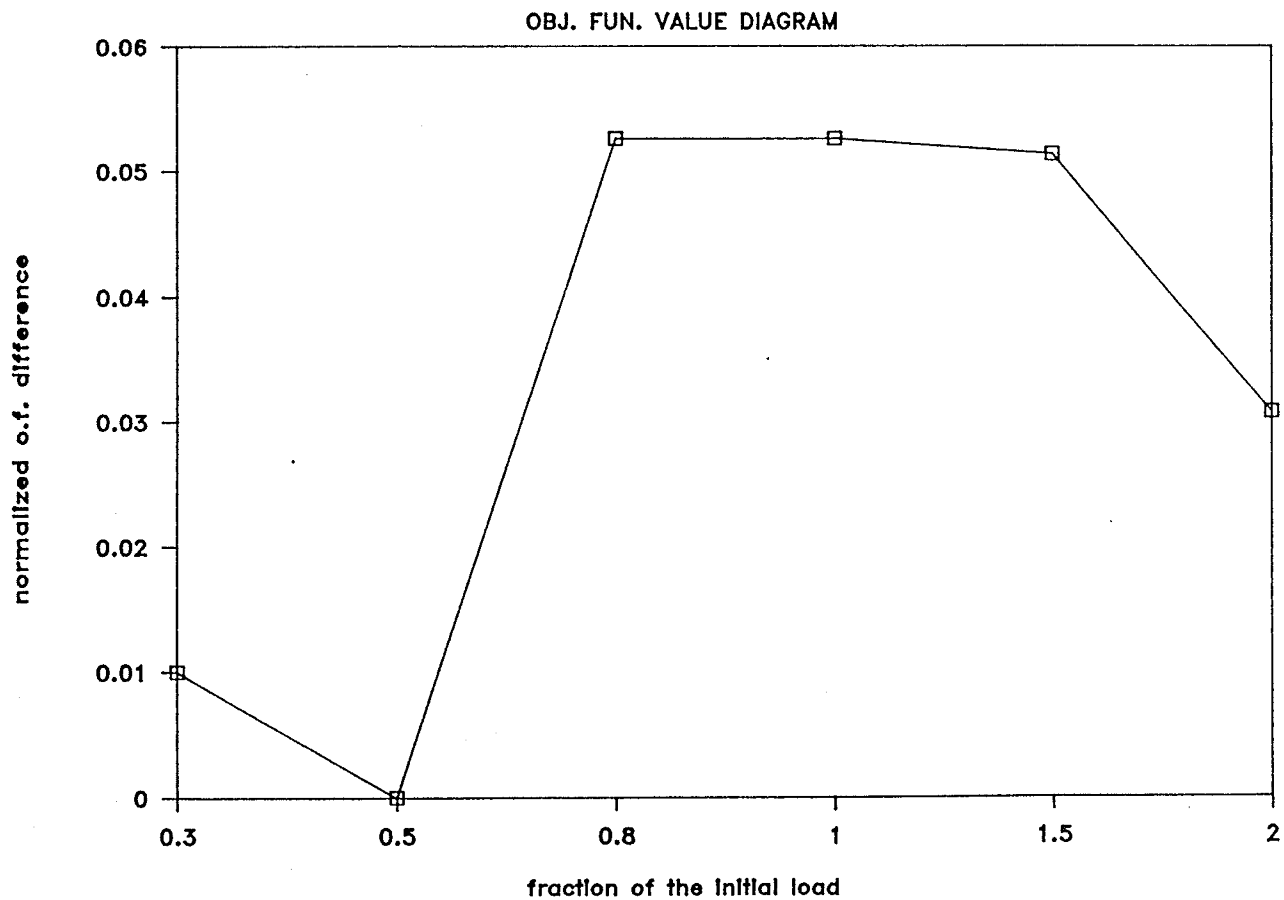


Figure 25. Normalized objective function difference for load variation

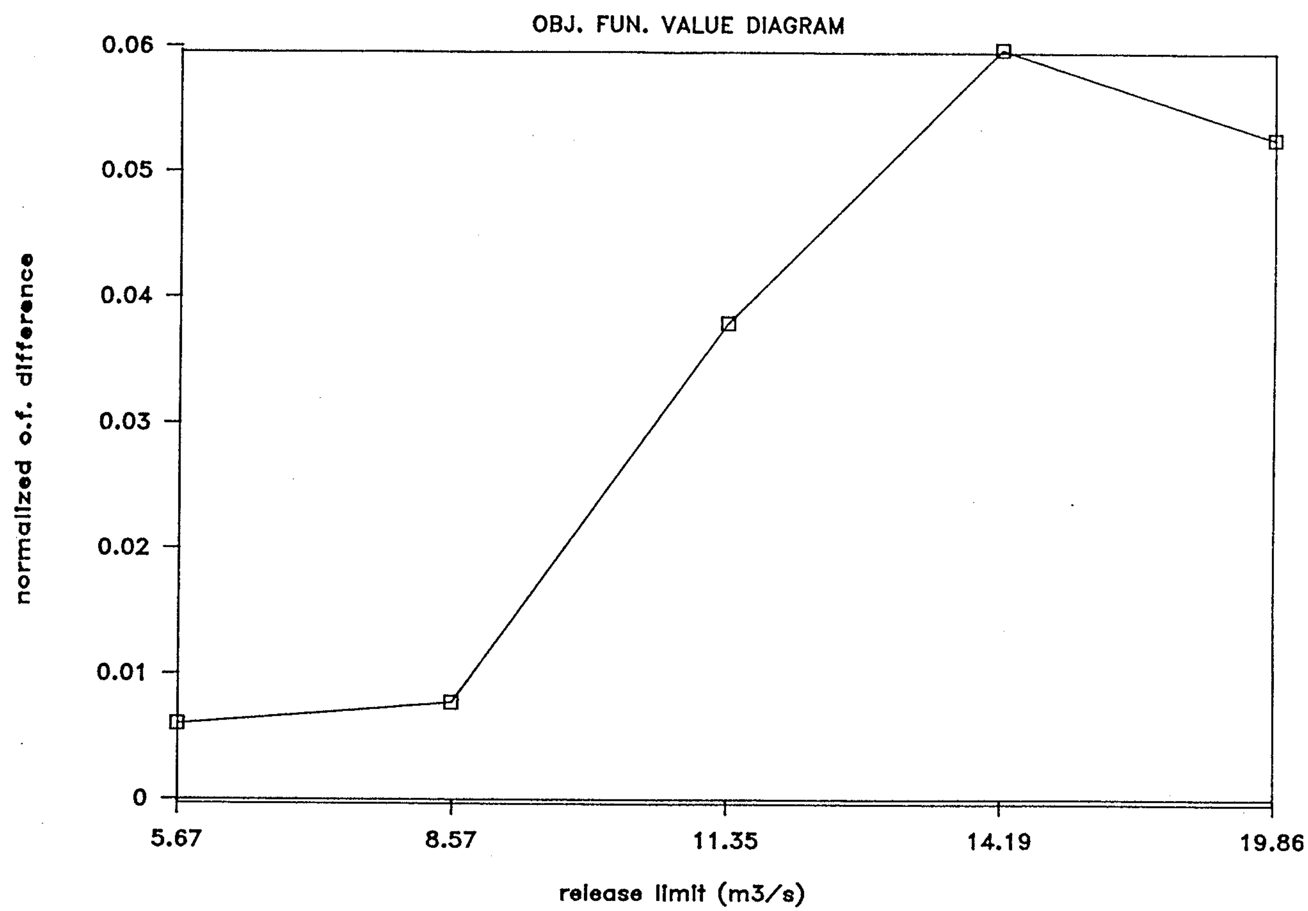


Figure 26. Normalized objective function difference for release limit variation

CHAPTER 5.

CONCLUSIONS

5.1 DISCUSSION OF THE EMSLP-EMMA COMPARISON RESULTS

The variation of input data indicated that the two algorithms identify similar solutions only when they are constrained to do so (e.g., very low release limit), or when the optimization problem is very straightforward (e.g., very low system demand). In cases when the requirement for trade off between production, export, import, and storage use was noticeable, EMSLP presented better results than EMMA (the final case in Tables C-1 and C-2 in Appendix C).

EMMA was not able to adjust the release policy to the existing price structure as successfully as the EMSLP algorithm. When the system load was high and energy had to be imported to satisfy the demand (the last two cases in Tables C-3 and C-4 in Appendix C) EMSLP tended to save the water during the time steps when the import energy price was lower (see Table 3) and to satisfy the demand mostly by import (see Figures 17 and 18). The water was released (i.e., the energy was produced at a domestic power source) at time steps when the import energy price was higher. This adjustment to the price structure on the energy market could not be experienced in the EMMA results. However, by repeating the model run and constraining the release the EMMA results can be improved. In general, EMSLP tended to maintain a higher power plant head during the planing period and therefore obtain a higher power output for the same release (see Figures 12, 17 and 18). The operation with a higher head in the EMSLP case can be attributed to the more important role of the storage variable in the hydro power modelling. EMSLP identified a more rational solution in the case when a mandatory spill had to occur from the reservoir during the planning period due to the very low limit of the turbine release capabilities (see the first cases in Tables C-5 and C-6). EMSLP spreads the spill throughout the planning period and keeps the reservoir full maintaining the high head. EMMA spills a huge amount in the first time step disregarding the impact on the head (see Figure 19). By interventions in the constraint set and re-executing the model these irrationalities can be avoided in the EMMA model.

The objective function value was the same in the cases of low system demand and release limit but the difference of up to 5% was obtained for the more complex situations with the trade-off between energy and storage. It has to be added that the size of the reservoir substantially influences the variation in the results: the bigger the reservoir operation range the bigger the variation. The resulting differences always favoured EMSLP, as it is illustrated in Figures 5 to 7. When the storage value or the load had extremely high values, the differences between the objective function values obtained by the two programs decreased. This is due to the overwhelming impact of that particular high input value on the objective function. For example, high system load requires very high energy import and the reservoir operation, whether its rational or not, has only minor impact on the objective function value (the final case in Tables C-3 and C-4, and the illustration in Figure 6). The number of iterations was very similar for both of the programs. The releases were substantially different.

The behaviour of both algorithms has been examined with different choice of the initial value for production coefficients. It is shown that the solutions of both algorithms are independent of the initial value of production coefficients. The EMSLP has also been tested on the impact of the allowed change in the release policy from one iteration to another. The allowed change in release policy of 30% seemed to give the best stability and the least number of iterations.

The newly formulated algorithm applied to the case study achieved better results than the original EMMA algorithm. However, it should be emphasized that the reservoir size and operation range play an important role in obtaining differences between the two results. EMMA and EMSLP results differ a little when the models are applied to a run-of-river plant configuration, although some irrationalities may be avoided by applying EMSLP. EMMA is applicable to run-of-river and low reservoir operation range plants exclusively. The advantage of EMSLP is that it is equally well suited for all kind of plant characteristics.

5.2 DEVELOPMENT OF THE EMSLP RESEARCH

An original SLP algorithm has been developed to optimize the operation of a hydro-electric utility. The algorithm identifies stable solutions due to the improved iterative modelling of the hydro production function. The approximation of the hydro production function with the first order Taylor expansion proved to be more efficient than the constant production coefficient approach. The limitation of the change in the release policy from one iteration to the other, and the introduction of a stability check has made possible to apply the ideas from Grygier's algorithm to the specific problem formulation of the interconnected hydro utility. These additions have resulted in a completely new, two level algorithm with the prime goal to identify a stable solution and to possibly improve the value of the objective function by looking for the optimum in the interior of the feasible region.

5.3 DIFFERENCES BETWEEN EMSLP AND GRYGIER'S SLP MODEL

The problem of modelling the optimization of an interconnected utility requires decisions about energy management and therefore the existence of energy variables in the model. Grygier's model does not incorporate energy variables since the only energy considered in the model is the hydro energy. The EMSLP objective function and hydro production constraint are significantly different from Grygier's formulation. The difference is due to the existence of energy variables in the EMSLP model. The different problem formulation requires a different guidance of the iterative solution procedure. Instead of a search in a steadily decreasing solution space (as in Grygier's algorithm), EMSLP follows a different path. At the first level, iterations are performed over a solution space of an initial width and occasionally the interior of the solution space is searched at the second level when the objective function drops. Besides, Grygier's algorithm accepts the LP calculated release policy to be the estimate for the next iteration regardless of the previous estimate. EMSLP evaluates the difference between the calculated and assumed release policy (used to obtain the calculated one in the previous iteration). The change of the estimated policy is limited to 30% in the direction of the calculated release policy. With this measure the stability of the iterative process is improved.

The introduced differences in the iteration process guidance and input updating proved to be fruitful for the optimization of interconnected utility.

5.4 DIRECTIONS OF FUTURE RESEARCH

The current EMSLP model will be expanded in the future to have more realistic representation of the physical system. The modelling of the power plant efficiency will incorporate its dependence on the discharge, as well. The power plant head modelling will be made more realistic with the inclusion of the tail water dependence on the discharge. The above modelling improvements will be introduced by a two dimensional energy rate function (function of storage and release). The new energy rate function will be derived from practical power plant measurements by regression analysis. The evaluation of the enhanced model will be done by comparing its performance to the EMMA model using the Manitoba Hydro system data.

The EMSLP model implementation requires a thorough understanding of the details related to input, solution algorithm guidance, etc. The model use can be made more user friendly by creating a support environment which would provide guidelines for the program execution. The capability of a knowledge based system to serve as a support environment for the EMSLP use will be assessed during the future research.

Appendix A: Notation

B_T	-	benefit from saving the water for future production (\$ (2.45 million m^3 or KCFS-day)$^{-1}$)
DERF	-	is the first derivative of ERF over ST
DPS_t	-	number of days in the t-th time step
e	-	efficiency of the hydro power plant
E, E_t	-	energy, energy in t-th time step (kW h)
$EB_{s,t}$	-	export energy benefit in s during t (\$ (GW h)$^{-1}$)
$EE_{s,t}$	-	interruptible export en. in s during t (GW h)
EEF	-	export efficiency
$EML_{s,t}$	-	maximum export load in s during t (MW)
ERF	-	energy rate function used in EMSLP (GW h (m^3 s^{-1} days)$^{-1}$)
γ	-	specific weight of water (kN m^{-3})
H	-	head of the hydro power plant (m)
$HC_{s,t}$	-	hydro energy production cost in s during t (\$ (GW h)$^{-1}$)
E_t	-	produced hydro en. in s during t (GW h)
$I_{s,t}$	-	reservoir inflow during t (m^3 s^{-1} days)
$IC_{s,t}$	-	import energy cost in s during t (\$ (GW h)$^{-1}$)
IEF	-	import efficiency
$IE_{s,t}$	-	interruptible import en. in s during t (GW h)

$L_{s,t}$ - system demand in s during t (MW)

Q - discharge (m^3 s^{-1})

PC - production coefficients used in EMMA (GW h (m^3 s^{-1} days)$^{-1}$)

R_t - released water through the turbines in T (m^3 s^{-1} days)

$\hat{R}_t$ - estimated release through the turbines in T (m^3 s^{-1} days)

RATIO - export and import tieline capacity ratio

S_t - released water through the spillway in t (m^3 s^{-1} days)

SC_t - cost of spilling water (\$ (2.45 million m^3 or KCFS-day)$^{-1}$)

ST_t - stored volume at the end of t (m^3 s^{-1} days)

ST_T - stored volume at the end of the final time step T ($m^3 s^{-1}$ days)

$S\hat{T}_t$ - estimated storage at the end of t (m^3 s^{-1} days)

$STMAX_t$ - maximum storage in T (m^3 s^{-1} days)

$STMIN_t$ - minimum storage in T (m^3 s^{-1} days)

VARMIN - tolerance limit for storage variation (m^3 s^{-1} days)

VARYMX- maximum allowed storage variation (m^3 s^{-1} days)

$W_{s,t}$ - load duration curve strip width for s during t

Appendix B: Bibliography

Barritt-Flatt, P.E. and A.D. Cormie, "A comprehensive optimization model for hydro-electric reservoir operations", presented to the *3rd. Water Resources Operation and Management Workshop: Computerized Decision Support Systems for Water Managers,* Fort Collins, Col., USA, 1988.

Bechard, D.I., I. Corbu, R. Gagnon, G.A. Nix, L.E. Parker, K. Stewart and M. Trinh (1981), "The Ottawa river regulation modelling system", *Proc. Int. Symp. on Real Time Operation of Hydrosystems*, Waterloo, June 24-26, 1981, vol 1, 179-198, Univ. of Waterloo, Waterloo, ONT, CAN, 1981.

Can, E.K., M.H. Houck, G.H. Toebes, "Optimal Real-Time Reservoir Systems Operation: Innovative Objectives and Implementation Problems", Purdue University Water Resources Research Center, Technical Report No. 150, 1982.

Cormie, A.D. and P.E. Barritt-Flatt, "HERMES--a decision support system for reservoir operations at Manitoba Hydro". Presented to the Operations planning section of Canadian Electrical Association, Vancouver, BC, CAN, March 1987.

Daellenbach, H.G. and E.G. Read, "Survey on optimization for the long-term scheduling of hydro-thermal power systems", presented to the ORSNZ conference, New Zealand, 1976.

Dagli, C.H., and J.F. Miles, "Determining operating policies for a water resource systems", *J. Hydrol.,* **47(34),** 297-306, 1980.

Draper, D.W. and Adamowski, K., "Application of linear programming optimization to a Northern Ontario hydro power system", *Can. J. Civ. Eng.,* **3(1),** 20-31, 1976.

Gustafsson, L., "Optimization of the production in a hydro-thermal power system", Paper presented at the VII World Power Conference, Moskow, April 1968.

Grygier, J.C., "Optimal monthly operation of hydrosystems", Ph.D Thesis Cornell Univ., Ithaca, NY, USA 1983.

Grygier, J.C. and J.R. Stedinger, "Algorithms for optimizing hydropower system operation". *Wat. Resour. Res.* **21(1),** 1-11, 1985.

Hall, W.A. and T.G. Roefs, "Hydropower project output optimization", *J. Power Div., ASCE* **92(PO1),** 67-79, 1966.

Heidari, M., V.T. Chow, P.V. Kokotovic and D.D. Meredith, "Discrete differential dynamic programming approach to water resource systems optimization", *Wat. Resour. Res.* **7(2),** 273-282, 1971.

Land, A.H. and S. Powell, *Fortran Codes for Mathematical Programming: Linear, Quadratic and Discrete*, John Wiley & Sons, N.Y., USA, 1973.

Larson, R.E. and W.G. Keckler, "Applications of dynamic programming to the control of water resource systems", *Automatica* **5(1),** 15-26, 1969.

Little, J.D.C., "The use of storage water in a hydroelectric system", *Oper. Res.* **3**, 187-197, 1955.

Loucks, D.P., and O.T. Sigvaldason, "Multiple-reservoir operation in North America", in *The Operation of Multiple Reservoir Systems*, edited by Z. Kaczmarek and J. Kindler, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1982.

Loucks, D.P., J.R. Stedinger and D.A. Haith, *Water Resource Systems Planning and Analysis*, Prentice Hall, Englewood Cliffs, NY, USA, 1981.

Manitoba Hydro, *EMMA-Energy Management and Maintenance Analysis guide.* Report CTP707-199, 1986.

Miller, R.H., and R.P. Thompson, "Long-range scheduling of power production", part I, IEEE PAS CP606, presented at the 1971 Summer Power Meeting.

Miller, R.H., and R.P. Thompson, "Long-range scheduling of power production", part II, IEEE PAS CP606, presented at the 1972 Winter Power Meeting.

Nopmongcol, P., and A.J. Askew, "Multi-level incremental dynamic programming, *Water Resour. Res.,* **12(6),** 1291-1297, 1976.

Pereira, M.V.F. and L.M.V.G. Pinto, "Application of decomposition techniques to the mid- and short-term scheduling of hydrosystems", *IEEE Trans. on Power and Apparatus Systems,* **PAS-102(11),** 3611-3618, 1983.

Reznicek, K.K. and S.P. Simonovic, "Practical application of Successive Linear Programming for reservoir operations at Manitoba Hydro", Paper prepared for the IAHS Third Scientific Assembly, 10-19 May 1989 in Baltimore, Maryland, USA, 1988a.

Reznicek, K.K. and S.P. Simonovic, "An improved algorithm for hydropower optimization", paper submitted to *Water Resour. Res.,* 1988b.

Rosen, J.B., "The gradient projection method for nonlinear programming, I, Linear constraints", *J. Soc. Indust. Appl. Math.,* **8(1),** 181-217, 1960.

Rosenthal, R.E., "A nonlinear network flow algorithm for maximization of benefits in a hydroelectric power system", *Oper. Res.* **29(4),** 763-786, 1981.

Sigvaldason, O.T., "A simulation model for operating a multipurpose multireservoir system", *Water Resour. Res.,* **12(2),** 263--278, 1976.

Takeuchi, K. and D.H. Moreau, "Optimal control of multi-unit interbasin water resource system", *Wat. Resour. Res.* **10(3),** 407-414, 1974.

Tennessee Valley Authority Water Resource Management Methods Staff, Weekly release scheduling by REDGRAD, Rep. B-27, Tenn. Valley Auth. Water Res. Manage. Methods Staff, Knoxville, 1976.

Turgeon, A., "Optimal operation of multireservoir power systems with stochastic inflows", *Water Resour. Res.,* **16(2),** 275-283, 1980.

Turgeon, A., "Optimum short-term hydro scheduling from the principle of progressive optimality", *Wat. Resour. Res.* **17(3),** 481-486, 1981.

Young, G.K., "Finding reservoir operating rules", *J. Hydraulic Div., ASCE,* **93(HY6), 297-321,** 1967.

Yeh, W. W-G, "Reservoir Management and Operations Models: A State of-the-Art-Review", *Wat. Resour. Res.* **21(12)**, 1797-1818, 1985.

APPENDIX C

Tables of the Case Study Results

Table C-1. EMSLP results for different ending storage values

Case Level Release		Planning Time Horizon 1	2	3	4	5	Objective Function (1000$)	Iterations 1st Level	Total
3600	(m)	91.3	91.5	90.8	88.1	85.4	-126	2	3
	(m^3 s^{-1} day)	218	288	283	422	384			
3700	(m)	91.3	91.5	91.5	88.9	86.3	-126	3	4
	(m^3 s^{-1} day)	218	288	214	414	377			
4000	(m)	91.3	91.5	91.5	88.9	86.3	-125	6	9
	(m^3 s^{-1} day)	218	288	213	414	377			
4200	(m)	91.3	91.5	91.5	91.2	88.7	-124	5	8
	(m^3 s^{-1} day)	218	288	213	208	358			
4500	(m)	91.3	91.5	91.3	91.1	88.7	-121	2	3
	(m^3 s^{-1} day)	218	288	229	195	195			
4600	(m)	91.3	91.5	91.5	91.5	91.0	-119	6	10
	(m^3 s^{-1} day)	218	288	213	176	176			
4700	(m)	91.3	91.5	91.5	91.5	91.5	-117	3	5
	(m^3 s^{-1} day)	218	288	214	178	129			
4800	(m)	91.3	91.5	91.5	91.5	91.5	-115	3	4
	(m^3 s^{-1} day)	218	288	214	178	129			

Table C-2. EMMA results for different ending storage values

Case Level Release	Planning Time Horizon 1	2	3	4	5	Objective Function (1000$)	Iterations 1st Level	Total
4200 (m)	89.3	90.0	88.6	88.1	85.4	-133	2	2
(m^3 s^{-1} day)	345	299	343	222	385			
4500 (m)	91.4	91.5	91.5	88.9	90.3	-125	4	4
(m^3 s^{-1} day)	209	298	213	415	0			
4600 (m)	89.9	90.0	91.5	91.5	89.2	-123	3	3
(m^3 s^{-1} day)	346	298	75	176	349			
4700 (m)	89.9	90.0	91.5	91.5	89.2	-122	3	3
(m^3 s^{-1} day)	346	298	75	176	349			
4800 (m)	89.9	90.0	91.5	91.5	91.5	-120	3	3
(m^3 s^{-1} day)	346	298	75	176	132			

Table C-3. EMSLP results for different system loads

Case Level Release	Planning Time Horizon 1	2	3	4	5	Objective Function (1000$)	Iterations 1st Level	Total
0.3 (m)	91.5	90.3	91.5	91.5	91.5	101	2	2
(m^3 s^{-1} day)	199	419	101	180	128			
0.5 (m)	91.5	91.0	91.5	91.2	90.8	42	1	1
(m^3 s^{-1} day)	199	353	168	202	174			
0.8 (m)	90.7	91.5	90.9	89.2	88.5	-54	3	5
(m^3 s^{-1} day)	274	232	269	332	297			
1.0 (m)	91.3	91.5	91.5	90.1	86.3	-126	5	9
(m^3 s^{-1} day)	218	288	214	414	377			
1.5 (m)	91.5	90.1	91.5	90.1	85.4	-314	6	11
(m^3 s^{-1} day)	199	438	83	311	565			
2.0 (m)	91.5	89.2	91.5	91.5	85.4	-504	6	10
(m^3 s^{-1} day)	199	518	2	176	699			

Table C-4. EMMA results for different system load

Case Level Release	Planning Time Horizon 1	2	3	4	5	Objective Function (1000$)	Iterations 1st Level	Total
0.3 (m)	91.5	90.3	91.5	91.5	91.5	100	3	3
(m^3 s^{-1} day)	199	418	102	176	132			
0.5 (m)	91.5	91.0	91.5	91.2	90.7	42	3	3
(m^3 s^{-1} day)	199	353	168	203	176			
0.8 (m)	90.7	91.5	90.9	89.1	87.3	-57	3	3
(m^3 s^{-1} day)	275	233	272	339	304			
1.0 (m)	89.9	90.0	88.6	88.1	85.4	-133	2	2
(m^3 s^{-1} day)	345	299	343	222	385			
1.5 (m)	87.9	86.1	88.4	85.4	85.4	-331	5	5
(m^3 s^{-1} day)	530	475	0	457	132			
2.0 (m)	89.0	85.4	87.7	85.4	85.4	-520	5	5
(m^3 s^{-1} day)	433	641	0	389	132			

Table C-5. EMSLP results for varying release limits

Case Level Release	Planning Time Horizon 1	2	3	4	5	Objective Function (1000$)	Iterations 1st Level	Total
5.67 (m)	91.5	91.5	91.5	91.5	91.0	-170	1	1
P(m^3 s^{-1} day)	170	176	170	176	176			
S(m^3 s^{-1} day)	28	132	43	0	0			
8.57 (m)	91.0	91.0	91.5	90.1	88.7	-131	3	5
(m^3 s^{-1} day)	243	264	255	264	264			
11.35 (m)	91.3	91.5	91.5	89.6	87.2	-127	4	7
(m^3 s^{-1} day)	218	288	217	352	352			
14.19 (m)	91.3	91.5	91.5	88.9	86.3	-126	5	9
(m^3 s^{-1} day)	^{2}18	288	214	414	377			
19.86 (m)	91.3	91.5	91.5	88.9	86.3	-126	5	9
(m^3 s^{-1} day)	346	298	352	440	158			

Table C-6. EMMA results on varying release limits

Case Level Release	Planning Time Horizon 1	2	3	4	5	Objective Function (1000$)	Iterations 1st Level	Total
5.67 (m)	89.6	91.0	91.5	91.5	91.0	-171	3	3
P(m^3 s^{-1} day)	170	176	170	176	176			
S(m^3 s^{-1} day)	203	0	0	0	0			
8.57 (m)	90.9	91.4	90.9	90.0	88.5	-132	3	3
(m^3 s^{-1} day)	255	264	255	264	264			
11.35 (m)	90.0	90.0	88.7	86.8	85.4	-132	3	3
(m^3 s^{-1} day)	340	298	340	352	263			
14.19 (m)	89.9	90.0	88.5	85.7	85.4	-134	4	4
(m^3 s^{-1} day)	346	298	352	440	158			
19.86 (m)	89.9	90.0	88.6	88.1	85.4	-133	2	2
(m^3 s^{-1} day)	345	299	343	222	385			

APPENDIX D.

The Fortran Program of EMSLP

Applied to the Case Study

D.1 PROGRAM STRUCTURE

The EMSLP algorithm is programmed in FORTRAN IV language. The program is composed of a main routine, six subroutines and the Land and Powell routines for LP solving. The program structure is identical to Grygier's SLP model. However, the routines are altered to model the interconnected hydro utility operation problem. The CHGB and SOLVER routines are abridged from Grygier (1983).

The MAIN routine controls the program execution: invokes routines to set and solve the initial problem, evaluates the solution, decides about the iteration procedure, terminates the run when the stopping criteria is satisfied. It calls the INPUT, SETA and SETABC subroutines to set or alter the LP matrices, CHGB and SOLVER to solve the LP problem, IPRINT and OUTPUT to give reports of the program execution.

The following subroutines are incorporated in the EMSLP program:

INPUT - reads in the input data file, prints the read data to the output file and initializes release bounds and right hand side values of some of the constraints; called by MAIN;

SETA - sets the coefficients of the left hand side of constraints and calculates the cost coefficients of the objective function; calls ERF function; called by MAIN;

SETABC - calculates the coefficients of the hydro production constraint (both right and left hand side), the right hand side of the storage constraints and calls the problem solving subroutines; calls ERF function and SOLVER; called by MAIN;

OUTPUT - calculates the objective function value and prints a report about the iteration in the output file; called by MAIN;

SOLVER - a driver for the Land and Powell (1973) routines which solve the LP; called by MAIN, SETABC;

CHGB - resolves the LP with the modified right hand side; called by MAIN;

LP SUBROUTINES - written by Land and Powell (1973) which solve the formulated LP problem (see Appendix C).

The program has about two thousand lines. One thousand lines are the MAIN routine and the six subroutines, while the Land and Powell routines represent the other thousand program lines in the total. The executable version takes 274 kB of memory on the hard disk of a personal computer. The execution time varies depending on the number of iterations, but the typical values are one to two minutes on an IBM/XT personal computer with a mathematical coprocessor. The FORTRAN source code is attached in Appendix C. A sample input and output file are presented in Appendix D.

D.2 VARIABLES

The following section describes the variables existing in the EMSLP program.

The variables used in the program other than in the Land and Powell subroutines are :

CEE(S,T) - benefit from the exported energy ($/GWh)

CEI(S,T) - cost of the energy import ($/GWh)

CJ(S) - the coefficients of the energy rate function

DPS(T) - days per time step (days)

EE(S,T) - interruptible export energy (GWh)

EEL(T) - export energy loss (%/100)

EEM(S,T) - maximum export energy capacity (MW)

EI(S,T) - interruptible import energy (GWh)

EIL(S,T) - import energy loss (%/100)

EIM(S,T) - import energy cost ($/GWh)

ENLO(S,T) - power load of the system (MW)

ENWI(S,T) - load duration curve strip width (%/100)

ESTFL(T) - estimated release (KCFS*days or $28.37 m^3/s$*days)

FLOBO(T) - discharge limit through the turbines (KCFS or $28.37 m^3/s$)

FLOW(T) - reservoir inflow (KCFS or $28.37 m^3/s$)

GFP(T) - release through the turbines(Generation Flow-Power) (KCFS*days or $28.37 m^3/s$*days)

HE(S,T) - the produced hydro energy (GWH)

IMONTH(T) - time step notation - string

IPEEK - a variable to control the printing of output reports

ITER - the iteration counter

ITERMX -the maximum allowed iteration number

LASTM - the last time step of the planning horizon

MSTART - the first time step of the planning horizon

NMONTH - number of time steps of the planning horizon

NSTRIP - number of load duration curve discretization strips

OB(ITER) - the value of the objective in the ITER-th iteration

OBJ1 - the accepted highest objective till the last iteration ($)

OBJECT - the last objective function value calculated ($)

OCO(T) - operation cost of the hydro power plant ($/GWh)

PROCHA - the allowed change of the release policy from one iteration to the other (%/100)

S - index of the load duration curve discretization strip

SCALE - a multiplication factor of the hydro production constraint to bring to scale the coefficients

SINTER - the bank interest rate, important for bringing all benefits and costs to the present value

SPICO(T) - the penalty for spilling water from the reservoir (\$/KCFS*days or \$/(28.37m^3/s*days))

STASTO -the slope of the storage-stage relationship (KCFS*days/ft or (28.37m^3/s*days)/0.305 m)

STMAX(T) - the maximum allowed storage in the reservoir (KCFS*days)

STMIN(T) - the minimum allowed storage in the reservoir (KCFS*days)

STO(T) - accepted reservoir storage (KCFS*days)

STOCA(T) - calculated reservoir storage (KCFS*days)

STOIN - the initial storage in the reservoir (KCFS*days)

STOVA(T) - the value of the stored water (\$/KCFS*days)

T - index of the time step

VARFAC - the factor to decrease the allowed storage variation (%/100)

VARMIN - the lower boundary of the storage variation (KCFS*days)

VARYMX - the storage variation (KCFS*days)

The Variables needed in the routines of Land and Powell to Solve the LP problem (for detailed description of the meaning of the variables see Land and Powell (1973)):

AA(LOOK) - the array of the non-zero elements of the A coefficient matrix

B(I) - the array of the right hand sides of the constraints

BIG - a large number treated as infinity

BOUND(J) - the array of upper bounds for each variable

C(J) - the elements of the linear function to be maximized

DRIVER - an indicator for new variable introduction to the basis

G(I) - the array of changes to SLACK(I) to be made at each basis change

GR(K) - the array of changes to the current basic variables,XR(K), to be made at each iteration

INBASE(J) - the array to indicate whether the j-th variable is basic or not

INREV - an indicator for the CHSLCK variables

INV(K,L) - the array of the reduced inverse matrix

IR - reinversion counter of the inverse matrix during the LP solution

IRMAX - the maximum reinversion number allowed

IROW(I) - the array of elements which signify the starting points of rows of A in AA

ISBIG - the maximum size of the inverse encountered during the calculation

ISDONE - end indicator to avoid stop anywhere except the main routine

ISEFF(I) - an array of elements indicating whether the i-th constraint is effective and represented in the inverse

ISTATE - condition indicator

ITR - iteration number counter during the LP solution

ITRMAX - the maximum number of simplex iterations allowed

JCOL(LOOK) - the array of column labels of the elements of A in AA

M - the number of original constraints in the problem

MARKI - identifies the constraint which is represented by a slack variable explicitly in the basis

MARKK - identifies the row of the inverse containing the slack variable indicated by MARKI

MAXA - the maximum number of elements that can be stored in the AA array

MAXM - the maximum number of the constraints allowed

MAXN - the maximum number of X variables

MNOW - the total number of constraints in the system

MORE - problem number indicator

MOREPR - printing control variable

MXSIZE - the maximum size of the inverse matrix

N - the number of the variables in the problem

NEGINV - the row of the inverse associated with an infeasible variable

NEGROW - the most infeasible row of the A matrix

NEWX - the next variable to be introduced to the basis

NEWY - the row which limits the basis change

NUMSLK - the number of slack variables that are explicitly present in the basis

OBJ - the objective function value

PIV(J) - the array of the pivotal row

R - the limit of the value of the entering variable

S(I) - the array of signs of constraints (1 for <=; 0 for =; -1 for >=)

SLACK(I) - the array of the slack variables

SIZE - the size of the inverse matrix

SIZE1 - the size of the inverse matrix plus one

SMALL - a very small value

TOL(JK) - an array of tolerances that are used in the LP subroutines

X(J) - the values of the variables

XBASIS(K) - row labels of the inverse matrix, containing the numbers of the currently basic variables

XR(K) - the values of the variables listed in XBASIS(K)

XKPOS - entering variable indicator

Y(I) - the values of the variables of the dual problem

YAC(J) - the updated function row of the LP calculation

YAMINC - the element in the updated function row of the entering variable

YBASIS(L) - the array of the column labels of the inverse matrix containing the numbers of the currently effective constraints

YR(L) - the values of the dual variables of the constraints listed in YBASIS(L)

D.3 INPUT DATA

The program needs a variety of information as input. The data have to be provided in the following order in the EMUL.DAT file.

- values of control variables of the Land & Powell routines (ITRMAX, IRMAX, TOL(I));

- planning horizon description (NMONTH, MSTART, DPS(T), IMONTH(T));

- the iterative process control data (ITERMX, VARYMX, VARFAC, VARMIN, PROCHA);

- the objective function cost coefficients (STOVA(T), SPICO(T), OCO(T), CEE(S,T), CEI(S,T));

- the system demand data (NSTRIP, ENLO(S,T), ENWI(S,T), EEM(S,T), EIM(S,T), EEL(S,T) ,EIL(S,T));

- the reservoir and power plant data (STOIN, STASTO, STMIN(T), STMAX(T), FLOBO(T), CJ(N));

- the initial release estimate (ESTFL(T));

- the deterministic reservoir inflow forecast (FLOW(T)); and

- miscellaneous (SCALE,SINTER).

There are 8 tolerance values to be input (Land and Powell (1973)). Care has to be taken that the number of data for each array complies with the specified problem size. In the case of the one dimensional array it is the number of time steps specified in NMONTH. In the case of two dimensional array it is the product of the strip number (given in NSTRIP) and time step number (NMONTH). The order is : specify all the data for one strip and then for the next strip.

D.4 PROGRAM EXECUTION

In this section the program run is presented in details. The execution starts with calling the INPUT subroutine from MAIN. The INPUT subroutine reads in all the pertinent data from the file EMUL.DAT. It also initializes some variables needed for the LP routines of Land & Powell, sets bounds on the reservoir release variables (the LP takes care about them implicitly, without requiring explicitly written constraints) and sets the right hand sides of the constraints (except for the hydro production constraint).

Further the MAIN routine calls the SETA subroutine. The matrix of the coefficients of the left hand sides of the constraints is set according to the requirements of the Land & Powell routines (in the one dimensional AA array). The only coefficients which are not calculated and assigned are the coefficients of the hydro production constraints, which are assigned later. This subroutine also sets the coefficients of the decision variables in the objective function.

After executing SETA, the MAIN routine calculates the estimates for storage levels. This is done using the initial storage level, release estimates and the deterministic inflow forecast balanced through the flow continuity equation. If the so calculated estimated storage would exceed the set boundaries the estimated release is altered to avoid the problem, and the storage is either at its lower or upper bound.

At this point the program enters the iteration loop of solving the optimization problem. The first iteration is specific since the coefficients of the hydro production constraints have to be calculated and assigned based on the estimated releases and storages. This is done in the SETABC subroutine. After the LP is completely set up, the SETABC activates the Land and Powell routines which solve the LP problem. From this point on, the program executes the procedure in the MAIN routine which is the same for all of the subsequent iterations.

The program checks whether the LP solution is feasible and optimal by looking to the ISTATE value. If the ISTATE is different than one, the program terminates the run reporting an infeasible solution. If it is feasible the OUTPUT subroutine is called. After printing the heading of the report to the output file, EMIZ.DAT, the routine enters a time loop. The computation in the loop is repeated for every time step. Besides the calculation of the monthly benefit, BEMO, (by summing the values of the decision variables of that particular month multiplied by the appropriate cost coefficients), the routine also calculates the release and storage estimates for the next iteration. This is performed by calculating the change in the release policy, PROMENA, between the accepted, ESTFL(MON) and the newly calculated GFP(MON) release. The value of PROMENA, actually its absolute value, CHANGE, is normalized by the ESTFL(MON) to give the value of VALTO. VALTO is compared to the allowed policy change fraction specified in PROCHA. If the VALTO value is greater than the PROCHA the GFP(MON) value is recalculated by adding/subtracting the PROCHA multiple of the EMSLP(MON) to the EMSLP(MON) value. The adding or subtracting depends on the original GFP(MON) value, i.e., on the sign of PROMENA: if it is less than zero subtraction takes place, if not addition. In this way the direction of the change is the same as indicated by the original GFP(MON) value.

If the value of VALTO is less than PROCHA the original value of GFP(MON) is unchanged. After updating the release estimates the estimates for the storage STOCA(MON) are calculated using the flow continuity equation. Checking is done to ensure that the obtained STOCA(MON) is within the specified bounds STMIN(MON) and STMAX(MON). If the bounds are violated, the value of GFP(MON) is recalculated to have a feasible STOCA(MON) value (either STMAX(MON) or STMIN(MON) depending whether the original was too high or too low).

It is important to note that the values in the GFP and STOCA arrays are just the candidates for the estimates of the next iteration. They are not accepted at this point, since the evaluation of the solution, based on the comparison of the accepted and calculated objective function, is yet to come.

After the calculation of monthly benefits, i.e., obtaining the objective function value OBJECT is printed to the output file. The control is returned to the MAIN routine.

The NODROP variable contains the information whether the previous iteration has brought improvement to the objective or not. The TRUE value corresponds to the improved objective. Before the first iteration a TRUE value is assigned to calculate the initial coefficients of the hydro production constraints in the SETABC routine. Depending on the value of NODROP the program branches into two directions.

If NODROP is TRUE the program compares the newly obtained objective function value with the so far accepted objective function values of the previous iterations. This is to avoid cycling in an infinite loop of iterations without achieving a stable solution. If a loop is identified the program stops after reporting the cause. If there is no loop the program joins to the "NODROP is FALSE" branch.

The difference between the newly obtained solution and the accepted one is compared to the tolerance limit. If the absolute value of the highest monthly storage difference (AD) is less than the set minimum storage variability (VARMIN) the iterative solution procedure terminates. A stable solution is identified. If not, the program evaluates the obtained solution.

The newly calculated OBJECT value is compared to the last accepted objective function value stored in OBJ1. If the new one is higher, the NODROP gets the TRUE value and the solution is accepted. The value of the OBJ1 is updated, and the storage variability is reset to its original, maximum value (VARYMX=VARMAX). The program goes into a new iteration at the first level. The SETABC recalculates the coefficients of the hydro production constraints, resets the bounds on the storage variables and finally solves the new LP by calling the SOLVER routine. The control is returned to the MAIN routine.

In the case that there is no improvement in the objective value and the NODROP gets a FALSE value after the comparison, the program enters the second level of iteration. The task is to search the interior of the original feasible region of the LP problem in order to possibly improve the objective function value. The storage variability (VARYMX) is decreased by multiplying it with the VARFAC factor. The new value of VARYMX is compared to the tolerance limit (VARMIN).

If it is greater or equal than VARMIN the bounds on the storages are recalculated according to the new VARYMX value and the LP is resolved. It is important to note that the coefficients of the hydro production constraints are not changed, only the feasible region is decreased by changing the range of the possible storage values.The execution continues from the beginning of the loop. Since the NODROP has a FALSE value the program goes directly to the OUTPUT routine without invoking the SETABC. From this point on the execution is the same as described above for the first iteration.

If the value of VARYMX is less than VARMIN the initial worse solution (where the objective started to decrease) is accepted, and the program returns to the first level of iteration. The value of VARYMX is reset to VARMAX, the storage bounds are recalculated and the LP is resolved. A TRUE value is assigned to the NODROP in order to accept the solution. The execution continues from the start of the loop as if an improvement in the objective has occurred.

To summarize, the program accepts solutions which improve the objective and solutions where the objective drops after it is determined that the interior of the original feasible region does not contain a better solution. The iteration continues until the difference between the storage trajectories of the calculated and the accepted solution is less than the defined accuracy.

D.5 OUTPUT REPORT

The output file named EMIZ.DAT contains the input data and reports on the iterative solution procedure. The input data are printed in the output report to be able to correct the potential errors and to be able to relate the solution to the input data (Input data sample is given in Appendix F).

Every solution in the iterative process is documented by giving the values of following variables for each time step: inflow to the reservoir, turbine release, spilled release, total release, on and off peak export and import energy, on and off peak produced hydro energy and the monthly benefit value. Besides the report gives the number of the solution, the number of simplex iterations done to obtain the solution the current value of the storage variation variable VARYMX and the objective function value (A sample of output file is enclosed in Appendix F).

The report can contain messages about accepting the worse solution at the end of the search at the second iteration level or about identifying an infinite loop of solutions. At the end of the output file the number of iterations at the first iteration level is also reported.

APPENDIX E

SOURCE CODE

```
C
C THE MAIN ROUTINE
C
      IMPLICIT REAL*8 (A-H,O-Z)
      LOGICAL NODROP
      REAL*8 INV,KMIN
      INTEGER SIZE,SIZE1,XBASIS,YBASIS
      COMMON/IO/ IOIN,IOOUT
      COMMON/LINPCO/
     1BOUND(156),C(156),INBASE(156),PIV(156),X(156),YAC(156),
     2B(156),G(156),GR(156),ISEFF(156),S(156),SLACK(156),Y(156),
     3INV(130,130),XBASIS(130),XR(130),YBASIS(130),YR(130),
     4      TOL(8),BIG,DRIVER,INREV,IR,
     6      IRMAX, ISBIG, ISDONE, ISTATE, ITR, ITRMAX, M,
     7      MARKI, MARKK, MAXM, MAXN, MNOW, MORE, MOREPR,
     8      MXSIZE, N, NEGINV, NEGROW, NEWX, NUMSLK, OBJ, R,
     9      SIZE, SIZE1, SMALL, XKPOS, YAMINC, NEWY, ISBND
      COMMON/DATA/ MSTART,LASTM,NSTRIP,SCALE,STOIN,SINTER,
     1STOVA(12),STASTO,SPICO(12),OCO(12),ENLO(2,12),ENWI(2,12),
     2DPS(12),FLOW(12),FLOBO(12),ESTFL(12),STMAX(12),STMIN(12),
     3HE(2,12),CJ(2),EE(2,12),EI(2,12),EEM(2,12),EIM(2,12),
     4EEL(12),EIL(12),CEE(2,12),CEI(2,12)
      COMMON/MAIN/ ITERMX,IPEEK,VARYMX,BETTER,VARFAC,VARMIN,VARFC1
      COMMON/RESULT/ STO(12),GFP(12),OBJECT,OBJ1,ITER,STOCA(12)
      COMMON /AREF/ AA(600),JCOL(600),IROW(157),MAXA
      COMMON /CONTROL/ KFLAG
      COMMON /CHANGES/ PROCHA
      DIMENSION OB(250)
      IC=1
C
      CALL INPUT
      CALL SETA
C
C A LOOP TO :
C   - CALC. MONTHLY INFLOWS/OUTFLOWS IN KCFS*DAYS
C   - CALCULATE THE INITIAL  STORAGE ESTIMATES
C
      BEG=STOIN
      DO 50 MN=MSTART,LASTM
      MON=MN
      IF(MON.GT.12) MON=MON-12
      FLOW(MON)=FLOW(MON)*DPS(MON)
      ESTFL(MON)=ESTFL(MON)*DPS(MON)
      STO(MON)=BEG+FLOW(MON)-ESTFL(MON)
      IF(STO(MON).LE.STMAX(MON)) GO TO 20
      STO(MON)=STMAX(MON)
      ESTFL(MON)=BEG-STO(MON)+FLOW(MON)
      IF(ESTFL(MON).GT.FLOBO(MON)) ESTFL(MON)=FLOBO(MON)
   20 IF(STO(MON).GE.STMIN(MON)) GO TO 50
      STO(MON)=STMIN(MON)
      ESTFL(MON)=BEG-STO(MON)+FLOW(MON)
      IF(ESTFL(MON).GT.FLOBO(MON)) ESTFL(MON)=FLOBO(MON)
   50 BEG=STO(MON)
C
```

```
C  INITIALIZE VARIABLES FOR THE FIRST ITERATION
C
      VARMAX=VARYMX
      OBJ1=-BIG
      NODROP=.TRUE.
C
C  THE LOOP OF THE ITERATIVE SOLUTION PROCEDURE
C
      DO 100 ITER=1,ITERMX
      MOREPR=IPEEK-1
      IF(MOREPR.LT.0.OR.MOREPR.GT.3) MOREPR=0
      KFLAG=0
      IF(NODROP) CALL SETABC
      IF(ISTATE.NE.1) GO TO 1000
      CALL OUTPUT
      IF(.NOT.NODROP) GO TO 500
C
C THIS PART IS TO AVOID CYRCLING IN AN INFINITE LOOP OF SOLUTIONS
C
      COM=BIG
      DO 451 IK=1,IC
      VAL=DABS(OBJECT-OB(IK))
 451  IF(VAL.LT.COM) COM=VAL
      IF(COM.GE.0.01)GO TO 550
      WRITE(6,*) 'LOOP FOUND,IT STOPS'
      GO TO 1100
 550  IC=IC+1
      OB(IC)=OBJECT
C
 500  CONTINUE
C
C CONVERGENCE CHECK : THE STOPPING CRITERIA
C
      AD=0.0
      J=9
      DO 55 MN=MSTART,LASTM
      MON=MN
      IF(MON.GT.12) MON=MON-12
      DIF=DABS(STO(MON)-X(J))
      IF(DIF.GT.AD) AD=DIF
  55  J=J+9
      IF(AD.LE.VARMIN) WRITE(6,*) 'CONVERGED VOLUME'
      IF(AD.LE.VARMIN) GO TO 1100
C
C  CONVERGENCE HAS NOT BEEN DETERMINED SO PREPARE
C  FOR THE NEXT ITERATION: DETERMINE WHETHER THERE WAS AN
C  IMPROVEMENT IN THE OBJECTIVE FUNCTION (NODROP=.TRUE.)
C
      NODROP=OBJECT.GT.OBJ1
C
C IF NODROP IS .TRUE. THE ITERATION CONTINUES AT THE
C FIRST LEVEL ( THE OBTAINED SOLUTION IS
C  ACCEPTED AND USED AS THE ESTIMATE FOR THE NEXT ONE)
C
```

```
      IF(.NOT.NODROP) GO TO 79
      VARYMX=VARMAX
      OBJ1=OBJECT
      GO TO 100
C
C THE OBJECTIVE FUNCTION VALUE GOT WORSE SO
C  ITERATE AT THE SECOND LEVEL (TRY TO IMPROVE THE OBJECTIVE
C  BY EXPLORING THE INTERIOR OF THE FEASIBLE REGION BY
C  LIMITING THE STORAGE VARIABILITY VARYMX
C
  79    VARYMX=VARYMX*VARFAC
C
C
C  CHECK WHETHER THE ITERATION AT SECOND LEVEL IS FINISHED
C
      IF(VARYMX.LE.VARMIN) GO TO 99
C
C  RECALCULATE THE RIGHT HAND SIDES OF THE STORAGE
C  LIMITING CONSTRAINTS AND RESOLVE THE LP USING THE UNCHANGED
C  ESTIMATE FOR STORAGES AND RELEASES (I.E. THE SAME
C  COEFFICIENTS IN THE HYDRO PRODUCTION CONTRAINTS)
C
      I=1
      DO 60 MN=MSTART,LASTM
      MON=MN
      IF(MON.GT.12) MON=MON-12
      PIV(I)=0.
      PIV(I+1)=0.
      PIV(I+2)=0.
      PIV(I+3)=0.
      PIV(I+4)=0.
      PIV(I+5)=0.
      PIV(I+6)=DMAX1(STO(MON)-VARYMX,STMIN(MON))-B(I+6)
      B(I+6)=B(I+6)+PIV(I+6)
      PIV(I+7)=DMIN1(STO(MON)+VARYMX,STMAX(MON))-B(I+7)
      B(I+7)=B(I+7)+PIV(I+7)
      PIV(I+8)=B(I+8)
   60 I=I+9
      NEWY=0
      CALL CHGB
      IF(ISTATE.NE.1) CALL SOLVER
      IF(ISTATE.EQ.1) GO TO 97
C
C AN OTHER TRY TO SOLVE THE LP
C
      WRITE(6,*) 'A TRY FROM SCRATCH'
      KFLAG=1
      CALL SETABC
      IF(ISTATE.NE.1) GO TO 1000
  97  IF(IPEEK.EQ.1) WRITE(6,902) (B(I+6),B(I+7),I=1,M,8)
      IF(IPEEK.EQ.14) CALL IPRINT
      IF(IPEEK.GE.10) IPEEK=IPEEK-10
      GO TO 100
C
C  THE ITERATIONS AT THE SECOND LEVEL DID NOT FIND
```

```
C  A BETTER OBJECTIVE THAN THE INITIAL DECREASED ONE,
C  SO ACCEPT THE FIRST DETERMINED SOLUTION AS AN
C  ESTIMATE FOR THE NEXT ONE
C
  99  NODROP=.TRUE.
      VARYMX=VARMAX
      WRITE(6,*) 'THE INITIAL DECREASE IS ACCEPTED'
      I=1
      DO 62 MN=MSTART,LASTM
      MON=MN
      IF(MON.GT.12) MON=MON-12
      PIV(I)=0.
      PIV(I+1)=0.
      PIV(I+2)=0.
      PIV(I+3)=0.
      PIV(I+4)=0.
      PIV(I+5)=0.
      PIV(I+6)=DMAX1(STO(MON)-VARYMX,STMIN(MON))-B(I+6)
      B(I+6)=B(I+6)+PIV(I+6)
      PIV(I+7)=DMIN1(STO(MON)+VARYMX,STMAX(MON))-B(I+7)
      B(I+7)=B(I+7)+PIV(I+7)
      PIV(I+8)=B(I+8)
  62  I=I+9
      NEWY=0
      CALL CHGB
      IF(ISTATE.NE.1) CALL SOLVER
      IF(ISTATE.EQ.1) GO TO 98
C
C AN OTHER TRY TO SOLVE THE LP
C
      WRITE(6,*) 'A TRY FROM SCRATCH'
      KFLAG=1
      CALL SETABC
      IF(ISTATE.NE.1) GO TO 1000
  98  CALL OUTPUT
      OBJ1=OBJECT
 100  CONTINUE
      GO TO 1100
1000  MOREPR=3
      CALL IPRINT
      WRITE(6,*) 'I AM BACK FROM IPRINT'
 902  FORMAT (8F10.3)
1100  CONTINUE
      IC=IC-1
      WRITE(6,129) IC
 129  FORMAT('  NET ITERATIONS  ',I4)
      END
```

```
      SUBROUTINE INPUT
      IMPLICIT REAL*8 (A-H,O-Z)
      CHARACTER*3 IMONTH
      LOGICAL NODROP
      REAL*8 INV,KMIN
      INTEGER SIZE,SIZE1,XBASIS,YBASIS
      COMMON/IO/ IOIN,IOOUT
      COMMON/LINPCO/
     1BOUND(156),C(156),INBASE(156),PIV(156),X(156),YAC(156),
     2B(156),G(156),GR(156),ISEFF(156),S(156),SLACK(156),Y(156),
     3INV(130,130),XBASIS(130),XR(130),YBASIS(130),YR(130),
     4      TOL(8),BIG,DRIVER,INREV,IR,
     6      IRMAX, ISBIG, ISDONE, ISTATE, ITR, ITRMAX, M,
     7      MARKI, MARKK, MAXM, MAXN, MNOW, MORE, MOREPR,
     8      MXSIZE, N, NEGINV, NEGROW, NEWX, NUMSLK, OBJ, R,
     9      SIZE, SIZE1, SMALL, XKPOS, YAMINC, NEWY, ISBND
      COMMON/DATA/ MSTART,LASTM,NSTRIP,SCALE,STOIN,SINTER,
     1STOVA(12),STASTO,SPICO(12),OCO(12),ENLO(2,12),ENWI(2,12),
     2DPS(12),FLOW(12),FLOBO(12),ESTFL(12),STMAX(12),STMIN(12),
     3HE(2,12),CJ(2),EE(2,12),EI(2,12),EEM(2,12),EIM(2,12),
     4EEL(12),EIL(12),CEE(2,12),CEI(2,12)
      COMMON/MAIN/ ITERMX,IPEEK,VARYMX,BETTER,VARFAC,VARMIN,VARFC1
      COMMON/RESULT/ STO(12),GFP(12),OBJECT,OBJ1,ITER,STOCA(12)
      COMMON /AREF/ AA(600),JCOL(600),IROW(157),MAXA
      COMMON/MONTH/ IMONTH(12)
      COMMON/CHANGES/ PROCHA
   10 FORMAT(10I5)
   20 FORMAT(6F11.2)
   30 FORMAT(6F12.6)
      OPEN(5,FILE='EMUL.DAT')
      OPEN(6,FILE='EMIZ.DAT')
      IOIN=5
      IOOUT=6
C
C  READING IN  THE NEEDED DATA AND WRITING IT TO THE
C  OUTPUT FILE
C
      WRITE(6,*) '   OUTPUT REPORT FROM THE EMSLP PROGRAM'
      WRITE(6,*) '   '
      WRITE(6,*) '   '
      WRITE(6,*) '  THE INPUT DATA  '
      WRITE(6,*) '-------------------'
      WRITE(6,*) '   '
      WRITE(6,*) '*  LAND & POWELL CONTROL DATA'
      WRITE(6,*) '   '
      READ(5,*) ITRMAX,IRMAX
      WRITE(6,10) ITRMAX,IRMAX
      READ(5,*) (TOL(I),I=1,8)
      WRITE(6,30) (TOL(I),I=1,8)
C
      WRITE(6,*) '   '
      WRITE(6,*) '*  PLANNING HORIZON DESCRIPTION'
      WRITE(6,*) '   '
      READ(5,*) NMONTH,MSTART
```

```
      WRITE(6,10) NMONTH,MSTART
      READ(5,*) (DPS(IT),IT=1,NMONTH)
      WRITE(6,20) (DPS(IT),IT=1,NMONTH)
      READ(5,*) (IMONTH(IT),IT=1,NMONTH)
      WRITE(6,*) (IMONTH(IT),IT=1,NMONTH)
C
      WRITE(6,*) '   '
      WRITE(6,*) '*  ITERATIVE PROCESS CONTROL DATA'
      WRITE(6,*) '   '
      READ(5,*) ITERMX,IPEEK
      WRITE(6,*) ITERMX,IPEEK
      READ(5,*) VARYMX,VARFAC,VARMIN,PROCHA
      WRITE(6,30) VARYMX,VARFAC,VARMIN,PROCHA
C
      WRITE(6,*) '   '
      WRITE(6,*) '*  OBJ. FUN COST COEFF.'
      WRITE(6,*) '   '
      READ(5,*) (STOVA(IT),IT=1,NMONTH)
      WRITE(6,20) (STOVA(IT),IT=1,NMONTH)
      READ(5,*) (SPICO(IT),IT=1,NMONTH)
      WRITE(6,30) (SPICO(IT),IT=1,NMONTH)
      READ(5,*) (OCO(IT),IT=1,NMONTH)
      WRITE(6,30) (OCO(IT),IT=1,NMONTH)
      READ(5,*) (CEE(1,IT),IT=1,NMONTH)
      WRITE(6,20) (CEE(1,IT),IT=1,NMONTH)
      READ(5,*) (CEE(2,IT),IT=1,NMONTH)
      WRITE(6,20) (CEE(2,IT),IT=1,NMONTH)
      READ(5,*) (CEI(1,IT),IT=1,NMONTH)
      WRITE(6,20) (CEI(1,IT),IT=1,NMONTH)
      READ(5,*) (CEI(2,IT),IT=1,NMONTH)
      WRITE(6,20) (CEI(2,IT),IT=1,NMONTH)
C
      WRITE(6,*) '   '
      WRITE(6,*) '*  SYSTEM DEMAND DATA'
      WRITE(6,*) '   '
      READ(5,*) NSTRIP
      WRITE(6,10) NSTRIP
      DO 50 NI=1,NSTRIP
      READ(5,*) (ENLO(NI,J),J=1,NMONTH)
   50 WRITE(6,30) (ENLO(NI,J),J=1,NMONTH)
      DO 60 NI=1,NSTRIP
      READ(5,*) (ENWI(NI,J),J=1,NMONTH)
   60 WRITE(6,30) (ENWI(NI,J),J=1,NMONTH)
      READ(5,*) (EEM(1,IT),IT=1,NMONTH)
      WRITE(6,20) (EEM(1,IT),IT=1,NMONTH)
      READ(5,*) (EEM(2,IT),IT=1,NMONTH)
      WRITE(6,20) (EEM(2,IT),IT=1,NMONTH)
      READ(5,*) (EIM(1,IT),IT=1,NMONTH)
      WRITE(6,20) (EIM(1,IT),IT=1,NMONTH)
      READ(5,*) (EIM(2,IT),IT=1,NMONTH)
      WRITE(6,20) (EIM(2,IT),IT=1,NMONTH)
      READ(5,*) (EEL(IT),IT=1,NMONTH)
      WRITE(6,30) (EEL(IT),IT=1,NMONTH)
      READ(5,*) (EIL(IT),IT=1,NMONTH)
```

```
      WRITE(6,30) (EIL(IT),IT=1,NMONTH)
C
      WRITE(6,*) '   '
      WRITE(6,*) '*  RESERVOIR & POWER PLANT DATA'
      WRITE(6,*) '   '
      READ(5,*) STOIN,STASTO
      WRITE(6,30) STOIN,STASTO
      READ(5,*) (STMIN(IT),IT=1,NMONTH)
      WRITE(6,20) (STMIN(IT),IT=1,NMONTH)
      READ(5,*) (STMAX(IT),IT=1,NMONTH)
      WRITE(6,20) (STMAX(IT),IT=1,NMONTH)
      READ(5,*) (FLOBO(IT),IT=1,NMONTH)
      WRITE(6,30) (FLOBO(IT),IT=1,NMONTH)
      READ(5,*) (CJ(IT),IT=1,2)
      WRITE(6,30) (CJ(IT),IT=1,2)
C
      WRITE(6,*) '   '
      WRITE(6,*) '*  RELEASE ESTIMATES'
      WRITE(6,*) '   '
      READ(5,*) (ESTFL(IT),IT=1,NMONTH)
      WRITE(6,30) (ESTFL(IT),IT=1,NMONTH)
C
      WRITE(6,*) '   '
      WRITE(6,*) '*  FORECASTED INFLOW'
      WRITE(6,*) '   '
      READ(5,*) (FLOW(IT),IT=1,NMONTH)
      WRITE(6,30) (FLOW(IT),IT=1,NMONTH)
C
      WRITE(6,*) '   '
      WRITE(6,*) '*  SCALE FOR LP & DISCOUNT RATE'
      WRITE(6,*) '   '
      READ(5,*) SCALE,SINTER
      WRITE(6,30) SCALE,SINTER
      WRITE(6,*) '---------------------------------------------'
      WRITE(6,*) '   '
      WRITE (6,*) '  REPORTS ON THE ITERATIVE SOLUTION PROCEDURE'
      WRITE(6,*) '   '
C
C INITIALIZE VARIABLES FOR LP ROUTINES
C
      BIG=1.E8
      SMALL=1.E-6
      M=9*NMONTH

C
C INITIALIZE SOME BOUNDS ON RELEASES , SOME RHS-S
C
      LASTM=MSTART+NMONTH-1
      J=1
      DO 100 MN=MSTART,LASTM
      I=(MN-MSTART)*9+1
      MON=MN
      IF(MON.GT.12) MON=MON-12
      FLOBO(MON)=FLOBO(MON)*DPS(MON)
```

```
      BOUND(J)=FLOBO(MON)
      BOUND(J+1)=-1.
      BOUND(J+2)=-1.
      BOUND(J+3)=-1.
      BOUND(J+4)=-1.
      BOUND(J+5)=-1.
      BOUND(J+6)=-1.
      BOUND(J+7)=-1.
      BOUND(J+8)=-1.
      B(I+1)=FLOW(MON)*DPS(MON)
      B(I+2)=EEM(1,MON)/(1-EEL(MON))*24./1000.*DPS(MON)*ENWI(1,MON)
      B(I+3)=EEM(2,MON)/(1-EEL(MON))*24./1000.*DPS(MON)*ENWI(2,MON)
      B(I+4)=ENLO(1,MON)*24./1000.*DPS(MON)*ENWI(1,MON)
      B(I+5)=ENLO(2,MON)*24./1000.*DPS(MON)*ENWI(2,MON)
      B(I+8)=0.0
  100 J=J+9
      B(2)=B(2)+STOIN
      RETURN
      END
```

```
      SUBROUTINE SETA
C
C THIS SUBROUTINE SETS UP THE A MATRIX AND THE C VALUES,TOO
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 INV,KMIN
      INTEGER SIZE,SIZE1,XBASIS,YBASIS
      COMMON/IO/ IOIN,IOOUT
      COMMON/LINPCO/
     1BOUND(156),C(156),INBASE(156),PIV(156),X(156),YAC(156),
     2B(156),G(156),GR(156),ISEFF(156),S(156),SLACK(156),Y(156),
     3INV(130,130),XBASIS(130),XR(130),YBASIS(130),YR(130),
     4      TOL(8),BIG,DRIVER,INREV,IR,
     6      IRMAX, ISBIG, ISDONE, ISTATE, ITR, ITRMAX, M,
     7      MARKI, MARKK, MAXM, MAXN, MNOW, MORE, MOREPR,
     8      MXSIZE, N, NEGINV, NEGROW, NEWX, NUMSLK, OBJ, R,
     9      SIZE, SIZE1, SMALL, XKPOS, YAMINC, NEWY, ISBND
      COMMON/DATA/ MSTART,LASTM,NSTRIP,SCALE,STOIN,SINTER,
     1STOVA(12),STASTO,SPICO(12),OCO(12),ENLO(2,12),ENWI(2,12),
     2DPS(12),FLOW(12),FLOBO(12),ESTFL(12),STMAX(12),STMIN(12),
     3HE(2,12),CJ(2),EE(2,12),EI(2,12),EEM(2,12),EIM(2,12),
     4EEL(12),EIL(12),CEE(2,12),CEI(2,12)
      COMMON/MAIN/ ITERMX,IPEEK,VARYMX,BETTER,VARFAC,VARMIN,VARFC1
      COMMON/RESULT/ STO(12),GFP(12),OBJECT,OBJ1,ITER,STOCA(12)
      COMMON /AREF/ AA(600),JCOL(600),IROW(157),MAXA
C
C AA(K)-IS A ONE DIMENSIONAL ARRAY CONTAINING THE
C       VALUES OF THE CONSTRAINT COEFFICIENTS
C JCOL(K)- DENOTES THE COLUMN OF THE K-TH COEFFICIENT
C IROW(I)- DENOTES THE K NUMBER OF THE FIRST COEFFICIENT IN THE I-TH ROW
C I- CONSTRAINT NUMBER
C J- VARIABLE NUMBER
C K- COEFFICIENT NUMBER IN AA ARRAY
C
C SEE LAND & POWELL TO UNDERSTAND MORE
C ABOUT CONSTRAINT MATRIX SPECIFICATION
C
      DO 10 JI=MSTART,LASTM
   10 TOTAL=DPS(JI)+TOTAL
      MAXA=300
      I=1
      J=1
      K=1
      SD=0.0
      DO 100 MN=MSTART,LASTM
      SD=SD+DPS(MON)
      MON=MN
      IF(MON.GT.12) MON=MON-12
      IF(K.GT.19) GO TO 20
      JCOL(1)=1
      JCOL(2)=3
      AA(2)=-2.0*SCALE
      JCOL(3)=4
      AA(3)=-2.0*SCALE
```

```
JCOL(4)=9
IROW(1)=1
S(1)=0.0
JCOL(5)=1
AA(5)=1.0
JCOL(6)=2
AA(6)=1.0
JCOL(7)=9
AA(7)=1.0
S(2)=0.0
IROW(2)=5
JCOL(8)=5
AA(8)=1.0
JCOL(9)=7
AA(9)=EEM(1,MON)/EIM(1,MON)/((1-EEL(MON))*(1-EIL(MON)))
S(3)=1.0
IROW(3)=8
JCOL(10)=6
AA(10)=1.0
JCOL(11)=8
AA(11)=EEM(2,MON)/EIM(2,MON)/((1-EEL(MON))*(1-EIL(MON)))
S(4)=1.0
IROW(4)=10
JCOL(12)=3
AA(12)=1.0
JCOL(13)=5
AA(13)=-1.0
JCOL(14)=7
AA(14)=1.0
S(5)=0.0
IROW(5)=12
JCOL(15)=4
AA(15)=1.0
JCOL(16)=6
AA(16)=-1.0
JCOL(17)=8
AA(17)=1.0
S(6)=0.0
IROW(6)=15
JCOL(18)=9
AA(18)=1.0
S(7)=-1.0
IROW(7)=18
JCOL(19)=9
AA(19)=1.0
S(8)=1.0
IROW(8)=19
JCOL(20)=1
AA(20)=-24./1000.*ERF(CJ,STMAX(MON))*SCALE
JCOL(21)=3
AA(21)=1.0*SCALE
JCOL(22)=4
AA(22)=1.0*SCALE
S(9)=1.0
```

```
      IROW(9)=20
      FACT=1./(1+SINTER)**((TOTAL-SD)/365.)
      C(1)=0.0
      C(2)=SPICO(MON)*FACT
      C(3)=OCO(MON)*FACT
      C(4)=OCO(MON)*FACT
      C(5)=CEE(1,MON)*FACT*(1-EEL(MON))
      C(6)=CEE(2,MON)*FACT*(1-EEL(MON))
      C(7)=CEI(1,MON)*FACT/(1-EIL(MON))
      C(8)=CEI(2,MON)*FACT/(1-EIL(MON))
      C(9)=STOVA(MON)*FACT
      J=J+9
      I=I+9
      K=K+22
      GO TO 100
   20 JCOL(K)=J-1
      JCOL(K+1)=J
      JCOL(K+2)=J+2
      AA(K+2)=-2.0*SCALE
      JCOL(K+3)=J+3
      AA(K+3)=-2.0*SCALE
      JCOL(K+4)=J+8
      S(I)=0.0
      IROW(I)=K
      JCOL(K+5)=J-1
      AA(K+5)=-1.0
      JCOL(K+6)=J
      AA(K+6)=1.0
      JCOL(K+7)=J+1
      AA(K+7)=1.0
      JCOL(K+8)=J+8
      AA(K+8)=1.0
      S(I+1)=0.0
      IROW(I+1)=K+5
      JCOL(K+9)=J+4
      AA(K+9)=1.0
      JCOL(K+10)=J+6
      AA(K+10)=EEM(1,MON)/EIM(1,MON)/((1-EEL(MON))*(1-EIL(MON)))
      S(I+2)=1.0
      IROW(I+2)=K+9
      JCOL(K+11)=J+5
      AA(K+11)=1.0
      JCOL(K+12)=J+7
      AA(K+12)=EEM(2,MON)/EIM(2,MON)/((1-EEL(MON))*(1-EIL(MON)))
      S(I+3)=1.0
      IROW(I+3)=K+11
      JCOL(K+13)=J+2
      AA(K+13)=1.0
      JCOL(K+14)=J+4
      AA(K+14)=-1.0
      JCOL(K+15)=J+6
      AA(K+15)=1.0
      S(I+4)=0.0
      IROW(I+4)=K+13
```

```
      JCOL(K+16)=J+3
      AA(K+16)=1.0
      JCOL(K+17)=J+5
      AA(K+17)=-1.0
      JCOL(K+18)=J+7
      AA(K+18)=1.0
      S(I+5)=0.0
      IROW(I+5)=K+16
      JCOL(K+19)=J+8
      AA(K+19)=1.0
      S(I+6)=-1.0
      IROW(I+6)=K+19
      JCOL(K+20)=J+8
      AA(K+20)=1.0
      S(I+7)=1.0
      IROW(I+7)=K+20
      JCOL(K+21)=J
      AA(K+21)=-24./1000.*ERF(CJ,STMAX(MON))*SCALE
      JCOL(K+22)=J+2
      AA(K+22)=1.0*SCALE
      JCOL(K+23)=J+3
      AA(K+23)=1.0*SCALE
      S(I+8)=1.0
      IROW(I+8)=K+21
      FACT=1/(1+SINTER)**((TOTAL-SD)/365.)
      C(J)=0.0
      C(J+1)=SPICO(MON)*FACT
      C(J+2)=OCO(MON)*FACT
      C(J+3)=OCO(MON)*FACT
      C(J+4)=CEE(1,MON)*FACT*(1-EEL(MON))
      C(J+5)=CEE(2,MON)*FACT*(1-EEL(MON))
      C(J+6)=CEI(1,MON)*FACT/(1-EIL(MON))
      C(J+7)=CEI(2,MON)*FACT/(1-EIL(MON))
      C(J+8)=STOVA(MON)*FACT
      J=J+9
      I=I+9
      K=K+24
  100 CONTINUE
      N=J-1
      IROW(I)=K
      RETURN
      END
C
      REAL*8 FUNCTION ERF(CC,S)
      REAL*8 CC,S
C
C     CC(1)=9.801*0.305**4*1000/1000*EFFICIENCY/STASTO
C     CC(2)=CC(1)*(-TAIL WATER LEVEL)
C
      DIMENSION CC(2)
      ERF=CC(1)*S+CC(2)
      RETURN
      END
```

```
      SUBROUTINE SETABC
C
C  THIS SUBROUTINE CALCULATES THE COEFF.'S OF THE
C  HYDRO PRODUCTION CONSTRAINT, AND ALTERS THE SAME IN THE
C  SUBSEQUENT ITERATIONS
C
      IMPLICIT REAL*8 (A-H,O-Z)
      LOGICAL CHANGB
      REAL*8 INV,KMIN
      INTEGER SIZE,SIZE1,XBASIS,YBASIS
      COMMON/IO/ IOIN,IOOUT
      COMMON/LINPCO/
     1BOUND(156),C(156),INBASE(156),PIV(156),X(156),YAC(156),
     2B(156),G(156),GR(156),ISEFF(156),S(156),SLACK(156),Y(156),
     3INV(130,130),XBASIS(130),XR(130),YBASIS(130),YR(130),
     4      TOL(8),BIG,DRIVER,INREV,IR,
     6      IRMAX, ISBIG, ISDONE, ISTATE, ITR, ITRMAX, M,
     7      MARKI, MARKK, MAXM, MAXN, MNOW, MORE, MOREPR,
     8      MXSIZE, N, NEGINV, NEGROW, NEWX, NUMSLK, OBJ, R,
     9      SIZE, SIZE1, SMALL, XKPOS, YAMINC, NEWY, ISBND
      COMMON/DATA/ MSTART,LASTM,NSTRIP,SCALE,STOIN,SINTER,
     1STOVA(12),STASTO,SPICO(12),OCO(12),ENLO(2,12),ENWI(2,12),
     2DPS(12),FLOW(12),FLOBO(12),ESTFL(12),STMAX(12),STMIN(12),
     3HE(2,12),CJ(2),EE(2,12),EI(2,12),EEM(2,12),EIM(2,12),
     4EEL(12),EIL(12),CEE(2,12),CEI(2,12)
      COMMON/MAIN/ ITERMX,IPEEK,VARYMX,BETTER,VARFAC,VARMIN,VARFC1
      COMMON/RESULT/ STO(12),GFP(12),OBJECT,OBJ1,ITER,STOCA(12)
      COMMON /AREF/ AA(600),JCOL(600),IROW(157),MAXA
      COMMON /CONTROL/ KFLAG
C
      DIMENSION T(84)
C
C THE KFLAG=1 MEANS THAT THE ROUTINE IS CALLED
C ONLY TO RESOLVE THE LP WITHOUT ALTERING IT BEFORE
C
      IF(KFLAG.EQ.1) GO TO 111
      IF(ITER.GT.1.AND.MOREPR.GT.0) CALL IPRINT
      BEG=0.
      I=1
      J=1
      K=1
      OERF=ERF(CJ,STOIN)
C
C  LOOP TO CALCULATE THE COEFFICIENTS FOR EVERY TIME STEP
C
      DO 200 MN=MSTART,LASTM
      MON=MN
      IF(MON.GT.12) MON=MON-12
C
C  THE RELEASE (GFP(MON)) AND STORAGE ESTIMATE (STOCA(MON)) CANDIDATES
C  CALCULATED IN OUTPUT SUBROUTINE ARE ACCEPTED TO
C  BE THE ESTIMATES FOR THE NEXT ITERATION
C
      IF(ITER.GT.1) STO(MON)=STOCA(MON)
```

```
      IF(ITER.GT.1) ESTFL(MON)=GFP(MON)
C
      SUM=0.0
      JK=K
      IF(I.EQ.1) JK=0
C
C TEMPORARY ARRAY OF RHS OF THE MN-TH MONTH'S CONSTRAINTS
C
      T(I)=((BEG+STO(MON))*CJ(1)*ESTFL(MON)*24/1000.)*SCALE
      T(I+1)=B(I+1)
      T(I+2)=B(I+2)
      T(I+3)=B(I+3)
      T(I+4)=B(I+4)
      T(I+5)=B(I+5)
      T(I+6)=DMAX1(STO(MON)-VARYMX,STMIN(MON))
      T(I+7)=DMIN1(STO(MON)+VARYMX,STMAX(MON))
      T(I+8)=B(I+8)
C
C  CALCULATION OF THE COEFFICIENTS
C  NOTE: THE HYDRO PRODUCTION CONSTR. FOR THE FIRST MONTH (I=1) IS
C  DIFFERENT THAN THE SUBSEQUENT ONES
C
      UERF=ERF(CJ,STO(MON))
      IF(I.EQ.1) GO TO 30
      AAJM1=CJ(1)*ESTFL(MON)*24/1000.*SCALE
   30 AAJ=(OERF+UERF)*24/1000.*SCALE
      AAJP2=CJ(1)*ESTFL(MON)*24/1000.*SCALE
      DO 35 IJ=1,5
   35 PIV(IJ)=0.0
      IF(I.EQ.1) GO TO 40
      PIV(1)=AAJM1-AA(JK)
   40 PIV(2)=AAJ-AA(JK+1)
      PIV(5)=AAJP2-AA(JK+4)
      IF(ITER.EQ.1) GO TO 180
      IF(I.NE.1) GO TO 45
      PIV(1)=PIV(2)
      PIV(2)=PIV(3)
      PIV(3)=PIV(4)
      PIV(4)=PIV(5)
      PIV(5)=0.0
45    IF(PIV(1)+PIV(2)+PIV(3)+PIV(4)+PIV(5).EQ.0.) GO TO 190
      K1=IROW(I)
      K2=IROW(I+1)-1
      DO 67 KP=K1,K2
      KV=KP-K1+1
 67     AA(KP)=AA(KP)+PIV(KV)
      GO TO 190
  180 IK=JK+4
      PIV(3)=-2.0*SCALE
      PIV(4)=-2.0*SCALE
      DO 185 L=JK,IK
      IF(I.EQ.1.AND.L.EQ.JK) GO TO 185
      AA(L)=PIV(L-JK+1)
  185 CONTINUE
```

```
  190 BEG=STO(MON)
      K=K+22
      IF(I.GT.1) K=K+2
      I=I+9
      J=J+9
 200  OERF=UERF
C
      DO 50 I=1,M
   50 B(I)=T(I)
C
C INITIALIZING THE NECESSARY LAND AND POWELL VARIABLES
C
 111     BIG=1.E8
      SMALL=1.E-6
      M=9*(LASTM-MSTART+1)
      N=9*(LASTM-MSTART+1)
      MXSIZE=81
      MAXM=108
      MAXN=108
      ISDONE=0
      INREV=0
      IR=0
      ISBND=1
      ITR=0
      MNOW=M
      NEGINV=0
      NEGROW=0
      NEWX=0
      NEWY=0
      R=0.0
      SIZE=0
      ISBIG=1
      YAMINC=0.0
      ISTATE=0
      DO 300 IK=1,60
      INBASE(IK)=0
      PIV(IK)=0.0
      X(IK)=0.0
      YAC(IK)=0.0
      G(IK)=0.0
      GR(IK)=0.0
      ISEFF(IK)=0.0
      SLACK(IK)=0.0
      Y(IK)=0.0
 300  CONTINUE
      DO 350 IK=1,100
      XBASIS(IK)=0.0
      XR(IK)=0.0
      YBASIS(IK)=0.0
      YR(IK)=0.0
      DO 360 IJ=1,70
 360  INV(IK,IJ)=0.0
 350  CONTINUE
C
```

```
C  INVOKING THE LAND AND POWELL ROUTINES TO
C  SOLVE THE FORMULATED LP PROBLEM
C
      CALL SOLVER
C
      IF(MOREPR.EQ.3) CALL IPRINT
      KFLAG=0
 1000 RETURN
      END
```

```
C     DRIVER FOR LP ROUTINES
      SUBROUTINE SOLVER
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER SIZE,SIZE1,XBASIS,YBASIS
      REAL*8 INV
      COMMON /LINPCO/
     1BOUND(156),C(156),INBASE(156),PIV(156),X(156),YAC(156),
     2B(156),G(156),GR(156),ISEFF(156),S(156),SLACK(156),Y(156),
     3INV(130,130),XBASIS(130),XR(130),YBASIS(130),YR(130),
     4       TOL(8),BIG,DRIVER,INREV,IR,
     6       IRMAX, ISBIG, ISDONE, ISTATE, ITR, ITRMAX, M,
     7       MARKI, MARKK, MAXM, MAXN, MNOW, MORE, MOREPR,
     8       MXSIZE, N, NEGINV, NEGROW, NEWX, NUMSLK, OBJ, R,
     9       SIZE, SIZE1, SMALL, XKPOS, YAMINC, NEWY, ISBND
      COMMON /ERRORS/ ERR
      ERR1=ERR
   10 CALL DOANLP
      IF(ISTATE.GT.1) GO TO 40
      CALL CHACC
      IF(ISTATE.EQ.1) GO TO 20
      IF(IR.GE.IRMAX) GO TO 40
C     IF WE COME OUT CLEAN OR SCREW UP TWICE IN A ROW IN THE SAME PLACE
C        (I.E. SAME ERROR > TOLERANCE) THEN EITHER IT WORKED OR WE'LL
C        NEVER DO ANY BETTER SO QUIT NOW ALREADY
      IF(ERR.EQ.ERR1 .OR. ERR1.EQ.-1.) GO TO 30
      CALL REVERT
      ISTATE=11
      ERR1=-1.
      GO TO 10
   20 ERR=0.
   30 ISTATE=1
   40 RETURN
      END
C     THESE ROUTINES RESOLVE THE LP WITH MODIFIED COEFFICIENTS.
C     IF YOU LOOK HARD ENOUGH THEY ARE ALL MADE UP OF PARTS OF
C     LAND AND POWELL ROUTINES, SO I'LL BE BRIEF
      SUBROUTINE CHGA
      IMPLICIT REAL*8(A-H,O-Z)
C     PIV CONTAINS CHANGES TO NEWY'TH ROW OF A MATRIX
      INTEGER SIZE,SIZE1,XBASIS,YBASIS
      REAL*8 INV
      COMMON /LINPCO/
     1BOUND(156),C(156),INBASE(156),PIV(156),X(156),YAC(156),
     2B(156),G(156),GR(156),ISEFF(156),S(156),SLACK(156),Y(156),
     3INV(130,130),XBASIS(130),XR(130),YBASIS(130),YR(130),
     4       TOL(8),BIG,DRIVER,INREV,IR,
     6       IRMAX, ISBIG, ISDONE, ISTATE, ITR, ITRMAX, M,
     7       MARKI, MARKK, MAXM, MAXN, MNOW, MORE, MOREPR,
     8       MXSIZE, N, NEGINV, NEGROW, NEWX, NUMSLK, OBJ, R,
     9       SIZE, SIZE1, SMALL, XKPOS, YAMINC, NEWY, ISBND
      COMMON /AREF/ AA(600),JCOL(600),IROW(157),MAXA
      DIMENSION PIVA(30)
      IF(MOREPR.GT.0) CALL IPRINT
      TOL1=TOL(1)
```

```
      TOL4=TOL(4)
      IY=NEWY
      K1=IROW(IY)
      K2=IROW(IY+1)-1
C
C     UPDATE BASIC COLUMNS
      K3=K2-K1+1
      DO 5 K=1,K3
      PIVA(K)=PIV(K)
    5 PIV(K)=0.
   10 DO 100 K=K1,K2
      NEWX=JCOL(K)
      INB=INBASE(NEWX)
      DELTAA=PIVA(K-K1+1)
      IF(DABS(DELTAA).LT.SMALL .OR. INB.LE.0) GO TO 100
C     THESE SHENANIGANS ARE TO AVOID CHANGING AA(K) TWICE WHEN WE GO
C        THROUGH THE LOOP AGAIN
      PIVA(K-K1+1)=0.
      AA(K)=AA(K)+DELTAA
      IF(ISEFF(IY).EQ.0) GO TO 100
      XNEW=XR(INB)
      YAMINC=DELTAA*Y(IY)
      IF(DABS(YAMINC).LT.TOL4) YAMINC=0.
      CALL NEWVEC
      IF(DABS(GR(INB)).GT.TOL1) GO TO 70
C
C     COP-OUT (RESTART FROM BEGINNING)
      ISTATE=0
      GO TO 110
   70 R=XNEW/GR(INB)
      DO 80 L=1,SIZE
   80 IF(DABS(GR(L)).LE.SMALL) GR(L)=0.
      NEWY=INB
      CALL CHBSIS
      INREV=1
      ISTATE=10
      CALL DOANLP
      IF(ISTATE.GT.1) GO TO 110
C     NEW SOLUTION MAY HAVE MADE SOME MORE COLUMNS BASIC, SO START
C         FROM BEGINNING AGAIN SO AS NOT TO LEAVE ANY OUT.
C         WE'LL EVENTUALLY STOP WHEN WE CYCLE THROUGH WITHOUT FINDING
C         ANY NEW BASIC COLUMNS TO CHANGE
      GO TO 10
  100 CONTINUE
      YAMINC=0.
      NEWX=0
      NEGROW=0
      NEGINV=0
      IYEFF=ISEFF(IY)
      SLKNEW=B(IY)
C     CHANGE NON-BASIC COLUMNS
      DO 20 K=K1,K2
      J=JCOL(K)
      DELTAA=PIVA(K-K1+1)
```

```
      IF(DABS(DELTAA).LE.SMALL) GO TO 20
      INB=INBASE(J)
      IF(INB.GT.0) GO TO 20
      AA(K)=AA(K)+DELTAA
      IF(IYEFF.EQ.0) GO TO 20
      YACJ=YAC(J)+DELTAA*Y(IY)
      IF(DABS(YACJ).LT.TOL4) YACJ=0.
      YAC(J)=YACJ
      IF(INB.EQ.-1) YACJ=-YACJ
C     LOOK FOR A VARIABLE TO BRING HOME TO THE BASIS, DEARIE]
      IF(YACJ.GE.YAMINC) GO TO 20
      YAMINC=YACJ
      NEWX=J
   20 SLKNEW=SLKNEW-X(J)*AA(K)
      IF(NEWX.NE.0) ISTATE=12
      IF(DABS(SLKNEW).LT.TOL(2)) SLKNEW=0.
      SLACK(IY)=SLKNEW
      SI=S(IY)
      IF(IYEFF.EQ.0 .AND. (SI*SLKNEW .GT.0. .OR. SLKNEW.EQ.0.)) GO TO 60
      IF(IYEFF.NE.0 .AND. SLKNEW.EQ.0.) GO TO 60
      IF(IYEFF.NE.0) GO TO 30
C     OOPS] WE MADE A FORMERLY INACTIVE CONSTRAINT INFEASIBLE
      NEGROW=IY
      ISTATE=11
      GO TO 60
C     RETURN SLACK OF ACTIVE CONSTRAINT TO ZERO, BY PRETENDING
C        TO CHANGE B INSTEAD]
   30 R=SLKNEW
      NEWY=IY
      CALL CHGB
      R=0.
   60 IF(ISTATE.EQ.11) YAMINC=0.
      IF(ISTATE.NE.1) CALL SOLVER
  110 IF(MOREPR.GT.0) CALL IPRINT
      RETURN
      END
C
      SUBROUTINE CHGB
      IMPLICIT REAL*8(A-H,O-Z)
C     B VECTOR HAS ALREADY BEEN CHANGED;  DELTA B IS IN PIV
      INTEGER SIZE,SIZE1,XBASIS,YBASIS
      REAL*8 INV
      COMMON /LINPCO/
     1BOUND(156),C(156),INBASE(156),PIV(156),X(156),YAC(156),
     2B(156),G(156),GR(156),ISEFF(156),S(156),SLACK(156),Y(156),
     3INV(130,130),XBASIS(130),XR(130),YBASIS(130),YR(130),
     4      TOL(8),BIG,DRIVER,INREV,IR,
     6      IRMAX, ISBIG, ISDONE, ISTATE, ITR, ITRMAX, M,
     7      MARKI, MARKK, MAXM, MAXN, MNOW, MORE, MOREPR,
     8      MXSIZE, N, NEGINV, NEGROW, NEWX, NUMSLK, OBJ, R,
     9      SIZE, SIZE1, SMALL, XKPOS, YAMINC, NEWY, ISBND
      COMMON /AREF/ AA(600),JCOL(600),IROW(157),MAXA
      IF(MOREPR.GT.0) CALL IPRINT
      TOL1=TOL(1)
```

```
      TOL2=TOL(2)
C     HOW BAD IS MOST INFEASIBLE VARIABLE OR CONSTRAINT? (WILL BE
C         NEGATIVE IF INFEASIBLE, 0 OTHERWISE)
      HOWNEG=0.
C     WHICH VARIABLE IS INFEASIBLE
      NEGINV=0
C     WHICH CONSTRAINT IS INFEASIBLE
      NEGROW=0
C
C     CHANGE BASIC VARIABLES
      DO 20 K=1,SIZE
      XRK=XR(K)
C     WE WERE CALLED FROM CHGA
      IF(NEWY.EQ.0) GO TO 5
      IYEFF=ISEFF(NEWY)
      XRK=XRK+R*INV(K,IYEFF)
      GO TO 12
    5 DO 10 L=1,SIZE
      I=YBASIS(L)
      DELBI=PIV(I)
      IF(DABS(DELBI).LT.TOL2) GO TO 10
      XRK=XRK+DELBI*INV(K,L)
   10 CONTINUE
   12 IF(DABS(XRK).LE.TOL1) XRK=0.
      IF(DABS(XRK-XR(K)).LE.SMALL) GO TO 20
      XR(K)=XRK
      J=XBASIS(K)
      X(J)=XRK
      IF(XRK.GE.HOWNEG) GO TO 15
      HOWNEG=XRK
      NEGINV=K
C     DRIVER=1 MEANS INCREASE VALUE OF XR(K) 'COS IT'S NEGATIVE
      DRIVER=1.0
      GO TO 20
   15 BOUNDJ=BOUND(J)
      IF(DABS(BOUNDJ-XRK).LT.TOL1) XRK=BOUNDJ
      IF(BOUNDJ.EQ.-1. .OR. BOUNDJ-XRK.GE.HOWNEG) GO TO 20
      HOWNEG=BOUNDJ-XRK
      NEGINV=K
C     DECREASE XR(K) 'COS IT'S ABOVE ITS BOUND
      DRIVER=-1.0
   20 CONTINUE
C
C     CHANGE SLACK VARIABLES
      DO 30 I=1,M
      IF(ISEFF(I).NE.0) GO TO 30
      SLKI=B(I)
      LAST=IROW(I+1)-1
      ISTART=IROW(I)
      DO 35 LOOK=ISTART,LAST
      J=JCOL(LOOK)
   35 IF(INBASE(J).NE.0) SLKI=SLKI-AA(LOOK)*X(J)
      IF(DABS(SLKI).LE.TOL2) SLKI=0.
      SLACK(I)=SLKI
```

```
      IF(NEGINV.NE.0) GO TO 30
      SI=S(I)
      ABSLKI=DABS(SLKI)
      IF(SI.NE.0. .AND. SI*SLKI.GE.HOWNEG .OR. SI.EQ.0. .AND. -ABSLKI
     1   .GE.HOWNEG) GO TO 30
      HOWNEG=-ABSLKI
      NEGROW=I
   30 CONTINUE
      IF(NEGINV.GT.0 .OR. NEGROW.GT.0) ISTATE=11
      IF(MOREPR.GT.0) CALL IPRINT
      RETURN
      END
      SUBROUTINE CHGC
      IMPLICIT REAL*8(A-H,O-Z)
C     C VECTOR HAS ALREADY BEEN CHANGED; DELTA C IS IN PIV
C     NOT SURPRISINGLY, THIS ROUTINE LOOKS A LOT LIKE THE DUAL OF CHGB
      INTEGER SIZE,SIZE1,XBASIS,YBASIS
      REAL*8 INV
      COMMON /LINPCO/
     1BOUND(156),C(156),INBASE(156),PIV(156),X(156),YAC(156),
     2B(156),G(156),GR(156),ISEFF(156),S(156),SLACK(156),Y(156),
     3INV(130,130),XBASIS(130),XR(130),YBASIS(130),YR(130),
     4      TOL(8),BIG,DRIVER,INREV,IR,
     6      IRMAX, ISBIG, ISDONE, ISTATE, ITR, ITRMAX, M,
     7      MARKI, MARKK, MAXM, MAXN, MNOW, MORE, MOREPR,
     8      MXSIZE, N, NEGINV, NEGROW, NEWX, NUMSLK, OBJ, R,
     9      SIZE, SIZE1, SMALL, XKPOS, YAMINC, NEWY, ISBND
      COMMON /AREF/ AA(600),JCOL(600),IROW(157),MAXA
      IF(MOREPR.GT.0) CALL IPRINT
      TOL3=TOL(3)
      TOL4=TOL(4)
      YAMINC=0.
      NEWX=0
C
C     CHANGE DUAL VARIABLES
      DO 20 L=1,SIZE
      YRL=YR(L)
      DO 10 K=1,SIZE
      J=XBASIS(K)
      DELTAC=PIV(J)
      IF(DABS(DELTAC).LT.TOL4) GO TO 10
C     UPDATE DUAL BASIC VARIABLES
      YRL=YRL+DELTAC*INV(K,L)
   10 CONTINUE
      IF(DABS(YRL).LT.TOL3) YRL=0.
      IF(DABS(YRL-YR(L)).LE.SMALL) GO TO 20
      YR(L)=YRL
      I=YBASIS(L)
      Y(I)=YRL
      YRL=YRL*S(I)
      IF(YRL.GE.YAMINC) GO TO 20
C     DUAL VARIABLE YR(L) IS DUAL INFEASIBLE
C     NEWX > N MEANS MAKE YR(NEWX-N) DUAL FEASIBLE
      YAMINC=YRL
```

```
      NEWX=I+N
   20 CONTINUE
C
C     CHANGE DUAL SLACKS
      DO 30 J=1,N
   30 IF(INBASE(J).LE.0) YAC(J)=-C(J)
      DO 50 L=1,SIZE
      YRL=YR(L)
      I=YBASIS(L)
      ISTART=IROW(I)
      LAST=IROW(I+1)-1
      DO 40 LOOK=ISTART,LAST
      J=JCOL(LOOK)
   40 IF(INBASE(J).LE.0) YAC(J)=YAC(J)+YRL*AA(LOOK)
   50 CONTINUE
      DO 60 J=1,N
      INJ=INBASE(J)
      IF(INJ.GT.0) GO TO 60
      YACJ=YAC(J)
      IF(DABS(YACJ).LT.TOL4) YACJ=0.
      YAC(J)=YACJ
      IF(NEWX.GT.N) GO TO 60
      IF(INJ.EQ.-1) YACJ=-YACJ
      IF(YACJ.GE.YAMINC) GO TO 60
      YAMINC=YACJ
C     PRIMAL VARIABLE X(J) SHOULD INCREASE TO REACH OPTIMALITY
C        (DUAL FEASIBILITY)
      NEWX=J
   60 CONTINUE
      IF(NEWX.NE.0) ISTATE=12
      IF(MOREPR.GT.0) CALL IPRINT
      RETURN
      END
```

```
      SUBROUTINE OUTPUT
C
C THIS ROUTINE CALCULATES THE OBJECTIVE FUNCTION VALUES
C AND GIVES THE OUTPUT FROM THE PROGRAM
C
      IMPLICIT REAL*8 (A-H,O-Z)
      CHARACTER*3 IMONTH
      REAL*8 INV,KMIN
      INTEGER SIZE,SIZE1,XBASIS,YBASIS
      COMMON/IO/ IOIN,IOOUT
      COMMON/LINPCO/
     1BOUND(156),C(156),INBASE(156),PIV(156),X(156),YAC(156),
     2B(156),G(156),GR(156),ISEFF(156),S(156),SLACK(156),Y(156),
     3INV(130,130),XBASIS(130),XR(130),YBASIS(130),YR(130),
     4      TOL(8),BIG,DRIVER,INREV,IR,
     6      IRMAX, ISBIG, ISDONE, ISTATE, ITR, ITRMAX, M,
     7      MARKI, MARKK, MAXM, MAXN, MNOW, MORE, MOREPR,
     8      MXSIZE, N, NEGINV, NEGROW, NEWX, NUMSLK, OBJ, R,
     9      SIZE, SIZE1, SMALL, XKPOS, YAMINC, NEWY, ISBND
      COMMON/DATA/ MSTART,LASTM,NSTRIP,SCALE,STOIN,SINTER,
     1STOVA(12),STASTO,SPICO(12),OCO(12),ENLO(2,12),ENWI(2,12),
     2DPS(12),FLOW(12),FLOBO(12),ESTFL(12),STMAX(12),STMIN(12),
     3HE(2,12),CJ(2),EE(2,12),EI(2,12),EEM(2,12),EIM(2,12),
     4EEL(12),EIL(12),CEE(2,12),CEI(2,12)
      COMMON/MAIN/ ITERMX,IPEEK,VARYMX,BETTER,VARFAC,VARMIN,VARFC1
      COMMON/RESULT/ STO(12),GFP(12),OBJECT,OBJ1,ITER,STOCA(12)
      COMMON /AREF/ AA(600),JCOL(600),IROW(157),MAXA
      COMMON /MONTH/ IMONTH(12)
      COMMON/CHANGES/ PROCHA
C
C WRITES THE HEADING TO THE OUTPUT FILE
C
      WRITE (6,900) ITER,ITR,STOIN,VARYMX
 900  FORMAT(/,2X,'ITERATION',I3,'SIMPLEX ITERATIONS',I5,
     1'INITIAL STORAGE',F7.1,'VARYMX',F6.1)
      WRITE(6,901)
 901  FORMAT(/,7X,'*',10X,'THE RESERVOIR DATA',10X,'*',5X)
      WRITE(6,905)
 905  FORMAT(' MONTH *  INFLOW  GEN.FL.POW  GEN.FL.SP TOT.OUTFL',
     1'  ON EX     OFF EX    ON IN    OFF IN  ',
     2'   HE ON    HE OFF      END STAGE',
     2'  MONTHLY BENEFIT')
      OBJECT=0.0
      J=1
      BEG=STOIN
C
C  THE LOOP TO CALCULATE MONTHLY BENEFITS AND THE
C  CANDIDATES FOR STORAGE (STOCA(MON)) AND RELEASE (GFP(MON))
C  ESTIMATES
C
      DO 100 MN=MSTART,LASTM
      MON=MN
      IF(MON.GT.12) MON=MON-12
      GFP(MON)=X(J)
```

```
      STOCA(MON)=X(J+8)
      BEMO=0.0
      IJ=J-1
      DO 70 KJ=1,9
   70 BEMO=BEMO+C(IJ+KJ)*X(IJ+KJ)
      BEMO=BEMO-STMIN(MON)*C(J+8)
      ZB=X(J)+X(J+1)
      STAGE=STOCA(MON)/STASTO
      WRITE(6,902) IMONTH(MON),FLOW(MON),GFP(MON),X(J+1),ZB,
     1X(J+4),X(J+5),X(J+6),X(J+7),X(J+2),X(J+3),STAGE,BEMO
  902 FORMAT(2X,A3,'  *  ',F6.2,4X,F6.2,4X,F6.2,4X,F6.2,4X,
     1F6.2,4X,F6.2,4X,F6.2,4X,F6.2,4X,F6.2,4X,F6.2,4X,F7.2,2X,F10.2)
      J=J+9
C
C  CALCULATION TO INSURE THAT THE CANDIDATES FOR THE NEW RELEASE
C  ESTIMATES (GFP(MON)) DO NOT DIFFER MORE THAN PROCHA TIMES FROM
C  THE ALREADY ACCEPTED ESTIMATES (ESTFL(MON))
C
      PROMENA=GFP(MON)-ESTFL(MON)
      CHANGE=DABS(PROMENA)
      VALTO=CHANGE/ESTFL(MON)
      IF(VALTO.GT.PROCHA) PROMENA=ESTFL(MON)*PROCHA*CHANGE/PROMENA
      GFP(MON)=PROMENA+ESTFL(MON)
      STOCA(MON)=BEG+FLOW(MON)-GFP(MON)
      IF(STOCA(MON).LT.STMAX(MON)) GO TO 75
      STOCA(MON)=STMAX(MON)
      GFP(MON)=BEG+FLOW(MON)-STOCA(MON)
      IF(GFP(MON).GT.FLOBO(MON)) GFP(MON)=FLOBO(MON)
   75 IF(STOCA(MON).GT.STMIN(MON)) GO TO 77
      STOCA(MON)=STMIN(MON)
      GFP(MON)=BEG+FLOW(MON)-STOCA(MON)
   77 BEG=STOCA(MON)
      OBJECT=OBJECT+BEMO
  100 CONTINUE
C
C

      WRITE(6,903) OBJECT
  903 FORMAT(2X,'THE OBJECTIVE FUNCTION VALUE IS  ',F12.2)
      RETURN
      END
```

LAND AND POWELL ROUTINES

```
      SUBROUTINE ADDCON
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 INV
      INTEGER SIZE, SIZE1, XBASIS, YBASIS
      COMMON /AREF/ AA(600),JCOL(600),IROW(157),MAXA
      COMMON/LINPCO/
     1BOUND(156),C(156),INBASE(156),PIV(156),X(156),YAC(156),
     2B(156),G(156),GR(156),ISEFF(156),S(156),SLACK(156),Y(156),
     3INV(130,130),XBASIS(130),XR(130),YBASIS(130),YR(130),
     4      TOL(8),BIG,DRIVER,INREV,IR,
     6      IRMAX, ISBIG, ISDONE, ISTATE, ITR, ITRMAX, M,
     7      MARKI, MARKK, MAXM, MAXN, MNOW, MORE, MOREPR,
     8      MXSIZE, N, NEGINV, NEGROW, NEWX, NUMSLK, OBJ, R,
     9      SIZE, SIZE1, SMALL, XKPOS, YAMINC, NEWY, ISBND
      IF (SIZE1.GT.MXSIZE) GO TO 40
      I = NEWY - SIZE
      DO 10 L = 1, SIZE
           INV(L, SIZE1) = 0.0
10         INV(SIZE1, L) = 0.0
      ISTART = IROW(I)
      LAST = IROW(I+1) - 1
      DO 30 LOOK = ISTART, LAST
           J = JCOL(LOOK)
           IF (INBASE(J).LE.0) GO TO 30
           K = INBASE(J)
           AIJ = AA(LOOK)
           DO 20 L = 1, SIZE
20              INV(SIZE1, L) = INV(SIZE1, L) - AIJ * INV(K, L)
30         CONTINUE
      INV(SIZE1, SIZE1) = 1.0
      XR(SIZE1) = SLACK(I)
      ISEFF(I) = SIZE1
      XBASIS(SIZE1) = I + N
      YBASIS(SIZE1) = I
      YR(SIZE1) = 0.0
      SIZE = SIZE1
      SIZE1 = SIZE1 + 1
      IF (SIZE.GT.ISBIG) ISBIG = SIZE
      NUMSLK = NUMSLK + 1
      NEWY = SIZE
      GO TO 50
40    ISTATE = 4
50    RETURN
      END
```

```
      SUBROUTINE CHBSIS
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 INV
      INTEGER SIZE, SIZE1, XBASIS, YBASIS
      COMMON/LINPCO/
     1BOUND(156),C(156),INBASE(156),PIV(156),X(156),YAC(156),
     2B(156),G(156),GR(156),ISEFF(156),S(156),SLACK(156),Y(156),
     3INV(130,130),XBASIS(130),XR(130),YBASIS(130),YR(130),
     4      TOL(8),BIG,DRIVER,INREV,IR,
     6      IRMAX, ISBIG, ISDONE, ISTATE, ITR, ITRMAX, M,
     7      MARKI, MARKK, MAXM, MAXN, MNOW, MORE, MOREPR,
     8      MXSIZE, N, NEGINV, NEGROW, NEWX, NUMSLK, OBJ, R,
     9      SIZE, SIZE1, SMALL, XKPOS, YAMINC, NEWY, ISBND
      COMMON/IO/IOIN, IOOUT
      ITR = ITR + 1
      MOSNEG = 0
      HOWNEG = 0.0
      XOFNEG = 0.0
      DRITEM = 0.0
      TOL1 = TOL(1)
      IF (INREV.EQ.1) GO TO 90
      IF (R.EQ.0.0) GO TO 40
      DO 30 K = 1, SIZE
           XR(K) = XR(K) - R * GR(K) * XKPOS
           XXX = XR(K)
           IF (DABS(XXX).LE.TOL1) XR(K) = 0.0
           J = XBASIS(K)
           IF (J.LE.N) GO TO 10
           XXX = XR(K)
           I = J - N
           SI = S(I)
           IF(SI.EQ.0.0.AND.XXX.GT.0.0.OR.SI.EQ.-1.0) XXX = -XXX
           GO TO 20
10         BOUNDJ = BOUND(J)
           IF (DABS(BOUNDJ-XXX).LE.TOL1) XR(K) = BOUNDJ
           XXX = XR(K)
           IF (XXX.LE.BOUNDJ .OR. BOUNDJ.EQ.-1.0) GO TO 20
           XXX = BOUNDJ - XXX
20         IF (K.EQ.NEGINV) XOFNEG = XXX
           IF (XXX.GE.HOWNEG.OR.K.EQ.NEGINV) GO TO 30
           MOSNEG = K
           DRITEM = 1.0
           IF (XR(K).GE.0.0) DRITEM = -1.0
           HOWNEG = XXX
30         CONTINUE
      IF (NEWY.NE.-1) GO TO 40
      IT = INBASE(NEWX)
      INBASE(NEWX) = -1
      IF(IT.EQ.-1) INBASE(NEWX) = 0
      IXOUT = NEWX
      OBJ = OBJ - R * YAMINC
      GO TO 120
40    IXOUT = XBASIS(NEWY)
      IF (IXOUT.GT.N) GO TO 50
```

```
      INBASE(IXOUT) = 0
      IF (GR(NEWY)*XKPOS.LT.0.0.AND.NEWY.NE.NEGINV) INBASE(IXOUT) = -1
      IF (NEWY.EQ.NEGINV.AND.XR(NEWY).GT.0.0) INBASE(IXOUT) = -1
50    IF (NEWX.GT.N) GO TO 60
      IHOLD = INBASE(NEWX)
      INBASE(NEWX) = NEWY
60    XBASIS(NEWY) = NEWX
      IF(NEWX.GT.N) NUMSLK = NUMSLK + 1
      IF (IXOUT.GT.N) NUMSLK = NUMSLK - 1
      XR(NEWY) = R
      IF (NEWX.LE.N) GO TO 70
      I = NEWX - N
      IF (S(I).EQ.-1.0) XR(NEWY) = -R
      GO TO 80
70    IF (IHOLD.EQ.-1) XR(NEWY) = BOUND(NEWX) - R
80    OBJ = OBJ - R * YAMINC
90    RR = 1.0/GR(NEWY)
      DO 110 L = 1, SIZE
           IF (DABS(INV(NEWY,L)).LT.SMALL) GO TO 110
           RL = INV(NEWY,L) * RR
           DO 100 K = 1, SIZE
                INV(K,L) = INV(K,L) - RL * GR(K)
100             CONTINUE
           INV(NEWY,L) = RL
           IF (INREV.NE.1) YR(L) = YR(L) - RL * YAMINC * XKPOS
110        CONTINUE
120   IF (R.EQ.0.0.OR.XOFNEG.LT.0.0.AND.NEWY.NE.NEGINV) GO TO 130
      NEGINV = MOSNEG
      DRIVER = DRITEM
130   RETURN
      END
```

```
      SUBROUTINE REVERT
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 INV
      INTEGER SIZE, SIZE1, XBASIS, YBASIS
      COMMON/IO/IOIN, IOOUT
      COMMON/LINPCO/
     1BOUND(156),C(156),INBASE(156),PIV(156),X(156),YAC(156),
     2B(156),G(156),GR(156),ISEFF(156),S(156),SLACK(156),Y(156),
     3INV(130,130),XBASIS(130),XR(130),YBASIS(130),YR(130),
     4      TOL(8),BIG,DRIVER,INREV,IR,
     6      IRMAX, ISBIG, ISDONE, ISTATE, ITR, ITRMAX, M,
     7      MARKI, MARKK, MAXM, MAXN, MNOW, MORE, MOREPR,
     8      MXSIZE, N, NEGINV, NEGROW, NEWX, NUMSLK, OBJ, R,
     9      SIZE, SIZE1, SMALL, XKPOS, YAMINC, NEWY, ISBND
9000  FORMAT (1X, 20X, 'REINVERTED AT ITERATION ', I6)
      IR = IR + 1
      ITHOLD = ITR
      INREV = 1
      HOLD = SMALL
      SMALL = 0.0
      TOL8 = TOL(8)
      DO 20 K = 1, SIZE
10        IF (XBASIS(K).LE.N) GO TO 20
          I = XBASIS(K) - N
          L = ISEFF(I)
          IF (K.EQ.L) GO TO 20
          XBASIS(K) = XBASIS(L)
          XBASIS(L) = I + N
          J = XBASIS(K)
          IF (J.GT.N) GO TO 10
20        CONTINUE
      DO 40 K = 1, SIZE
          IF (XBASIS(K).LE.N) XBASIS(K) = -XBASIS(K)
          DO 30 L = 1, SIZE
30            INV(K,L) = 0.0
40        INV(K,K) = 1.0
      DO 50 J = 1, N
          IF (INBASE(J).NE.-1) INBASE(J) = 0
50        CONTINUE
      DO 90 K = 1, SIZE
          NEWX = -XBASIS(K)
          IF (NEWX.LT.0.OR.NEWX.GT.N) GO TO 90
60        CALL NEWVEC
          NEWY = 0
          DO 80 KK = 1, SIZE
              IF (XBASIS(KK).GT.0) GO TO 80
              ABDIF = DABS(GR(KK))
              IF (ABDIF.LT.TOL8) GO TO 80
              IF (NEWY.NE.0) GO TO 70
              BEST = DABS(1.0 - ABDIF)
              NEWY = KK
              GO TO 80
70            ABDIF = DABS (1.0 - ABDIF)
              IF (ABDIF.GE.BEST) GO TO 80
```

```
            BEST = ABDIF
            NEWY = KK
80          CONTINUE
         IF (NEWY.EQ.0) GO TO 90
         IHOLD = -XBASIS(NEWY)
         IF (IHOLD.EQ.NEWX) IHOLD = 0
         CALL CHBSIS
         XBASIS(NEWY) = NEWX
         IF (IHOLD.EQ.0) GO TO 90
         NEWX = IHOLD
         GO TO 60
90       CONTINUE
      NUMSLK = 0
      DO 110 K = 1, SIZE
         J = XBASIS(K)
         IF (J.GT.0) GO TO 100
         I = YBASIS(K)
         XBASIS(K) = N + I
         NUMSLK = NUMSLK + 1
         GO TO 110
100      IF (J.LE.N) INBASE(J) = K
         IF (J.GT.N) NUMSLK = NUMSLK + 1
110      CONTINUE
      SMALL = HOLD
      DO 120 K = 1, SIZE
         XR(K) = 0.0
120      YR(K) = 0.0
      DO 140 K = 1, SIZE
         I = YBASIS(K)
         J = XBASIS(K)
         TC = 0.0
         IF (J.LE.N) TC = C(J)
         TB = B(I)
         DO 130 JJ = 1, N
            IF (INBASE(JJ).NE.-1) GO TO 130
            TB = TB - BOUND(JJ) * A(I,JJ)
130      CONTINUE
         DO 140 L = 1, SIZE
            XR(L) = XR(L) + TB * INV(L,K)
            YR(L) = YR(L) + TC * INV(K,L)
            IF (DABS(YR(L)).LE.SMALL) YR(L) = 0.0
            IF (DABS(XR(L)).LE.SMALL) XR(L) = 0.0
140      CONTINUE
      NEGINV = 0
      T = 0.0
      DO 180 K = 1, SIZE
         XRK = XR(K)
         J = XBASIS(K)
         IF (J.GT.N) GO TO 160
         IF (ISBND.EQ.0) GO TO 150
         IF (BOUND(J).EQ.-1.0) GO TO 150
         IF (XRK.GT.BOUND(J)) XRK = BOUND(J) - XR(K)
150      IF (XRK.GE.T) GO TO 180
         GO TO 170
```

```
160       I = J - N
          IF (S(I).NE.0.0.AND.XRK*S(I).GE.T.OR.S(I).EQ.0.0.AND.
     1        DABS(XRK)*(-1.0).GE.T) GO TO 180
170       T = -1.0 * DABS(XRK)
          NEGINV = K
          DRIVER = 1.0
          IF (XR(K).GT.0.0) DRIVER = -1.0
180       CONTINUE
      IF (NUMSLK.GE.1) CALL REDUCE
      CALL CHSLCK
      CALL ISOPT
      ITR = ITHOLD
      INREV = 0
      OBJ = 0.0
      DO 190 J = 1, N
          IF (INBASE(J).EQ.0) GO TO 190
          OBJ = OBJ + X(J) * C(J)
190       CONTINUE
C     WRITE (IOOUT, 9000) ITR
      RETURN
      END
```

```
      SUBROUTINE CHACC
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 INV
      INTEGER SIZE, SIZE1, XBASIS, YBASIS
      COMMON/LINPCO/
     1BOUND(156),C(156),INBASE(156),PIV(156),X(156),YAC(156),
     2B(156),G(156),GR(156),ISEFF(156),S(156),SLACK(156),Y(156),
     3INV(130,130),XBASIS(130),XR(130),YBASIS(130),YR(130),
     4      TOL(8),BIG,DRIVER,INREV,IR,
     6      IRMAX, ISBIG, ISDONE, ISTATE, ITR, ITRMAX, M,
     7      MARKI, MARKK, MAXM, MAXN, MNOW, MORE, MOREPR,
     8      MXSIZE, N, NEGINV, NEGROW, NEWX, NUMSLK, OBJ, R,
     9      SIZE, SIZE1, SMALL, XKPOS, YAMINC, NEWY, ISBND
      COMMON/IO/IOIN, IOOUT
      COMMON /AREF/ AA(600),JCOL(600),IROW(157),MAXA
      COMMON /ERRORS/ ERR
 9000 FORMAT (1H0,'UNACCEPTABLE ERROR OF ',F16.8,' FOUND IN B-SLACK-AX O
     1F CONSTRAINT',I6)
 9004 FORMAT (1H0,'UNACCEPTABLE RELATIVE ERROR OF ',F16.8,' FOUND IN B-S
     1LACK-AX OF CONSTRAINT',I6/1H ,'THE ABSOLUTE ERROR IS ',F16.8,' AND
     2 B(I) IS ',F16.8)
 9008 FORMAT (1H0,'UNACCEPTABLE ERROR OF ',F16.8,' FOUND IN YA-C OF BASI
     1C VARIABLE ',I6)
 9012 FORMAT (1H0,'UNACCEPTABLE RELATIVE ERROR OF ',F16.8,' FOUND IN YA-
     1C OF BASIC VARIABLE, ',I6/1H ,'THE ABSOLUTE ERROR IS ',F16.8,' AND
     2 C(J) IS ',F16.8)
      IF (NUMSLK.EQ.0) GO TO 10
      DO 5 K = 1, SIZE
           IF (XBASIS(K).LE.N) GO TO 5
           I = XBASIS(K) - N
           SLACK(I) = XR(K)
 5         CONTINUE
 10   DO 20 J = 1, N
           IF (INBASE(J).LE.0) GO TO 20
           YAC(J) = -C(J)
 20        CONTINUE
      TOL2 = TOL(2)
      TOL6 = TOL(6)
      DO 40 I = 1, MNOW
           ISEFFI = ISEFF(I)
           YI = Y(I)
           BAXSL = B(I) - SLACK(I)
           ISTART = IROW(I)
           LAST = IROW(I+1) - 1
           DO 30 LOOK = ISTART, LAST
                J = JCOL(LOOK)
                INJ = INBASE(J)
                IF (INJ.EQ.0) GO TO 30
                AIJ = AA(LOOK)
                BAXSL = BAXSL - X(J) * AIJ
                IF (INJ.GT.0.AND.ISEFFI.NE.0) YAC(J) = YAC(J) + YI*AIJ
 30             CONTINUE
           ERR = DABS(BAXSL)
```

```
            IF (ERR.GT.TOL2) GO TO 60
            ABSB = DABS(B(I))
            IF (ABSB.LT.1.0) ABSB = 1.0
            IF (ERR / ABSB .GT. TOL6) GO TO 65
 40         CONTINUE
       TOL7 = TOL(7)
       TOL4 = TOL(4)
       DO 50 J = 1, N
            IF (INBASE(J) .LE. 0) GO TO 50
            ERR = DABS(YAC(J))
            IF (ERR .GT. TOL4) GO TO 70
            ABSC = DABS(C(J))
            IF (ABSC.LT.1.0) ABSC = 1.0
            IF (ERR / ABSC .GT. TOL7) GO TO 75
 50         CONTINUE
       GO TO 90
 60    WRITE(IOOUT,9000) ERR,I
       GO TO 80
 65    RELERR = ERR / ABSB
       WRITE (IOOUT, 9004) RELERR, I, ERR, ABSB
       GO TO 80
 70    WRITE (IOOUT, 9008) ERR, J
       GO TO 80
 75    RELERR = ERR / ABSC
       WRITE (IOOUT, 9012) RELERR, J, ERR, ABSC
 80    ISTATE = 7
 90    RETURN
       END
```

```
      SUBROUTINE DOANLP
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 INV
      INTEGER SIZE, SIZE1, XBASIS, YBASIS
      COMMON/LINPCO/
     1BOUND(156),C(156),INBASE(156),PIV(156),X(156),YAC(156),
     2B(156),G(156),GR(156),ISEFF(156),S(156),SLACK(156),Y(156),
     3INV(130,130),XBASIS(130),XR(130),YBASIS(130),YR(130),
     4      TOL(8),BIG,DRIVER,INREV,IR,
     6      IRMAX, ISBIG, ISDONE, ISTATE, ITR, ITRMAX, M,
     7      MARKI, MARKK, MAXM, MAXN, MNOW, MORE, MOREPR,
     8      MXSIZE, N, NEGINV, NEGROW, NEWX, NUMSLK, OBJ, R,
     9      SIZE, SIZE1, SMALL, XKPOS, YAMINC, NEWY, ISBND
      COMMON/IO/IOIN, IOOUT
      IF (ISTATE.EQ.0) CALL FIRSTB
      IF (ISTATE.EQ.11) GO TO 20
      IF (ISTATE.EQ.12) GO TO 50
10    CALL CHSLCK
      IF(MOREPR.GT.0) CALL IPRINT
      IF (ITR.LE.ITRMAX) GO TO 20
      ISTATE = 5
      GO TO 80
20    IF (NEGROW.EQ.0 .AND. NEGINV.EQ.0) GO TO 40
      IF (NEGINV.NE.0) GO TO 30
      NEWY = NEGROW + SIZE
      DRIVER = 1.0
      IF (SLACK(NEGROW).GT.0.0) DRIVER = -1.0
      NEGINV = SIZE1
      CALL ADDCON
      IF (ISTATE.EQ.4) GO TO 80
30    CALL SEEKX
      IF (NEWX.NE.0) GO TO 50
      ISTATE = 2
      GO TO 80
40    CALL ISOPT
      IF (NEWX.NE.0) GO TO 50
      ISTATE = 1
      GO TO 80
50    CALL NEWVEC
      CALL SEEKY
      IF (NEWY.NE.0) GO TO 60
      ISTATE = 3
      GO TO 80
60    IF (NEWY.LE.SIZE) GO TO 70
      CALL ADDCON
      IF (ISTATE.EQ.4) GO TO 80
70    CALL CHBSIS
      CALL REDUCE
      GO TO 10
80    RETURN
      END
      REAL*8 FUNCTION A(I,J)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /AREF/ AA(600),JCOL(600),IROW(157),MAXA
```

```
      COMMON/IO/IOIN, IOOUT
      ISTART = IROW(I)
      LAST = IROW(I+1) - 1
      A = 0.0
      DO 1 LOOK = ISTART, LAST
           JHERE = JCOL(LOOK)
           IF (JHERE.LT.J) GO TO 1
           IF (JHERE.GT.J) RETURN
           A = AA(LOOK)
           RETURN
1          CONTINUE
      RETURN
      END
```

```
      SUBROUTINE IPRINT
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 INV
      INTEGER SIZE, SIZE1, XBASIS, YBASIS
      COMMON /LINPCO/
     1BOUND(156),C(156),INBASE(156),PIV(156),X(156),YAC(156),
     2B(156),G(156),GR(156),ISEFF(156),S(156),SLACK(156),Y(156),
     3INV(130,130),XBASIS(130),XR(130),YBASIS(130),YR(130),
     4      TOL(8),BIG,DRIVER,INREV,IR,
     6      IRMAX, ISBIG, ISDONE, ISTATE, ITR, ITRMAX, M,
     7      MARKI, MARKK, MAXM, MAXN, MNOW, MORE, MOREPR,
     8      MXSIZE, N, NEGINV, NEGROW, NEWX, NUMSLK, OBJ, R,
     9      SIZE, SIZE1, SMALL, XKPOS, YAMINC, NEWY, ISBND
      COMMON /AREF/ AA(600),JCOL(600),IROW(157),MAXA
      COMMON/IO/IOIN, IOOUT
 8000 FORMAT ('0',' THE SIGN(I) VECTOR INDICATES THE SIGN OF THE I-TH CO
     2NSTRAINT, 0 FOR EQ, 1 FOR LE, -1 FOR GE.')
 9000 FORMAT ('1','NON-ZERO ELEMENTS OF THE A MATRIX, FOLLOWED BY THEIR
     1COLUMN LABELS....')
 9001 FORMAT ('0',12(4X,I6))
 9002 FORMAT(1X, 12F10.3)
 9003 FORMAT (1X, 12(4X,I6))
 9004 FORMAT ('0','THE FOLLOWING VECTORS SHOW THE STARTING POINTS OF THE
     1 SUCCESSIVE ROWS OF A IN THE ABOVE LIST OF THE NON-ZERO ELEMENTS.'
     2'..')
 9005 FORMAT ('0', 24(I5))
 9006 FORMAT (1X, 24(I5))
 9200 FORMAT ('0', 'OBJECTIVE ',F22.8)
 9204 FORMAT ('0', 18X, 'J ', 10(3X,I3,4X))
 9205 FORMAT('0', 18X, 'I ', 10(3X,I3,4X))
 9208 FORMAT ('0', 11X, 'C VECTOR ', 10(F9.1, 1X))
 9212 FORMAT('0', 7X, 'BOUND VECTOR ', 10(F9.4,1X))
 9220 FORMAT ('0', 11X, 'X VECTOR ', 10(F9.4, 1X))
 9228 FORMAT ('0', 14X, 5HY*A-C,1X,10(F9.2,1X))
 9232 FORMAT (///)
 9234 FORMAT ('0', 12(6X,I3, 1X))
 9236 FORMAT ('0', 12(F9.1,1X))
 9238 FORMAT ('0', 12(F9.0,1X))
 9240 FORMAT ('0', 12(F9.4,1X))
 9244 FORMAT ('0', 12(F9.2,1X))
 9300 FORMAT ('0', 11X, 'B VECTOR ', 10(F9.1, 1X))
 9304 FORMAT ('0', 15X, 'SIGN ', 10(F9.0,1X))
 9308 FORMAT ('0', 11X, 'Y VECTOR ', 10(F9.4,1X))
 9312 FORMAT ('0', 14X, ' B-AX ', 10(F9.4,1X))
 9404 FORMAT ('1', 12X, 'COLUMN ', 7(6X,I2,6X))
 9408 FORMAT ('0', 12X, 'YBASIS ', 7(5X, I3, 6X))
 9412 FORMAT ('0', 16X, 'YR ', 2X, 7(F12.4, 2X))
 9416 FORMAT ('0', 'ROW XBS ', 4X, 'XR', 5X, 'INVERSE MATRIX'/)
 9420 FORMAT (2(I3,1X), 8(F12.4, 2X))
 9424 FORMAT (///5X, 8(5X, I3, 6X))
 9428 FORMAT ('0', 5X, 8(5X, I3, 6X))
 9432 FORMAT ('0', 5X, 8(F12.4, 2X))
 9436 FORMAT ('0', 'ROW', 5X, 'INVERSE MATRIX CONTINUES'/)
 9438 FORMAT (1X, I3, 2X, 8(F12.4, 2X))
```

```
9500  FORMAT ('1')
9504  FORMAT('0','    BIG',E12.4,', DRIVER',F12.1,',  INREV',I12,',      I
     1R',I12,',   IRMAX',I12,',   ISBND',I12/1H ,'ISDONE',I12,', ISTATE',I
     212,',     ITR',I12,', ITRMAX',I12,',       M',I12,',   MARKI',I12/1H
     3,' MARKK',I12,',    MAXA',I12,',    MAXM',I12,',    MAXN',I12,',    MO
     4RE',I12,', MXSIZE',I12/1H ,'      N',I12,', NEGINV',I12,', NEGROW',
     5I12,',    NEWX',I12,',    NEWY',I12,', NUMSLK',I12/1H ,'      R',
     6F12.5,',    SIZE',I12,',   SMALL',E12.4,', TOL(1)',E12.4,', TOL(2)',
     7E12.4,', TOL(3)',E12.4/1H ,'TOL(4)',E12.4,', TOL(5)',E12.4,
     8', TOL(6)',E12.4,', TOL(7)',E12.4,', TOL(8)',E12.4,',  XKPOS',
     9F12.1/1H ,'YAMINC',F12.5)
9516  FORMAT ('0','ISEFF'/1H ,40I3)
9520  FORMAT ('0', 'INBASE'/1X, 40I3)
9600  FORMAT ('0', I5, ' SIMPLEX ITERATIONS.')
9604  FORMAT ('0','(N.B., THE MAXIMUM SIZE OF THE INVERSE DURING THE CAL
     1CULATION WAS ',I4,')')
      IF(MOREPR.LE.0) RETURN
      IF(MOREPR.EQ.1) GO TO 400
C     IF(N.GE.8) GO TO 90
C     CALL SPRINT
C     GO TO 400
90    WRITE (IOOUT, 9000)
      LAST = IROW(MNOW + 1) - 1
      ISTART = 1
100   IEND = ISTART + 11
      IF (IEND.GT.LAST) IEND = LAST
      WRITE (IOOUT, 9001) (IJ, IJ = ISTART, IEND)
      WRITE (IOOUT, 9002) (AA(IJ), IJ = ISTART, IEND)
      WRITE (IOOUT, 9003) (JCOL(IJ), IJ = ISTART, IEND)
      IF (IEND.EQ.LAST) GO TO 105
      ISTART = IEND + 1
      GO TO 100
105   WRITE (IOOUT, 9004)
      ISTART = 1
110   IEND = ISTART + 23
      IF (IEND.GT.MNOW) IEND = MNOW
      WRITE (IOOUT, 9005) (I, I = ISTART, IEND)
      WRITE (IOOUT, 9006) (IROW(I), I = ISTART, IEND)
      IF (IEND.EQ.MNOW) GO TO 200
      ISTART = IEND + 1
      GO TO 110
200   WRITE (IOOUT, 9200) OBJ
      IEND = 10
      IF (N.LE.IEND) IEND = N
      WRITE (IOOUT, 9204) (J, J = 1, IEND)
      WRITE (IOOUT, 9208) (C(J), J = 1, IEND)
      IF (ISBND.EQ.0) GO TO 210
      WRITE (IOOUT, 9212) (BOUND(J), J = 1, IEND)
210   WRITE (IOOUT, 9220) (X(J), J = 1, IEND)
      WRITE (IOOUT, 9228) (YAC(J), J = 1, IEND)
230   IF (N.LE.IEND) GO TO 300
      WRITE (IOOUT, 9232)
      ISTART = IEND + 1
      IEND = IEND + 12
```

```
      IF (N.LE.IEND) IEND = N
      WRITE (IOOUT, 9234) (J, J = ISTART, IEND)
      WRITE (IOOUT, 9236) (C(J), J = ISTART, IEND)
      IF (ISBND.EQ.0) GO TO 235
      WRITE (IOOUT, 9240) (BOUND(J), J = ISTART, IEND)
  235 WRITE (IOOUT, 9240) (X(J), J = ISTART, IEND)
      WRITE (IOOUT, 9244) (YAC(J), J = ISTART, IEND)
      GO TO 230
  300 IEND = 10
      IF (MNOW.LE.IEND) IEND = MNOW
      WRITE (IOOUT, 9232)
      WRITE (IOOUT, 8000)
      WRITE (IOOUT, 9205) (I, I = 1, IEND)
      WRITE (IOOUT, 9300) (B(I), I = 1, IEND)
      WRITE (IOOUT, 9304) (S(I), I = 1, IEND)
      WRITE (IOOUT, 9308) (Y(I), I = 1, IEND)
      WRITE (IOOUT, 9312) (SLACK(I), I = 1, IEND)
  310 IF (MNOW.LE.IEND) GO TO 400
      WRITE (IOOUT, 9232)
      ISTART = IEND + 1
      IEND = IEND + 12
      IF (MNOW.LE.IEND) IEND = MNOW
      WRITE (IOOUT, 9234) (I, I = ISTART, IEND)
      WRITE (IOOUT , 9236) (B(I), I = ISTART, IEND)
      WRITE (IOOUT, 9238) (S(I), I = ISTART, IEND)
      WRITE (IOOUT, 9240) (Y(I), I = ISTART, IEND)
      WRITE (IOOUT, 9240) (SLACK(I), I = ISTART, IEND)
      GO TO 310
  400 IF(MOREPR.EQ.2) GO TO 600
      IEND = 7
      IF (SIZE.LE.IEND) IEND = SIZE
      WRITE (IOOUT, 9404) (L, L = 1, IEND)
  410 WRITE (IOOUT, 9408) (YBASIS(L), L = 1, IEND)
      WRITE (IOOUT, 9412) (YR(L), L = 1, IEND)
      WRITE (IOOUT, 9416)
      DO 430 K = 1, SIZE
  430      WRITE (IOOUT, 9420) K,XBASIS(K),XR(K),(INV(K,L),L=1,IEND)
  440 IF (SIZE.LE.IEND) GO TO 500
      ISTART = IEND + 1
      IEND = IEND + 8
      IF (SIZE.LE.IEND) IEND = SIZE
      WRITE (IOOUT, 9424) (L, L = ISTART, IEND)
      WRITE (IOOUT, 9428) (YBASIS(L), L = ISTART, IEND)
      WRITE (IOOUT, 9432) (YR(L), L = ISTART, IEND)
      WRITE (IOOUT, 9436)
      DO 445 K = 1, SIZE
  445      WRITE (IOOUT, 9438) K, (INV(K,L), L = ISTART, IEND)
      GO TO 440
  500 WRITE (IOOUT, 9232)
      WRITE(IOOUT,9504)BIG,DRIVER,INREV,IR,IRMAX,ISBND,ISDONE,ISTATE,
     1     ITR,ITRMAX,M,MARKI,MARKK,MAXA,MAXM,MAXN,MORE,MXSIZE,N,
     2     NEGINV,NEGROW,NEWX,NEWY,NUMSLK,R,SIZE,SMALL,(TOL(K),K=1,8),
     3     XKPOS,YAMINC
  600 WRITE (IOOUT, 9516) (ISEFF(I), I = 1, MNOW)
```

```
WRITE (IOOUT, 9520) (INBASE(J), J = 1, N)
WRITE (IOOUT, 9600) ITR
WRITE (IOOUT, 9604) ISBIG
RETURN
END
```

```
      SUBROUTINE CHSLCK
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 INV
      INTEGER SIZE, SIZE1, XBASIS, YBASIS
      COMMON /AREF/ AA(600),JCOL(600),IROW(157),MAXA
      COMMON/LINPCO/
     1BOUND(156),C(156),INBASE(156),PIV(156),X(156),YAC(156),
     2B(156),G(156),GR(156),ISEFF(156),S(156),SLACK(156),Y(156),
     3INV(130,130),XBASIS(130),XR(130),YBASIS(130),YR(130),
     4      TOL(8),BIG,DRIVER,INREV,IR,
     6      IRMAX, ISBIG, ISDONE, ISTATE, ITR, ITRMAX, M,
     7      MARKI, MARKK, MAXM, MAXN, MNOW, MORE, MOREPR,
     8      MXSIZE, N, NEGINV, NEGROW, NEWX, NUMSLK, OBJ, R,
     9      SIZE, SIZE1, SMALL, XKPOS, YAMINC, NEWY, ISBND
      IF (R.NE.0.0) NEGROW = 0
      HOWNEG = 0.0
      DO 10 J = 1, N
           YACJ = 0.0
           K = INBASE(J)
           IF(K.LE.0) YACJ = -C(J)
           YAC(J) = YACJ
           XJ = 0.0
           IF (K.EQ.-1) XJ = BOUND(J)
           IF (K.GT.0) XJ = XR(K)
10         X(J) = XJ
      TOL2 = TOL(2)
      DO 70 I = 1, MNOW
           L = ISEFF(I)
           Y(I) = 0.0
           IF (L.EQ.0) GO TO 30
           YI = YR(L)
           Y(I) = YI
           SLACK(I) = 0.0
           LAST = IROW(I+1) - 1
           ISTART = IROW(I)
           DO 20 LOOK = ISTART, LAST
                J = JCOL(LOOK)
                IF (INBASE(J).GT.0) GO TO 20
                AIJ = AA(LOOK)
                YAC(J) = YAC(J) + YI * AIJ
20              CONTINUE
           GO TO 70
30         IF (INREV.NE.1) GÓ TO 50
           SLKI = B(I)
           DO 40 J = 1, N
                IF (INBASE(J).EQ.0) GO TO 40
                SLKI = SLKI - A(I,J) * X(J)
40              CONTINUE
           GO TO 60
50         IF (R.EQ.0.0) GO TO 70
           SLKI = SLACK(I) - R * G(I) * XKPOS
60         IF (DABS(SLKI).LE.TOL2) SLKI = 0.0
           SLACK(I) = SLKI
           IF (NEGINV.NE.0) GO TO 70
```

```
          SI = S(I)
          ABSLKI = DABS(SLKI)
          IF (SI.NE.0.0.AND.SI*SLKI.GE.HOWNEG.OR.SI.EQ.0.0.AND.
     1        -ABSLKI.GE.HOWNEG) GO TO 70
          HOWNEG = -ABSLKI
          NEGROW = I
70        CONTINUE
      INREV = 0
      IF (MARKI.NE.0) SLACK(MARKI) = XR(MARKK)
      RETURN
      END
```

```
      SUBROUTINE FIRSTB
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 INV
      INTEGER SIZE, SIZE1, XBASIS, YBASIS
      COMMON/LINPCO/
     1BOUND(156),C(156),INBASE(156),PIV(156),X(156),YAC(156),
     2B(156),G(156),GR(156),ISEFF(156),S(156),SLACK(156),Y(156),
     3INV(130,130),XBASIS(130),XR(130),YBASIS(130),YR(130),
     4      TOL(8),BIG,DRIVER,INREV,IR,
     6      IRMAX, ISBIG, ISDONE, ISTATE, ITR, ITRMAX, M,
     7      MARKI, MARKK, MAXM, MAXN, MNOW, MORE, MOREPR,
     8      MXSIZE, N, NEGINV, NEGROW, NEWX, NUMSLK, OBJ, R,
     9      SIZE, SIZE1, SMALL, XKPOS, YAMINC, NEWY, ISBND
      COMMON/IO/IOIN, IOOUT
      DO 10 J = 1, N
10        INBASE(J) = 0
      DO 20 I = 2, MNOW
20        ISEFF(I) = 0
      DRIVER = 0.0
      NEGINV = 0
      SS = S(1)
      BB = B(1)
      IF(SS.EQ.1.0.AND.BB.GE.0.0.OR.SS.EQ.-1.0.AND.BB.LE.
     10.0.OR.SS.EQ.0.0.AND.BB.EQ.0.0) GO TO 30
      NEGINV = 1
      DRIVER = 1.0
      IF (BB.GT.0.0) DRIVER = -1.0
30    SIZE = 1
      SIZE1 = 2
      NEWX = N + 1
      XBASIS(1) = N + 1
      INV(1,1) = 1.0
      XR(1) = BB
      OBJ = 0.0
      YR(1) = 0.0
      YBASIS(1) = 1
      ISEFF(1) = 1
      NUMSLK = 1
      MARKI = 1
      MARKK = 1
      ITR = ITR + 1
      INREV = 1
      RETURN
      END
```

```
      SUBROUTINE ISOPT
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 INV
      INTEGER SIZE, SIZE1, XBASIS, YBASIS
      COMMON/IO/IOIN, IOOUT
      COMMON/LINPCO/
     1BOUND(156),C(156),INBASE(156),PIV(156),X(156),YAC(156),
     2B(156),G(156),GR(156),ISEFF(156),S(156),SLACK(156),Y(156),
     3INV(130,130),XBASIS(130),XR(130),YBASIS(130),YR(130),
     4      TOL(8),BIG,DRIVER,INREV,IR,
     6      IRMAX, ISBIG, ISDONE, ISTATE, ITR, ITRMAX, M,
     7      MARKI, MARKK, MAXM, MAXN, MNOW, MORE, MOREPR,
     8      MXSIZE, N, NEGINV, NEGROW, NEWX, NUMSLK, OBJ, R,
     9      SIZE, SIZE1, SMALL, XKPOS, YAMINC, NEWY, ISBND
      YAMINC = -TOL(3)
      NEWX = 0
      DO 10 L = 1, SIZE
           I = YBASIS(L)
           SI = S(I)
           IF (SI.EQ.0.0) GO TO 10
           YRL = YR(L) * SI
           IF (YRL.GE.YAMINC) GO TO 10
           YAMINC = YRL
           NEWX = I + N
10         CONTINUE
      TOL4 = TOL(4)
      DO 20 J = 1, N
           INBJ = INBASE(J)
           IF (INBJ.GT.0.OR.BOUND(J).EQ.0.0) GO TO 20
           T = YAC(J)
           IF (DABS(T).LE.TOL4) T = 0.0
           YAC(J) = T
           IF (INBJ.EQ.-1) T = -T
           IF (T.GE.YAMINC) GO TO 20
           YAMINC = T
           NEWX = J
20         CONTINUE
      RETURN
      END
```

```
      SUBROUTINE NEWVEC
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 INV
      INTEGER SIZE, SIZE1, XBASIS, YBASIS
      COMMON/IO/IOIN, IOOUT
      COMMON/LINPCO/
     1BOUND(156),C(156),INBASE(156),PIV(156),X(156),YAC(156),
     2B(156),G(156),GR(156),ISEFF(156),S(156),SLACK(156),Y(156),
     3INV(130,130),XBASIS(130),XR(130),YBASIS(130),YR(130),
     4      TOL(8),BIG,DRIVER,INREV,IR,
     6      IRMAX, ISBIG, ISDONE, ISTATE, ITR, ITRMAX, M,
     7      MARKI, MARKK, MAXM, MAXN, MNOW, MORE, MOREPR,
     8      MXSIZE, N, NEGINV, NEGROW, NEWX, NUMSLK, OBJ, R,
     9      SIZE, SIZE1, SMALL, XKPOS, YAMINC, NEWY, ISBND
      XKPOS = 1.0
      IF (NEWX.GT.N) GO TO 40
      DO 10 K = 1, SIZE
10         GR(K) = 0.0
      DO 30 L = 1, SIZE
           I = YBASIS(L)
           AIJ = A(I,NEWX)
           IF (AIJ.EQ.0.0) GO TO 30
           DO 20 K = 1, SIZE
20              GR(K) = GR(K) + AIJ * INV(K,L)
30         CONTINUE
      IF (INBASE(NEWX).EQ.-1) XKPOS = -1.0
      GO TO 60
40    I = NEWX - N
      L = ISEFF(I)
      DO 50 K = 1, SIZE
50         GR(K) = INV(K,L)
      IF (S(I).EQ.-1.0) XKPOS = -1.0
60    RETURN
      END
```

```
      SUBROUTINE REDUCE
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 INV
      INTEGER SIZE, SIZE1, XBASIS, YBASIS
      COMMON/IO/IOIN, IOOUT
      COMMON/LINPCO/
     1BOUND(156),C(156),INBASE(156),PIV(156),X(156),YAC(156),
     2B(156),G(156),GR(156),ISEFF(156),S(156),SLACK(156),Y(156),
     3INV(130,130),XBASIS(130),XR(130),YBASIS(130),YR(130),
     4      TOL(8),BIG,DRIVER,INREV,IR,
     6      IRMAX, ISBIG, ISDONE, ISTATE, ITR, ITRMAX, M,
     7      MARKI, MARKK, MAXM, MAXN, MNOW, MORE, MOREPR,
     8      MXSIZE, N, NEGINV, NEGROW, NEWX, NUMSLK, OBJ, R,
     9      SIZE, SIZE1, SMALL, XKPOS, YAMINC, NEWY, ISBND
      MARKI = 0
      MARKK = 0
      IF (NUMSLK.EQ.0) GO TO 80
      IT = SIZE
      DO 60 K = 1, IT
10          IF (SIZE.LE.1) GO TO 70
            J = XBASIS(K)
            IF (J.LE.N) GO TO 60
            I = J - N
            SI = S(I)
            IF(SI*XR(K).LT.0.0.OR.SI.EQ.0.0.AND.XR(K).NE.0.0) GO TO 60
            IF (K.EQ.SIZE) GO TO 30
            DO 20 L = 1, SIZE
20                INV(K, L) = INV(SIZE, L)
            J = XBASIS(SIZE)
            XBASIS(K) = J
            IF (J.LE.N) INBASE(J) = K
30          SLACK(I) = XR(K)
            XR(K) = XR(SIZE)
            IF (NEGINV.EQ.SIZE) NEGINV = K
            L = ISEFF(I)
            ISEFF(I) = 0
            IF (L.EQ.SIZE) GO TO 50
            DO 40 KK = 1, SIZE
40                INV(KK,L) = INV(KK,SIZE)
            YR(L) = YR(SIZE)
            YBASIS(L) = YBASIS(SIZE)
            I = YBASIS(SIZE)
            ISEFF(I) = L
50          XBASIS(SIZE) = 0
            SIZE = SIZE - 1
            SIZE1 = SIZE1 - 1
            NUMSLK = NUMSLK - 1
            GO TO 10
60          CONTINUE
70    IF (SIZE.LT.2.AND.XBASIS(1).GT.N) MARKK = 1
      IF (NEGINV.EQ.0.AND.MARKK.EQ.0) GO TO 80
      J = 0
      IF (NEGINV.NE.0) J = XBASIS(NEGINV)
      IF (J.GT.N) MARKK = NEGINV
```

```
      IF (MARKK.EQ.0) GO TO 80
      MARKI = XBASIS(MARKK) - N
80    RETURN
      END
```

```
      SUBROUTINE SEEKX
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 INV
      INTEGER SIZE,SIZE1,XBASIS,YBASIS
      COMMON /LINPCO/
     1BOUND(156),C(156),INBASE(156),PIV(156),X(156),YAC(156),
     2B(156),G(156),GR(156),ISEFF(156),S(156),SLACK(156),Y(156),
     3INV(130,130),XBASIS(130),XR(130),YBASIS(130),YR(130),
     4      TOL(8),BIG,DRIVER,INREV,IR,
     6      IRMAX, ISBIG, ISDONE, ISTATE, ITR, ITRMAX, M,
     7      MARKI, MARKK, MAXM, MAXN, MNOW, MORE, MOREPR,
     8      MXSIZE, N, NEGINV, NEGROW, NEWX, NUMSLK, OBJ, R,
     9      SIZE, SIZE1, SMALL, XKPOS, YAMINC, NEWY, ISBND
      COMMON /AREF/ AA(600),JCOL(600),IROW(157),MAXA
      NEWX=0
      R=-BIG
      PIVMAX=0.0
      JMAXP=0
      BESPIV=0.0
      TOL3=TOL(3)
      TOL4=TOL(4)
      TOL5=TOL(5)
      DO 10 J=1,N
   10 PIV(J)=0.0
      DO 40 L=1,SIZE
      I=YBASIS(L)
      SI=S(I)
      YI=YR(L)*SI
      IF(DABS(YI).LT.TOL3) YI=0.0
      RINVL=INV(NEGINV,L)
      ISTART=IROW(I)
      LAST=IROW(I+1)-1
      DO 20 LOOK=ISTART,LAST
      J=JCOL(LOOK)
      IF(INBASE(J).GE.1 .OR. BOUND(J).EQ.0.0) GO TO 20
      AIJ=AA(LOOK)
      PIV(J)=PIV(J)+AIJ*RINVL
   20 CONTINUE
      IF(SI.EQ.0.0) GO TO 40
      PIVOT=RINVL*SI*DRIVER
      IF(PIVOT.GE.-TOL5 .OR. PIVOT.GE.-0.5 .AND.NEWX.NE.0
     1   .AND. YI.LT.0.0) GO TO 40
      IF(PIVOT.GE.-0.5 .AND. YI.LT.0.0) GO TO 30
      RATIO=YI/PIVOT
      IF(RATIO.LT.R .AND. NEWX.NE.0) GO TO 40
      IF(RATIO.EQ.0.0 .AND. PIVOT.GE.BESPIV) GO TO 40
      IF(RATIO.EQ.0.0) BESPIV=PIVOT
      R=RATIO
      YAMINC=YI
      NEWX=N+I
      GO TO 40
   30 IF(PIVOT.GE.PIVMAX) GO TO 40
      YACP=YI
      JMAXP=N+I
```

```
      PIVMAX=PIVOT
   40 CONTINUE
      DO 60 J=1,N
      INJ=INBASE(J)
      IF(INJ.GE.1 .OR. BOUND(J).EQ. 0.0) GO TO 60
      SJ=1.0
      IF(INJ.EQ.-1) SJ=-1.0
      FUNC=YAC(J)*SJ
      IF(DABS(FUNC).LT.TOL4) FUNC=0.0
      PIVOT=PIV(J)*SJ*DRIVER
      IF(PIVOT.GE.-TOL5 .OR. PIVOT.GE.-0.5 .AND.NEWX.NE.0
     1   .AND. FUNC.LT.0.0) GO TO 60
      IF(PIVOT.GE.-0.5 .AND. FUNC.LT.0.0) GO TO 50
      RATIO=FUNC/PIVOT
      IF(RATIO.LT.R.AND. NEWX.NE.0) GO TO 60
      IF(RATIO.EQ.0.0 .AND. PIVOT.GE.BESPIV) GO TO 60
      IF(RATIO.EQ.0.0) BESPIV=PIVOT
      R=RATIO
      YAMINC=FUNC
      NEWX=J
      GO TO 60
   50 IF(PIVOT.GE.PIVMAX) GO TO 60
      PIVMAX=PIVOT
      YACP=FUNC
      JMAXP=J
   60 CONTINUE
      IF(NEWX.NE.0) GO TO 70
      NEWX=JMAXP
      YAMINC=YACP
      IF(NEWX.NE.0) R=YAMINC/PIVMAX
   70 RETURN
      END
      SUBROUTINE SEEKY
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 INV
      INTEGER SIZE,SIZE1,XBASIS,YBASIS
      COMMON /LINPCO/
     1BOUND(156),C(156),INBASE(156),PIV(156),X(156),YAC(156),
     2B(156),G(156),GR(156),ISEFF(156),S(156),SLACK(156),Y(156),
     3INV(130,130),XBASIS(130),XR(130),YBASIS(130),YR(130),
     4      TOL(8),BIG,DRIVER,INREV,IR,
     6      IRMAX, ISBIG, ISDONE, ISTATE, ITR, ITRMAX, M,
     7      MARKI, MARKK, MAXM, MAXN, MNOW, MORE, MOREPR,
     8      MXSIZE, N, NEGINV, NEGROW, NEWX, NUMSLK, OBJ, R,
     9      SIZE, SIZE1, SMALL, XKPOS, YAMINC, NEWY, ISBND
      COMMON /AREF/ AA(600),JCOL(600),IROW(157),MAXA
      SI=1.0
      BOUNDJ=-1.0
      R=BIG
      NEWY=0
      TOL5=TOL(5)
      IF(ISBND.EQ.0 .OR. NEWX.GT.N) GO TO 10
      IF(BOUND(NEWX).EQ.-1.0) GO TO 10
      R=BOUND(NEWX)
```

```
      NEWY=-1
   10 IF(NEGINV.EQ.0) GO TO 30
      XRNEG=XR(NEGINV)
      J=XBASIS(NEGINV)
      IF(J.GT.N) GO TO 20
      BOUNDJ=BOUND(J)
      IF(BOUNDJ.GE.XRNEG .OR. BOUNDJ.EQ.-1.0) GO TO 20
      XRNEG=XRNEG-BOUNDJ
   20 RTRY=XRNEG/(XKPOS*GR(NEGINV))
      IF(RTRY.GT.R) GO TO 30
      R=RTRY
      IF(R.LE.SMALL) R=0.0
      NEWY=NEGINV
      IF(R.EQ.0.0) GO TO 140
   30 DO 90 K=1,SIZE
      IF(K.EQ.NEGINV) GO TO 90
      GK=GR(K)*XKPOS
      IF(DABS(GK).LE.TOL5) GO TO 90
      J=XBASIS(K)
      IF(J.GT.N) SI=S(J-N)
      IF(J.LE.N) BOUNDJ=BOUND(J)
      XX=XR(K)
      IF(GK.LE.0.0) GO TO 70
      IF(XX.LT.0.0) GO TO 90
      IF(J.LE.N.AND.BOUNDJ.EQ.-1.0.OR.J.LE.N.AND.XX.LE.BOUNDJ) GO TO 40
      IF(J.GT.N .AND. SI.EQ.1.0) GO TO 40
      GO TO 90
   40 IF(XX.GE.GK*R) GO TO 90
   50 R=XX/GK
   60 IF(R.LE.SMALL) R=0.0
      NEWY=K
      IF(R.EQ.0.0) GO TO 140
      GO TO 90
   70 IF(J.GT.N) GO TO 80
      IF(BOUNDJ.EQ.-1.0 .OR. XX.LT.0.0 .OR. XX.GT.BOUNDJ) GO TO 90
      IF((XX-GK*R).LE.BOUNDJ) GO TO 90
      R=(BOUNDJ-XX)/(-1.0*GK)
      GO TO 60
   80 IF(XX.GE.0.0 .OR. S(J-N).GE.0.0) GO TO 90
      IF((XX-GK*R).LE.0.0) GO TO 90
      GO TO 50
   90 CONTINUE
      DO 130 I=1,MNOW
      IF(ISEFF(I).EQ.0) GO TO 100
      G(I)=0.0
      GO TO 130
  100 SLACKI=SLACK(I)
      SI=S(I)
      GI=0.0
      ISTART=IROW(I)
      LAST=IROW(I+1)-1
      DO 120 LOOK=ISTART,LAST
      J=JCOL(LOOK)
      INJ=INBASE(J)
```

```
      IF(INJ.LE.0) GO TO 110
      GI=GI-AA(LOOK)*GR(INJ)
      GO TO 120
  110 IF(J.EQ.NEWX) GI=GI+AA(LOOK)
  120 CONTINUE
      G(I)=GI
      IF(DABS(GI).LE.TOL5) GO TO 130
      IF(SI.EQ.0.0 .AND. SLACKI.NE.0.0) GO TO 130
      IF(SI*SLACKI.LT.0.0) GO TO 130
      GI=GI*XKPOS
      T=SLACKI-GI*R
      IF(T.GE.0. .AND. SI.EQ.1. .OR. T.LE.0. .AND. SI.EQ.-1.) GO TO 130
      R=SLACKI/GI
      IF(R.LE.SMALL) R=0.0
      NEWY=SIZE+I
      GR(SIZE1)=GI*XKPOS
      IF(R.EQ.0.0) GO TO 140
  130 CONTINUE
  140 RETURN
      END
```

APPENDIX F

SAMPLE INPUT AND OUTPUT FILE

```
500,20
0.0014,0.002,0.001,0.0015,0.0007,0.002,0.002,0.01
5,1
30.,31.,30.,31.,31.
'SEP'  'OCT'  'NOV'  'DEC'  'JAN'
 30,0
 20.,0.3,2.,0.3
 0.0,0.0,0.0,0.0,3700.
0.0,0.0,0.0,0.0,0.0
-534.6,-534.6,-534.6,-534.6,-534.6
 14000. 18000. 14000. 18000. 18000.
 8000. 8000. 8000. 10000. 9000.
-20000. -22000. -20000. -21000. -21000.
-12000. -12000. -12000. -14000. -12000.
2
6.0,5.0,6.0,7.0,6.
4.0,3.0,4.0,5.0,5.0
0.57,0.57,0.57,0.57,0.57
0.43,0.43,0.43,0.43,0.43
40.0 40.0 40.0 40.0 40.0
40.0 40.0 40.0 40.0 40.0
30.0 30.0 30.0 30.0 30.0
30.0 30.0 30.0 30.0 30.0
0.1 0.1 0.1 0.1 0.1
0.1 0.1 0.1 0.1 0.1
295.,1.
280.,280.,280.,280.,280.
300.,300.,300.,300.,300
1.5,1.5,1.5,1.5,1.5
0.0593701,-9.0836254
0.20,0.20,0.20,0.20,0.20
0.4,0.35,0.25,0.2,0.15
10.,0.
```

```
OUTPUT REPORT FROM THE EMSLP PROGRAM


 THE INPUT DATA
-------------------


*  LAND & POWELL CONTROL DATA

 500   20
   0.001400    0.002000    0.001000    0.001500    0.000700    0.002000
   0.002000    0.010000

*  PLANNING HORIZON DESCRIPTION

   5    1
     30.00       31.00       30.00       31.00       31.00
SEPOCTNOVDECJAN

*  ITERATIVE PROCESS CONTROL DATA

         30           0
  20.000000    0.300000    2.000000    0.300000

*  OBJ. FUN COST COEFF.

      0.00        0.00        0.00        0.00     3700.00
   0.000000    0.000000    0.000000    0.000000    0.000000
-534.600000 -534.600000 -534.600000 -534.600000 -534.600000
  14000.00    18000.00    14000.00    18000.00    18000.00
   8000.00     8000.00     8000.00    10000.00     9000.00
 -20000.00   -22000.00   -20000.00   -21000.00   -21000.00
 -12000.00   -12000.00   -12000.00   -14000.00   -12000.00

*  SYSTEM DEMAND DATA

   2
   6.000000    5.000000    6.000000    7.000000    6.000000
   4.000000    3.000000    4.000000    5.000000    5.000000
   0.570000    0.570000    0.570000    0.570000    0.570000
   0.430000    0.430000    0.430000    0.430000    0.430000
     40.00       40.00       40.00       40.00       40.00
     40.00       40.00       40.00       40.00       40.00
     30.00       30.00       30.00       30.00       30.00
     30.00       30.00       30.00       30.00       30.00
   0.100000    0.100000    0.100000    0.100000    0.100000
   0.100000    0.100000    0.100000    0.100000    0.100000
```

* RESERVOIR & POWER PLANT DATA

295.000000	1.000000			
280.00	280.00	280.00	280.00	280.00
300.00	300.00	300.00	300.00	300.00
1.500000	1.500000	1.500000	1.500000	1.500000
0.059370	-9.083625			

* RELEASE ESTIMATES

0.200000 0.200000 0.200000 0.200000 0.200000

* FORECASTED INFLOW

0.400000 0.350000 0.250000 0.200000 0.150000

* SCALE FOR LP & DISCOUNT RATE

10.000000 0.000000

REPORTS ON THE ITERATIVE SOLUTION PROCEDURE

ITERATION 1SIMPLEX ITERATIONS 42INITIAL STORAGE 295.OVARYMX 20.0

* THE RESERVOIR DATA *

MONTH *	INFLOW	GEN.FL.POW	GEN.FL.SP	TOT.OUTFL	ON EX	OFF EX	ON IN	OFF IN	HE ON	HE OFF	END STAGE	MONTHLY BENEFIT
SEP *	12.00	7.70	0.00	7.70	0.00	0.00	0.88	1.24	1.58	0.00	299.30	-36920.5
OCT *	10.85	10.15	0.00	10.15	0.00	0.00	0.00	0.96	2.12	0.00	300.00	-13930.3
NOV *	7.50	11.00	0.00	11.00	0.00	0.00	0.18	1.24	2.29	0.00	296.50	-21663.0
DEC *	6.20	14.50	0.00	14.50	0.00	0.00	0.00	1.60	2.97	0.00	288.20	-26469.6
JAN *	4.65	12.85	0.00	12.85	0.00	0.00	0.00	1.60	2.54	0.00	280.00	-22688.2

THE OBJECTIVE FUNCTION VALUE IS -121671.92

ITERATION 2SIMPLEX ITERATIONS 43INITIAL STORAGE 295.OVARYMX 20.0

* THE RESERVOIR DATA *

MONTH *	INFLOW	GEN.FL.POW	GEN.FL.SP	TOT.OUTFL	ON EX	OFF EX	ON IN	OFF IN	HE ON	HE OFF	END STAGE	MONTHLY BENEFIT
SEP *	12.00	7.70	0.00	7.70	0.00	0.00	0.88	1.24	1.58	0.00	299.30	-36921.1
OCT *	10.85	10.15	0.00	10.15	0.00	0.00	0.00	0.96	2.12	0.00	300.00	-13930.3
NOV *	7.50	10.43	0.00	10.43	0.00	0.00	0.30	1.24	2.16	0.00	297.07	-24334.0
DEC *	6.20	14.71	0.00	14.71	0.00	0.00	0.00	1.60	2.97	0.00	288.56	-26469.6
JAN *	4.65	13.21	0.00	13.21	0.00	0.00	0.00	1.60	2.54	0.00	280.00	-22688.2

THE OBJECTIVE FUNCTION VALUE IS -124343.48

ITERATION 3SIMPLEX ITERATIONS 45INITIAL STORAGE 295.OVARYMX 6.0

* THE RESERVOIR DATA *

MONTH *	INFLOW	GEN.FL.POW	GEN.FL.SP	TOT.OUTFL	ON EX	OFF EX	ON IN	OFF IN	HE ON	HE OFF	END STAGE	MONTHLY BENEFIT
SEP *	12.00	7.70	0.00	7.70	0.00	0.00	0.88	1.24	1.58	0.00	299.30	-36921.12
OCT *	10.85	10.15	0.00	10.15	0.00	0.00	0.00	0.96	2.12	0.00	300.00	-13930.37
NOV *	7.50	7.92	0.00	7.92	0.00	0.00	0.80	1.24	1.66	0.00	299.58	-35245.78
DEC *	6.20	11.09	0.00	11.09	0.00	0.00	0.69	1.60	2.28	0.00	294.68	-42234.98
JAN *	4.65	12.85	0.00	12.85	0.00	0.00	0.00	1.60	2.54	0.00	286.48	1287.72

THE OBJECTIVE FUNCTION VALUE IS -127044.53

THE INITIAL DECREASE IS ACCEPTED

ITERATION 3SIMPLEX ITERATIONS 46INITIAL STORAGE 295.0VARYMX 20.0

* THE RESERVOIR DATA *

MONTH *	INFLOW	GEN.FL.POW	GEN.FL.SP	TOT.OUTFL	ON EX	OFF EX	ON IN	OFF IN	HE ON	HE OFF	END STAGE	MONTHLY BENEFIT
SEP *	12.00	7.70	0.00	7.70	0.00	0.00	0.88	1.24	1.58	0.00	299.30	-36921.12
OCT *	10.85	10.15	0.00	10.15	0.00	0.00	0.00	0.96	2.12	0.00	300.00	-13930.37
NOV *	7.50	10.43	0.00	10.43	0.00	0.00	0.30	1.24	2.16	0.00	297.07	-24334.05
DEC *	6.20	14.71	0.00	14.71	0.00	0.00	0.00	1.60	2.97	0.00	288.56	-26469.66
JAN *	4.65	13.21	0.00	13.21	0.00	0.00	0.00	1.60	2.54	0.00	280.00	-22688.28

THE OBJECTIVE FUNCTION VALUE IS -124343.48

ITERATION 4SIMPLEX ITERATIONS 43INITIAL STORAGE 295.0VARYMX 20.0

* THE RESERVOIR DATA *

MONTH *	INFLOW	GEN.FL.POW	GEN.FL.SP	TOT.OUTFL	ON EX	OFF EX	ON IN	OFF IN	HE ON	HE OFF	END STAGE	MONTHLY BENEFIT
SEP *	12.00	7.70	0.00	7.70	0.00	0.00	0.88	1.24	1.58	0.00	299.30	-36921.1
OCT *	10.85	10.15	0.00	10.15	0.00	0.00	0.00	0.96	2.12	0.00	300.00	-13930.3
NOV *	7.50	10.07	0.00	10.07	0.00	0.00	0.37	1.24	2.09	0.00	297.43	-25873.4
DEC *	6.20	14.81	0.00	14.81	0.00	0.00	0.00	1.60	2.97	0.00	288.81	-26469.6
JAN *	4.65	13.46	0.00	13.46	0.00	0.00	0.00	1.60	2.54	0.00	280.00	-22688.2

THE OBJECTIVE FUNCTION VALUE IS -125882.90

ITERATION 5SIMPLEX ITERATIONS 43INITIAL STORAGE 295.0VARYMX 6.0

* THE RESERVOIR DATA *

MONTH *	INFLOW	GEN.FL.POW	GEN.FL.SP	TOT.OUTFL	ON EX	OFF EX	ON IN	OFF IN	HE ON	HE OFF	END STAGE	MONTHLY BENEFIT
SEP *	12.00	7.70	0.00	7.70	0.00	0.00	0.88	1.24	1.58	0.00	299.30	-36921.12
OCT *	10.85	10.15	0.00	10.15	0.00	0.00	0.00	0.96	2.12	0.00	300.00	-13930.37
NOV *	7.50	9.24	0.00	9.24	0.00	0.00	0.54	1.24	1.92	0.00	298.26	-29484.18
DEC *	6.20	14.75	0.00	14.75	0.00	0.00	0.00	1.60	2.97	0.00	289.71	-26469.66
JAN *	4.65	13.39	0.00	13.39	0.00	0.00	0.00	1.60	2.54	0.00	280.97	-19107.26

THE OBJECTIVE FUNCTION VALUE IS -125912.58

THE INITIAL DECREASE IS ACCEPTED

ITERATION 5SIMPLEX ITERATIONS 43INITIAL STORAGE 295.0VARYMX 20.0

* THE RESERVOIR DATA *

MONTH *	INFLOW	GEN.FL.POW	GEN.FL.SP	TOT.OUTFL	ON EX	OFF EX	ON IN	OFF IN	HE ON	HE OFF	END STAGE	MONTHLY BENEFIT
SEP *	12.00	7.70	0.00	7.70	0.00	0.00	0.88	1.24	1.58	0.00	299.30	-36921.12
OCT *	10.85	10.15	0.00	10.15	0.00	0.00	0.00	0.96	2.12	0.00	300.00	-13930.37
NOV *	7.50	10.07	0.00	10.07	0.00	0.00	0.37	1.24	2.09	0.00	297.43	-25873.48
DEC *	6.20	14.81	0.00	14.81	0.00	0.00	0.00	1.60	2.97	0.00	288.81	-26469.66
JAN *	4.65	13.46	0.00	13.46	0.00	0.00	0.00	1.60	2.54	0.00	280.00	-22688.28

THE OBJECTIVE FUNCTION VALUE IS -125882.90

ITERATION 6SIMPLEX ITERATIONS 44INITIAL STORAGE 295.0VARYMX 20.0

* THE RESERVOIR DATA *

MONTH *	INFLOW	GEN.FL.POW	GEN.FL.SP	TOT.OUTFL	ON EX	OFF EX	ON IN	OFF IN	HE ON	HE OFF	END STAGE	MONTHLY BENEFIT
SEP *	12.00	7.70	0.00	7.70	0.00	0.00	0.88	1.24	1.58	0.00	299.30	-36921.1
OCT *	10.85	10.15	0.00	10.15	0.00	0.00	0.00	0.96	2.12	0.00	300.00	-13930.3
NOV *	7.50	8.02	0.00	8.02	0.00	0.00	0.78	1.24	1.68	0.00	299.48	-34784.4
DEC *	6.20	14.66	0.00	14.66	0.00	0.00	0.00	1.60	2.97	0.00	291.02	-26469.6
JAN *	4.65	13.36	0.00	13.36	0.00	0.00	0.00	1.60	2.54	0.00	282.31	-14140.1

THE OBJECTIVE FUNCTION VALUE IS -126245.70

ITERATION 7SIMPLEX ITERATIONS 44INITIAL STORAGE 295.0VARYMX 6.0

* THE RESERVOIR DATA *

MONTH *	INFLOW	GEN.FL.POW	GEN.FL.SP	TOT.OUTFL	ON EX	OFF EX	ON IN	OFF IN	HE ON	HE OFF	END STAGE	MONTHLY BENEFIT
SEP *	12.00	7.70	0.00	7.70	0.00	0.00	0.88	1.24	1.58	0.00	299.30	-36921.1
OCT *	10.85	10.15	0.00	10.15	0.00	0.00	0.00	0.96	2.12	0.00	300.00	-13930.3
NOV *	7.50	8.02	0.00	8.02	0.00	0.00	0.78	1.24	1.68	0.00	299.48	-34784.4
DEC *	6.20	14.66	0.00	14.66	0.00	0.00	0.00	1.60	2.97	0.00	291.02	-26469.6
JAN *	4.65	13.36	0.00	13.36	0.00	0.00	0.00	1.60	2.54	0.00	282.31	-14140.1

THE OBJECTIVE FUNCTION VALUE IS -126245.70

THE INITIAL DECREASE IS ACCEPTED

ITERATION 7SIMPLEX ITERATIONS 44INITIAL STORAGE 295.0VARYMX 20.0

* THE RESERVOIR DATA *

MONTH *	INFLOW	GEN.FL.POW	GEN.FL.SP	TOT.OUTFL	ON EX	OFF EX	ON IN	OFF IN	HE ON	HE OFF	END STAGE	MONTHLY BENEFIT
SEP *	12.00	7.70	0.00	7.70	0.00	0.00	0.88	1.24	1.58	0.00	299.30	-36921.1
OCT *	10.85	10.15	0.00	10.15	0.00	0.00	0.00	0.96	2.12	0.00	300.00	-13930.3
NOV *	7.50	8.02	0.00	8.02	0.00	0.00	0.78	1.24	1.68	0.00	299.48	-34784.4
DEC *	6.20	14.66	0.00	14.66	0.00	0.00	0.00	1.60	2.97	0.00	291.02	-26469.6
JAN *	4.65	13.36	0.00	13.36	0.00	0.00	0.00	1.60	2.54	0.00	282.31	-14140.1

THE OBJECTIVE FUNCTION VALUE IS -126245.70

ITERATION 8SIMPLEX ITERATIONS 44INITIAL STORAGE 295.0VARYMX 20.0

* THE RESERVOIR DATA *

MONTH *	INFLOW	GEN.FL.POW	GEN.FL.SP	TOT.OUTFL	ON EX	OFF EX	ON IN	OFF IN	HE ON	HE OFF	END STAGE	MONTHLY BENEFIT
SEP *	12.00	7.70	0.00	7.70	0.00	0.00	0.88	1.24	1.58	0.00	299.30	-36921.12
OCT *	10.85	10.15	0.00	10.15	0.00	0.00	0.00	0.96	2.12	0.00	300.00	-13930.37
NOV *	7.50	7.53	0.00	7.53	0.00	0.00	0.88	1.24	1.58	0.00	299.97	-37016.59
DEC *	6.20	14.59	0.00	14.59	0.00	0.00	0.00	1.60	2.97	0.00	291.58	-26469.66
JAN *	4.65	13.30	0.00	13.30	0.00	0.00	0.00	1.60	2.54	0.00	282.92	-11867.68

THE OBJECTIVE FUNCTION VALUE IS -126205.41

CONVERGED VOLUME

NET ITERATIONS 5